UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

Center For International Economics

Working Paper Series

Working Paper No. 2026-04

Estimating, Forecasting and Backtesting a Family of Exponential and Other GARCH Models Using the fEGarch Package

Dominik Schulz, Yuanhua Feng, Christian Peitz, Oliver Kojo Ayensu

März 2026

Center For International Economics

Estimating, Forecasting and Backtesting a Family of Exponential and Other GARCH Models Using the `fEGarch` Package

Dominik Schulz[1], Yuanhua Feng[1], Christian Peitz[2], Oliver Kojo Ayensu[1]

[1]Faculty of Business Administration and Economics, Paderborn University

[2]Financial Intelligence Unit Germany

March 31, 2026

Abstract

A new R package called `fEGarch` is introduced that allows for the estimation of a broad family of short- and long-memory EGARCH models, as well as other popular short- and long-memory GARCH-type models, such as the (FI)GARCH and (FI)APARCH models. The EGARCH-family includes the well-established EGARCH and FIEGARCH models as special cases. An overview of the EGARCH-family models, their semiparametric extension and their extension to dual models with simultaneous mean modelling is given. Backtesting strategies for value-at-risk and expected shortfall in this context are summarized. A practical comparison with the `rugarch` package is drawn based on EGARCH models and the package's capabilities are compared to those of *OxMetrics* and the R package `ufRisk`. A simulation study highlights the consistency of the `fEGarch` estimators for the EGARCH-family models and their applications to real-world log-returns from stock markets underline the models' convincing performance. The application of `fEGarch` for backtesting multivariate GARCH and non-GARCH volatility models is presented briefly. The appendix includes relevant theoretical information, such as on available conditional distributions, including the new (scaled) average Laplace distribution and its skewed variant, as well as on estimation via conditional QMLE and on value-at-risk and expected shortfall forecasting.

Keywords: QMLE, `fEGarch`, exponential GARCH, dual long-memory processes

JEL Codes: C51, C58, C87

1 Introduction

Modelling and forecasting volatility in financial time series is a key task in risk management because of the inherent randomness associated with the rise and fall of stock prices. Major financial institutions investing in the stock market could, in the worst case, face insolvency, potentially leading to the destabilization of entire financial systems, if stock market losses were to reach sudden and unexpectedly extreme magnitudes (McNeil et al., 2015, p. 30). To forecast volatility and hence risk measures or to forecast risk measures directly, several approaches have been proposed over the last decades (see for example J.P. Morgan, 1996; Poon and Granger, 2003). In particular, generalized autoregressive conditional heteroskedasticity (GARCH, Bollerslev, 1986; Engle, 1982) models and their extensions have gained popularity since their introduction. This development was accompanied by the creation of a vast selection of different statistical software packages to make GARCH model estimation and forecasting easily applicable. We contribute to this collection of statistical software through the introduction of the `R` (R Core Team, 2026) package `fEGarch` (Schulz et al., 2026). `fEGarch` provides functionality that is not yet available:

(i) it allows for fitting of a vast class of univariate short- and long-memory models from the novel exponential GARCH family (EGF, see Ayensu et al., 2026; Peitz et al., 2026);

(ii) it allows for fitting further popular short- and long-memory GARCH models, like the short-memory GARCH and APARCH (asymmetric power ARCH, Ding et al., 1993), but also a selection of long-memory GARCH-type models aside from a standard FIGARCH (fractionally integrated GARCH, Baillie et al., 1996), like the fractionally integrated APARCH (FIAPARCH, Tse, 1998) and the fractionally integrated exponential GARCH (FIEGARCH, Bollerslev and Mikkelsen, 1996), among others;

(iii) it makes further model extensions such as simultaneous modelling in the mean and the variance or semiparametric model extensions easily applicable;

(iv) it introduces a new conditional distribution in GARCH modelling that has advantageous theoretical properties for (FI)EGARCH processes; and

(v) it gives access to one-step value-at-risk (VaR) and expected shortfall (ES) forecasting and traffic light backtesting (and to other backtests) for all its models as required by Basel regulations.

A comparison to other `R` packages for GARCH modelling shows the following:

rugarch (Galanos, 2024) provides access to a variety of univariate GARCH-type models including the standard GARCH, the exponential GARCH (EGARCH, Nelson, 1991) and the APARCH models under several common conditional distributions. Furthermore, it contains testing capabilities for fitted GARCH-type models and was, before the release of fEGarch, the only R package to explicitly allow for fitting a long-memory GARCH-type model, more precisely the FIGARCH. tseries (Trapletti and Hornik, 2024) allows for standard GARCH fitting under a conditional normal distribution. With fGarch (Wuertz et al., 2024), standard GARCH and APARCH models and special cases of them can be fitted under a selection of common conditional distributions. lgarch (Sucarrat, 2025) can be used to fit Log-GARCH (logarithmic GARCH) models via their autoregressive moving-average (ARMA) representation. A selection of multivariate GARCH models, including CCC-GARCH (constant conditional correlation GARCH, Bollerslev, 1990), DCC-GARCH (dynamic conditional correlation GARCH, Engle, 2002) and aDCC-GARCH (asymmetric DCC-GARCH, Cappiello et al., 2006), can be fitted with rmgarch (Galanos, 2025). ufRisk (Feng et al., 2023c) gives access to short-memory parametric and semiparametric GARCH-type models (under conditional normal and t-distribution only) by calling rugarch for the parametric model components. Moreover, it allows for backtesting its fitted models on test sets using traffic light tests for VaR and ES (see Costanzino and Curran, 2018, for a review of the traffic light test for VaR and the introduction of the traffic light test for ES). In conclusion and notwithstanding the vast options for GARCH modelling in R, the functionalities of fEGarch are, to our knowledge, in their entirety not yet provided by any other R package on CRAN (Comprehensive R Archive Network), most notably the option to fit a variety of popular univariate long-memory GARCH-type models.

The article is structured as follows: Section 2 provides a brief review of short- and long-memory EGF models, while also briefly presenting extensions to semiparametric models and parametric models for simultaneous mean and variance modelling. In Section 3, common backtesting procedures for VaR and ES, including traffic light tests, are reviewed as implemented in fEGarch. The main functions of fEGarch and their arguments are described in Section 4. Section 5 gives example applications to real-world daily log-returns. In Section 6, a simulation study to validate the implemented QMLE in fEGarch for the EGF models is presented. Section 7 provides a comparison of our package to rugarch based on estimated coefficients and computation time for the conventional EGARCH model fitting. Likewise, it gives a brief comparison of the capabilities of fEGarch to the related *G@RCH* extension in the statistical software *OxMetrics* (Laurent, 2018; Laurent and Peters, 2002) and to the R package `ufRisk` (Feng et al., 2023c), which fulfill related purposes. Section 8 concludes. The Appendix includes detailed simulation results and further theoretical topics on available conditional distributions as well as EGF properties, estimation and forecasting.

2 Selected available models in the `fEGarch` package

The `fEGarch` package allows for the estimation of a variety of short- and long-memory models. An overview of available GARCH-type models is provided in Table 1 excluding unnamed EGF specifications. Volatility-model estimation (see Appendix C.3), semi-parametric model extensions (see Section 2.2.1), simultaneous modelling in the mean and the variance (see Section 2.2.2), and risk measure forecasting (see Appendix C.4) as well as backtesting procedures (see Section 3) are available for all models in the package. However, within this article they will only be described in the context of the novel EGF models. For further information on the remaining models, we refer the readers to the original sources on those models.

Table 1: Available GARCH-type volatility models in `fEGarch`.

Model	Sources
EGARCH	Nelson (1991)
Log-GARCH	Pantula (1986), Geweke (1986), Milhøj (1987)
MEGARCH	Ayensu et al. (2026), Peitz et al. (2026)
MLog-GARCH	Ayensu et al. (2026), Feng et al. (2023a), Peitz et al. (2026)
GARCH	Engle (1982), Bollerslev (1986)
APARCH	Ding et al. (1993)
TGARCH	Zakoian (1994)
GJR-GARCH	Glosten et al. (1993)
FIEGARCH	Bollerslev and Mikkelsen (1996)
FILog-GARCH	Feng et al. (2020a)
FIMEGARCH	Ayensu et al. (2026), Peitz et al. (2026)
FIMLog-GARCH	Ayensu et al. (2026), Feng et al. (2023a), Peitz et al. (2026)
FIGARCH	Baillie et al. (1996)
FIAPARCH	Tse (1998)
FITGARCH	Ayensu et al. (2026)
FIGJR-GARCH	Ayensu et al. (2026)

2.1 The EGARCH-family models

The EGARCH was first introduced by Nelson (1991), and was stated in terms of a return series (r_t), $t \in \mathbb{Z}$, as an infinite-order exponential ARCH through

$$r_t = \sigma_t \eta_t, \quad \eta_t \sim \mathrm{IID}(0,1), \quad \text{with} \tag{1}$$

$$\ln\left(\sigma_t^2\right) = \omega_\sigma + \theta(B) g(\eta_{t-1}), \tag{2}$$

where $\eta_t \sim \mathrm{IID}(0,1)$ means that the innovations are independent and identically distributed (iid) random variables with mean zero and variance one, in particular in `fEGarch` from the distributions discussed in Appendix C.1. In this context, $\sigma_t = \sqrt{\mathrm{var}\left(r_t \mid \mathcal{F}_{t-1}\right)} > 0$ are the conditional standard deviations that depend on the corresponding set of past information $\mathcal{F}_{t-1}$ up until (and including) information at time point $t-1$, and

$$g(\eta_t) = \kappa\left[\eta_t - E(\eta_t)\right] + \gamma\left[|\eta_t| - E\left(|\eta_t|\right)\right] \tag{3}$$

is a transformation of the innovations, which splits into an asymmetry term $(\kappa\left[\eta_t - E(\eta_t)\right])$ and a magnitude term $(\gamma\left[|\eta_t| - E\left(|\eta_t|\right)\right])$. By definition of η_t, $E(\eta_t) = 0$ can be dropped from (3). ω_σ, κ and γ are real-valued coefficients, where $\omega_\sigma = E\left[\ln\left(\sigma_t^2\right)\right]$ due to $E\left[g\left(\eta_t\right)\right] = 0$ and where commonly κ and γ are assumed to not both be zero at the same time, while, assuming that the model order p satisfies $p \geq 1$, $\theta(B) = 1 + \sum_{i=1}^{\infty} \theta_i B^i$ is an infinite-order polynomial defined in more detail through

$$\theta(B) = \phi^{-1}(B)\psi(B) \tag{4}$$

with $\psi(B) = 1 + \sum_{j=1}^{q-1} \psi_j B^j$ and $\phi(B) = 1 - \sum_{i=1}^{p} \phi_i B^i$, where B is the backshift operator and ϕ_i, $i = 1, \ldots, p$, as well as ψ_j, $j = 1, \ldots, q-1$, are also real-valued. For $p = 0$, $\theta(B) = \psi(B)$ can be interpreted to be of infinite order but with $\theta_i = 0$ for all $i > q-1$. $\psi(B)$ and $\phi(B)$ are assumed to be without common roots. Following concerns indicated for example in Francq and Zakoïan (2019, p. 78), considering $q-1$ as the upper bound of the running variable in $\psi(B)$ is one possible way to account for an otherwise occurring identifiability problem in the model, however at the cost of the need for $q \geq 1$ in this representation. Other potential solutions can be to let the running variable in $\psi(B)$ go from $j = 1$ to $j = q$ and instead fix either κ or γ to one.

Another equivalent and more intuitive representation of (2) is

$$\ln\left(\sigma_t^2\right) = \omega + \sum_{j=0}^{q-1} \psi_j g(\eta_{t-j-1}) + \sum_{i=1}^{p} \phi_i \ln\left(\sigma_{t-i}^2\right) \tag{5}$$

with $\omega = \omega_\sigma \phi(1)$.

Bollerslev and Mikkelsen (1996) extended the EGARCH to the FIEGARCH by introducing the fractional differencing operator $(1-B)^d = \sum_{i=0}^{\infty} b_i B^i$ with $b_0 = 1$ and $b_i = (-1)^i \binom{d}{i}$ for $i \geq 1$, where $\binom{d}{i}$ is the common binomial coefficient, into (4), such that

$$\theta(B) = \phi^{-1}(B)(1-B)^{-d}\psi(B) = \sum_{i=0}^{\infty} \theta_i B^i \tag{6}$$

with $d \in [0,1]$ and $\theta_0 = 1$, where the model collapses to a short-memory EGARCH for $d = 0$, $d \in (0,1)$ describes the long-memory case with $d \in (0, 0.5)$ representing weak stationarity, if other conditions are met simultaneously, and an integrated EGARCH is with $d = 1$. Note that in order for $\theta(B)$ to be a true infinite-order polynomial, we require either $p \geq 1$ or $d > 0$. If $p = 0$ with $d = 0$, then like in the short-memory case $\theta(B) = \psi(B)$ with $\theta_i = 0$ for all $i > q-1$.

To extend and generalize the family of EGARCH models even further to how they can be applied in `fEGarch`, we consider (1) with (2) and (6). However, instead of (3), we generalize $g(\eta_t)$ through

$$g(\eta_t) = \kappa \left\{ g_{\text{asy}}(\eta_t) - E\left[g_{\text{asy}}(\eta_t)\right] \right\} + \gamma \left\{ g_{\text{mag}}(\eta_t) - E\left[g_{\text{mag}}(\eta_t)\right] \right\} \qquad \text{with} \tag{7}$$

$$g_{\text{asy}}(\eta_t) = \left\{ \begin{array}{ll} \operatorname{sgn}(\eta_t) \ln\left(|\eta_t| + M_{\text{asy}}\right), & \text{for } p_{\text{asy}} = 0, \\ \operatorname{sgn}(\eta_t) \left[\left(|\eta_t| + M_{\text{asy}}\right)^{p_{\text{asy}}} - M_{\text{asy}}\right] / p_{\text{asy}}, & \text{for } p_{\text{asy}} > 0, \end{array} \right\} \quad \text{and} \tag{8}$$

$$g_{\text{mag}}(\eta_t) = \left\{ \begin{array}{ll} \ln\left(|\eta_t| + M_{\text{mag}}\right), & \text{for } p_{\text{mag}} = 0, \\ \left[\left(|\eta_t| + M_{\text{mag}}\right)^{p_{\text{mag}}} - M_{\text{mag}}\right] / p_{\text{mag}}, & \text{for } p_{\text{mag}} > 0, \end{array} \right\} \tag{9}$$

where M_{asy} and M_{mag} are constants that are equal to one, if a modulus transformation is wanted in the asymmetry or magnitude term, respectively, and zero otherwise, and where $p_{\text{asy}}, p_{\text{mag}} \geq 0$ are two selectable power parameters. $\operatorname{sgn}(\cdot)$ is a function that returns the sign of its input. Note that for the special cases $p_{\text{asy}} = M_{\text{asy}} = 0$ or $p_{\text{mag}} = M_{\text{mag}} = 0$, we require the innovations to be different from zero almost surely for the model to be well-defined. A simple (FI)EGARCH is then given through the special case of selecting $M_{\text{asy}} = M_{\text{mag}} = 0$ together with $p_{\text{asy}} = p_{\text{mag}} = 1$. Another alternative, the (FI)MLog-GARCH (fractionally integrated modified Log-GARCH, Ayensu et al., 2026; Feng et al., 2023a; Peitz et al., 2026), is with $M_{\text{asy}} = M_{\text{mag}} = 1$ together with $p_{\text{asy}} = p_{\text{mag}} = 0$. Therefore, the (FI)MLog-GARCH model allows for a log-modulus-transformation in both its asymmetry and its magnitude terms. This ensures that $(|\eta_t| + 1) \geq 1$ and consequently the applicability of the natural logarithm to this transformed random variable. A third special case is the (FI)MEGARCH (fractionally integrated modified EGARCH, Peitz et al., 2026), for which $M_{\text{asy}} = 1$ with $p_{\text{asy}} = 0$ and simultaneously $M_{\text{mag}} = 0$ with $p_{\text{mag}} = 1$, i.e. a log-modulus-transformation in the asymmetry term while keeping the (FI)EGARCH specification of the magnitude term, are considered. Other combinations of M_{asy}, M_{mag}, p_{asy} and p_{mag} are also possible and applicable with `fEGarch`, however (FI)EGARCH, (FI)MEGARCH and (FI)MLog-GARCH have built-in specification shortcuts in the package and their characteristics have been studied in the literature at least to some extent. Note that this extended model type sharing resemblance to the classic EGARCH model is called an EGF model of **Type I** hereinafter. In the current package version, p_{asy} and p_{mag} are selected by the user manually before the estimation routine, so that they do not belong to the set of estimated parameters. M_{asy} and M_{mag} are selected manually as well.

An EGF model of **Type II** is, in contrast, based on the Log-GARCH (Geweke, 1986; Milhøj, 1987; Pantula, 1986) and FILog-GARCH (Feng et al., 2020a) models. Instead of (2), we have

$$\ln\left(\sigma_t^2\right) = \omega_\sigma + \gamma(B)\xi_t \qquad \text{with} \tag{10}$$

$$\gamma(B) = \sum_{i=1}^{\infty} \gamma_i B^i = \phi^{-1}(B)(1-B)^{-d}\psi(B) - 1, \tag{11}$$

where $\xi_t = \ln\left(\eta_t^2\right) - E\left[\ln\left(\eta_t^2\right)\right]$ and $\psi(B) = 1 + \sum_{j=1}^{q} \psi_j B^j$ up to lag q. $\phi(B)$ is defined as for the Type-I EGF case. In order for $\gamma(B)$ to be a true infinite-order polynomial, it is required that either $p \geq 1$ or $d > 0$. In contrast, for $p = 0$ with $d = 0$, $\gamma(B) = \psi(B) - 1$ with $\gamma_i = 0$ for all $i > q$. In addition, we require η_t to be different from zero almost surely for the model to be well-defined. Another representation of (10) is given by

$$\ln(\sigma_t^2) = \omega_\sigma \phi_{\inf}(1) + \left[\psi(B) - \phi_{\inf}(B)\right]\xi_t + \left[1 - \phi_{\inf}(B)\right]\ln(\sigma_t^2) \tag{12}$$

with $\phi_{\inf}(B) = 1 - \sum_{i=1}^{\infty} \phi_{\inf,i} B^i$ for $d > 0$ as the infinite-order coefficient polynomial resulting from $\phi(B)(1-B)^d$ and where missing coefficients in $\psi(B)$ are replaced by zero. This representation is particularly useful in fitting and forecasting with short-memory

Log-GARCH models, where then for $d = 0$,

$$\ln\left(\sigma_t^2\right) = \omega_\sigma \left(1 - \sum_{i=1}^{p} \phi_i\right) + \sum_{i=1}^{p} \phi_i \ln\left(\sigma_{t-i}^2\right) + \sum_{j=1}^{\max(p,q)} \left(\psi_j + \phi_j\right) \xi_{t-j}. \tag{13}$$

is the ARMA-representation of $\ln\left(\sigma_t^2\right)$ in a short-memory Log-GARCH, where it is assumed that $\phi_j = 0$ for $j > p$ or $\psi_j = 0$ for $j > q$.

Nonetheless, (10) with (11) also possesses a representation corresponding to (2) that has

$$g(\eta_t) = 2\gamma_1 \left\{\ln\left(|\eta_t|\right) - E\left[\ln\left(|\eta_t|\right)\right]\right\} \qquad \text{and} \tag{14}$$

$$\theta(B) = \gamma(B)/(\gamma_1 B), \tag{15}$$

where γ_1, assumed to be different from zero, is the first coefficient of the infinite-length polynomial $\gamma(B)$ in (10). Therefore, the key difference between Type-I and Type-II models is the definition of $g(\eta_t)$. A Type-II EGF model can hence be thought of as a Type-I EGF model, where in (7) through (9) we set $\kappa = p_{\text{mag}} = M_{\text{mag}} = 0$. Thus, a Type-II model does not include an asymmetry term in its definition of $g(\eta)$ and considers a simple log-transformation of $|\eta_t|$ for the magnitude term. However, for simplicity, for Type-II models of the EGF, the more familiar representation (10) with (11) for Log-GARCH-type models is used to state the estimation output in the `fEGarch` package, whereas the estimation output for Type-I models is stated following representation (2).

2.2 Further model extensions

2.2.1 Semiparametric models

A natural extension of the EGF is by introducing a smooth deterministic scale function $s(x_t) > 0$, where $x_t = t/n$ is the rescaled time with $t = 1, \ldots, n$. The adjusted definition of (1) reads

$$y_t = \mu + s(x_t) r_t, \tag{16}$$

with $\mu = E(y_t)$, where r_t is now governed by a process of the EGF and where we impose $\text{var}(r_t) = 1$, so that $s(x_t)$ is indeed a function of the unconditional standard deviation in (y_t) at time point t. Related ideas have for example been discussed by Feng (2004b), Sucarrat (2019), Feng et al. (2022) and Letmathe et al. (2022). A simple transformation of y_t invokes an additive component model

$$w_t \equiv \ln\left[(y_t - \mu)^2\right] = m(x_t) + u_t \tag{17}$$

with $m(x_t) = \ln\left\{[s(x_t)]^2\right\} + C_{\mathrm{T}}$ as a smooth, deterministic trend and $u_t = \ln(r_t^2) - C_{\mathrm{T}}$ as the zero-mean remainder due to $C_{\mathrm{T}} = E\left[\ln(r_t^2)\right]$. The inclusion of μ is reasonable, because an observed return can in practice sometimes be exactly zero. Let $\bar{y}$ be the arithmetic mean over the observed (y_t), then $\Pr(y_t - \bar{y} = 0) = 0$, which makes $\tilde{w}_t \equiv \ln\left[(y_t - \bar{y})^2\right]$ as an approximation to w_t be well-defined at all observation time points. The whole multi-step procedure to fit a semiparametric volatility model and to estimate the total volatilities $\Omega_t = s(x_t)\sigma_t$ then reads as follows.

(1) Obtain $\tilde{w}_t$ for all $t = 1, \ldots, n$.

(2) Estimate $m(x_t)$ in (17) from $(\tilde{w}_t)$ using some appropriate method, for example local polynomial regression (Fan and Gijbels, 1996) with automated bandwidth selection using the `smoots` (Feng et al., 2023b) or `esemifar` (Feng et al., 2024) packages under short- or long-memory errors as in the `fEGarch` package. Denote the estimates of $m(x_t)$ by $\hat{m}(x_t)$.

(3) Let $\hat{u}_t = \tilde{w}_t - \hat{m}(x_t)$ for all $t = 1, \ldots, n$. Compute $\hat{C}_{\mathrm{T}} = -\ln\left[n^{-1}\sum_{t=1}^{n}\exp(\hat{u}_t)\right]$ and subsequently $\hat{s}(x_t) = \exp\left\{\left(\hat{m}(x_t) - \hat{C}_{\mathrm{T}}\right)/2\right\}$.

(4) Obtain the estimated standardized returns $\hat{r}_t = (y_t - \bar{y})/\hat{s}(x_t)$.

(5) Fit a parametric GARCH-type model, including those of the EGF, to the collection of $\hat{r}_t$ and denote the estimated conditional volatilities $\tilde{\sigma}_t$ as estimates of the conditional volatilities σ_t.

(6) As estimates of the total volatilities Ω_t obtain $\hat{\Omega}_t = \hat{s}(x_t)\tilde{\sigma}_t$.

This procedure ensures that the estimated standardized returns $(\hat{r}_t)$ have roughly unit-sample-variance. Moreover, note that the scale of $(\hat{r}_t)$ is however irrelevant for $\hat{\Omega}_t$, as multiplication of $\hat{s}(x_t)$ by some constant $C^* \in \mathbb{R}$ - and thus division of $\hat{r}_t$ by C^* - will be absorbed by the estimate of ω_σ, when fitting a parametric EGF model to $(\hat{r}_t)$. Thus, the retransformation so that $(\hat{r}_t)$ has approximately unconditional sample variance one is solely made to adhere to the identifiability of the exact elements $s(x_t)$ and σ_t.

A further point is that, since the estimation of Ω_t as described here involves a two-step procedure, where firstly the nonparametric scale is estimated and then secondly a parametric model is fitted to the residuals of the first step, errors in the nonparametric step are carried forward to $(\hat{r}_t)$ and thus to the parametric step. Firstly, all members of the EGF share the property that the long-memory parameter is not affected by power- or log-transformations (Feng et al., 2020a; Surgailis and Viano, 2002). Moreover, it is well-established (Beran and Feng, 2002; Feng, 2004a; Feng et al., 2020b) that $\hat{m}(x_t)$ in (17) is consistent for $m(x_t)$, if the bandwidths are selected automatically by `smoots` and `esemifar`, however under the assumption of the existence of the eighth moment of u_t in case of `smoots` and under the assumption that (u_t) follows a FARIMA (fractionally integrated ARMA, Granger and Joyeux, 1980; Hosking, 1981) in case of `esemifar`, among others. While these assumptions are yet to be proven generally for the EGF, it is assumed that they hold at least approximately, so that we can obtain consistent $(\hat{u}_t)$ in our setting. In case of a semiparametric FILog-GARCH, however, it is known that (u_t) in (17) follows a FARIMA (Feng et al., 2020a). Now assume $E(r_t^4) < \infty$ and that $\hat{\Delta}$ as the QMLE based on the unknown (r_t) is consistent for the true parameter vector Δ_0 of the underlying EGF-process. Since $\hat{u}_t$ is consistent for u_t under given assumptions, $\hat{r}_t$ as given before is consistent for r_t, implying $\tilde{\Delta} \xrightarrow{p} \Delta_0$, where $\tilde{\Delta}$ is the QMLE of Δ based on $(\hat{r}_t)$.

Furthermore, note that in theory, in order for $\text{var}(r_t) = 1$ to hold, the parameter ω_σ is no longer free following (43). By setting $E(r_t^2) \overset{!}{=} 1$, we can deduce

$$\omega_\sigma = -\sum_{i=0}^{\infty} C_g(\theta_i) \tag{18}$$

as the restriction on ω_σ under a semiparametric model, where $C_g(\cdot)$ is the cumulant-generating function of $g(\eta_t)$.

2.2.2 Parametric models in the mean and the variance

As an alternative, the `fEGarch` package also provides access to simultaneous parametric modelling of the mean and the variance. Assume

$$\beta(B)(1-B)^D(y_t - \mu) = \alpha(B)r_t, \tag{19}$$

where the observed y_t follows a FARIMA with orders (P, D, Q) and with once again $\mu = E(y_t)$. $\beta(B) = 1 - \sum_{i=1}^{P} \beta_i B^i$ and $\alpha(B) = 1 + \sum_{j=1}^{Q} \alpha_j B^j$ as the common polynomials of an ARMA-type model without common roots and $D \in [0, 1]$ is the fractional differencing parameter with $D = 0$ implying a short-memory ARMA and $D = 1$ an ARIMA process. Then r_t is the error in (19) and is assumed to be governed by a process from the EGF and to be with $E(r_t) = 0$. Alternatively, the extended model can be represented through

$$y_t = \mu_t + r_t \qquad \text{with} \tag{20}$$

$$\mu_t = E\left(y_t \mid \mathcal{F}_{t-1}\right) = \mu + \left[1 - \alpha^{-1}(B)\beta(B)(1-B)^D\right](y_t - \mu), \tag{21}$$

where $\left[1 - \alpha^{-1}(B)\beta(B)(1-B)^D\right] = \sum_{i=1}^{\infty} \delta_i B^i$ is an infinite-order polynomial, if $Q \geq 1$ or $D > 0$. (20) with (21) describes the infinite-order AR-representation of a FARIMA with EGF-errors. Given that $r_t = \sigma_t \eta_t$, where σ_t is the conditional standard deviation, $y_t = \mu_t + \sigma_t \eta_t$ highlights the simultaneous modelling of (conditional) mean and (conditional) variance. Note that under D = 0, the representation

$$y_t - \mu = \sum_{i=1}^{P} \beta_i (y_{t-i} - \mu) + \sum_{j=1}^{Q} \alpha_j r_{t-j} + r_t \tag{22}$$

with finite-length polynomials holds.

By default, the `fEGarch` package considers Model (19) with $P = Q = D = 0$, so that only μ is considered by default from the mean model extension. For further discussion of theoretical concepts behind these ideas, we refer the readers to Feng et al. (2023a) and Karanasos and Kim (2003).

3 Backtesting VaR and ES

As a consequence of various financial crises in recent history, stronger regulations for financial institutions were implemented to account for market risk. As a means of market risk assessment, the Basel Committee on Banking Supervision (BCBS) introduced the value-at-risk (VaR, J.P. Morgan, 1996) into official regulations at the end of the 20th century (BCBS, 1996a), which is a quantile (or more precisely an estimate of it) of the (conditional) loss function. As a consequence of the global financial crisis in 2007, it became apparent that the VaR was not sufficient as a market risk measure, because it does not contain any information about the expected loss once the VaR is exceeded (in magnitude). Hence, the framework was adjusted to incorporate the coherent expected shortfall (ES, Acerbi and Tasche, 2002a, 2002b) as a risk measure (BCBS, 2016, 2019) as part of the Basel III international standards, which is expected to be adopted within the European Union with the beginning of the year 2027 (European Commission, 2025).

Assume a log-return series $(y_t)_{t=1}^{n+n_{\text{te}}}$ is observed, where $(y_t)_{t=1}^{n}$ are the observations used for the model estimation and where $(y_t)_{t=n+1}^{n+n_{\text{te}}}$ are the observations reserved for backtesting. Denote by

$$I_t = \begin{cases} 1, & \text{for } y_t < \widehat{\text{VaR}}_{\alpha}^{y}(t)\,, \\ 0, & \text{otherwise,} \end{cases} \tag{23}$$

$t = n+1, \ldots, n+n_{\text{te}}$, the hit sequence of the test period, in which the realized log-return lies below the corresponding (one-step-ahead forecast of the) VaR at level α. Please note that in this text, we consider VaR and ES at level α to be located in the lower tail of the conditional log-return distribution, so that these risk measures are based on the $1-\alpha$ quantile and therefore they usually take on negative values for common $\alpha = 0.975, 0.99$. If the I_t are uncorrelated, then $\Sigma_I \sim \text{Bin}(n_{\text{te}}, 1-\alpha)$ with $\Sigma_I = \sum_{t=n+1}^{n+n_{\text{te}}} I_t$, i.e. the sum of violations is binomially distributed with n_{te} trials and violation probability $1-\alpha$ in each trial with probability mass function $f_{\text{Bin}}(x, n_{\text{te}}, \alpha) = \binom{n_{\text{te}}}{x}(1-\alpha)^x \alpha^{n_{\text{te}}-x}$ and cumulative distribution function $F_{\text{Bin}}(x, n_{\text{te}}, \alpha) = \sum_{i=0}^{x} f_{\text{Bin}}(i, n_{\text{te}}, \alpha)$. For given n_{te} and α and the hit sequence $(I_t)_{t=n+1}^{n+n_{\text{te}}}$ based upon a fitted model, the traffic light test for VaR (BCBS, 1996b) sorts a model for forecasting the VaR at level α into the green zone for $F_{\text{Bin}}(\Sigma_I, n_{\text{te}}, \alpha) < 0.95$, into the yellow zone for $0.95 \leq F_{\text{Bin}}(\Sigma_I, n_{\text{te}}, \alpha) < 0.9999$ and into the red zone for $F_{\text{Bin}}(\Sigma_I, n_{\text{te}}, \alpha) \geq 0.9999$. Consequently, for the commonly used $n_{\text{te}} = 250$, a model for forecasting the 97.5%-VaR (99%-VaR) is sorted into the green zone for $\Sigma_I \leq 10$ ($\Sigma_I \leq 4$), into the yellow zone for $11 \leq \Sigma_I \leq 16$ ($5 \leq \Sigma_I \leq 9$) and into the red zone for $\Sigma_I \geq 17$ ($\Sigma_I \geq 10$).

A traffic light test for ES was suggested by Costanzino and Curran (2018). Let $\hat{\eta}_t = (y_t - \hat{\mu}_t)/\hat{\sigma}_t$ (for parametric models) or $\hat{\eta}_t = (y_t - \bar{y})/\hat{\Omega}_t$ (for semiparametric models), $t = n+1, \ldots, n+n_{\text{te}}$, be the (standardized) residuals in the test period following the rolling one-step-ahead volatility forecasts ($\hat{\sigma}_t$ or $\hat{\Omega}_t$) and mean forecasts ($\hat{\mu}_t$ or $\bar{y}$) of a fitted model. Reusing the hit sequence $(I_t)_{t=n+1}^{n+n_{\text{te}}}$,

$$X_{\text{ES}} = \sum_{t=n+1}^{n+n_{\text{te}}} \frac{[1 - F(\hat{\eta}_t)] - \alpha}{1-\alpha} I_t \tag{24}$$

is the severity of the VaR-breaches, where $F(\cdot)$ is the (possibly estimated) cumulative distribution function (cdf) of the model innovations. They state that $n_{\text{te}}^{-1/2}(X_{\text{ES}} - \mu_{\text{ES}}) \xrightarrow{d} N(0, \sigma_{\text{ES}}^2)$ with $\mu_{\text{ES}} = 0.5(1-\alpha)n_{\text{te}}$ and $\sigma_{\text{ES}}^2 = (1-\alpha)[(1+3\alpha)/12]$. Given $n_{\text{te}} = 250$, a model for forecasting the 97.5%-ES is sorted into the green zone for approximately $X_{\text{ES}} < 5.4768$, into the yellow zone for approximately $5.4768 \leq X_{\text{ES}} <$

8.4424 and into the red zone for approximately $X_{\mathrm{ES}} \geq 8.4424$ by imposing the same zone boundaries on the cumulative probabilities of observing the severity of breaches as for the VaR-traffic light test. Other ideas for backtesting the ES have also been proposed in the literature. For example the t-test for ES has been introduced in Section 9.3.2 in McNeil et al. (2015), which is, however, currently not yet implemented in `fEGarch`. Further backtests for VaR (Christoffersen, 1998; Kupiec, 1995) are, however, available within the package.

As a further model selection criterion, Feng et al. (2020a) introduced the weighted absolute deviation (WAD) as a criterion that accumulates information from the forecasted 97.5%-VaR, 97.5%-ES and 99%-VaR. Let $\mu_{\mathrm{VaR},0.975} = 0.025 n_{\mathrm{te}}$ and $\mu_{\mathrm{VaR},0.99} = 0.01 n_{\mathrm{te}}$ be the expected number of 97.5%- and 99%-VaR breaches in the test set. Then

$$\mathrm{WAD} = \frac{|\Sigma_{I,0.975} - \mu_{\mathrm{VaR},0.975}|}{\mu_{\mathrm{VaR},0.975}} + \frac{|\Sigma_{I,0.99} - \mu_{\mathrm{VaR},0.99}|}{\mu_{\mathrm{VaR},0.99}} + \frac{|X_{\mathrm{ES}} - \mu_{\mathrm{ES}}|}{\mu_{\mathrm{ES}}}. \tag{25}$$

The best model according to the WAD criterion can be considered to be the one that minimizes the WAD among the available models.

Further capabilities of `fEGarch` include the computation of so-called *loss functions* with respect to VaR and ES. Let R_t, $t = n+1, \ldots, n+n_{\mathrm{te}}$, be a (forecasted) risk measure, i.e. either VaR or ES, at time point t of the test period. Denote by $k_{\mathrm{pen}} \geq 0$ a constant penalty term. The loss function is then generally

$$\mathrm{LF}_i = \sum_{t=n+1}^{n+n_{\mathrm{te}}} L_{t,i}, \qquad i = 0,1,2,3, \quad \text{with} \tag{26}$$

$$L_{t,i} = (R_t - y_t)^2, \quad i = 0,1,2,3, \quad \text{for } y_t < R_t. \tag{27}$$

Different specifications of the loss function differ in how the case $y_t \geq R_t$ is treated. For $y_t \geq R_t$, we have $L_{t,0} = 0$ for the regulatory loss function (RLF, Sarma et al., 2003), we impose $L_{t,1} = k_{\mathrm{pen}} |R_t|$ for the firm's loss function (FLF, Sarma et al., 2003), we consider $L_{t,2} = k_{\mathrm{pen}} |R_t - y_t|$ for the adjusted loss function (ALF, Abad et al., 2015) and we use $L_{t,3} = k_{\mathrm{pen}} \min(|R_t - y_t|, |R_t|)$ as the corrected loss function (CLF, Feng and Peitz, 2025).

4 Overview of the `fEGarch` package

The `fEGarch` package (Schulz et al., 2026) uses specification functions to set up model specifications before the actual estimation step. In particular, for models of the EGF, the function `fEGarch_spec()` is used to provide details about the volatility model and its conditional distribution. The function's arguments are as follows.

`model_type`: either `"egarch"` or `"loggarch"` to specify the subtype (I or II) of the EGF.

`orders`: a two-element vector with the volatility model orders following the form $(p, q)'$.

`long_memo`: a logical value indicating whether to consider fractional differencing in the volatility model or not.

`cond_dist`: a character value from `"norm"`, `"std"`, `"ged"`, `"ald"`, `"snorm"`, `"sstd"`, `"sged"` and `"sald"` specifying the conditional distribution to consider; the options correspond to a normal distribution, a t-distribution, a generalized error distribution (GED), an average Laplace distribution (ALD) and their skewed variants.

`powers`: a two-element vector $(p_{\text{asy}}, p_{\text{mag}})'$ giving the power parameters for a Type-I EGF model.

`modulus`: a two-element logical vector for a Type I EGF model indicating whether the elements in the vector $(M_{\text{asy}}, M_{\text{mag}})'$ should be set to one (for `TRUE`) or to zero (for `FALSE`).

For convenience, the package includes a variety of wrapper functions around `fEGarch_spec()` that directly specify certain named submodels of the EGF as mentioned in Section 2.1. Those wrapper specification functions then only contain the arguments `orders` and `cond_dist`. Among these functions are `egarch_spec()`, `loggarch_spec()`, `megarch_spec()`, `mloggarch_spec()`, `fiegarch_spec()`, `filoggarch_spec()`, `fimegarch_spec()` and `fimloggarch_spec()`.

To further specify the conditional mean function to estimate simultaneously, the function `mean_spec()` should be considered with the following arguments.

`orders`: a two-element vector with the conditional mean model orders following the form $(P, Q)'$.

`long_memo`: a logical value indicating whether to consider fractional differencing in the conditional mean model or not.

`include_mean`: a single logical value indicating whether to include a model parameter for the unconditional mean or not; if it is not included for `include_mean = FALSE`, it is assumed that $\mu = 0$ in the model.

By default, `include_mean = TRUE`, `orders = c(0, 0)` and `long_memo = FALSE` in `mean_spec()`, i.e. the unconditional mean parameter is considered, ARMA-orders P and Q are set to zero and fractional differencing is not included.

In addition, `locpol_spec()` can be used to define settings for the optional scale function estimation via local polynomial regression.

`poly_order:` can be set to either `1` or `3` as the local polynomial order.

`kernel_order:` the smoothness-order of the second-order kernel weighting function implemented; can be one of the integers from `0` to `3`.

`boundary_method:` either `"extend"` or `"shorten`, where `"extend"` keeps the amount of observations used for smoothing constant even at boundary points by extending the window created by the bandwidth toward the interior accordingly; for `"shorten"`, the width of the smoothing window at boundary points keeps decreasing when approaching the first or last observation time point.

`bwidth:` the relative smoothing bandwidth; for the default `NULL`, automatic bandwidth selection algorithms are applied.

The main estimation function for models of the EGF is then `fEGarch()`. For simplicity, we reduce the subsequent display of function arguments to the most relevant ones.

`spec:` a volatility specification object as returned by `fEGarch_spec()` and related wrapper functions.

`rt:` a numeric vector or a time series of class `"ts"` or of class `"zoo"`.

`meanspec:` an object returned by `mean_spec()` with conditional mean model specifications.

`nonparspec:` the specification of the nonparametric model part as through `locpol_spec()`.

`use_nonpar:` a logical value indicating whether to include the nonparametric estimation step or not.

`n_test:` the number of observations from `rt` to reserve at the end of the series for testing; these observations will be excluded from the fitting process.

If not further specified in a function call to `fEGarch()`, the arguments `spec`, `meanspec` and `nonparspec` are set to the defaults of the calls to `egarch_spec()`, `mean_spec()` and `locpol_spec()`, respectively.

Further models included in the `fEGarch` package are the GARCH, the APARCH, the TGARCH (threshold GARCH), the GJR-GARCH (Glosten-Jagannathan-Runkle GARCH), and their corresponding fractionally integrated model versions. Accordingly, the functions `garch()`, `aparch()`, `tgarch()`, `gjrgarch()`, `figarch()`, `fiaparch()`, `fitgarch()` and `figjrgarch()` are made available. As an alternative, a call to the function `garchm_estim()` as a wrapper allows access to all these other models via selection through the argument `model`.

Moreover, the package provides access to forecasting functions for the available models. In that context, `predict()` provides general point forecasting capabilities for the conditional standard deviation and the conditional mean, where the argument `n.ahead` can be set to the desired forecasting horizon. On the other hand, `predict_roll()` grants the option to produce rolling point forecasts for a selectable step size, where one-step forecasts are the default. Nonetheless, in order to apply `predict_roll()` to a fitted model object from this package, the argument `n_test` must have been set to a value greater than zero previously in the model estimation step. Then the rolling point forecasts will be produced at the time points of those reserved observations. By default, the model will not be refitted throughout the test period, however setting the argument `refit_after` to a positive integer forces refitting the model each time after the specified number of test observations through `refit_after`. The additional argument `steady_window` (with default `FALSE`) grants the possibility to adjust the window of training observations when models are being refitted. For `steady_window = FALSE`, the first available observation is always the starting point of the training set, while the training set is only extended to the right-hand side in accordance with `refit_after`. For `steady_window = TRUE`, the width of the training window is kept constant throughout, so that when a model is being refitted the left-hand side of the window of the training set is also being shortened by length defined through `refit_after`.

The output from `predict_roll()` may then be passed to `measure_risk()`, where optionally value-at-risk (VaR) and expected shortfall (ES) can be computed for the test period in accordance with the forecasted conditional volatilities and conditional means. Model refitting, if enabled in `predict_roll()`, is also considered correctly for the corresponding test period subsets in VaR and ES computation via `measure_risk()`. `"VaR"` and `"ES"` can be selected, either individually or both at the same time, via the argument `measure`. The risk measures are then obtained at the confidence levels defined via the argument `level`. Once again note that VaR- and ES-forecasts should only be produced in context of a one-step-ahead setting.

An object with risk measures as returned by `measure_risk()` can then again be passed to the function `backtest_suite()`, which automatically applies the traffic light tests for VaR and ES (BCBS, 1996b; Costanzino and Curran, 2018) as well as unconditional and conditional coverage tests and an independence test for VaR-breaches (Christoffersen, 1998). Individual functions for these tests are also available within the package, for example in form of `trafflight_test()`. Once again, changes in conditional distribution due to model refitting in `predict_roll()` are considered in the package's backtesting functions. Loss functions are available through `loss_functions()` and can be applied to the output of `measure_risk()`.

The package is designed to allow for an intuitive application of a variety of (extended) volatility models and is therefore aimed at both researchers and practitioners in financial econometrics. A typical workflow for backtesting a model consists of the sequential application of a model specification function, of `fEGarch()`, then of `predict_roll()`, subsequently of `measure_risk()` and ultimately of `backtest_suite()`, clearly conveying a specific purpose within the backtesting methodology with each step. A set of S4-classes in `fEGarch` simplifies the reuse of estimation, forecasting and backtesting functions internally and provides user-level methods to easily obtain estimation and backtesting results within the R console. Additionally, designated plotting methods help users in visualizing key output from `fEGarch()` and `measure_risk()`, where both plots in the style of base R and `ggplot2` (Wickham, 2016) are supported through `plot()` and `autoplot()`, respectively. Although the package relies on C++ via `Rcpp` (Eddelbuettel and François, 2011) and `RcppArmadillo` (Eddelbuettel and Sanderson, 2014) for improved computational efficiency, computation times can increase notably under a (conditional) ALD or skew-ALD, under semiparametric model extensions, if the bandwidth in local polynomial regression is being estimated, and under long-memory models, especially for increasing numbers of observations. For common lengths of daily log-return series up to about twenty years, we, however, find computation times with `fEGarch` still reasonable.

5 Example applications

5.1 Backtesting parametric models for VaR and ES forecasting

We apply the `fEGarch` package to the daily log-returns of Apple (AAPL) from January 05, 2010, to December 31, 2024. A Log-GARCH, an EGARCH, an MEGARCH, an MLog-GARCH, all with orders $(1, 1)$, and a FILog-GARCH, a FIEGARCH, a FIMEGARCH and a FIMLog-GARCH, all with orders $(1, d, 1)$, are fitted under a conditional GED to the log-return series with the exception of the last 250 log-returns, which are reserved for backtesting. The corresponding returns for the initial training and test split are displayed in Figure 1, where clear shocks in the AAPL-returns are visible in 2020 due to the COVID-19 pandemic.

In the following, the data is read into R using base R's `read.csv()`, while the series is transformed into a `"zoo"`-class time series object (Zeileis and Grothendieck, 2005) immediately afterwards, since the `fEGarch` package, which is also attached in the following code simply by `library("fEGarch")`, also allows for fitting models to such time series objects and formats its output accordingly.

```
library(zoo); library(fEGarch)
data0 <- read.csv("Apple_20100104to20241231.csv")[-1, ]
rt <- data0$lret %>% zoo(order.by = as.Date(data0$ref_date))
```

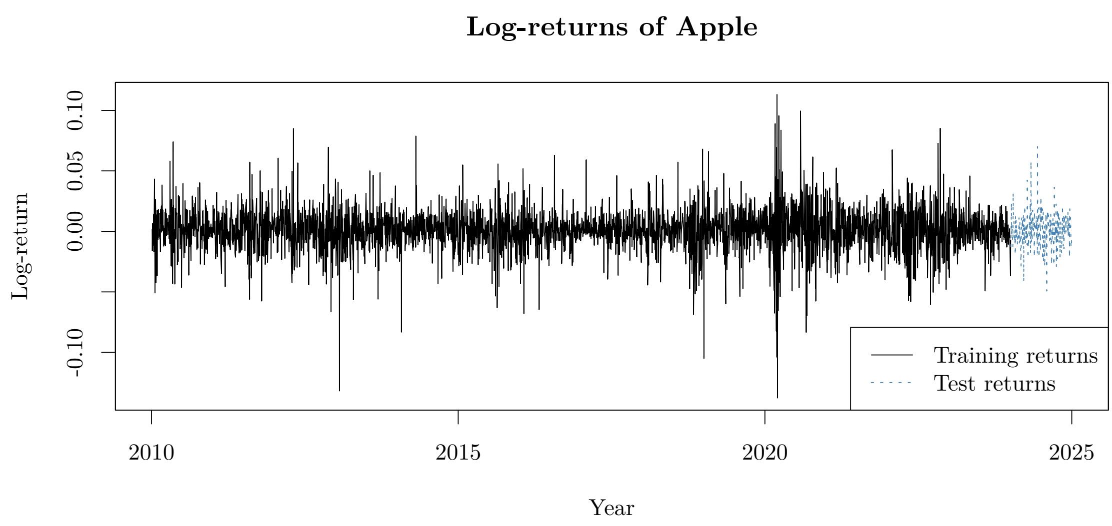

Figure 1: The initial training and test log-returns of Apple.

Given an input time series, which could also be a numeric vector or a `"ts"`-class time series object, an individual model can be fitted as follows, where a Log-GARCH$(1, 1)$ model with conditional GED is fitted.

```
# Fit individual model
model_single <- loggarch_spec(cond_dist = "ged") %>%
                  fEGarch(rt, n_test = 250)
```

Firstly, a specification function needs to be called. This can be either the more general `fEGarch_spec()` or, if a specific named EGF model should be fitted, a named specification function like `loggarch_spec()`. Within the specification function, the conditional distribution is adjusted, here to `cond_dist = "ged"` for considering a conditional GED. The specification output is then plugged directly into the main EGF-fitting function `fEGarch()`. This can be done either directly or, as shown here, using the pipe operator, which is imported by `fEGarch` from the `magrittr` package (Bache and Wickham, 2022) and re-exported to simplify workflows. Within the function `fEGarch()`, the full time series object - training and test observations - needs to be stated as well as the number of test observations `n_test`, which reserves the last `n_test` observations directly from the input time series. The default is `n_test = 0` to use all observations for fitting.

Alternatively, `fEGarch` also works well with common `apply`-functionalities in R to fit multiple models to the same series. A corresponding workflow is illustrated in the following, where at first a named list with the different model specification functions is created, which then `lapply()` iterates through.

```
# Fit multiple models
spec_funs <- list(
  "Log-GARCH" = loggarch_spec, "EGARCH" = egarch_spec,
  "MEGARCH" = megarch_spec, "MLog-GARCH" = mloggarch_spec,
  "FILog-GARCH" = filoggarch_spec, "FIEGARCH" = fiegarch_spec,
  "FIMEGARCH" = fimegarch_spec, "FIMLog-GARCH" = fimloggarch_spec
)
models <- spec_funs %>% lapply(FUN = function(.fun, rt) {
    .fun(cond_dist = "ged") %>% fEGarch(rt, n_test = 250)}, rt = rt)
```

Analogously, key results, such as the fitted conditional volatility series for the in-sample period, can be easily obtained and saved in a separate object.

```
volas <- models %>% lapply(FUN = function(.model) {.model@sigt}) %>%
  do.call(what = merge, args = .)
plot.zoo(volas, xlab = "Year", ylim = c(min(volas), max(volas)),
         main = "Fitted conditional standard deviations")
```

These series are shown in Figure 2, where all subfigures share the same scaling for the y-axis. It becomes clear that the Log-GARCH and the FILog-GARCH do not react as strongly to the COVID-shock, because the conditional volatility series show a significantly smaller peak around that time compared to the other models. Moreover, for this example, we can observe that the long-memory models produce slightly smaller volatility peaks in the year 2020 relative to the corresponding short-memory model variants, while keeping a similar volatility structure throughout. Only the Type-II EGF models - Log-GARCH and FILog-GARCH - produce estimated conditional volatilities that do not change as abruptly, potentially due to the lack of an asymmetry term in the model definition.

The estimated coefficients with corresponding standard errors as well as the normalized log-likelihood at the optimum $\left(l\left(\hat{\Delta}\right)/n\right)$, the normalized AIC (AIC$/n$) and the normalized BIC (BIC$/n$) can be observed in Table 2. The estimated coefficients can be accessed from each individual model output via `@pars`, the standard errors via `@se`, the (not normalized) log-likelihood via `@llhood` and the normalized AIC as well as the normalized BIC via `@inf_criteria`. Alternatively, correspondingly named methods can be applied to the estimation output of `fEGarch()`. Aside from the Log-GARCH and the FILog-GARCH - note also the different parameterization for those models -, all models produce similar coefficient estimates, where however, for long-memory models $\hat{\phi}_1$ is generally reduced in comparison to the corresponding short-memory models, however at the cost of an increased standard error. The estimated d-values range from 0.2412 to 0.3151 between long-memory models, while all except that of the FIMLog-GARCH are statistically significant - sometimes barely - at the 5%-level of significance. Nonetheless, the corresponding standard errors of $\hat{d}$ are quite large with values ranging from 0.0310 for the FILog-GARCH to 0.1887 for the FIMLog-GARCH, meaning the true d could be notably different still, even $d \geq 0.5$ is not implausible in some instances. Nonetheless, the estimated d-parameters are at least always more than one standard error away from 0.5. Moreover, the Log-GARCH and the FILog-GARCH models appear to be less suitable overall: they have the smallest (normalized) log-likelihood as well as the largest (normalized) AIC and BIC. In comparison, by log-likelihood the MEGARCH and the FIMEGARCH perform the best with approximately identical log-likelihood values, whereas by AIC and BIC the MEGARCH describes the training returns best. However, note that the information criteria for the Type-I EGF models are generally all close to each other for this example, i.e. they perform similarly well.

To produce one-step rolling conditional volatility point forecasts for the test period, according to VaR and ES forecasts and backtest results, we use the previous model outputs and the functions `predict_roll()`, `measure_risk()` and

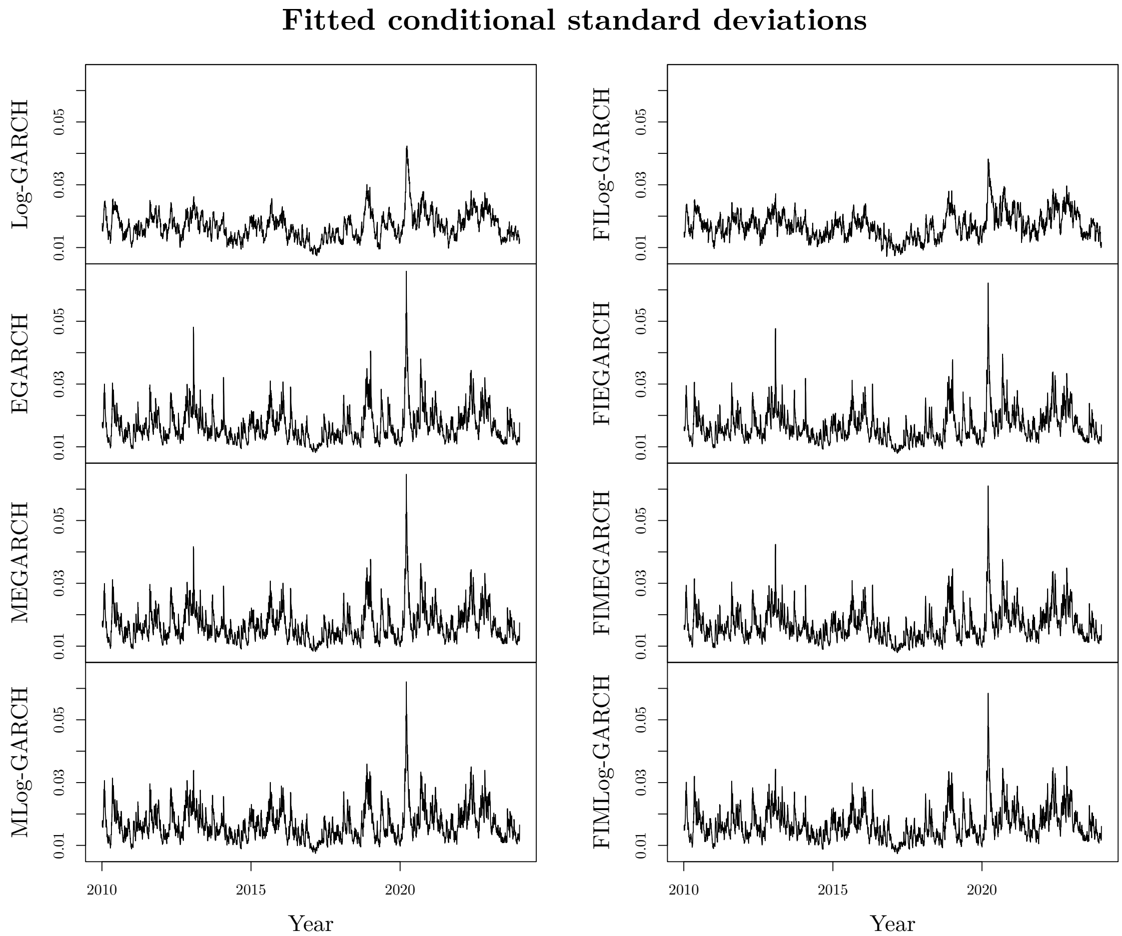

Figure 2: The fitted conditional standard deviations for the AAPL-log-returns following the eight main models of the EGF with conditional GED.

Table 2: Estimated parameters, normalized log-likelihood, normalized AIC and normalized BIC for the eight models with conditional GED (standard errors underneath in parentheses).

Model	$\hat{\mu}$	$\hat{\omega}_\sigma$	$\hat{\phi}_1$	$\hat{\psi}_1$	$\hat{\kappa}$	$\hat{\gamma}$	$\hat{d}$	$\hat{\beta}$	$l\left(\hat{\Delta}\right)/n$	AIC/n	BIC/n
Log-GARCH	0.0013	-8.3521	0.9739	-0.9254	—	—	—	1.2083	2.7227	-5.4426	-5.4338
	(0.0000)	(0.0977)	(0.0065)	(0.0109)				(0.0360)			
EGARCH	0.0010	-8.2642	0.9492	—	-0.1038	0.1845	—	1.2669	2.7382	-5.4729	-5.4624
	(0.0002)	(0.0774)	(0.0095)		(0.0155)	(0.0227)		(0.0385)			
MEGARCH	0.0010	-8.2536	0.9483	—	-0.1823	0.1883	—	1.2696	**2.7387**	**-5.4739**	**-5.4634**
	(0.0002)	(0.0783)	(0.0097)		(0.0266)	(0.0228)		(0.0386)			
MLog-GARCH	0.0010	-8.2629	0.9473	—	-0.1807	0.3940	—	1.2705	2.7386	-5.4738	-5.4633
	(0.0002)	(0.0780)	(0.0096)		(0.0268)	(0.0465)		(0.0387)			
FILog-GARCH	-0.0008	-8.4276	0.3291	-0.5421	—	—	0.2683	1.2003	2.7130	-5.4225	-5.4120
	(0.0000)	(0.1793)	(0.0465)	(0.0573)			(0.0310)	(0.0353)			
FIEGARCH	0.0010	-8.2578	0.8058	—	-0.0942	0.1652	0.3127	1.2662	2.7382	-5.4725	-5.4603
	(0.0001)	(0.1378)	(0.1159)		(0.0160)	(0.0243)	(0.1461)	(0.0384)			
FIMEGARCH	0.0010	-8.2317	0.7991	—	-0.1657	0.1713	0.3151	1.2685	**2.7387**	-5.4734	-5.4611
	(0.0002)	(0.1585)	(0.1227)		(0.0277)	(0.0246)	(0.1522)	(0.0386)			
FIMLog-GARCH	0.0010	-8.2632	0.8502	—	-0.1646	0.3548	0.2412	1.2693	2.7386	-5.4732	-5.4609
	(0.0002)	(0.1299)	(0.1123)		(0.0270)	(0.0476)	(0.1887)	(0.0386)			

Table 3: Out-of-sample results of traffic light tests for 97.5%-VaR, 99%-VaR and 97.5%-ES together with WAD for the eight main EGF models.

	97.5%-VaR		99%-VaR		97.5%-ES		
Model	Cumul. prob.	Zone	Cumul. prob.	Zone	Cumul. prob.	Zone	WAD
Log-GARCH	0.4040	Green	0.5432	Green	0.3728	Green	0.5484
EGARCH	0.5657	Green	0.5432	Green	0.5039	Green	0.2444
MEGARCH	0.5657	Green	0.5432	Green	0.5541	Green	0.3022
MLog-GARCH	0.5657	Green	0.5432	Green	0.6077	Green	0.3651
FILog-GARCH	0.4040	Green	0.5432	Green	0.4564	Green	0.4502
FIEGARCH	0.5657	Green	0.5432	Green	0.4994	Green	**0.2407**
FIMEGARCH	0.5657	Green	0.5432	Green	0.5659	Green	0.3160
FIMLog-GARCH	0.5657	Green	0.5432	Green	0.6161	Green	0.3751

`trafflight_test()` of the `fEGarch` package. `predict_roll()` produces the rolling volatility forecasts, `measure_risk()` computes the corresponding VaR and ES series, and `trafflight_test()` runs traffic light tests for VaR and ES. Setting `refit_after = 50` allows for the initially fitted models to be fitted repeatedly after every 50 test observations. Using common `apply` approaches, this can be done for multiple models, where the model output object is simply passed to `predict_roll()`, whose output is fed into `measure_risk()`, whose output is yet again simply the input for `trafflight_test()`. This workflow can also be implemented using the pipe-operator as shown below. Commonly by default `silent = FALSE`, each call to `trafflight_test()` will print the test results directly to the console. As a next step, we apply `WAD()` to compute the WAD-criterion for each model. Alternatively, however not shown here, one may also apply coverage and independence tests via `cov_tests()` or the entire backtesting suite with traffic light tests, coverage and independence tests and WAD with `backtest_suite()`. Moreover, loss functions can be implemented by applying `loss_functions()` to the output of `measure_risk()`, which is however also omitted here. The backtesting results for `trafflight_test()` and `WAD()` are displayed in Table 3. All models are sorted into the green zone for the 97.5%-VaR, the 99%-VaR and the 97.5%-ES. Interestingly, following the WAD-criterion, the FIEGARCH model performs best on the test returns among the tested models with a WAD-value of 0.2407 in contrast to the best-performing models on the training data, which however still perform well on the test series.

```
risks <- models %>% lapply(FUN = function(.model) {
    .model %>% predict_roll(refit_after = 50) %>% measure_risk()})
backtests <- risks %>% lapply(FUN = trafflight_test, silent = TRUE)
wads <- risks %>% lapply(FUN = WAD, silent = TRUE)
```

Plots of risk measure results can be easily created by applying `plot()` to the output of `measure_risk()` as shown below. Via the argument `which`, we may select the confidence level of the risk measures among the available options in the risk output.

```
plot(risks$`FIEGARCH`, which = 0.975, xlab = "Year",
     main = "FIEGARCH")
```

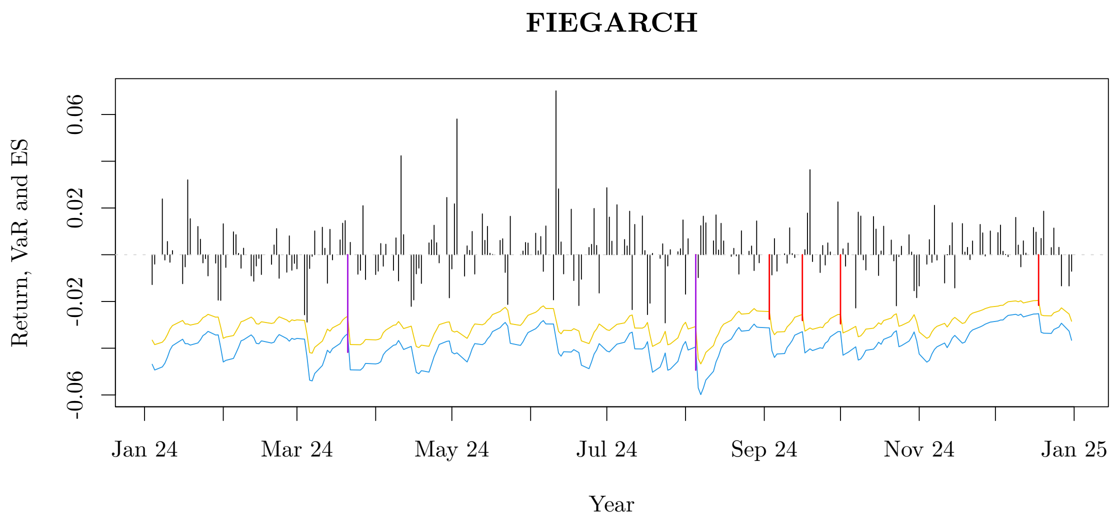

Figure 3: AAPL-test returns together with one-step forecasted 97.5%-VaR and 97.5%-ES following a FILog-GARCH$(1, d, 1)$ and a FIEGARCH$(1, d, 1)$ and corresponding VaR (red) and ES (purple) violations.

As can be seen in Figure 3, the FIEGARCH works well for forecasting the 97.5%-VaR and the 97.5%-ES in the AAPL-returns. There are only six cases, in which the VaR is violated, and two cases, where the returns are marginally smaller than the ES.

5.2 Semiparametric model implementation

We implement a semiparametric MLog-GARCH$(1, 1)$ model - in short a Semi-MLog-GARCH$(1, 1)$ model - for the daily log-returns of the Standard & Poor's 500 index from January 03, 2006, to December 31, 2021. The data is a subset of the object `SP500` in the `fEGarch` package and was originally collected from Yahoo Finance. The subset of `SP500` is selected to reduce computation time. Figure 4(a) shows the log-return series with the initial split into training and test observations. The model is estimated by calling `mloggarch_spec()` and the usage of a conditional skew-ALD by `cond_dist = "sald"` within that call is illustrated below. The actual semiparametric extension is enabled by setting `use_nonpar = TRUE` in `fEGarch()`.

Moreover, `nonparspec = locpol_spec(poly_order = 3)` allows to implement an automated local cubic regression (see, for example, Letmathe et al., 2024, for more information) for the scale function estimation. Furthermore, the test period is set to 250 observations by setting `n_test = 250` within `fEGarch()`. By default, the bandwidth relevant for this purpose is estimated automatically, however, it could also be set manually using the argument `bwidth` in `locpol_spec()`.

```
library(zoo); library(fEGarch)
# data included in the fEGarch package
rt <- window(SP500, start = "2006-01-01", end = "2021-12-31")
model_sp <- mloggarch_spec(cond_dist = "sald") %>%
  fEGarch(rt, use_nonpar = TRUE, n_test = 250,
          nonparspec = locpol_spec(poly_order = 3))
model_sp
```

The estimated total volatility following the Semi-MLog-GARCH$(1, 1)$ is displayed together with the estimated scale function in Figure 4(b), whereas the estimated conditional volatility from the parametric model part is being shown in Figure 4(c). The fitted in-sample total volatilities reflect high- and low-risk periods of the training returns suitably, while the estimated scale function provides estimates of the slowly changing unconditional standard deviation that causes the training returns to be non-stationary. In particular at around the years 2009 and 2020 strong increases in the scale function and therefore also in volatility are visible. Moreover, we observe a clearly stabilizing effect of the semiparametric extension on the parametric model part, because by removing the local component, the conditional volatilities in Figure 4(c) seem to be stable.

Risk measure forecasts, backtests and test period plots can be generated as before using `predict_roll()` with `measure_risk()`, `trafflight_test()` and `plot()`, respectively. We refit the model after every 50 test observations and use a fixed moving window width for the training periods.

```
risk <- model_sp %>%
  predict_roll(refit_after = 50, steady_window = TRUE) %>%
  measure_risk()
backtest <- risk %>% trafflight_test()
plot(risk, which = 0.975, xlab = "Month")
```

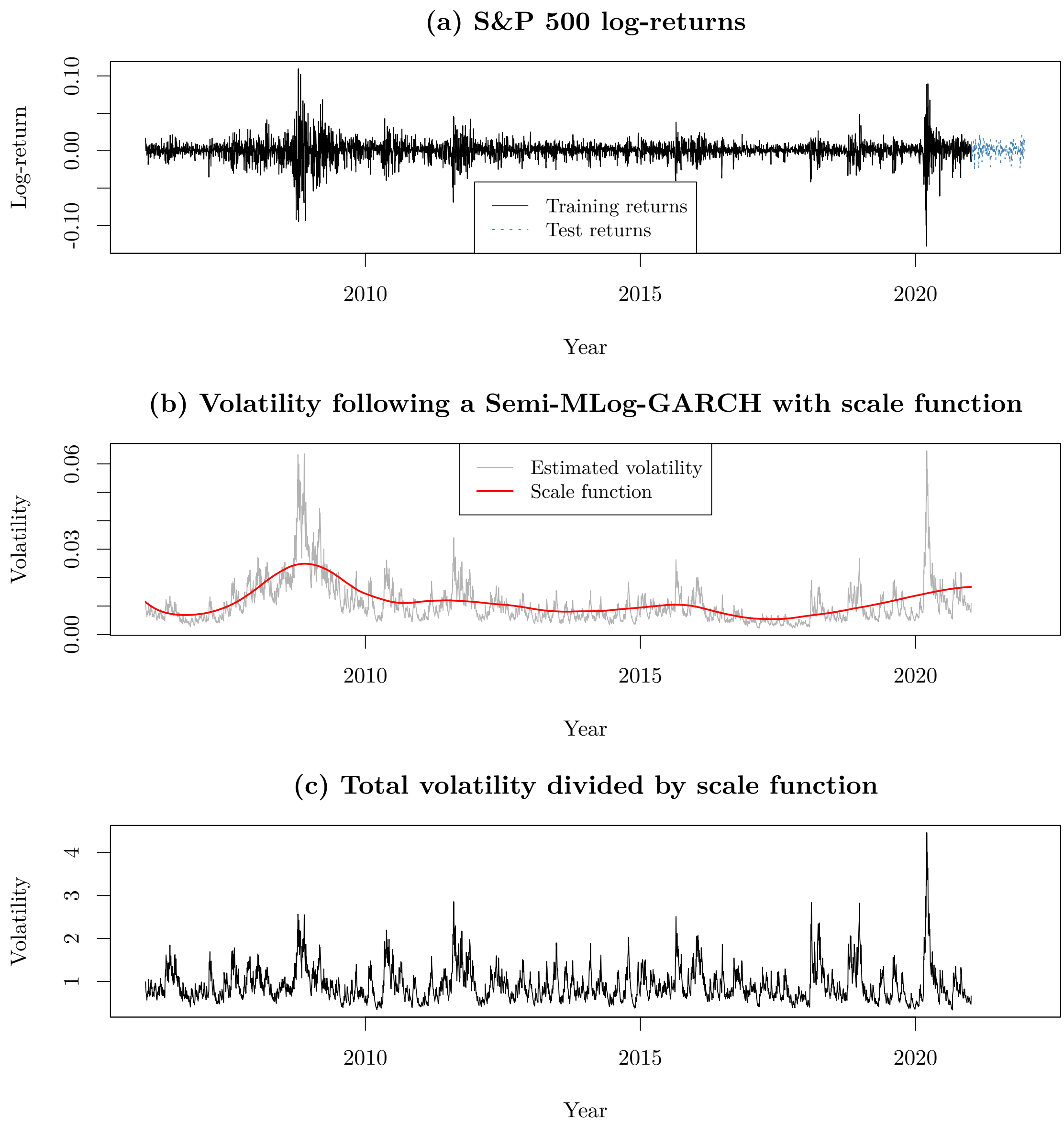

Figure 4: Results of a fitted Semi-MLog-GARCH$(1, 1)$ model with conditional skew-ALD. The estimated scale function captures the slowly moving unconditional standard deviation over time, the estimated conditional volatility following the parametric model part appears to be stable.

As can be seen from Figure 5, however, is that the estimated Semi-MLog-GARCH$(1, 1)$ model performs well in forecasting the 97.5%-VaR and the 97.5%-ES. Only a few violations of these risk measures occur over the test period and they only fall marginally below the risk measures at the corresponding time points. Backtesting results from `trafflight_test()` show that the forecasted 97.5%-VaR, the 97.5%-ES and the 99%-VaR would all be sorted into the green zone for this series following this model, thus implying the suitability of the model for forecasting these risk measures from a regulatory perspective.

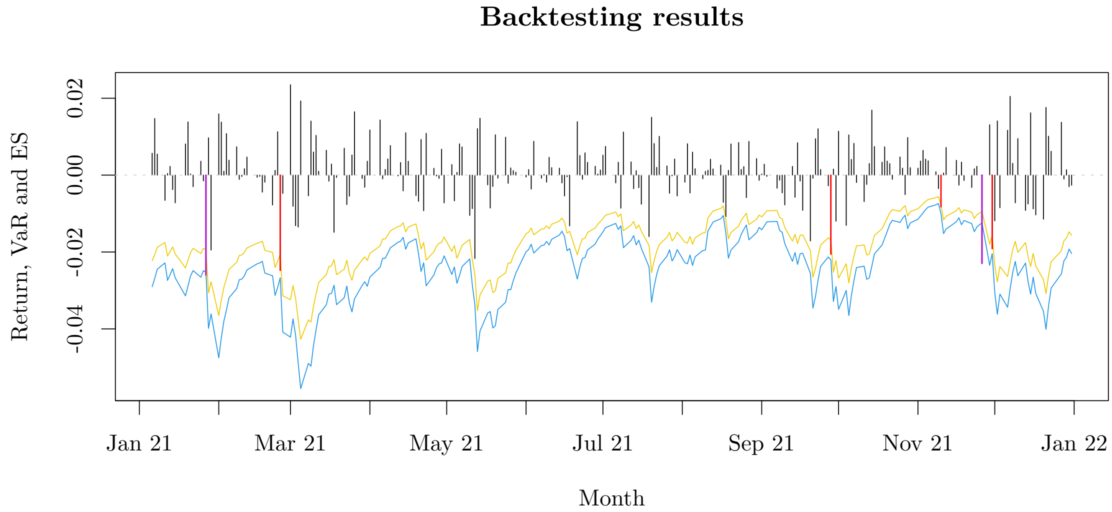

Figure 5: Backtesting results at 97.5%-level for a semiparametric MLog-GARCH$(1,1)$ with conditional skew-ALD.

5.3 Backtesting CCC- and DCC-GARCH models

Multivariate GARCH (MGARCH) models are crucial in many real-world applications, as risk managers often need to assess the risk associated with entire portfolios instead of individual stocks. Due to covariances between stocks, risk measures for portfolios usually cannot simply be obtained from the risk measures of the individual portfolio elements. Therefore, various MGARCH models have been proposed in the literature with the VEC-GARCH (vectorized GARCH, Bollerslev et al., 1988) and the BEKK-GARCH (Baba, Engle, Kraft and Kroner GARCH, Engle and Kroner, 1995) being early proposals that suffer from large numbers of parameters to estimate already for portfolios consisting of a relatively small number of series. The need for stable and computationally feasible MGARCH models led to the development of the aforementioned CCC-GARCH (Bollerslev, 1990) and DCC-GARCH (Engle, 2002) models. Usually, see for example Galanos (2025), these models are being estimated in a two-step procedure, where initially univariate GARCH-type models are fitted to the univariate log-return series of each of the individual stocks of a portfolio. In a second-step, the correlations between the residuals from the first step are being modelled, either through an estimate of a constant correlation matrix over the entire estimation period in case of a CCC-GARCH or by means of a time-changing conditional correlation matrix, if a DCC-GARCH is employed. Further details on the models are omitted at this point.

While `fEGarch` is not designed to fit CCC-GARCH and DCC-GARCH models directly and produce volatility forecasts with them over a test period, it can still be used to extend those models and generally any model that produces one-step volatility forecasts over a test period for VaR and ES backtesting. Assume that $\hat{\mu}_t^{\mathrm{g}}$ and $\hat{\sigma}_t^{\mathrm{g}}$, $t = 1, \ldots, n + n_{\mathrm{te}}$, are the general fitted conditional means and conditional standard deviations, respectively, over the training period for $t = 1, \ldots, n$ and the general rolling one-step-ahead point forecasts of the conditional mean and of the conditional standard deviation, correspondingly, over the test period for $t = n + 1, \ldots, n + n_{\mathrm{te}}$ according to some appropriate model. Apart from that, let the observed return series, which may also be a series of portfolio returns, over the training period be described through y_t, $t = 1, \ldots, n$. Then the standardized residuals are given through $\hat{\eta}_t^{\mathrm{g}} = (y_t - \hat{\mu}_t^{\mathrm{g}})/\hat{\sigma}_t^{\mathrm{g}}$, $t = 1, \ldots, n$, for the training period. `fEGarch` provides the function `find_dist()`, which fits the various available distributions in `fEGarch` to a series via QMLE and selects the best distribution model according to an information criterion like BIC. This way, an approximately best distribution model can be selected for a residual series, even if the underlying volatility model works without distribution assumption or if the true distribution is unknown. The output of `find_dist()` can be plugged into `measure_risk()` and then `trafflight_test()` again. As further options, the package also gives access to individual functions for fitting specific distributions via QMLE to a numerical input vector. These functions are `norm_est()`, `std_est()`, `ged_est()`, `ald_est()`, `snorm_est()`, `sstd_est()`, `sged_est()` and `sald_est()`.

This makes fitting and backtesting a CCC-GARCH model feasible using `fEGarch` for estimating the initial models on the univariate stock returns and the final backtesting. The steps in between need to be coded by the practitioner. Assume we have the two-element numeric vector of portfolio weights `w <- c(0.6, 0.4)`, where the first element is the weight of the first series and the second element is that for the second series in the portfolio. Note that the following ideas may also analogously be easily extended to portfolios with more series. Given the two log-return series in form of numeric vectors `yt1` and `yt2`, we can fit any univariate GARCH-type model to them including semiparametric extensions or dual models; the types of these models are allowed to differ, i.e. one may fit a FIEGARCH$(1, d, 1)$ model with any of the available conditional distributions to Series 1 and a FIMLog-GARCH$(1, d, 1)$ model with the same or another one of the available conditional distributions to Series 2, while also already reserving observations for testing via the argument `n_test` in `fEGarch()`. Afterwards, we can estimate the constant correlation matrix from the in-sample residuals of the fitted models. A workflow may look as follows after attaching the `fEGarch` package.

```
model1 <- fiegarch_spec(cond_dist = "ged") %>%
  fEGarch(yt1, n_test = 250)
model2 <- fimloggarch_spec(cond_dist = "sstd") %>%
  fEGarch(yt2, n_test = 250)
cor_mat <- cor(cbind(model1@etat, model2@etat))
```

If no model refitting is conducted, then one may simply produce the one-step mean and volatility forecasts over the test period as for the univariate case and then combine the forecasts.

```
fc1 <- model1 %>% predict_roll(refit_after = NULL)
fc2 <- model2 %>% predict_roll(refit_after = NULL)
mean_fc <- (cbind(fc1@cmeans, fc2@cmeans) %*% w) %>% c()
vol_fc <- vapply(1:length(mean_fc),
  FUN = function(.x, fc_mat, w, cor_mat) {
    Dt <- diag(fc_mat[.x, ])
    sqrt(t(w) %*% Dt %*% cor_mat %*% Dt %*% w)
  }, fc_mat = cbind(as.numeric(fc1@sigt), as.numeric(fc2@sigt)),
  w = w, cor_mat = cor_mat, FUN.VALUE = numeric(1))
```

If at least one of the conditional distributions considered for the univariate series is different from normal, then the conditional distribution of the portfolio returns is unknown. In such a case, one may apply `find_dist()` with simultaneously fixing the mean and the standard deviation of the distribution to zero and one, respectively, via the settings `fix_mean = 0` and `fix_sdev = 1`. This will identify the best distribution for the portfolio residuals. If all conditional distributions for the univariate series are normal, the portfolio returns will also have a conditional normal distribution in theory. Then one may apply `norm_est()` with `fix_mean = 0` and `fix_sdev = 1` directly without the need to identify the type of distribution. In both cases, this approach allows us to produce one-step VaR and ES forecasts from the forecasted means, the forecasted volatilities and the identified residual distribution. In the following, since we considered non-normal conditional distributions, the workflow under usage of `find_dist()` is demonstrated.

```
yt_pf <- (cbind(yt1, yt2) %*% w) %>% c()  # Portfolio returns
yt_pf_train <- head(yt_pf, -250) # Training portion
yt_pf_test <- tail(yt_pf, 250)   # Test portion
cmeans_train <- (cbind(model1@cmeans, model2@cmeans) %*% w) %>% c()
```

```
vol_train <- vapply(1:length(cmeans_train),
  FUN = function(.x, vol_mat, w, cor_mat) {
    Dt <- diag(vol_mat[.x, ])
    sqrt(t(w) %*% Dt %*% cor_mat %*% Dt %*% w)
  },
  vol_mat = cbind(as.numeric(model1@sigt), as.numeric(model2@sigt)),
  w = w, cor_mat = cor_mat, FUN.VALUE = numeric(1))

resid_pf <- (yt_pf_train - cmeans_train) / vol_train
risk <- resid_pf %>% find_dist(fix_mean = 0, fix_sdev = 1) %>%
  measure_risk(test_obs = yt_pf_test, sigt = vol_fc,
               cmeans = mean_fc)
```

`cmeans_train` and `vol_train` are the in-sample fitted conditional means and standard deviations, respectively. Graphics can again be produced from the output of `measure_risk()`, which can have an automatically formatted x-axis, if `resid_pf`, `rt_pf_test`, `vol_fc` and `mean_fc` are formatted as objects of class `"zoo"` after their generation, which is not done in the previous example. Furthermore, backtesting functions from the `fEGarch` package can be applied to the object `risk`.

Optionally, one may also forecast the means and volatilities of the univariate series with refitting after a certain amount of test observations. The element `@model` in the output `predict_roll()` is then a list that contains the various fitted model objects. For each of the training periods, one may then estimate the correlation matrix between the corresponding residual series separately and fit a distribution to the corresponding portfolio residuals for each training period. From this information, one may then construct VaR and ES forecasts according to each test period subset individually using the functions `VaR_calc()` and `ES_calc()` from `fEGarch` to obtain $\widehat{\mathrm{VaR}}_{\alpha}^{\eta}$ and $\widehat{\mathrm{ES}}_{\alpha}^{\eta}$ as the estimated VaR and and ES, respectively, for the innovations and the Equations (61) and (62) from Appendix C.4 to then derive the total VaR and ES forecasts. VaR and ES forecasts from the various test subperiods can then be combined. Nonetheless, a backtesting function from `fEGarch` is not directly applicable under such a refitting scheme in the context of MGARCH models. Therefore, test quantities need to be aggregated and checked manually.

Analogous ideas as presented in this section can also be applied under a DCC-GARCH model; however, since `fEGarch` does not allow for the estimation of the conditional correlations, users may need to fit the DCC-GARCH, both its univariate and its multivariate steps, and to produce $\hat{\mu}_t^{\mathrm{g}}$ and $\hat{\sigma}_t^{\mathrm{g}}$, $t = 1, \ldots, n+n_{\mathrm{te}}$, as portfolio quantities using other software like `rmgarch`, which excludes many of the univariate GARCH-type models from `fEGarch`. Nonetheless, `fEGarch` can be considered subsequently to fit a distribution to the portfolio residuals, produce corresponding VaR and ES forecasts and to backtest these quantities.

5.4 Backtesting other models than GARCH

As described throughout this article, the core models to fit and backtest with `fEGarch` are those of the GARCH-family. However, recent developments, for example in machine and deep learning, have given rise to new models that can be used to forecast volatility on the stock market that are mostly unrelated to the GARCH-class. A representative is the long short-term memory recurrent neural network (LSTM-RNN, Hochreiter and Schmidhuber, 1997). Using the same idea as in the previous subsection, users may produce and backtest one-step VaR and ES forecasts for such models under consideration of `fEGarch` given externally obtained mean and volatility forecasts as well as in-sample residuals. Let `yt_in`, `yt_out`, `mu_in`, `mu_out`, `sig_in` and `sig_out` be the in-sample training observations, the test observations, the estimated in-sample conditional means, the forecasted out-of-sample conditional means, the fitted in-sample conditional standard deviations and the forecasted out-of-sample conditional standard deviations, respectively, and all as numeric vectors. Then a typical workflow would read as follows after attaching the `fEGarch` package.

```
eta_in <- (yt_in - mu_in) / sig_in
eta_in %>% find_dist(fix_mean = 0, fix_sdev = 1) %>%
  measure_risk(test_obs = yt_out, sigt = sig_out,
               cmeans = mu_out) %>%
  trafflight_test()
```

`trafflight_test()` at the end of the previous pipe can be replaced by any other backtesting or loss function in `fEGarch`, for example by `backtest_suite()` and `loss_functions()`. Moreover, instead of numeric vectors, also objects of class `"zoo"` and `"ts"` are supported and the output of `measure_risk()` can once again be visualized easily through a designated `plot()` method. The method was successfully implemented for example by Li (2026) to backtest various LSTM-RNN. Notwithstanding the simple workflow, the approach has one key disadvantage: `fEGarch` cannot automate a refitting procedure for those models. Therefore, the backtesting of such models is only available without refitting through `fEGarch`. If refitting is wished, the provided functions need to be combined into a more complex workflow by the user and the results need to be aggregated correctly.

5.5 Dual long-memory models for macroeconomic time series

A dual long-memory model combines a (parametric) long-memory model in the mean, i.e. a FARIMA model, with a (parametric) long-memory model in the variance, i.e. a long-memory GARCH-type model. These models are therefore a special case of the model extension discussed in Section 2.2.2. While not commonly used for analyzing return data, these models can be particularly useful for selected other time series, for example for some macroeconomic time series. In the following, we consider a member of this class of models, in detail a FARIMA$(0, d, 1)$-FIMLog-GARCH$(1, d, 1)$ model with conditional skew-ALD, to estimate the conditional mean and simultaneously the conditional standard deviation in the monthly inflation rate of the UK from January 1956 to December 2000. The data is included in the `fEGarch` package and was originally downloaded from the databank of the Federal Reserve Bank of St. Louis.

Once again, the volatility model together with the conditional distribution can be specified with `fimloggarch_spec(cond_dist = "sald")`. Nonetheless, the model in the mean now also needs to be specified, here with `mean_spec(orders = c(0, 1), long_memo = TRUE)`, where `orders = c(0, 1)` sets the AR- and MA-orders, whereas `long_memo = TRUE` allows for the consideration of the fractional differencing parameter in the mean. Both specifications need to be passed to `fEGarch()` for fitting, where the mean specification result of `mean_spec()` is passed to the argument `meanspec`.

```
library(fEGarch)
library(zoo)
xt <- UKinflation     # data in the fEGarch package
volspec <- fimloggarch_spec(cond_dist = "sald")
meanspec <- mean_spec(orders = c(0, 1), long_memo = TRUE)
dual_lm_model <- volspec %>%
  fEGarch(xt, meanspec = meanspec)
dual_lm_model
TS <- cbind(xt, dual_lm_model@cmeans, dual_lm_model@sigt)
plot.zoo(TS, screens = c(1, 1, 2), col = c("grey64", "red", "black"),
```

```
xlab = "Year", ylab = c("Inflation", "Cond. volatility"),
main = paste0("UK-inflation rate with fitted conditional\n",
              "means and standard deviations"))
```

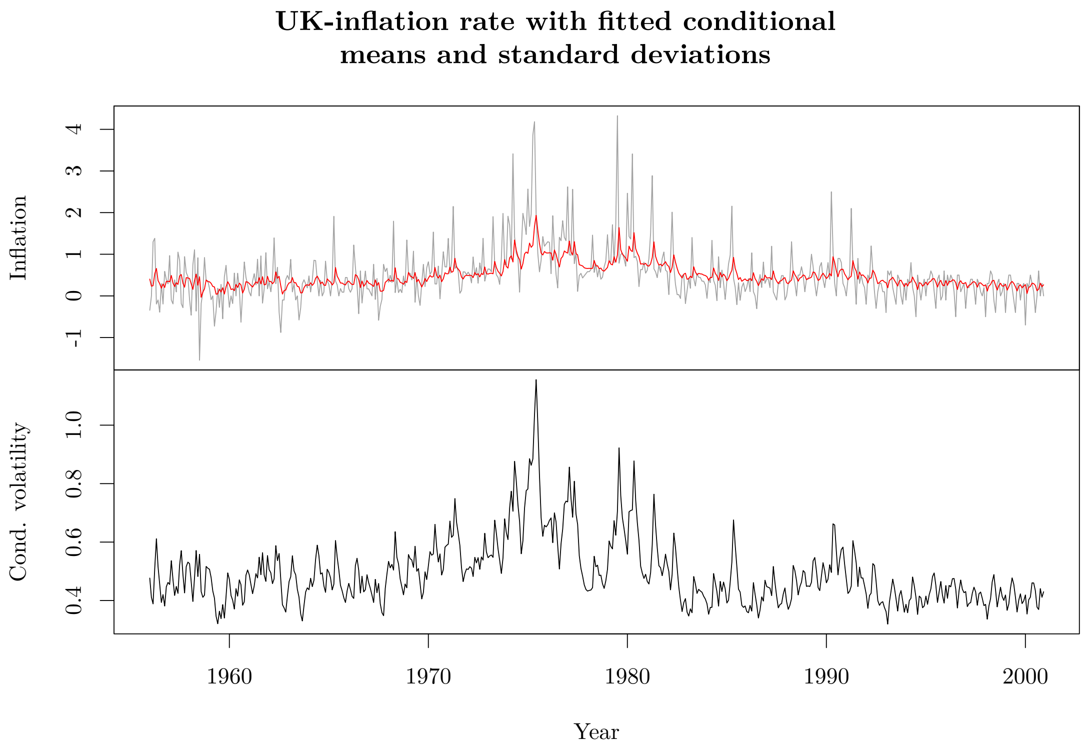

Figure 6: Results of a fitted FARIMA$(0, d, 1)$-FIMLog-GARCH$(1, d, 1)$ model with conditional skew-ALD.

Detailed estimation results are omitted to save space but can be investigated easily by running the previous code. The fitting results are observable in Figure 6, where in the first subplot the inflation data is displayed together with the estimated conditional means. The means follow the rise and fall of the inflation rate in a plausible way, as can for example be seen around the years 1975 to 1980, where the estimated conditional means capture the high-level period of the series. The estimated conditional volatilities in the second subplot of Figure 6 similarly reflect the change in the variation of the inflation series well. During the 1975 to about 1980, the volatility increases notably, at all other times it is smaller.

6 Simulation study

In the following, the validity of the (conditional) QML estimator implemented in the `fEGarch` package is confirmed through a simulation study. We draw 1000 samples each from an EGARCH, a Log-GARCH, an MEGARCH and an MLog-GARCH process - each of orders $p = q = 1$ - and also their long-memory variants each under a conditional normal distribution, a conditional t-distribution, a conditional GED and a conditional ALD as well as their skewed variants. The corresponding model is then fitted to the data in order to collect the parameter estimates. This procedure is repeated under numbers of observations $n_1 = 750$, $n_2 = 1500$ and $n_3 = 3000$, representing roughly three, six and twelve years of returns for a stock observed on each work day. These specifications lead to 192 different settings. For models of Type I, we consider as true parameters $\mu = 1 \times 10^{-4}$, $\omega_\sigma = -8.5$, $\phi_1 = 0.65$, $\kappa = -0.25$ and $\gamma = 0.35$ with $d = 0.3$ for long-memory models and with $\nu = 6$, $\beta = 1.4$, $P = 2$ and $s = 0.9$ as the parameters for conditional distributions aside from the pure normal distribution. On the other hand, for models of Type II we consider $\phi_1 = 0.85$ and $\psi_1 = -0.8$ in the short-memory case and $\phi_1 = 0.21$ and $\psi_1 = -0.51$ in the long-memory case, omit κ and γ, and keep the remaining parameter settings. For each individual series, we simulate $n_\mathrm{b} + n_i$, $i \in \{1, 2, 3\}$, observations with $n_\mathrm{b} = 10000$ and only keep the last n_i observations for estimation.

An example path and the corresponding volatility series of a FIMLog-GARCH$(1, d, 1)$-process with the previously given specifications are shown in Figure 7. The simulated series shows signs of volatility clustering, for example at around time point 2900. However generally, the series seems to be stable and the conditional volatilities appear to revert to a volatility level of at around 0.1 to 0.15.

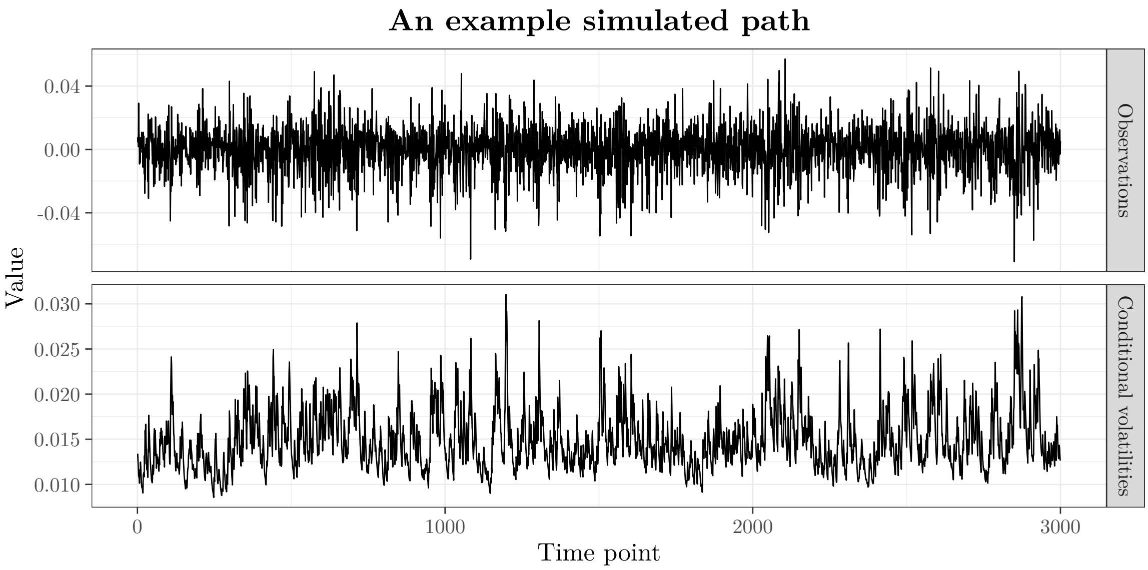

Figure 7: An example path simulated from a FIMLog-GARCH$(1, d, 1)$-process with conditional skew-ALD as considered in the simulation study.

For each of the 192 settings in the simulation, the sample mean, sample standard deviation and the sample mean squared error (MSE) are computed for each of the parameters and displayed in Tables 7 to 14 in Appendix A. In contrast, Figures 8 to 15 in Appendix B show kernel density plots of all coefficient estimators under all settings. Note that Table 6 in Appendix A summarizes potential issues in the fitting process for each setting, including the number of errors, where estimation could not be conducted at all, and convergence problems, where the fitting functions ran without error but convergence of the log-likelihood function was not reached. Errors occurred only for Log-GARCH and FILog-GARCH models, convergence issues occurred mostly for a FILog-GARCH setting. For other short-memory models and even more so for other long-memory models, convergence issues happened rarely for smaller numbers of observations and they reduced or disappeared for n_3. For better comparison, Tables 7 to 14 in Appendix A and Figures 8 to 15 in Appendix B do not include results from series, where either errors or convergence issues appeared, and are therefore in some cases based on marginally fewer sample paths.

From the tables in Appendix A, we can observe that all estimators show signs of convergence toward the true underlying parameters under all settings, as the MSE-values are decreasing clearly for an increase in the number of observations. In case of the FILog-GARCH, the sample MSE can sometimes increase for $\hat{\mu}$ when going from n_1 to n_2, however then followed by a decrease in MSE when going from n_2 to n_3. On the other hand, for a Log-GARCH, the MSE under n_1 can sometimes decrease notably when going from n_1 to n_2. Both of these features can be assigned to the instability of the (FI)Log-GARCH as the numerical solver may sometimes face problems when the logarithm of values is approximately zero in the computation of the conditional standard deviation, which, from a purely numerical perspective, causes problems with calculating the log-likelihood. (FI)EGARCH, (FI)MEGARCH and (FI)MLog-GARCH do not face such problems and show plausible decreases in MSE throughout.

The figures in Appendix B show the simulated sample distributions of the parameter estimators under all settings. The convergence for an increase in the number of observations can also be observed from the densities getting narrower and centered more clearly around the true parameter values for increasing n. Moreover, for larger number of observations, the densities of the continuous estimators seem to become more bell-shaped, supporting the idea of asymptotic normality of these estimators (at least under the specifications of the simulation). For Type-I long-memory models, the lower bound set for $\hat{d}$ is zero, which, due to the large variance of the estimator, leads to a smaller peak of the distribution near zero for smaller number of observations. As the number of observations is increased and the density becomes narrower, the density peak near zero disappears slowly. Under the conditional ALD or skew-ALD, the discrete $\hat{P}$ also clearly converges to the true value P. For Log-GARCH and FILog-GARCH models, the solver may sometimes be stuck on the starting values, which can be seen for example in small additional peaks in the estimator distributions, in particular regarding $\hat{\mu}$, $\hat{\phi}_1$, $\hat{\psi}_1$ or $\hat{\beta}$. For these two models, we can also often observe a slower convergence for some estimators, for example with respect to $\hat{d}$ or $\hat{P}$. A bell-shaped distribution for a large number of observations can be observed for all continuous estimators with the exception of $\hat{\mu}$ under a FILog-GARCH, where the distribution shows a slower decay toward the tails than in a normal case.

Overall, the QMLE in the context of the Type-I EGF models seems to show all desirable characteristics, namely consistency and asymptotic normality, following our simulation study, whereas the QMLE regarding the Type-II EGF models shares these properties mostly but is generally more unstable.

7 Comparison to related statistical software

7.1 Practical comparison to EGARCH in `rugarch`

One of the most popular R packages for GARCH-modelling is `rugarch` (Galanos, 2024), which contains options for fitting standard EGARCH models under various conditional distributions. It makes use of an EGARCH representation similar to (5) to state its estimation output, i.e. with intercept term ω instead of ω_σ. In the following in Table 4, we report the estimated EGARCH$(1,1)$-coefficients - rounded to four decimals - under a conditional normal distribution (norm), a conditional t-distribution (std), a conditional GED (ged) and their skewed variants (snorm / sstd / sged) for the daily log-returns of the S&P 500 (SP500), the DAX and the FTSE 100 Index (FTSE) from the beginning of January 2000 to the end of August 2025. The data was downloaded from Yahoo Finance. For better comparison, the `rugarch` estimates of ω are replaced by those of ω_σ after the estimation following the relationship between those two quantities.

As observable in Table 4, the estimates are almost identical throughout. Aside from estimates for ω_σ and ν, for which results sometimes differ slightly at the third decimal, differences in other parameter estimates are either non-observable for the rounded values or at most at the fourth decimal. These marginal differences can be

Table 4: Comparison of EGARCH$(1, 1)$ estimates for daily S&P 500, DAX and FTSE 100 Index log-returns for various conditional distributions between `rugarch` and `fEGarch`.

		Estimate											
		norm		std		ged		snorm		sstd		sged	
Series	Parameter	`rugarch`	`fEGarch`	`rugarch`	`fEGarch`	`rugarch`	`fEGarch`	`rugarch`	`fEGarch`	`rugarch`	`fEGarch`	`rugarch`	`fEGarch`
SP500	μ	0.0002	0.0002	0.0005	0.0005	0.0005	0.0005	0.0001	0.0001	0.0002	0.0002	0.0001	0.0001
	ω_σ	-9.1489	-9.1482	-9.6185	-9.6181	-9.5852	-9.5848	-9.1059	-9.1054	-9.3729	-9.3724	-9.3256	-9.3253
	κ	-0.1441	-0.1441	-0.1669	-0.1669	-0.1552	-0.1552	-0.1463	-0.1463	-0.1704	-0.1704	-0.1609	-0.1609
	ϕ_1	0.9714	0.9715	0.9787	0.9787	0.9764	0.9764	0.9727	0.9728	0.9757	0.9757	0.9737	0.9737
	γ	0.1547	0.1547	0.1402	0.1402	0.1486	0.1486	0.1462	0.1462	0.1415	0.1415	0.1473	0.1473
	ν	—	—	6.8230	6.8220	—	—	—	—	7.5329	7.5330	—	—
	β	—	—	—	—	1.3690	1.3689	—	—	—	—	1.4215	1.4215
	s	—	—	—	—	—	—	0.8251	0.8251	0.8581	0.8581	0.8588	0.8588
DAX	μ	0.0002	0.0002	0.0005	0.0005	0.0005	0.0005	0.0001	0.0001	0.0002	0.0002	0.0002	0.0002
	ω_σ	-8.7609	-8.7603	-9.0975	-9.0966	-9.1125	-9.1118	-8.7169	-8.7170	-8.9278	-8.9270	-8.9330	-8.9323
	κ	-0.1233	-0.1233	-0.1344	-0.1344	-0.1288	-0.1288	-0.1241	-0.1241	-0.1367	-0.1367	-0.1321	-0.1321
	ϕ_1	0.9747	0.9747	0.9801	0.9801	0.9782	0.9782	0.9749	0.9749	0.9783	0.9783	0.9766	0.9766
	γ	0.1352	0.1352	0.1374	0.1375	0.1369	0.1369	0.1330	0.1331	0.1374	0.1375	0.1364	0.1364
	ν	—	—	7.7551	7.7534	—	—	—	—	8.2126	8.2099	—	—
	β	—	—	—	—	1.4446	1.4447	—	—	—	—	1.4608	1.4607
	s	—	—	—	—	—	—	0.8718	0.8718	0.8812	0.8812	0.8837	0.8836
FTSE	μ	-0.0000	-0.0000	0.0001	0.0001	0.0001	0.0001	-0.0001	-0.0001	-0.0001	-0.0001	-0.0001	-0.0001
	ω_σ	-9.2517	-9.2515	-9.5807	-9.5803	-9.5866	-9.5864	-9.2090	-9.2089	-9.3982	-9.3980	-9.3911	-9.3909
	κ	-0.1346	-0.1346	-0.1415	-0.1415	-0.1369	-0.1369	-0.1351	-0.1351	-0.1440	-0.1440	-0.1411	-0.1411
	ϕ_1	0.9772	0.9773	0.9808	0.9808	0.9795	0.9795	0.9782	0.9782	0.9796	0.9797	0.9784	0.9784
	γ	0.1284	0.1284	0.1277	0.1277	0.1286	0.1286	0.1211	0.1211	0.1255	0.1254	0.1250	0.1250
	ν	—	—	8.6478	8.6487	—	—	—	—	9.2005	9.2025	—	—
	β	—	—	—	—	1.4996	1.4996	—	—	—	—	1.5154	1.5154
	s	—	—	—	—	—	—	0.8540	0.8540	0.8645	0.8644	0.8628	0.8628

explained through potentially different settings regarding starting values of parameters in the numerical solving procedure and related reasons. Overall, the differences are, however, clearly negligible and show that `fEGarch` agrees with `rugarch` regarding the estimation of EGARCH$(1, 1)$-models.

Each model was fitted ten times and the computation times were recorded using the package `bench` (Hester and Vaughan, 2025). In Table 5, the median computation times - in seconds and rounded to two decimals - are displayed for all settings. In almost all cases, `fEGarch` beats `rugarch` in computation time, however, it has to be noted that `rugarch` does not only fit models but also directly runs further tests on the estimation results, which is not done in the fitting function of `fEGarch`. Given Table 5, `fEGarch` performs on average only slightly better for conditional normal and skew-normal distributions leading to an average reduction in computation time of roughly 22.49% under a conditional normal distribution and of about 14.03% under a skew-normal distribution. For all the other more complicated distributions, the average time reductions lie in between roughly 48.65% and 59.59% indicating overall at least a comparable computational efficiency for EGARCH$(1, 1)$-models of `fEGarch` to `rugarch` that is even in some cases clearly improved.

Additionally, a direct comparison of package functionalities shows clear advantages of `fEGarch`. `fEGarch`, unlike `rugarch`, offers explicit support for a huge selection of

Table 5: Comparison of median computation times for EGARCH(1, 1) estimation under various conditional distributions between `rugarch` and `fEGarch` for daily S&P500, DAX and FTSE 100 Index log-returns.

	Median computation time (in seconds)											
	norm		std		ged		snorm		sstd		sged	
Series	rugarch	fEGarch	rugarch	fEGarch	rugarch	fEGarch	rugarch	fEGarch	rugarch	fEGarch	rugarch	fEGarch
SP500	1.38	0.78	2.21	0.97	4.00	1.50	2.65	1.97	4.39	1.49	5.45	2.24
DAX	0.71	0.68	1.71	1.02	2.42	1.11	1.17	1.18	3.25	1.55	6.31	2.73
FTSE	0.91	0.73	1.98	1.00	3.01	1.26	1.85	1.53	4.58	1.91	8.22	3.03

long-memory GARCH-type models and a fully integrated VaR/ES backtesting pipeline. The backtesting pipeline of `fEGarch` is highly flexible and can therefore also be applied easily to other GARCH and non-GARCH volatility models. Further advantages include the design of `fEGarch` to work with pipe operators, so that typical application workflows are simple to implement line by line and hence easy to maintain.

7.2 Comparison to other software

The R package `ufRisk` (Feng et al., 2023c), available on CRAN, gives access to functionalities related to those of `fEGarch`. First and foremost, `ufRisk` can be viewed mainly as an extension to `rugarch` that allows to fit GARCH, APARCH, EGARCH and FIGARCH models via `rugarch` internally. However, the conditional distributions in `ufRisk` are restricted to those of a conditional normal distribution and a conditional t-distribution. In contrast, `fEGarch` uses its own coded estimation procedure but ultimately also relies, like the default in `rugarch`, on the `Rsolnp` package for numerical optimization. Nonetheless, `fEGarch` gives access to six further conditional distributions, where the conditional ALD and its skewed variant are novelties in GARCH fitting, and to further GARCH-type models, where in particular the long-memory variants (aside from FIGARCH) are a new addition to the R cosmos. Additionally, `ufRisk` provides access to the estimation of Log-GARCH and FILog-GARCH models through QMLE based on their ARMA or FARIMA representation, respectively, and then applying corresponding retransformations (see for example Feng et al., 2020a) to obtain the conditional standard deviations, whereas `fEGarch` implements Log-GARCH and FILog-GARCH models by QMLE according to the log-likelihood of the observed (return) series itself and under consideration of the definition of $\ln\left(\sigma_t^2\right)$. Semiparametric extensions for all of the GARCH-type models available in `ufRisk` are also implemented, as in `fEGarch`, via nonparametric scale estimation by the `smoots` and `esemifar` R packages. However, only `fEGarch` among the two packages also allows for estimating parametric models with simultaneous modelling in the mean and the variance. For a test period, one-step-ahead rolling point forecasts for VaR and ES can be obtained (without refitting) for all models in `ufRisk`, while the package also gives access to common VaR and ES backtesting procedures discussed in Section 3 in this paper. `fEGarch` provides the same functionalities in this regard and extends them by allowing for model refitting over the test period. Overall, `fEGarch` can, in contrast to `ufRisk`, be seen as independent of `rugarch` and as a package that provides access to further important long-memory GARCH-type models and some of their extensions as well as to more selectable conditional distributions.

OxMetrics (Laurent, 2018), in particular its *G@RCH* module, is another popular statistical software, outside of the R universe, for fitting and forecasting a selection of GARCH-type models. Example applications of the software in the context of purely parametric and semiparametric GARCH-type models can, for example, be found in Feng et al. (2020a) and Letmathe et al. (2022). A major drawback is that *OxMetrics* is a product that needs to be licensed against payment, while `fEGarch` is freely available online on CRAN. *OxMetrics* gives access to standard GARCH, EGARCH, GJR-GARCH, APARCH, IGARCH (integrated GARCH, Engle and Bollerslev, 1986), FIGARCH, HYGARCH (hyperbolic GARCH, Davidson, 2004), FIEGARCH, FIAPARCH, and to other related volatility models. In particular, a Spline-GARCH (Engle and Rangel, 2008) as a semiparametric model extension based on spline estimation, in comparison to local polynomial regression in `fEGarch`, is also available. Hence, *OxMetrics* also allows for fitting a variety GARCH-type models, including a broad selection of fractionally integrated models. While a Spline-GARCH, IGARCH and HYGARCH, are not yet available in `fEGarch`, the other aforementioned GARCH-type models are. As for conditional distributions, *OxMetrics* only has a conditional normal distribution, a conditional t-distribution, a conditional GED and a conditional skewed t-distribution as options. Therefore, it is clearly limited in this regard in comparison to `fEGarch`. On a further note, *OxMetrics* can be used to fit GARCH-type models with simultaneous modelling of the mean through ARMA and FARIMA models like `fEGarch`. Another similarity is that one-step rolling forecasts of the VaR can be obtained using both *OxMetrics* and `fEGarch`, where both allow for the re-estimation of the considered model after a selectable amount of observations. Among the two, ES forecasting is only available in `fEGarch`. Another disadvantage is that VaR backtesting in *OxMetrics* is restricted to the unconditional coverage test by Kupiec (1995) and the average exceedance of the VaR, if the return falls below the VaR, is stated. Here, `fEGarch` provides much more functionality with a variety of unconditional and conditional VaR backtests according to Christoffersen (1998), loss functions, and traffic light test for VaR and ES. Thus, `fEGarch` is a full GARCH-type model estimation, VaR and ES forecasting and backtesting software that supports the whole modelling cycle. Notwithstanding this position of `fEGarch`, a clear advantage of *OxMetrics* over `fEGarch` is that common multivariate GARCH-type models are already coded into the program, so that it is easy to apply for example CCC-GARCH and DCC-GARCH models on portfolio data. Nonetheless, a simple program for CCC-GARCH can be easily produced in R as was shown in Section 5.3, where `fEGarch` can be employed to fit models to the univariate series, and `fEGarch` allows for backtesting externally estimated DCC-GARCH models.

8 Concluding remarks

With this article, we have introduced the R package `fEGarch` for modelling, forecasting and backtesting of short- and long-memory GARCH-type models, most notably the models of the EGF. We have summarized well-known facts about the EGF and extended them to semiparametric models and to models with simultaneous modelling in the mean. Furthermore, the estimation algorithms for all discussed EGF models were described in detail under all eight different conditional distributions, including the new ALD and its skewed variant, which allow for the return series of (FI)EGARCH and (FI)MEGARCH processes to have finite moments until an arbitrary order. Moreover, corresponding forecasting and backtesting approaches were presented. In the application section we showed that the `fEGarch` package can be used to fit and backtest a variety of models easily with a simple and consistent code framework. In addition, we could show that the models from the EGF fulfill regulatory requirements for VaR and ES for at least the two selected return series in this article. The simulation study showed that the QMLE for the EGF models in `fEGarch` are consistent and in the context of the Type-I EGF models asymptotically normally distributed (for real-valued parameter estimators) following the conducted study. For Type-II EGF models, asymptotic normality was shown to be fulfilled for most parameter estimators through the simulation. Moreover, computation speed of `fEGarch` is comparable to, or better than, that of `rugarch` for standard EGARCH models under a conditional normal distribution, a conditional t-distribution, a conditional GED and their skewed variants, making the package an alternative to other established software like `rugarch`.

The `fEGarch` package can, however, be improved in the future: the problem of unbiased multi-step forecasts of conditional standard deviations has not yet been solved in the context of the EGF, and also the implementation of so-called robust standard errors for QML estimators would be a useful addition. Moreover, the estimation algorithms could be extended to also introduce the parameters p_{asy}, p_{mag}, M_{asy} and M_{mag} into the set of unknown parameters to be estimated.

Computational details

The numerical and visual results in this paper were obtained using R 4.5.2 with the `fEGarch` 1.0.6 package. R itself and all packages used are available from CRAN at https://CRAN.R-project.org/.

Acknowledgments: We thank the CRA Network for their support in making the `fEGarch` package available on their platform. Moreover, we would also like to thank Dr. Sebastian Letmathe and Ms. Shujie Li for helpful discussions.

References

Abad, P., Muela, S. B., and Martín, C. L. (2015). The Role of the Loss Function in Value-at-Risk Comparisons. *The Journal of Risk Model Validation*, *9*(1), 1–19. DOI: 10.21314/JRMV.2015.132.

Acerbi, C., and Tasche, D. (2002a). Expected Shortfall: A Natural Coherent Alternative to Value at Risk. *Economic Notes*, *31*(2), 379–388. DOI: 10.1111/1468-0300.00091.

Acerbi, C., and Tasche, D. (2002b). On the Coherence of Expected Shortfall. *Journal of Banking & Finance*, *26*(7), 1487–1503. DOI: 10.1016/S0378-4266(02)00283-2.

Ayensu, O. K., Feng, Y., and Schulz, D. (2026). *Well-Known and Recent Long-Memory GARCH Models and Their Semiparametric Extensions*. Unpublished manuscript at Paderborn University.

Bache, S. M., and Wickham, H. (2022). *magrittr: A Forward-Pipe Operator for R*. DOI: 10.32614/CRAN.package.magrittr. R package version 2.0.3.

Baillie, R. T., Bollerslev, T., and Mikkelsen, H. O. (1996). Fractionally Integrated Generalized Autoregressive Conditional Heteroskedasticity. *Journal of Econometrics*, *74*(1), 3–30. DOI: 10.1016/S0304-4076(95)01749-6.

BCBS (1996a). *Amendment to the Capital Accord to Incorporate Market Risks*. Bank for International Settlements (BIS). URL: https://www.bis.org/publ/bcbsc222.pdf. Updated April 1998.

BCBS (1996b). *Supervisory Framework for the Use of "Backtesting" in Conjunction with the Internal Models Approach to Market Risk Capital Requirements*. Bank for International Settlements (BIS). URL: https://www.bis.org/publ/bcbs22.pdf.

BCBS (2016). *Minimum Capital Requirements for Market Risk*. Bank for International Settlements (BIS). URL: https://www.bis.org/bcbs/publ/d352.pdf. Fundamental Review of the Trading Book (FRTB) — standards (revised version published Jan 2019).

BCBS (2019). *Explanatory Note on the Minimum Capital Requirements for Market Risk*. Bank for International Settlements (BIS). URL: https://www.bis.org/bcbs/publ/d457_note.pdf.

Beran, J., and Feng, Y. (2002). SEMIFAR Models—A Semiparametric Approach to Modelling Trends, Long-Range Dependence and Nonstationarity. *Computational Statistics & Data Analysis*, *40*(2), 393–419. DOI: 10.1016/S0167-9473(02)00007-5.

Bollerslev, T. (1986). Generalized Autoregressive Conditional Heteroskedasticity. *Journal of Econometrics*, *31*(3), 307–327. DOI: 10.1016/0304-4076(86)90063-1.

Bollerslev, T. (1990). Modelling the Coherence in Short-Run Nominal Exchange Rates:

A Multivariate Generalized ARCH Model. *The Review of Economics and Statistics*, *72*(3), 498–505. DOI: 10.2307/2109358.

Bollerslev, T., Engle, R. F., and Wooldridge, J. M. (1988). Capital Asset Pricing Model with Time-Varying Covariances. *Journal of Political Economy*, *96*(1), 116–131. DOI: 10.1086/261527.

Bollerslev, T., and Mikkelsen, H. O. (1996). Modeling and Pricing Long Memory in Stock Market Volatility. *Journal of Econometrics*, *73*(1), 151–184. DOI: 10.1016/0304-4076(95)01736-4.

Box, G. E. P., Jenkins, G. M., and Reinsel, G. C. (2008). *Time Series Analysis: Forecasting and Control* (4th ed.). John Wiley & Sons, Hoboken, NJ.

Cappiello, L., Engle, R. F., and Sheppard, K. (2006). Asymmetric Dynamics in the Correlations of Global Equity and Bond Returns. *Journal of Financial Econometrics*, *4*(4), 537–572. DOI: 10.1093/jjfinec/nbl005.

Christoffersen, P. F. (1998). Evaluating Interval Forecasts. *International Economic Review*, *39*(4), 841–862. DOI: 10.2307/2527341.

Costanzino, N., and Curran, M. (2018). A Simple Traffic Light Approach to Backtesting Expected Shortfall. *Risks*, *6*(1). DOI: 10.3390/risks6010002.

Davidson, J. E. (2004). Moment and Memory Properties of Linear Conditional Heteroscedasticity Models, and a New Model. *Journal of Business & Economic Statistics*, *22*(1), 16–29. DOI: 10.1198/073500103288619359.

Ding, Z., Granger, C. W., and Engle, R. F. (1993). A Long Memory Property of Stock Market Returns and a New Model. *Journal of Empirical Finance*, *1*(1), 83–106. DOI: 10.1016/0927-5398(93)90006-D.

Eddelbuettel, D., and François, R. (2011). Rcpp: Seamless R and C++ Integration. *Journal of Statistical Software*, *40*(8), 1–18. DOI: 10.18637/jss.v040.i08.

Eddelbuettel, D., and Sanderson, C. (2014). RcppArmadillo: Accelerating R with High-Performance C++ Linear Algebra. *Computational Statistics and Data Analysis*, *71*, 1054–1063. DOI: 10.1016/j.csda.2013.02.005.

Engle, R. F. (1982). Autoregressive Conditional Heteroskedasticity with Estimates of the Variance of United Kingdom Inflation. *Econometrica*, *50*(4), 987–1007. DOI: 10.2307/1912773.

Engle, R. F. (2002). Dynamic Conditional Correlation: A Simple Class of Multivariate GARCH Models. *Journal of Business & Economic Statistics*, *20*(3), 339–350. DOI: 10.1198/073500102288618487.

Engle, R. F., and Bollerslev, T. (1986). Modelling the Persistence of Conditional

Variances. *Econometric Reviews*, *5*(1), 1–50. DOI: 10.1080/07474938608800095.

Engle, R. F., and Kroner, K. F. (1995). Multivariate Simultaneous Generalized ARCH. *Econometric Theory*, *11*(1), 122–150. DOI: 10.1017/S0266466600009063.

Engle, R. F., and Rangel, J. G. (2008). The Spline-GARCH Model for Low-Frequency Volatility and Its Global Macroeconomic Causes. *The Review of Financial Studies*, *21*(3), 1187–1222. DOI: 10.1093/rfs/hhn004.

European Commission (2025). *Commission Proposes to Postpone by One Additional Year the Market Risk Prudential Requirements under Basel III*. Press release. URL: https://ec.europa.eu/commission/presscorner/api/files/document/print/en/ip_25_1478/IP_25_1478_EN.pdf.

Fan, J., and Gijbels, I. (1996). *Local Polynomial Modelling and Its Applications*. Chapman & Hall / CRC, New York, NY. DOI: 10.1201/9780203748725.

Feng, Y. (2004a). *Non- and Semiparametric Regression with Fractional Time Series Errors: Theory and Applications to Financial Data* [Habilitation thesis]. University of Konstanz.

Feng, Y. (2004b). Simultaneously Modeling Conditional Heteroskedasticity and Scale Change. *Econometric Theory*, *20*(3), 563–596. DOI: 10.1017/S0266466604203061.

Feng, Y., Beran, J., Letmathe, S., and Ghosh, S. (2020a). *Fractionally Integrated Log-GARCH with Application to Value at Risk and Expected Shortfall* (No. 137; CIE Working Paper Series). Center for International Economics, Universität Paderborn. URL: https://groups.uni-paderborn.de/wp-wiwi/RePEc/pdf/ciepap/WP137.pdf.

Feng, Y., Beran, J., Letmathe, S., and Schulz, D. (2024). *esemifar: Smoothing Long-Memory Time Series*. DOI: 10.32614/CRAN.package.esemifar. R package version 2.0.1.

Feng, Y., Gries, T., and Fritz, M. (2020b). Data-Driven Local Polynomial for the Trend and Its Derivatives in Economic Time Series. *Journal of Nonparametric Statistics*, *32*(2), 510–533. DOI: 10.1080/10485252.2020.1759598.

Feng, Y., Gries, T., and Letmathe, S. (2023a). FIEGARCH, Modulus Asymmetric FILog-GARCH and Trend-Stationary Dual Long Memory Time Series. *CIE Working Papers, Paderborn University, 156*. URL: https://groups.uni-paderborn.de/wp-wiwi/RePEc/pdf/ciepap/WP156.pdf.

Feng, Y., Gries, T., Letmathe, S., and Schulz, D. (2022). The smoots Package in R for Semiparametric Modeling of Trend Stationary Time Series. *The R Journal*, *14*(1), 182–195. DOI: 10.32614/rj-2022-017.

Feng, Y., Letmathe, S., and Schulz, D. (2023b). *smoots: Nonparametric Estimation* *of the Trend and Its Derivatives in TS*. DOI: 10.32614/CRAN.package.smoots. R package version 1.1.4.

Feng, Y., and Peitz, C. (2025). Semiparametric GARCH Models for Value-at-Risk and Expected Shortfall: An Object-Driven Procedure. *Journal of Risk*, *27*(5), 33–61. DOI: 10.21314/JOR.2025.007.

Feng, Y., Zhang, X., Peitz, C., Schulz, D., Li, S., and Letmathe, S. (2023c). *ufRisk: Risk Measure Calculation in Financial TS*. DOI: 10.32614/CRAN.package.ufRisk. R package version 1.0.7.

Feng, Y., and Zhou, C. (2015). Forecasting Financial Market Activity Using a Semiparametric Fractionally Integrated Log-ACD. *International Journal of Forecasting*, *31*(2), 349–363. DOI: 10.1016/j.ijforecast.2014.09.001.

Fernández, C., and Steel, M. F. J. (1998). On Bayesian Modelling of Fat Tails and Skewness. *Journal of the American Statistical Association*, *93*(441), 359–371. DOI: 10.1080/01621459.1998.10474117.

Francq, C., and Zakoïan, J.-M. (2019). *GARCH Models: Structure, Statistical Inference and Financial Applications* (2nd ed.). John Wiley & Sons, Hoboken, NJ.

Galanos, A. (2024). *rugarch: Univariate GARCH Models*. DOI: 10.32614/CRAN.package.rugarch. R package version 1.5-3.

Galanos, A. (2025). *rmgarch: Multivariate GARCH Models*. DOI: 10.32614/CRAN.package.rmgarch. R package version 1.4-2.

Galanos, A., and Theussl, S. (2015). *Rsolnp: General Non-linear Optimization Using Augmented Lagrange Multiplier Method*. DOI: 10.32614/CRAN.package.Rsolnp. R package version 1.16.

Geweke, J. (1986). Modelling Persistence in Conditional Variances: A Comment. *Econometric Reviews*, *5*(1), 57–61. DOI: 10.1080/07474938608800097.

Gilbert, P., and Varadhan, R. (2019). *numDeriv: Accurate Numerical Derivatives*. DOI: 10.32614/CRAN.package.numDeriv. R package version 2016.8-1.1.

Glosten, L. R., Jagannathan, R., and Runkle, D. E. (1993). On the Relation Between the Expected Value and the Volatility of Nominal Excess Return on Stocks. *Journal of Finance*, *48*(5), 1779–1801. DOI: 10.1111/j.1540-6261.1993.tb05128.x.

Granger, C. W. J., and Joyeux, R. (1980). An Introduction to Long-Memory Time Series Models and Fractional Differencing. *Journal of Time Series Analysis*, *1*(1), 15–29. DOI: 10.1111/j.1467-9892.1980.tb00297.x.

Hester, J., and Vaughan, D. (2025). *bench: High Precision Timing of R Expressions*. DOI: 10.32614/CRAN.package.bench. R package version 1.1.4.

Hochreiter, S., and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780. DOI: 10.1162/neco.1997.9.8.1735.

Hosking, J. R. M. (1981). Fractional Differencing. *Biometrika*, *68*(1), 165–176. DOI: 10.1093/biomet/68.1.165.

J.P. Morgan (1996). *RiskMetrics – Technical Document* (4th ed.). Morgan Guaranty Trust Company / RiskMetrics Group, New York. URL: https://www.msci.com/documents/10199/5915b101-4206-4ba0-aee2-3449d5c7e95a.

Karanasos, M., and Kim, J. (2003). Moments of the ARMA–EGARCH Model. *The Econometrics Journal*, *6*(1), 146–166. DOI: 10.1111/1368-423X.00104.

Kupiec, P. H. (1995). Techniques for verifying the accuracy of risk measurement models. *The Journal of Derivatives*, *3*(2), 73–84. DOI: 10.3905/jod.1995.407942.

Laurent, S. (2018). *Estimating and Forecasting ARCH Models Using G@RCH™ 8*. Timberlake Consultants Ltd., London, UK. URL: http://slaurent777.free.fr/pdf/Book8.pdf. OxMetrics 8 software manual.

Laurent, S., and Peters, J. (2002). G@RCH 2.2: An Ox Package for Estimating and Forecasting Various ARCH Models. *Journal of Economic Surveys*, *16*(3), 447–484. DOI: 10.1111/1467-6419.00174.

Letmathe, S., Beran, J., and Feng, Y. (2024). An Extended Exponential SEMIFAR Model with Application in R. *Communications in Statistics - Theory and Methods*, *53*(22), 7914–7926. DOI: 10.1080/03610926.2023.2276049.

Letmathe, S., Feng, Y., and Uhde, A. (2022). Semiparametric GARCH Models with Long Memory Applied to Value at Risk and Expected Shortfall. *Journal of Risk*, *25*(2), 75–105. DOI: 10.21314/JOR.2022.044.

Li, S. (2026). *Comparing the Estimation of Value at Risk and Expected Shortfall with LSTM and EGARCH Family Members.* Unpublished manuscript at Paderborn University.

Lopes, S. R. C., and Prass, T. S. (2014). Theoretical Results on Fractionally Integrated Exponential Generalized Autoregressive Conditional Heteroskedastic Processes. *Physica A: Statistical Mechanics and Its Applications*, *401*, 278–307. DOI: 10.1016/j.physa.2014.01.029.

McNeil, A. J., Frey, R., and Embrechts, P. (2015). *Quantitative Risk Management: Concepts, Techniques and Tools* (2nd ed.). Princeton University Press, Princeton, NJ.

Milhøj, A. (1987). *A Multiplicative Parametrization of ARCH Models* (No. 101; Research Report). Department of Statistics, University of Copenhagen.

Millar, R. B. (2011). *Maximum Likelihood Estimation and Inference: With Examples in R, SAS and ADMB*. John Wiley & Sons, Chichester, West Sussex, UK. DOI: 10.1002/9780470094846.

Missiakoulis, S. (1983). Sargan Densities: Which One? *Journal of Econometrics*, *23*(2), 223–233. DOI: 10.1016/0304-4076(93)90078-J.

Nelson, D. B. (1991). Conditional Heteroskedasticity in Asset Returns: A New Approach. *Econometrica*, *59*(2), 347–370. DOI: 10.2307/2938260.URL: https://www.jstor.org/stable/2938260.

Nielsen, M. Ø., and Noël, A. L. (2021). To Infinity and Beyond: Efficient Computation of ARCH(∞) Models. *Journal of Time Series Analysis*, *42*(3), 338–354. DOI: 10.1111/jtsa.12570.

Pantula, S. G. (1986). Modelling Persistence in Conditional Variances: A Comment. *Econometric Reviews*, *5*(1), 71–74. DOI: 10.1080/07474938608800099.

Peitz, C., Feng, Y., and Siddiqui, S. (2026). *Extending the EGARCH Family: Modulus and Long-Memory Log-Volatility Models*. Unpublished manuscript at Paderborn University.

Poon, S.-H., and Granger, C. W. J. (2003). Forecasting Volatility in Financial Markets: A Review. *Journal of Economic Literature*, *41*(2), 478–539. DOI: 10.1257/002205103765762743.

R Core Team (2026). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL: https://www.R-project.org/.

Sarma, M., Thomas, S., and Shah, A. (2003). Selection of Value-at-Risk Models. *Journal of Forecasting*, *22*(4), 337–358. DOI: 10.1002/for.868.

Schulz, D., Feng, Y., Peitz, C., and Ayensu, O. K. (2026). *fEGarch: SM/LM EGARCH & GARCH, VaR/ES Backtesting & Dual LM Extensions*. DOI: 10.32614/CRAN.package.fEGarch. R package version 1.0.6.

Sidi, A. (2003). *Practical Extrapolation Methods: Theory and Applications* (1st ed.). Cambridge University Press, Cambridge, UK. DOI: 10.1017/CBO9780511546815.

Sucarrat, G. (2019). The Log-GARCH Model via ARMA Representations. In: J. Chevallier, S. Goutte, D. Guerreiro, S. Saglio and B. Sanhaji (Eds.), *Financial mathematics, volatility and covariance modelling: Volume 2* (pp. 336–359). Routledge. DOI: 10.4324/9781315162737-14.

Sucarrat, G. (2025). *lgarch: Simulation and Estimation of Log-GARCH Models*. DOI: 10.32614/CRAN.package.lgarch. R package version 0.7.

Surgailis, D., and Viano, M.-C. (2002). Long Memory Properties and Covariance Structure of the EGARCH Model. *ESAIM: Probability and Statistics*, *6*, 311–329. DOI: 10.1051/ps:2002017.

Trapletti, A., and Hornik, K. (2024). *tseries: Time Series Analysis and Computational Finance*. DOI: 10.32614/CRAN.package.tseries. R package version 0.10-58.

Tsay, R. S. (2005). *Analysis of Financial Time Series* (2nd ed.). John Wiley & Sons, Hoboken, NJ. DOI: 10.1002/0471746193.

Tse, Y. K. (1987). A Note on Sargan Densities. *Journal of Econometrics*, *34*(3), 349–354. DOI: 10.1016/0304-4076(87)90017-0.

Tse, Y. K. (1998). The Conditional Heteroscedasticity of the Yen–Dollar Exchange Rate. *Journal of Applied Econometrics*, *13*(1), 49–55. DOI: 10.1002/(SICI)1099-1255(199801/02)13:1<49::AID-JAE459>3.0.CO;2-O.

Wickham, H. (2016). *ggplot2: Elegant Graphics for Data Analysis* (2nd ed.). Springer, Cham, Switzerland. DOI: doi.org/10.1007/978-3-319-24277-4.

Wuertz, D., Chalabi, Y., Setz, T., Maechler, M., and Boshnakov, G. N. (2024). *fGarch: Rmetrics - Autoregressive Conditional Heteroskedastic Modelling*. DOI: 10.32614/CRAN.package.fGarch. R package version 4033.92.

Ye, Y. (1989). *SOLNP Users' Guide: A Nonlinear Optimization Program in MATLAB*. URL: https://web.stanford.edu/~yyye/matlab.html. Department of Management Sciences, College of Business Administration, University of Iowa.

Zakoian, J.-M. (1994). Threshold Heteroskedastic Models. *Journal of Economic Dynamics and Control*, *18*(5), 931–955. DOI: 10.1016/0165-1889(94)90039-6.

Zeileis, A., and Grothendieck, G. (2005). zoo: S3 Infrastructure for Regular and Irregular Time Series. *Journal of Statistical Software*, *14*(6), 1–27. DOI: 10.18637/jss.v014.i06.

Appendix

A Simulation tables

The following pages contain the Tables 7 to 14 with the numeric results of the simulation study of Section 6. Each table contains the results for one of the main models from the EGF. The first table, Table 6, however provides information on how many of the 1000 simulated estimations for each setting had convergence issues and where estimation even may have failed entirely.

Table 6: Overview of number of cases in the simulation where convergence was not reached or estimation failed. Larger numbers of observations lead to more numerical stability, short-memory models are more stable than long-memory ones and Type-I models are more stable than Type-II models.

Short-memory								Long-memory							
		Conv. issues			Fails					Conv. issues			Fails		
Model	Cond. distr.	n_1	n_2	n_3	n_1	n_2	n_3	Model	Cond. distr.	n_1	n_2	n_3	n_1	n_2	n_3
EGARCH	norm	0	0	0	0	0	0	FIEGARCH	norm	0	0	0	0	0	0
EGARCH	std	0	0	0	0	0	0	FIEGARCH	std	0	0	0	0	0	0
EGARCH	ged	0	0	0	0	0	0	FIEGARCH	ged	0	1	0	0	0	0
EGARCH	ald	1	0	0	0	0	0	FIEGARCH	ald	0	0	0	0	0	0
EGARCH	snorm	0	0	0	0	0	0	FIEGARCH	snorm	0	0	0	0	0	0
EGARCH	sstd	0	0	0	0	0	0	FIEGARCH	sstd	0	0	0	0	0	0
EGARCH	sged	0	0	0	0	0	0	FIEGARCH	sged	0	0	0	0	0	0
EGARCH	sald	0	0	0	0	0	0	FIEGARCH	sald	2	0	0	0	0	0
MLog-GARCH	norm	1	0	0	0	0	0	FIMLog-GARCH	norm	1	0	0	0	0	0
MLog-GARCH	std	0	0	0	0	0	0	FIMLog-GARCH	std	0	2	0	0	0	0
MLog-GARCH	ged	1	1	0	0	0	0	FIMLog-GARCH	ged	3	0	0	0	0	0
MLog-GARCH	ald	0	0	0	0	0	0	FIMLog-GARCH	ald	2	3	1	0	0	0
MLog-GARCH	snorm	0	0	0	0	0	0	FIMLog-GARCH	snorm	2	0	0	0	0	0
MLog-GARCH	sstd	3	1	0	0	0	0	FIMLog-GARCH	sstd	1	1	0	0	0	0
MLog-GARCH	sged	0	0	0	0	0	0	FIMLog-GARCH	sged	1	1	0	0	0	0
MLog-GARCH	sald	1	0	0	0	0	0	FIMLog-GARCH	sald	2	0	0	0	0	0
MEGARCH	norm	0	0	0	0	0	0	FIMEGARCH	norm	0	0	0	0	0	0
MEGARCH	std	0	0	0	0	0	0	FIMEGARCH	std	1	0	0	0	0	0
MEGARCH	ged	0	0	0	0	0	0	FIMEGARCH	ged	2	0	0	0	0	0
MEGARCH	ald	0	0	0	0	0	0	FIMEGARCH	ald	0	0	0	0	0	0
MEGARCH	snorm	0	0	0	0	0	0	FIMEGARCH	snorm	0	0	0	0	0	0
MEGARCH	sstd	0	0	0	0	0	0	FIMEGARCH	sstd	0	0	0	0	0	0
MEGARCH	sged	0	0	0	0	0	0	FIMEGARCH	sged	2	0	0	0	0	0
MEGARCH	sald	0	0	0	0	0	0	FIMEGARCH	sald	0	0	0	0	0	0
Log-GARCH	norm	0	1	0	0	0	0	FILog-GARCH	norm	6	4	1	0	0	0
Log-GARCH	std	0	0	0	0	0	0	FILog-GARCH	std	18	1	0	0	0	0
Log-GARCH	ged	0	0	0	0	0	0	FILog-GARCH	ged	9	2	0	0	0	0
Log-GARCH	ald	0	0	0	0	0	0	FILog-GARCH	ald	7	2	1	0	0	0
Log-GARCH	snorm	0	0	0	0	0	0	FILog-GARCH	snorm	10	4	0	0	0	0
Log-GARCH	sstd	0	0	0	1	0	0	FILog-GARCH	sstd	11	5	0	1	0	0
Log-GARCH	sged	0	0	0	0	0	0	FILog-GARCH	sged	11	2	0	0	0	0
Log-GARCH	sald	0	0	0	0	0	0	FILog-GARCH	sald	6	0	0	0	0	0

Table 7: Simulation study results regarding EGARCH$(1,1)$. MSE-values are multiplied by 1000. True values are $\mu = 0.0001$, $\omega_\sigma = -8.5$, $\phi_1 = 0.65$, $\kappa = -0.25$, $\gamma = 0.35$, $\nu = 6$, $\beta = 1.4$, $P = 2$ and $s = 0.9$.

		EGARCH$(1,1)$																							
		norm			std			ged			ald			snorm			sstd			sged			sald		
Estimator	Measure	n_1	n_2	n_3	n_1	n_2	n_3	n_1	n_2	n_3	n_1	n_2	n_3	n_1	n_2	n_3	n_1	n_2	n_3	n_1	n_2	n_3	n_1	n_2	n_3
$\hat{\mu}$	Mean	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
	SD	0.0005	0.0003	0.0002	0.0004	0.0003	0.0002	0.0005	0.0003	0.0002	0.0005	0.0003	0.0002	0.0005	0.0004	0.0002	0.0005	0.0003	0.0002	0.0005	0.0003	0.0002	0.0005	0.0004	0.0002
	MSE	0.0002	0.0001	0.0001	0.0002	0.0001	0.0001	0.0002	0.0001	0.0001	0.0002	0.0001	0.0001	0.0003	0.0001	0.0001	0.0002	0.0001	0.0001	0.0003	0.0001	0.0001	0.0002	0.0001	0.0001
$\hat{\omega}_\sigma$	Mean	-8.5026	-8.5030	-8.5031	-8.5036	-8.5000	-8.5000	-8.5032	-8.5060	-8.5019	-8.5088	-8.5045	-8.5039	-8.5039	-8.5050	-8.5019	-8.5090	-8.5044	-8.5020	-8.5021	-8.5039	-8.5041	-8.5043	-8.5022	-8.5008
	SD	0.0737	0.0548	0.0383	0.0993	0.0693	0.0509	0.0862	0.0607	0.0450	0.0850	0.0622	0.0449	0.0801	0.0572	0.0394	0.1095	0.0716	0.0537	0.0941	0.0610	0.0476	0.0953	0.0664	0.0465
	MSE	5.4261	3.0076	1.4748	9.8634	4.8023	2.5853	7.4288	3.7176	2.0283	7.2976	3.8881	2.0321	6.4324	3.2906	1.5533	12.0512	5.1380	2.8881	8.8547	3.7330	2.2808	9.0894	4.4091	2.1597
$\hat{\phi}_1$	Mean	0.6408	0.6413	0.6460	0.6345	0.6425	0.6458	0.6382	0.6399	0.6472	0.6336	0.6413	0.6449	0.6364	0.6420	0.6468	0.6354	0.6460	0.6477	0.6334	0.6446	0.6476	0.6335	0.6442	0.6489
	SD	0.0887	0.0585	0.0403	0.1072	0.0677	0.0460	0.1031	0.0679	0.0456	0.1030	0.0681	0.0472	0.0850	0.0574	0.0402	0.0993	0.0638	0.0443	0.0965	0.0624	0.0425	0.0975	0.0645	0.0440
	MSE	7.9407	3.4925	1.6380	11.7240	4.6351	2.1321	10.7589	4.7057	2.0836	10.8664	4.7089	2.2499	7.3972	3.3556	1.6274	10.0547	4.0865	1.9631	9.5726	3.9152	1.8119	9.7632	4.1838	1.9330
$\hat{\kappa}$	Mean	-0.2532	-0.2525	-0.2511	-0.2524	-0.2526	-0.2516	-0.2532	-0.2495	-0.2503	-0.2513	-0.2504	-0.2502	-0.2527	-0.2529	-0.2515	-0.2541	-0.2489	-0.2505	-0.2541	-0.2538	-0.2511	-0.2569	-0.2518	-0.2511
	SD	0.0497	0.0343	0.0241	0.0601	0.0396	0.0285	0.0609	0.0405	0.0290	0.0571	0.0400	0.0284	0.0502	0.0341	0.0238	0.0585	0.0411	0.0284	0.0579	0.0396	0.0284	0.0576	0.0383	0.0274
	MSE	2.4766	1.1822	0.5822	3.6103	1.5701	0.8135	3.7179	1.6378	0.8415	3.2588	1.5993	0.8056	2.5246	1.1729	0.5660	3.4300	1.6858	0.8075	3.3653	1.5845	0.8088	3.3583	1.4684	0.7492
$\hat{\gamma}$	Mean	0.3422	0.3470	0.3499	0.3399	0.3461	0.3501	0.3370	0.3470	0.3462	0.3415	0.3444	0.3490	0.3428	0.3478	0.3483	0.3393	0.3442	0.3488	0.3373	0.3463	0.3475	0.3394	0.3434	0.3474
	SD	0.0825	0.0534	0.0398	0.0913	0.0618	0.0411	0.0879	0.0590	0.0438	0.0860	0.0589	0.0416	0.0792	0.0547	0.0377	0.0894	0.0607	0.0434	0.0873	0.0575	0.0425	0.0862	0.0596	0.0411
	MSE	6.8653	2.8589	1.5787	8.4287	3.8250	1.6885	7.8847	3.4868	1.9268	7.4525	3.4991	1.7332	6.3196	2.9905	1.4198	8.0946	3.7147	1.8803	7.7785	3.3169	1.8130	7.5386	3.5963	1.6940
$\hat{\nu}$	Mean	—	—	—	6.5640	6.2437	6.1399	—	—	—	—	—	—	—	—	—	6.6876	6.2893	6.1117	—	—	—	—	—	—
	SD	—	—	—	1.8407	0.9964	0.7068	—	—	—	—	—	—	—	—	—	1.8738	1.0894	0.7134	—	—	—	—	—	—
	MSE	—	—	—	3702.8668	1051.3156	518.6462	—	—	—	—	—	—	—	—	—	3980.4095	1269.2083	520.8734	—	—	—	—	—	—
$\hat{\beta}$	Mean	—	—	—	—	—	—	1.4228	1.4087	1.4045	—	—	—	—	—	—	—	—	—	1.4210	1.4073	1.4056	—	—	—
	SD	—	—	—	—	—	—	0.1119	0.0751	0.0552	—	—	—	—	—	—	—	—	—	0.1137	0.0744	0.0549	—	—	—
	MSE	—	—	—	—	—	—	13.0330	5.7047	3.0657	—	—	—	—	—	—	—	—	—	13.3499	5.5829	3.0433	—	—	—
$\hat{P}$	Mean	—	—	—	—	—	—	—	—	—	2.5225	2.3160	2.1880	—	—	—	—	—	—	—	—	—	2.4910	2.3420	2.2010
	SD	—	—	—	—	—	—	—	—	—	1.2026	0.8467	0.5166	—	—	—	—	—	—	—	—	—	1.1905	0.8257	0.5718
	MSE	—	—	—	—	—	—	—	—	—	1717.7177	816.0000	302.0000	—	—	—	—	—	—	—	—	—	1657.0000	798.0000	367.0000
$\hat{s}$	Mean	—	—	—	—	—	—	—	—	—	—	—	—	0.8961	0.9014	0.9003	0.8974	0.9005	0.9003	0.8992	0.9012	0.9004	0.8976	0.9007	0.9001
	SD	—	—	—	—	—	—	—	—	—	—	—	—	0.0494	0.0361	0.0249	0.0480	0.0330	0.0239	0.0431	0.0291	0.0194	0.0461	0.0317	0.0221
	MSE	—	—	—	—	—	—	—	—	—	—	—	—	2.4490	1.3067	0.6186	2.3118	1.0873	0.5695	1.8550	0.8472	0.3748	2.1334	1.0019	0.4871

Table 8: Simulation study results regarding MEGARCH$(1,1)$. MSE-values are multiplied by 1000. True values are $\mu = 0.0001$, $\omega_\sigma = -8.5$, $\phi_1 = 0.65$, $\kappa = -0.25$, $\gamma = 0.35$, $\nu = 6$, $\beta = 1.4$, $P = 2$ and $s = 0.9$.

		MEGARCH$(1,1)$																							
		norm			std			ged			ald			snorm			sstd			sged			sald		
Estimator	Measure	n_1	n_2	n_3	n_1	n_2	n_3	n_1	n_2	n_3	n_1	n_2	n_3	n_1	n_2	n_3	n_1	n_2	n_3	n_1	n_2	n_3	n_1	n_2	n_3
$\hat{\mu}$	Mean	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
	SD	0.0005	0.0004	0.0003	0.0004	0.0003	0.0002	0.0005	0.0003	0.0002	0.0005	0.0003	0.0002	0.0005	0.0004	0.0002	0.0005	0.0003	0.0002	0.0005	0.0004	0.0002	0.0005	0.0004	0.0002
	MSE	0.0002	0.0001	0.0001	0.0002	0.0001	0.0001	0.0002	0.0001	0.0001	0.0002	0.0001	0.0001	0.0003	0.0001	0.0001	0.0002	0.0001	0.0001	0.0003	0.0001	0.0001	0.0002	0.0001	0.0001
$\hat{\omega}_\sigma$	Mean	-8.5028	-8.5034	-8.5033	-8.5043	-8.5005	-8.4999	-8.5032	-8.5058	-8.5018	-8.5092	-8.5047	-8.5040	-8.5045	-8.5052	-8.5019	-8.5089	-8.5046	-8.5018	-8.5027	-8.5039	-8.5041	-8.5050	-8.5024	-8.5009
	SD	0.0718	0.0528	0.0369	0.0978	0.0684	0.0500	0.0841	0.0597	0.0443	0.0829	0.0607	0.0438	0.0759	0.0539	0.0373	0.1061	0.0693	0.0518	0.0893	0.0583	0.0459	0.0909	0.0633	0.0446
	MSE	5.1595	2.7943	1.3729	9.5718	4.6750	2.5024	7.0776	3.5902	1.9641	6.9564	3.7022	1.9331	5.7685	2.9298	1.3924	11.3176	4.8147	2.6830	7.9821	3.4084	2.1196	8.2861	4.0131	1.9844
$\hat{\phi}_1$	Mean	0.6335	0.6366	0.6432	0.6237	0.6383	0.6432	0.6270	0.6347	0.6458	0.6243	0.6366	0.6434	0.6297	0.6372	0.6454	0.6271	0.6429	0.6459	0.6223	0.6395	0.6461	0.6251	0.6410	0.6473
	SD	0.1180	0.0769	0.0517	0.1473	0.0883	0.0586	0.1414	0.0866	0.0579	0.1313	0.0865	0.0585	0.1122	0.0743	0.0514	0.1333	0.0841	0.0570	0.1323	0.0809	0.0550	0.1267	0.0833	0.0565
	MSE	14.1827	6.0905	2.7198	22.3525	7.9293	3.4729	20.4887	7.7242	3.3689	17.8916	7.6478	3.4630	12.9874	5.6789	2.6587	18.2724	7.1089	3.2640	18.2637	6.6462	3.0371	16.6451	7.0072	3.1968
$\hat{\kappa}$	Mean	-0.2541	-0.2533	-0.2512	-0.2543	-0.2527	-0.2523	-0.2534	-0.2491	-0.2499	-0.2522	-0.2499	-0.2502	-0.2532	-0.2544	-0.2521	-0.2544	-0.2470	-0.2505	-0.2541	-0.2556	-0.2522	-0.2598	-0.2524	-0.2517
	SD	0.0785	0.0543	0.0384	0.0989	0.0648	0.0462	0.0989	0.0660	0.0472	0.0925	0.0656	0.0455	0.0801	0.0549	0.0374	0.0950	0.0670	0.0458	0.0945	0.0650	0.0462	0.0931	0.0623	0.0443
	MSE	6.1742	2.9601	1.4748	9.7822	4.1997	2.1417	9.7868	4.3540	2.2254	8.5584	4.3032	2.0711	6.4124	3.0279	1.3998	9.0340	4.4916	2.0952	8.9350	4.2473	2.1404	8.7542	3.8820	1.9596
$\hat{\gamma}$	Mean	0.3439	0.3484	0.3506	0.3415	0.3475	0.3511	0.3394	0.3487	0.3467	0.3441	0.3459	0.3493	0.3446	0.3492	0.3489	0.3406	0.3449	0.3493	0.3396	0.3482	0.3482	0.3415	0.3441	0.3482
	SD	0.0851	0.0552	0.0414	0.0928	0.0639	0.0426	0.0912	0.0609	0.0451	0.0883	0.0606	0.0427	0.0814	0.0566	0.0390	0.0916	0.0627	0.0446	0.0903	0.0593	0.0441	0.0893	0.0610	0.0427
	MSE	7.2661	3.0465	1.7088	8.6779	4.0889	1.8138	8.4202	3.7013	2.0402	7.8178	3.6869	1.8241	6.6480	3.2017	1.5225	8.4619	3.9503	1.9886	8.2478	3.5105	1.9427	8.0327	3.7557	1.8249
$\hat{\nu}$	Mean	—	—	—	6.5658	6.2446	6.1402	—	—	—	—	—	—	—	—	—	6.6890	6.2920	6.1125	—	—	—	—	—	—
	SD	—	—	—	1.8333	0.9972	0.7070	—	—	—	—	—	—	—	—	—	1.8794	1.0906	0.7126	—	—	—	—	—	—
	MSE	—	—	—	3677.8283	1053.1530	518.9825	—	—	—	—	—	—	—	—	—	4003.1624	1273.5852	519.9317	—	—	—	—	—	—
$\hat{\beta}$	Mean	—	—	—	—	—	—	1.4230	1.4088	1.4045	—	—	—	—	—	—	—	—	—	1.4213	1.4074	1.4058	—	—	—
	SD	—	—	—	—	—	—	0.1118	0.0752	0.0552	—	—	—	—	—	—	—	—	—	0.1137	0.0743	0.0549	—	—	—
	MSE	—	—	—	—	—	—	13.0267	5.7301	3.0648	—	—	—	—	—	—	—	—	—	13.3680	5.5722	3.0438	—	—	—
$\hat{P}$	Mean	—	—	—	—	—	—	—	—	—	2.5220	2.3220	2.1880	—	—	—	—	—	—	—	—	—	2.4880	2.3400	2.2070
	SD	—	—	—	—	—	—	—	—	—	1.1996	0.8444	0.5186	—	—	—	—	—	—	—	—	—	1.1897	0.8192	0.5731
	MSE	—	—	—	—	—	—	—	—	—	1710.0000	816.0000	304.0000	—	—	—	—	—	—	—	—	—	1652.0000	786.0000	371.0000
$\hat{s}$	Mean	—	—	—	—	—	—	—	—	—	—	—	—	0.8960	0.9013	0.9002	0.8974	0.9005	0.9003	0.8992	0.9012	0.9004	0.8976	0.9007	0.9001
	SD	—	—	—	—	—	—	—	—	—	—	—	—	0.0494	0.0361	0.0249	0.0481	0.0330	0.0239	0.0433	0.0291	0.0194	0.0461	0.0316	0.0221
	MSE	—	—	—	—	—	—	—	—	—	—	—	—	2.4513	1.3051	0.6179	2.3199	1.0896	0.5693	1.8700	0.8487	0.3761	2.1300	1.0006	0.4879

Table 9: Simulation study results regarding MLog-GARCH$(1,1)$. MSE-values are multiplied by 1000. True values are $\mu = 0.0001$, $\omega_\sigma = -8.5$, $\phi_1 = 0.65$, $\kappa = -0.25$, $\gamma = 0.35$, $\nu = 6$, $\beta = 1.4$, $P = 2$ and $s = 0.9$.

		MLog-GARCH$(1,1)$																							
		norm			std			ged			ald			snorm			sstd			sged			sald		
Estimator	Measure	n_1	n_2	n_3	n_1	n_2	n_3	n_1	n_2	n_3	n_1	n_2	n_3	n_1	n_2	n_3	n_1	n_2	n_3	n_1	n_2	n_3	n_1	n_2	n_3
$\hat{\mu}$	Mean	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
	SD	0.0005	0.0004	0.0003	0.0004	0.0003	0.0002	0.0005	0.0003	0.0002	0.0005	0.0003	0.0002	0.0005	0.0004	0.0003	0.0005	0.0004	0.0003	0.0005	0.0004	0.0002	0.0005	0.0004	0.0002
	MSE	0.0002	0.0001	0.0001	0.0002	0.0001	0.0001	0.0002	0.0001	0.0001	0.0002	0.0001	0.0001	0.0003	0.0001	0.0001	0.0002	0.0001	0.0001	0.0003	0.0001	0.0001	0.0002	0.0001	0.0001
$\hat{\omega}_\sigma$	Mean	-8.5033	-8.5033	-8.5030	-8.5038	-8.5008	-8.5003	-8.5030	-8.5057	-8.5020	-8.5092	-8.5049	-8.5039	-8.5048	-8.5048	-8.5018	-8.5093	-8.5045	-8.5018	-8.5037	-8.5038	-8.5038	-8.5052	-8.5028	-8.5010
	SD	0.0616	0.0455	0.0318	0.0903	0.0599	0.0439	0.0827	0.0514	0.0382	0.0719	0.0526	0.0379	0.0657	0.0466	0.0322	0.0933	0.0610	0.0456	0.0778	0.0507	0.0397	0.0794	0.0551	0.0387
	MSE	3.7990	2.0759	1.0173	8.1553	3.5800	1.9272	6.8493	2.6690	1.4607	5.2536	2.7847	1.4520	4.3344	2.1961	1.0420	8.7868	3.7412	2.0820	6.0637	2.5820	1.5904	6.3220	3.0448	1.4947
$\hat{\phi}_1$	Mean	0.6245	0.6314	0.6423	0.6018	0.6243	0.6384	0.6215	0.6226	0.6434	0.6046	0.6324	0.6391	0.6198	0.6310	0.6428	0.6189	0.6355	0.6426	0.6022	0.6347	0.6421	0.6099	0.6354	0.6446
	SD	0.1739	0.1058	0.0663	0.2275	0.1410	0.0840	0.2021	0.1420	0.0819	0.2131	0.1280	0.0829	0.1651	0.1033	0.0669	0.2023	0.1254	0.0818	0.2122	0.1182	0.0757	0.1891	0.1206	0.0778
	MSE	30.8730	11.5318	4.4444	54.0126	20.5279	7.1868	41.6327	20.9008	6.7399	47.4221	16.6763	6.9857	28.1396	11.0307	4.5259	41.8602	15.9135	6.7385	47.2514	14.2035	5.7918	37.3441	14.7476	6.0694
$\hat{\kappa}$	Mean	-0.2558	-0.2543	-0.2515	-0.2570	-0.2547	-0.2535	-0.2549	-0.2510	-0.2507	-0.2537	-0.2509	-0.2511	-0.2546	-0.2555	-0.2526	-0.2559	-0.2488	-0.2515	-0.2555	-0.2564	-0.2533	-0.2614	-0.2532	-0.2523
	SD	0.0792	0.0532	0.0377	0.1002	0.0658	0.0465	0.0992	0.0665	0.0472	0.0945	0.0659	0.0456	0.0795	0.0536	0.0367	0.0972	0.0666	0.0460	0.0938	0.0647	0.0459	0.0928	0.0624	0.0442
	MSE	6.3052	2.8480	1.4196	10.0688	4.3491	2.1718	9.8554	4.4155	2.2259	8.9313	4.3356	2.0821	6.3349	2.9019	1.3533	9.4747	4.4332	2.1187	8.8207	4.2179	2.1191	8.7405	3.9037	1.9544
$\hat{\gamma}$	Mean	0.3386	0.3465	0.3506	0.3289	0.3444	0.3512	0.3244	0.3462	0.3445	0.3367	0.3369	0.3490	0.3378	0.3477	0.3481	0.3233	0.3389	0.3496	0.3281	0.3454	0.3478	0.3297	0.3382	0.3482
	SD	0.1577	0.0991	0.0730	0.1841	0.1224	0.0814	0.1729	0.1123	0.0834	0.1715	0.1171	0.0808	0.1527	0.1012	0.0688	0.1803	0.1214	0.0853	0.1735	0.1103	0.0813	0.1697	0.1148	0.0794
	MSE	24.9741	9.8173	5.3234	34.3040	14.9995	6.6263	30.5204	12.6209	6.9768	29.5738	13.8672	6.5301	23.4571	10.2429	4.7373	33.1950	14.8543	7.2731	30.5404	12.1814	6.6003	29.1739	13.3140	6.2967
$\hat{\nu}$	Mean	—	—	—	6.5689	6.2439	6.1404	—	—	—	—	—	—	—	—	—	6.7547	6.2964	6.1131	—	—	—	—	—	—
	SD	—	—	—	1.8456	0.9971	0.7065	—	—	—	—	—	—	—	—	—	2.7919	1.0897	0.7126	—	—	—	—	—	—
	MSE	—	—	—	3726.3305	1052.7448	518.3927	—	—	—	—	—	—	—	—	—	8356.4739	1273.9901	520.0799	—	—	—	—	—	—
$\hat{\beta}$	Mean	—	—	—	—	—	—	1.4232	1.4090	1.4046	—	—	—	—	—	—	—	—	—	1.4221	1.4077	1.4058	—	—	—
	SD	—	—	—	—	—	—	0.1118	0.0752	0.0552	—	—	—	—	—	—	—	—	—	0.1158	0.0744	0.0549	—	—	—
	MSE	—	—	—	—	—	—	13.0148	5.7324	3.0656	—	—	—	—	—	—	—	—	—	13.8806	5.5879	3.0415	—	—	—
$\hat{P}$	Mean	—	—	—	—	—	—	—	—	—	2.5270	2.3230	2.1880	—	—	—	—	—	—	—	—	—	2.5005	2.3500	2.2020
	SD	—	—	—	—	—	—	—	—	—	1.2111	0.8387	0.5128	—	—	—	—	—	—	—	—	—	1.1936	0.8247	0.5705
	MSE	—	—	—	—	—	—	—	—	—	1743.0000	807.0000	298.0000	—	—	—	—	—	—	—	—	—	1673.6737	802.0000	366.0000
$\hat{s}$	Mean	—	—	—	—	—	—	—	—	—	—	—	—	0.8957	0.9013	0.9002	0.8974	0.9005	0.9002	0.8993	0.9012	0.9004	0.8972	0.9008	0.9000
	SD	—	—	—	—	—	—	—	—	—	—	—	—	0.0495	0.0361	0.0249	0.0484	0.0330	0.0239	0.0435	0.0292	0.0194	0.0463	0.0317	0.0221
	MSE	—	—	—	—	—	—	—	—	—	—	—	—	2.4676	1.3032	0.6181	2.3431	1.0894	0.5718	1.8876	0.8551	0.3753	2.1497	1.0046	0.4872

Table 10: Simulation study results regarding Log-GARCH(1, 1). MSE-values are multiplied by 1000. True values are $\mu = 0.0001$, $\omega_\sigma = -8.5$, $\phi_1 = 0.85$, $\psi_1 = -0.8$, $\nu = 6$, $\beta = 1.4$, $P = 2$ and $s = 0.9$.

		Log-GARCH(1, 1)																							
		norm			std			ged			ald			snorm			sstd			sged			sald		
Estimator	Measure	n_1	n_2	n_3	n_1	n_2	n_3	n_1	n_2	n_3	n_1	n_2	n_3	n_1	n_2	n_3	n_1	n_2	n_3	n_1	n_2	n_3	n_1	n_2	n_3
$\hat{\mu}$	Mean	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0002	0.0002	0.0002	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
	SD	0.0005	0.0004	0.0003	0.0005	0.0004	0.0003	0.0005	0.0004	0.0003	0.0005	0.0004	0.0003	0.0006	0.0004	0.0003	0.0005	0.0004	0.0003	0.0006	0.0004	0.0003	0.0005	0.0004	0.0003
	MSE	0.0003	0.0002	0.0001	0.0003	0.0002	0.0001	0.0003	0.0002	0.0001	0.0003	0.0001	0.0001	0.0003	0.0002	0.0001	0.0003	0.0002	0.0001	0.0003	0.0002	0.0001	0.0003	0.0002	0.0001
$\hat{\omega}_\sigma$	Mean	-8.5006	-8.5000	-8.5010	-8.4933	-8.5005	-8.5013	-8.4895	-8.4957	-8.4943	-8.4963	-8.5015	-8.5014	-8.4999	-8.5020	-8.4999	-8.5029	-8.5048	-8.5032	-8.4940	-8.4996	-8.5009	-8.5061	-8.4993	-8.4991
	SD	0.1404	0.0508	0.0352	0.3316	0.0651	0.0465	0.0862	0.0629	0.0483	0.2502	0.0579	0.0414	0.0719	0.0498	0.0348	0.1029	0.0648	0.0477	0.0893	0.0574	0.0460	0.2131	0.0587	0.0416
	MSE	19.7011	2.5796	1.2416	109.9117	4.2297	2.1647	7.5335	3.9767	2.3644	62.5504	3.3519	1.7106	5.1715	2.4850	1.2067	10.5857	4.2159	2.2868	8.0036	3.2959	2.1131	45.4160	3.4395	1.7312
$\hat{\phi}_1$	Mean	0.8485	0.8552	0.8562	0.8480	0.8545	0.8570	0.8475	0.8541	0.8571	0.8384	0.8407	0.8489	0.8452	0.8498	0.8567	0.8428	0.8550	0.8561	0.8429	0.8535	0.8534	0.8326	0.8440	0.8455
	SD	0.1079	0.0775	0.0657	0.1452	0.0967	0.0749	0.1226	0.0954	0.0651	0.1392	0.0918	0.0538	0.1216	0.0816	0.0537	0.1512	0.0832	0.0644	0.1260	0.0787	0.0555	0.1378	0.0803	0.0577
	MSE	11.6391	6.0226	4.3443	21.0605	9.3691	5.6480	15.0299	9.1004	4.2786	19.4939	8.5120	2.8894	14.7832	6.6441	2.9258	22.8767	6.9328	4.1749	15.9011	6.2035	3.0881	19.2633	6.4795	3.3433
$\hat{\psi}_1$	Mean	-0.8008	-0.8084	-0.8089	-0.8027	-0.8081	-0.8100	-0.8008	-0.8072	-0.8096	-0.7896	-0.7919	-0.7994	-0.7980	-0.8023	-0.8090	-0.7969	-0.8087	-0.8086	-0.7965	-0.8056	-0.8049	-0.7835	-0.7944	-0.7955
	SD	0.1224	0.0868	0.0735	0.1594	0.1073	0.0831	0.1367	0.1044	0.0733	0.1513	0.1016	0.0611	0.1354	0.0911	0.0616	0.1651	0.0939	0.0729	0.1388	0.0880	0.0636	0.1515	0.0891	0.0642
	MSE	14.9739	7.5936	5.4712	25.3894	11.5628	6.9947	18.6794	10.9326	5.4613	22.9716	10.3730	3.7336	18.3169	8.2891	3.8682	27.2446	8.8922	5.3846	19.2691	7.7623	4.0616	23.2029	7.9631	4.1322
$\hat{\nu}$	Mean	—	—	—	6.6536	6.3804	6.2837	—	—	—	—	—	—	—	—	—	6.6893	6.3754	6.1910	—	—	—	—	—	—
	SD	—	—	—	1.5958	0.9362	0.7165	—	—	—	—	—	—	—	—	—	1.7591	1.0102	0.7253	—	—	—	—	—	—
	MSE	—	—	—	2971.1533	1020.2182	593.3444	—	—	—	—	—	—	—	—	—	3566.4853	1160.4135	562.0556	—	—	—	—	—	—
$\hat{\beta}$	Mean	—	—	—	—	—	—	1.4557	1.4404	1.4333	—	—	—	—	—	—	—	—	—	1.4417	1.4261	1.4203	—	—	—
	SD	—	—	—	—	—	—	0.1372	0.1156	0.1022	—	—	—	—	—	—	—	—	—	0.1319	0.1007	0.0840	—	—	—
	MSE	—	—	—	—	—	—	21.9105	14.9812	11.5338	—	—	—	—	—	—	—	—	—	19.1221	10.8162	7.4653	—	—	—
$\hat{P}$	Mean	—	—	—	—	—	—	—	—	—	2.5900	2.4520	2.3170	—	—	—	—	—	—	—	—	—	2.5810	2.4910	2.3210
	SD	—	—	—	—	—	—	—	—	—	1.2138	0.9943	0.7396	—	—	—	—	—	—	—	—	—	1.2365	0.9934	0.7297
	MSE	—	—	—	—	—	—	—	—	—	1820.0000	1192.0000	647.0000	—	—	—	—	—	—	—	—	—	1865.0000	1227.0000	635.0000
$\hat{s}$	Mean	—	—	—	—	—	—	—	—	—	—	—	—	0.9008	0.9045	0.9035	0.9027	0.9059	0.9052	0.9026	0.9043	0.9040	0.8994	0.9022	0.9010
	SD	—	—	—	—	—	—	—	—	—	—	—	—	0.0515	0.0391	0.0292	0.0497	0.0358	0.0278	0.0454	0.0320	0.0237	0.0465	0.0321	0.0224
	MSE	—	—	—	—	—	—	—	—	—	—	—	—	2.6548	1.5483	0.8626	2.4726	1.3164	0.7984	2.0683	1.0390	0.5751	2.1632	1.0348	0.5032

Table 11: Simulation study results regarding FIEGARCH$(1, d, 1)$. MSE-values are multiplied by 1000. True values are $\mu = 0.0001$, $\omega_\sigma = -8.5$, $\phi_1 = 0.65$, $\kappa = -0.25$, $\gamma = 0.35$, $d = 0.3$, $\nu = 6$, $\beta = 1.4$, $P = 2$ and $s = 0.9$.

		FIEGARCH$(1, d, 1)$											
		norm			std			ged			ald		
Estimator	Measure	n_1	n_2	n_3	n_1	n_2	n_3	n_1	n_2	n_3	n_1	n_2	n_3
$\hat{\mu}$	Mean	0.0000	0.0001	0.0001	0.0000	0.0000	0.0001	0.0001	0.0001	0.0001	0.0000	0.0001	0.0001
	SD	0.0005	0.0003	0.0002	0.0004	0.0003	0.0002	0.0004	0.0003	0.0002	0.0005	0.0003	0.0002
	MSE	0.0002	0.0001	0.0001	0.0002	0.0001	0.0000	0.0002	0.0001	0.0000	0.0002	0.0001	0.0000
$\hat{\omega}_\sigma$	Mean	-8.4849	-8.4998	-8.4961	-8.4857	-8.4811	-8.4924	-8.4773	-8.5059	-8.4925	-8.4946	-8.4937	-8.4982
	SD	0.2968	0.2599	0.2081	0.3220	0.2604	0.2207	0.3179	0.2564	0.2104	0.3230	0.2706	0.2188
	MSE	88.2206	67.4587	43.2707	103.7812	68.1144	48.7313	101.4843	65.7081	44.3000	104.2549	73.1924	47.8393
$\hat{\phi}_1$	Mean	0.6758	0.6735	0.6586	0.6844	0.6845	0.6644	0.6820	0.6745	0.6671	0.6778	0.6741	0.6656
	SD	0.1552	0.1094	0.0719	0.1624	0.1176	0.0824	0.1697	0.1210	0.0821	0.1697	0.1244	0.0854
	MSE	24.7137	12.5052	5.2362	27.5311	15.0022	6.9886	29.8104	15.2265	7.0224	29.5532	16.0288	7.5347
$\hat{\kappa}$	Mean	-0.2606	-0.2555	-0.2529	-0.2602	-0.2562	-0.2535	-0.2620	-0.2537	-0.2520	-0.2601	-0.2542	-0.2517
	SD	0.0420	0.0293	0.0207	0.0518	0.0327	0.0240	0.0520	0.0348	0.0245	0.0486	0.0337	0.0235
	MSE	1.8774	0.8904	0.4374	2.7813	1.1076	0.5878	2.8501	1.2213	0.6014	2.4586	1.1528	0.5536
$\hat{\gamma}$	Mean	0.3455	0.3478	0.3493	0.3427	0.3476	0.3500	0.3409	0.3474	0.3474	0.3438	0.3452	0.3496
	SD	0.0698	0.0452	0.0327	0.0762	0.0501	0.0348	0.0751	0.0497	0.0366	0.0725	0.0490	0.0348
	MSE	4.8848	2.0455	1.0715	5.8471	2.5138	1.2106	5.7114	2.4778	1.3434	5.2940	2.4167	1.2099
$\hat{d}$	Mean	0.2358	0.2543	0.2821	0.2209	0.2400	0.2733	0.2230	0.2493	0.2724	0.2233	0.2508	0.2713
	SD	0.1500	0.1128	0.0709	0.1565	0.1237	0.0837	0.1601	0.1262	0.0825	0.1587	0.1253	0.0864
	MSE	26.5996	14.7896	5.3486	30.7272	18.8811	7.7152	31.5361	18.4873	7.5677	31.0322	18.0994	8.2786
$\hat{\nu}$	Mean	—	—	—	6.6159	6.2622	6.1483	—	—	—	—	—	—
	SD	—	—	—	1.8507	1.0031	0.7094	—	—	—	—	—	—
	MSE	—	—	—	3800.7918	1073.9544	524.8042	—	—	—	—	—	—
$\hat{\beta}$	Mean	—	—	—	—	—	—	1.4270	1.4108	1.4053	—	—	—
	SD	—	—	—	—	—	—	0.1129	0.0751	0.0552	—	—	—
	MSE	—	—	—	—	—	—	13.4709	5.7574	3.0705	—	—	—
$\hat{P}$	Mean	—	—	—	—	—	—	—	—	—	2.5570	2.3420	2.1990
	SD	—	—	—	—	—	—	—	—	—	1.2067	0.8437	0.5212
	MSE	—	—	—	—	—	—	—	—	—	1765.0000	828.0000	311.0000
$\hat{s}$	Mean	—	—	—	—	—	—	—	—	—	—	—	—
	SD	—	—	—	—	—	—	—	—	—	—	—	—
	MSE	—	—	—	—	—	—	—	—	—	—	—	—

		FIEGARCH$(1, d, 1)$											
		snorm			sstd			sged			sald		
Estimator	Measure	n_1	n_2	n_3	n_1	n_2	n_3	n_1	n_2	n_3	n_1	n_2	n_3
$\hat{\mu}$	Mean	0.0000	0.0001	0.0001	0.0001	0.0001	0.0001	0.0000	0.0001	0.0001	0.0000	0.0001	0.0001
	SD	0.0005	0.0003	0.0002	0.0004	0.0003	0.0002	0.0005	0.0003	0.0002	0.0005	0.0003	0.0002
	MSE	0.0002	0.0001	0.0000	0.0002	0.0001	0.0000	0.0002	0.0001	0.0000	0.0002	0.0001	0.0000
$\hat{\omega}_\sigma$	Mean	-8.4750	-8.4965	-8.4936	-8.5056	-8.4874	-8.4960	-8.4729	-8.4878	-8.4952	-8.4757	-8.4772	-8.4978
	SD	0.3269	0.2681	0.2134	0.3475	0.2789	0.2336	0.3353	0.2650	0.2275	0.3381	0.2954	0.2320
	MSE	107.3789	71.7956	45.5332	120.6541	77.8882	54.5440	113.0500	70.3132	51.7255	114.8012	87.7000	53.7651
$\hat{\phi}_1$	Mean	0.6718	0.6664	0.6607	0.6742	0.6800	0.6693	0.6682	0.6765	0.6612	0.6752	0.6694	0.6652
	SD	0.1454	0.1032	0.0689	0.1633	0.1086	0.0769	0.1549	0.1130	0.0740	0.1597	0.1111	0.0748
	MSE	21.5793	10.9079	4.8541	27.2156	12.6697	6.2723	24.2914	13.4585	5.5902	26.1135	12.7021	5.8270
$\hat{\kappa}$	Mean	-0.2599	-0.2566	-0.2525	-0.2619	-0.2534	-0.2522	-0.2626	-0.2579	-0.2529	-0.2640	-0.2554	-0.2526
	SD	0.0423	0.0287	0.0200	0.0500	0.0339	0.0240	0.0489	0.0333	0.0241	0.0478	0.0325	0.0228
	MSE	1.8847	0.8664	0.4058	2.6392	1.1595	0.5812	2.5475	1.1707	0.5890	2.4823	1.0820	0.5276
$\hat{\gamma}$	Mean	0.3441	0.3475	0.3487	0.3415	0.3455	0.3490	0.3393	0.3461	0.3483	0.3412	0.3450	0.3477
	SD	0.0686	0.0455	0.0313	0.0755	0.0507	0.0357	0.0747	0.0489	0.0350	0.0731	0.0502	0.0337
	MSE	4.7358	2.0762	0.9822	5.7641	2.5911	1.2724	5.6841	2.4001	1.2277	5.4103	2.5377	1.1377
$\hat{d}$	Mean	0.2407	0.2632	0.2813	0.2324	0.2502	0.2708	0.2405	0.2516	0.2798	0.2311	0.2608	0.2765
	SD	0.1472	0.1069	0.0687	0.1562	0.1158	0.0779	0.1521	0.1176	0.0736	0.1535	0.1139	0.0730
	MSE	25.1696	12.7703	5.0644	28.9385	15.8667	6.9193	26.6486	16.1499	5.8245	28.2935	14.4854	5.8757
$\hat{\nu}$	Mean	—	—	—	6.7505	6.3033	6.1167	—	—	—	—	—	—
	SD	—	—	—	1.9580	1.1053	0.7124	—	—	—	—	—	—
	MSE	—	—	—	4393.2588	1312.3886	520.6873	—	—	—	—	—	—
$\hat{\beta}$	Mean	—	—	—	—	—	—	1.4240	1.4095	1.4062	—	—	—
	SD	—	—	—	—	—	—	0.1145	0.0748	0.0547	—	—	—
	MSE	—	—	—	—	—	—	13.6798	5.6837	3.0306	—	—	—
$\hat{P}$	Mean	—	—	—	—	—	—	—	—	—	2.5240	2.3520	2.2120
	SD	—	—	—	—	—	—	—	—	—	1.2072	0.8383	0.5774
	MSE	—	—	—	—	—	—	—	—	—	1730.4609	826.0000	378.0000
$\hat{s}$	Mean	0.8954	0.9011	0.9002	0.8966	0.8998	0.8998	0.8980	0.9007	0.8999	0.8964	0.9001	0.9001
	SD	0.0500	0.0365	0.0249	0.0482	0.0328	0.0237	0.0430	0.0288	0.0192	0.0466	0.0315	0.0220
	MSE	2.5163	1.3319	0.6184	2.3353	1.0751	0.5606	1.8523	0.8278	0.3686	2.1793	0.9904	0.4849

Table 12: Simulation study results regarding FIMEGARCH$(1, d, 1)$. MSE-values are multiplied by 1000. True values are $\mu = 0.0001$, $\omega_\sigma = -8.5$, $\phi_1 = 0.65$, $\kappa = -0.25$, $\gamma = 0.35$, $d = 0.3$, $\nu = 6$, $\beta = 1.4$, $P = 2$ and $s = 0.9$.

		FIMEGARCH$(1, d, 1)$																							
		norm			std			ged			ald			snorm			sstd			sged			sald		
Estimator	Measure	n_1	n_2	n_3	n_1	n_2	n_3	n_1	n_2	n_3	n_1	n_2	n_3	n_1	n_2	n_3	n_1	n_2	n_3	n_1	n_2	n_3	n_1	n_2	n_3
$\hat{\mu}$	Mean	0.0001	0.0001	0.0001	0.0000	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0000	0.0001	0.0001	0.0001	0.0001	0.0001	0.0000	0.0001	0.0001	0.0000	0.0001	0.0001
	SD	0.0005	0.0003	0.0002	0.0004	0.0003	0.0002	0.0005	0.0003	0.0002	0.0005	0.0003	0.0002	0.0005	0.0003	0.0002	0.0005	0.0003	0.0002	0.0005	0.0003	0.0002	0.0005	0.0003	0.0002
	MSE	0.0002	0.0001	0.0001	0.0002	0.0001	0.0000	0.0002	0.0001	0.0000	0.0002	0.0001	0.0001	0.0003	0.0001	0.0001	0.0002	0.0001	0.0001	0.0003	0.0001	0.0000	0.0002	0.0001	0.0001
$\hat{\omega}_\sigma$	Mean	-8.4926	-8.5041	-8.5023	-8.4913	-8.4870	-8.4947	-8.4831	-8.5067	-8.4957	-8.5029	-8.4982	-8.5000	-8.4876	-8.5033	-8.4986	-8.5085	-8.4946	-8.4981	-8.4810	-8.4956	-8.4993	-8.4878	-8.4841	-8.4965
	SD	0.2451	0.2124	0.1706	0.2816	0.2258	0.1910	0.2748	0.2197	0.1804	0.2762	0.2333	0.1870	0.2654	0.2186	0.1739	0.2962	0.2288	0.1953	0.2796	0.2203	0.1909	0.2914	0.2463	0.1947
	MSE	60.0803	45.1070	29.0952	79.2700	51.1115	36.4543	75.7007	48.2683	32.5226	76.2101	54.3763	34.9359	70.5405	47.7524	30.2054	87.7364	52.3348	38.1183	78.4444	48.5218	36.4239	85.0046	60.8579	37.8799
$\hat{\phi}_1$	Mean	0.6878	0.6875	0.6687	0.6936	0.7016	0.6761	0.6942	0.6860	0.6808	0.6887	0.6860	0.6783	0.6792	0.6788	0.6705	0.6786	0.6932	0.6821	0.6822	0.6905	0.6736	0.6771	0.6808	0.6764
	SD	0.1795	0.1313	0.0935	0.1883	0.1417	0.1053	0.1949	0.1473	0.1042	0.1946	0.1488	0.1046	0.1781	0.1259	0.0872	0.1976	0.1335	0.0999	0.1872	0.1376	0.0965	0.1966	0.1368	0.0978
	MSE	33.6368	18.6407	9.0796	37.3255	22.7139	11.7492	39.8912	22.9755	11.7991	39.3465	23.4135	11.7305	32.5347	16.6506	8.0184	39.8190	19.6788	11.0042	36.0594	20.5412	9.8670	39.3444	19.6575	10.2493
$\hat{\kappa}$	Mean	-0.2641	-0.2581	-0.2543	-0.2653	-0.2582	-0.2552	-0.2662	-0.2558	-0.2531	-0.2638	-0.2563	-0.2526	-0.2632	-0.2600	-0.2537	-0.2665	-0.2546	-0.2535	-0.2662	-0.2620	-0.2550	-0.2704	-0.2580	-0.2544
	SD	0.0652	0.0451	0.0321	0.0839	0.0525	0.0379	0.0824	0.0551	0.0385	0.0772	0.0546	0.0365	0.0669	0.0458	0.0310	0.0799	0.0543	0.0379	0.0789	0.0532	0.0383	0.0775	0.0518	0.0360
	MSE	4.4522	2.0967	1.0479	7.2697	2.8167	1.4652	7.0498	3.0715	1.4923	6.1471	3.0191	1.3388	4.6449	2.1921	0.9759	6.6432	2.9638	1.4490	6.4741	2.9684	1.4914	6.4163	2.7435	1.3169
$\hat{\gamma}$	Mean	0.3505	0.3512	0.3511	0.3480	0.3513	0.3523	0.3470	0.3516	0.3492	0.3505	0.3489	0.3511	0.3496	0.3510	0.3500	0.3476	0.3486	0.3510	0.3452	0.3504	0.3502	0.3473	0.3481	0.3498
	SD	0.0732	0.0478	0.0351	0.0793	0.0534	0.0371	0.0779	0.0524	0.0391	0.0763	0.0518	0.0369	0.0716	0.0485	0.0334	0.0796	0.0535	0.0380	0.0776	0.0510	0.0377	0.0766	0.0524	0.0361
	MSE	5.3470	2.2855	1.2299	6.2888	2.8493	1.3813	6.0700	2.7420	1.5260	5.8145	2.6805	1.3607	5.1215	2.3521	1.1140	6.3300	2.8632	1.4445	6.0380	2.5943	1.4234	5.8656	2.7459	1.3044
$\hat{d}$	Mean	0.2075	0.2278	0.2644	0.1943	0.2087	0.2529	0.1924	0.2238	0.2503	0.1933	0.2259	0.2512	0.2161	0.2387	0.2661	0.2088	0.2241	0.2493	0.2072	0.2233	0.2597	0.2097	0.2370	0.2575
	SD	0.1658	0.1346	0.0959	0.1698	0.1457	0.1098	0.1733	0.1465	0.1075	0.1737	0.1461	0.1080	0.1666	0.1306	0.0886	0.1763	0.1388	0.1050	0.1701	0.1412	0.0994	0.1700	0.1371	0.0974
	MSE	36.0352	23.3267	10.4599	39.9689	29.5518	14.2490	41.5776	27.2385	14.0160	41.5241	26.8041	14.0429	34.7728	20.7887	8.9893	39.3490	25.0055	13.5763	37.5298	25.7985	11.4845	37.0288	22.7442	11.2875
$\hat{\nu}$	Mean	—	—	—	6.6183	6.2675	6.1510	—	—	—	—	—	—	—	—	—	6.7510	6.3126	6.1212	—	—	—	—	—	—
	SD	—	—	—	1.8454	1.0051	0.7095	—	—	—	—	—	—	—	—	—	1.9429	1.1066	0.7140	—	—	—	—	—	—
	MSE	—	—	—	3784.4592	1080.7431	525.7259	—	—	—	—	—	—	—	—	—	4334.9084	1321.0422	523.9231	—	—	—	—	—	—
$\hat{\beta}$	Mean	—	—	—	—	—	—	1.4277	1.4110	1.4056	—	—	—	—	—	—	—	—	—	1.4249	1.4100	1.4067	—	—	—
	SD	—	—	—	—	—	—	0.1129	0.0751	0.0552	—	—	—	—	—	—	—	—	—	0.1142	0.0748	0.0548	—	—	—
	MSE	—	—	—	—	—	—	13.5021	5.7616	3.0742	—	—	—	—	—	—	—	—	—	13.6375	5.6872	3.0399	—	—	—
$\hat{P}$	Mean	—	—	—	—	—	—	—	—	—	2.5640	2.3400	2.1980	—	—	—	—	—	—	—	—	—	2.5300	2.3520	2.2130
	SD	—	—	—	—	—	—	—	—	—	1.2064	0.8433	0.5148	—	—	—	—	—	—	—	—	—	1.2176	0.8359	0.5779
	MSE	—	—	—	—	—	—	—	—	—	1772.0000	826.0000	304.0000	—	—	—	—	—	—	—	—	—	1762.0000	822.0000	379.0000
$\hat{s}$	Mean	—	—	—	—	—	—	—	—	—	—	—	—	0.8958	0.9012	0.9002	0.8969	0.9000	0.9000	0.8986	0.9009	0.9001	0.8968	0.9004	0.9000
	SD	—	—	—	—	—	—	—	—	—	—	—	—	0.0500	0.0364	0.0249	0.0483	0.0329	0.0238	0.0433	0.0289	0.0194	0.0465	0.0316	0.0221
	MSE	—	—	—	—	—	—	—	—	—	—	—	—	2.5141	1.3222	0.6187	2.3399	1.0796	0.5674	1.8739	0.8347	0.3748	2.1691	0.9968	0.4879

Table 13: Simulation study results regarding FIMLog-GARCH$(1, d, 1)$. MSE-values are multiplied by 1000. True values are $\mu = 0.0001$, $\omega_\sigma = -8.5$, $\phi_1 = 0.65$, $\kappa = -0.25$, $\gamma = 0.35$, $d = 0.3$, $\nu = 6$, $\beta = 1.4$, $P = 2$ and $s = 0.9$.

		FIMLog-GARCH$(1, d, 1)$											
		norm			std			ged			ald		
Estimator	Measure	n_1	n_2	n_3	n_1	n_2	n_3	n_1	n_2	n_3	n_1	n_2	n_3
$\hat{\mu}$	Mean	0.0001	0.0001	0.0001	0.0000	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
	SD	0.0005	0.0003	0.0002	0.0004	0.0003	0.0002	0.0005	0.0003	0.0002	0.0005	0.0003	0.0002
	MSE	0.0002	0.0001	0.0001	0.0002	0.0001	0.0001	0.0002	0.0001	0.0001	0.0002	0.0001	0.0001
$\hat{\omega}_\sigma$	Mean	-8.4997	-8.5049	-8.5032	-8.4936	-8.4896	-8.4966	-8.4938	-8.5077	-8.4989	-8.5079	-8.5002	-8.5010
	SD	0.1828	0.1527	0.1245	0.2041	0.1573	0.1313	0.2028	0.1565	0.1257	0.2037	0.1659	0.1302
	MSE	33.3697	23.3263	15.4982	41.6525	24.8208	17.2326	41.1221	24.5393	15.7974	41.5312	27.4823	16.9306
$\hat{\phi}_1$	Mean	0.6534	0.6718	0.6638	0.6575	0.6806	0.6661	0.6646	0.6667	0.6690	0.6476	0.6767	0.6691
	SD	0.2171	0.1521	0.1023	0.2485	0.1790	0.1214	0.2333	0.1837	0.1210	0.2513	0.1729	0.1195
	MSE	47.0948	23.5780	10.6389	61.7451	32.9577	14.9805	54.6035	33.9940	14.9927	63.0986	30.5858	14.6318
$\hat{\kappa}$	Mean	-0.2686	-0.2599	-0.2545	-0.2704	-0.2621	-0.2574	-0.2705	-0.2595	-0.2545	-0.2685	-0.2587	-0.2544
	SD	0.0659	0.0449	0.0322	0.0877	0.0542	0.0392	0.0851	0.0566	0.0394	0.0805	0.0547	0.0370
	MSE	4.6817	2.1076	1.0561	8.0991	3.0801	1.5885	7.6510	3.2884	1.5731	6.8242	3.0669	1.3889
$\hat{\gamma}$	Mean	0.3396	0.3437	0.3471	0.3195	0.3420	0.3486	0.3228	0.3426	0.3453	0.3267	0.3366	0.3484
	SD	0.1289	0.0816	0.0575	0.1696	0.0961	0.0665	0.1545	0.0928	0.0677	0.1535	0.0931	0.0632
	MSE	16.6930	6.6913	3.3138	29.6576	9.2841	4.4178	24.5957	8.6519	4.6044	24.0852	8.8449	3.9988
$\hat{d}$	Mean	0.2323	0.2401	0.2696	0.2168	0.2197	0.2582	0.2145	0.2354	0.2601	0.2238	0.2305	0.2568
	SD	0.1710	0.1356	0.0934	0.1924	0.1533	0.1114	0.1853	0.1507	0.1065	0.1934	0.1517	0.1102
	MSE	33.7802	21.9609	9.6397	43.8880	29.9286	14.1402	41.6256	26.8466	12.9264	43.1872	27.8062	13.9968
$\hat{\nu}$	Mean	—	—	—	6.6301	6.2652	6.1518	—	—	—	—	—	—
	SD	—	—	—	1.8548	0.9999	0.7085	—	—	—	—	—	—
	MSE	—	—	—	3833.8717	1069.2349	524.5408	—	—	—	—	—	—
$\hat{\beta}$	Mean	—	—	—	—	—	—	1.4274	1.4111	1.4056	—	—	—
	SD	—	—	—	—	—	—	0.1125	0.0753	0.0551	—	—	—
	MSE	—	—	—	—	—	—	13.3984	5.7832	3.0648	—	—	—
$\hat{P}$	Mean	—	—	—	—	—	—	—	—	—	2.5782	2.3501	2.1982
	SD	—	—	—	—	—	—	—	—	—	1.2148	0.8460	0.5247
	MSE	—	—	—	—	—	—	—	—	—	1808.6172	837.5125	314.3143
$\hat{s}$	Mean	—	—	—	—	—	—	—	—	—	—	—	—
	SD	—	—	—	—	—	—	—	—	—	—	—	—
	MSE	—	—	—	—	—	—	—	—	—	—	—	—

		FIMLog-GARCH$(1, d, 1)$											
		snorm			sstd			sged			sald		
Estimator	Measure	n_1	n_2	n_3	n_1	n_2	n_3	n_1	n_2	n_3	n_1	n_2	n_3
$\hat{\mu}$	Mean	0.0000	0.0001	0.0001	0.0001	0.0001	0.0001	0.0000	0.0001	0.0001	0.0000	0.0001	0.0001
	SD	0.0005	0.0003	0.0002	0.0005	0.0003	0.0002	0.0005	0.0003	0.0002	0.0005	0.0004	0.0002
	MSE	0.0003	0.0001	0.0001	0.0002	0.0001	0.0001	0.0003	0.0001	0.0001	0.0002	0.0001	0.0001
$\hat{\omega}_\sigma$	Mean	-8.4935	-8.5008	-8.5002	-8.5070	-8.4957	-8.4997	-8.4907	-8.4961	-8.5006	-8.4946	-8.4891	-8.4977
	SD	0.1999	0.1581	0.1260	0.2113	0.1662	0.1358	0.2118	0.1578	0.1319	0.2086	0.1740	0.1348
	MSE	39.9712	24.9738	15.8514	44.6508	27.6045	18.4216	44.8806	24.8955	17.3739	43.5121	30.3815	18.1598
$\hat{\phi}_1$	Mean	0.6525	0.6616	0.6652	0.6538	0.6812	0.6765	0.6443	0.6742	0.6642	0.6406	0.6698	0.6696
	SD	0.2093	0.1471	0.0977	0.2340	0.1643	0.1170	0.2377	0.1655	0.1125	0.2445	0.1673	0.1146
	MSE	43.7751	21.7526	9.7711	54.7101	27.9434	14.3772	56.4752	27.9389	12.8486	59.7858	28.3483	13.5132
$\hat{\kappa}$	Mean	-0.2664	-0.2619	-0.2544	-0.2706	-0.2575	-0.2556	-0.2705	-0.2644	-0.2565	-0.2738	-0.2606	-0.2561
	SD	0.0672	0.0454	0.0311	0.0821	0.0552	0.0382	0.0808	0.0540	0.0389	0.0789	0.0514	0.0366
	MSE	4.7746	2.2022	0.9838	7.1670	3.0943	1.4899	6.9469	3.1255	1.5549	6.7862	2.7549	1.3740
$\hat{\gamma}$	Mean	0.3328	0.3425	0.3469	0.3184	0.3369	0.3478	0.3147	0.3413	0.3472	0.3192	0.3387	0.3473
	SD	0.1326	0.0820	0.0547	0.1611	0.0985	0.0669	0.1629	0.0911	0.0642	0.1602	0.0922	0.0628
	MSE	17.8486	6.7770	2.9989	26.9368	9.8681	4.4743	27.7587	8.3597	4.1274	26.5939	8.6200	3.9433
$\hat{d}$	Mean	0.2361	0.2526	0.2694	0.2260	0.2296	0.2503	0.2359	0.2347	0.2655	0.2351	0.2402	0.2598
	SD	0.1747	0.1310	0.0904	0.1900	0.1483	0.1093	0.1911	0.1451	0.1029	0.1876	0.1455	0.1007
	MSE	34.5580	19.3798	9.1046	41.5374	26.9153	14.4021	40.5797	25.2956	11.7692	39.3773	24.7311	11.7532
$\hat{\nu}$	Mean	—	—	—	6.7692	6.3211	6.1238	—	—	—	—	—	—
	SD	—	—	—	1.9490	1.1029	0.7144	—	—	—	—	—	—
	MSE	—	—	—	4386.6074	1318.2287	525.1536	—	—	—	—	—	—
$\hat{\beta}$	Mean	—	—	—	—	—	—	1.4257	1.4102	1.4068	—	—	—
	SD	—	—	—	—	—	—	0.1149	0.0748	0.0545	—	—	—
	MSE	—	—	—	—	—	—	13.8442	5.6973	3.0125	—	—	—
$\hat{P}$	Mean	—	—	—	—	—	—	—	—	—	2.5281	2.3560	2.2130
	SD	—	—	—	—	—	—	—	—	—	1.2269	0.8438	0.5692
	MSE	—	—	—	—	—	—	—	—	—	1782.5651	838.0000	369.0000
$\hat{s}$	Mean	0.8954	0.9011	0.9002	0.8961	0.9000	0.8999	0.8979	0.9009	0.9001	0.8957	0.9004	0.9000
	SD	0.0503	0.0364	0.0250	0.0490	0.0330	0.0239	0.0439	0.0290	0.0193	0.0470	0.0316	0.0222
	MSE	2.5497	1.3235	0.6221	2.4112	1.0903	0.5701	1.9310	0.8429	0.3722	2.2227	1.0005	0.4909

Table 14: Simulation study results regarding FILog-GARCH$(1, d, 1)$. MSE-values are multiplied by 1000. True values are $\mu = 0.0001$, $\omega_\sigma = -8.5$, $\phi_1 = 0.21$, $\psi_1 = -0.51$, $d = 0.3$, $\nu = 6$, $\beta = 1.4$, $P = 2$ and $s = 0.9$.

		FILog-GARCH$(1, d, 1)$																							
		norm			std			ged			ald			snorm			sstd			sged			sald		
Estimator	Measure	n_1	n_2	n_3	n_1	n_2	n_3	n_1	n_2	n_3	n_1	n_2	n_3	n_1	n_2	n_3	n_1	n_2	n_3	n_1	n_2	n_3	n_1	n_2	n_3
$\hat{\mu}$	Mean	0.0001	0.0001	0.0001	0.0001	0.0001	0.0000	0.0002	0.0001	0.0002	0.0001	0.0001	0.0001	-0.0000	0.0001	0.0001	0.0003	0.0001	0.0002	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
	SD	0.0022	0.0023	0.0023	0.0014	0.0014	0.0014	0.0022	0.0022	0.0021	0.0011	0.0010	0.0008	0.0023	0.0023	0.0022	0.0015	0.0014	0.0014	0.0022	0.0022	0.0021	0.0011	0.0009	0.0008
	MSE	0.0049	0.0055	0.0053	0.0021	0.0020	0.0019	0.0049	0.0047	0.0043	0.0011	0.0009	0.0006	0.0052	0.0052	0.0050	0.0022	0.0020	0.0019	0.0050	0.0047	0.0045	0.0012	0.0009	0.0007
$\hat{\omega}_\sigma$	Mean	-8.4243	-8.4293	-8.4202	-8.4714	-8.4623	-8.4673	-8.4321	-8.4743	-8.4792	-8.4724	-8.4680	-8.4540	-8.4198	-8.4258	-8.4014	-8.4407	-8.4271	-8.4262	-8.3978	-8.4107	-8.4093	-8.4421	-8.4299	-8.4478
	SD	0.3375	0.2882	0.2436	0.3683	0.3084	0.2736	0.3866	0.3398	0.2906	0.3749	0.3385	0.3090	0.3464	0.2919	0.2508	0.3628	0.3157	0.2747	0.4025	0.3359	0.3209	0.3838	0.3381	0.2956
	MSE	119.5384	87.9821	65.6322	136.2984	96.4533	75.8288	153.9374	115.9749	84.7928	141.1356	115.4733	97.5310	126.2800	90.6478	72.5756	135.0232	104.8740	80.8262	172.2950	120.6829	111.0811	150.5045	119.1194	90.0063
$\hat{\phi}_1$	Mean	0.2487	0.2520	0.2532	0.2414	0.2402	0.2419	0.2495	0.2498	0.2482	0.2272	0.2257	0.2242	0.2508	0.2503	0.2539	0.2397	0.2432	0.2410	0.2468	0.2468	0.2496	0.2213	0.2217	0.2250
	SD	0.1284	0.0984	0.0782	0.1454	0.0995	0.0729	0.1395	0.1013	0.0598	0.1266	0.0705	0.0519	0.1329	0.0876	0.0808	0.1508	0.0982	0.0714	0.1594	0.0955	0.0678	0.1255	0.0710	0.0501
	MSE	17.9762	11.4465	7.9718	22.1204	10.8132	6.3199	21.0158	11.8351	5.0323	16.2957	5.2121	2.8926	19.2946	9.2952	8.4459	23.5972	10.7378	6.0505	26.7335	10.4645	6.1688	15.8637	5.1708	2.7366
$\hat{\psi}_1$	Mean	-0.4914	-0.5026	-0.5072	-0.4927	-0.5024	-0.5099	-0.4919	-0.5042	-0.5112	-0.4888	-0.5028	-0.5063	-0.4954	-0.4966	-0.5092	-0.4882	-0.5050	-0.5086	-0.4853	-0.4951	-0.5068	-0.4866	-0.4968	-0.5071
	SD	0.1443	0.1083	0.0839	0.1617	0.1090	0.0756	0.1571	0.1162	0.0673	0.1495	0.0875	0.0614	0.1489	0.1007	0.0814	0.1729	0.1075	0.0762	0.1817	0.1069	0.0727	0.1531	0.0883	0.0612
	MSE	21.1568	11.7701	7.0351	26.4301	11.9375	5.7066	24.9856	13.5145	4.5201	22.7753	7.7006	3.7847	22.3695	10.3076	6.6143	30.3459	11.5715	5.8065	33.6056	11.6490	5.2835	23.9748	7.9646	3.7524
$\hat{d}$	Mean	0.2551	0.2608	0.2639	0.2619	0.2707	0.2765	0.2547	0.2654	0.2718	0.2697	0.2812	0.2865	0.2563	0.2568	0.2648	0.2590	0.2699	0.2757	0.2506	0.2582	0.2660	0.2716	0.2794	0.2863
	SD	0.0716	0.0523	0.0424	0.0770	0.0539	0.0419	0.0757	0.0567	0.0435	0.0714	0.0506	0.0347	0.0708	0.0532	0.0416	0.0836	0.0544	0.0417	0.0790	0.0560	0.0443	0.0736	0.0487	0.0353
	MSE	7.1411	4.2657	3.1019	7.3771	3.7646	2.3097	7.7851	4.4125	2.6816	6.0063	2.9151	1.3847	6.9227	4.6923	2.9701	8.6627	3.8612	2.3250	8.6762	4.8768	3.1210	6.2128	2.7906	1.4326
$\hat{\nu}$	Mean	—	—	—	6.5563	6.1709	6.0734	—	—	—	—	—	—	—	—	—	6.3802	6.0199	5.8270	—	—	—	—	—	—
	SD	—	—	—	1.8491	0.9747	0.6995	—	—	—	—	—	—	—	—	—	1.9839	1.0313	0.7191	—	—	—	—	—	—
	MSE	—	—	—	3725.2367	978.3291	494.2557	—	—	—	—	—	—	—	—	—	4076.5145	1062.8616	546.5324	—	—	—	—	—	—
$\hat{\beta}$	Mean	—	—	—	—	—	—	1.4364	1.4151	1.4079	—	—	—	—	—	—	—	—	—	1.3858	1.3637	1.3603	—	—	—
	SD	—	—	—	—	—	—	0.1225	0.0851	0.0616	—	—	—	—	—	—	—	—	—	0.1303	0.0965	0.0793	—	—	—
	MSE	—	—	—	—	—	—	16.3217	7.4568	3.8496	—	—	—	—	—	—	—	—	—	17.1535	10.6124	7.8625	—	—	—
$\hat{P}$	Mean	—	—	—	—	—	—	—	—	—	2.6445	2.5030	2.5095	—	—	—	—	—	—	—	—	—	2.5342	2.6220	2.4830
	SD	—	—	—	—	—	—	—	—	—	1.3158	1.2666	1.1724	—	—	—	—	—	—	—	—	—	1.2974	1.2801	1.1803
	MSE	—	—	—	—	—	—	—	—	—	2145.0151	1855.7114	1632.6326	—	—	—	—	—	—	—	—	—	1966.8008	2024.0000	1625.0000
$\hat{s}$	Mean	—	—	—	—	—	—	—	—	—	—	—	—	0.8979	0.9041	0.9031	0.9063	0.9068	0.9084	0.9085	0.9091	0.9064	0.9006	0.9032	0.9030
	SD	—	—	—	—	—	—	—	—	—	—	—	—	0.0545	0.0417	0.0306	0.0686	0.0578	0.0550	0.0854	0.0817	0.0761	0.0541	0.0384	0.0302
	MSE	—	—	—	—	—	—	—	—	—	—	—	—	2.9702	1.7543	0.9478	4.7339	3.3854	3.0961	7.3533	6.7585	5.8259	2.9296	1.4836	0.9192

B Simulation graphics

The following pages contain the density figures of the (conditional) QML-estimators of the EGF from the simulation study of Section 6. Each main figure contains the results for one of the parameter estimators. Black vertical lines show true parameter values.

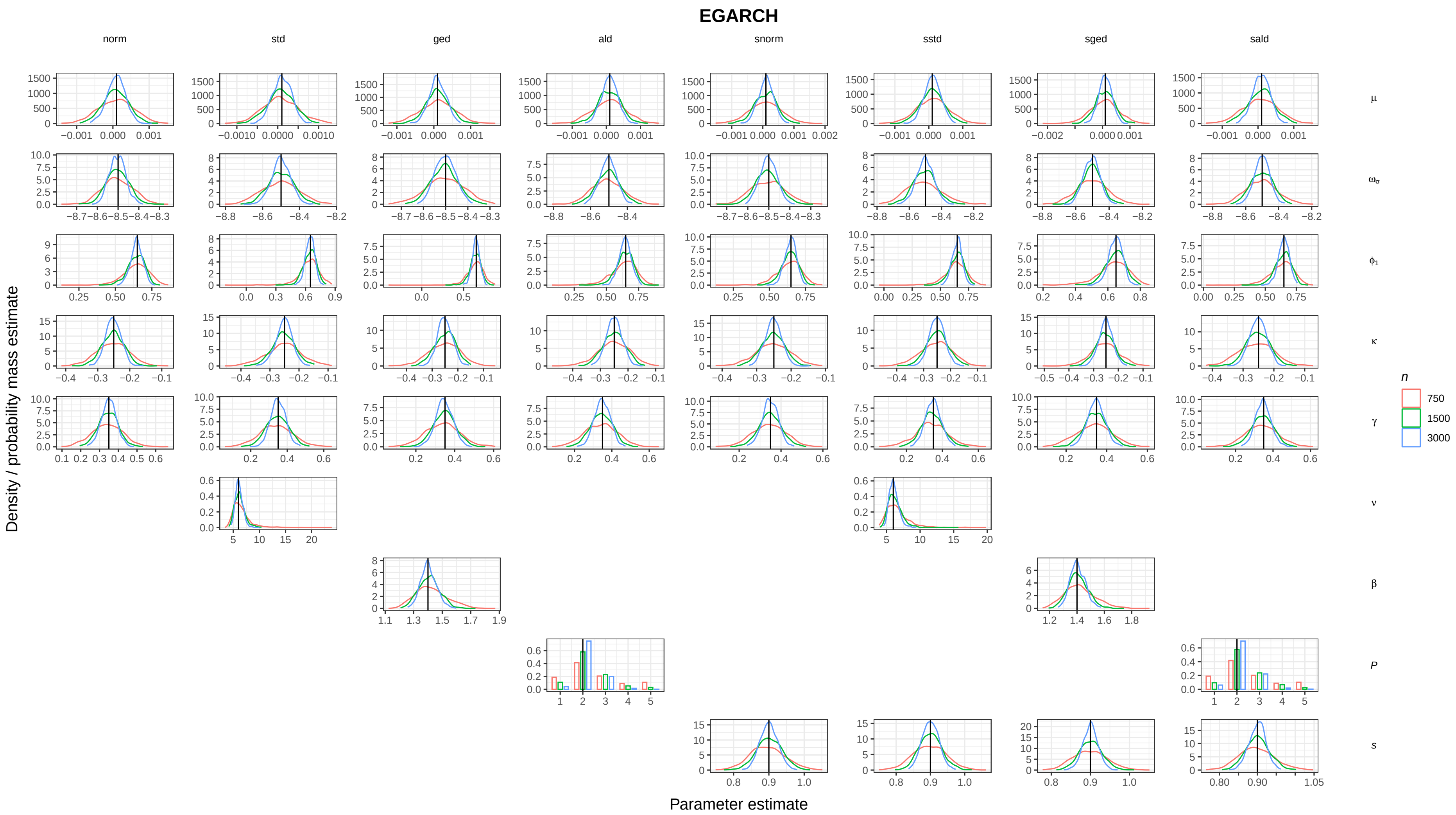

Figure 8: Simulated distributions of the EGARCH$(1, 1)$-parameter-estimators.

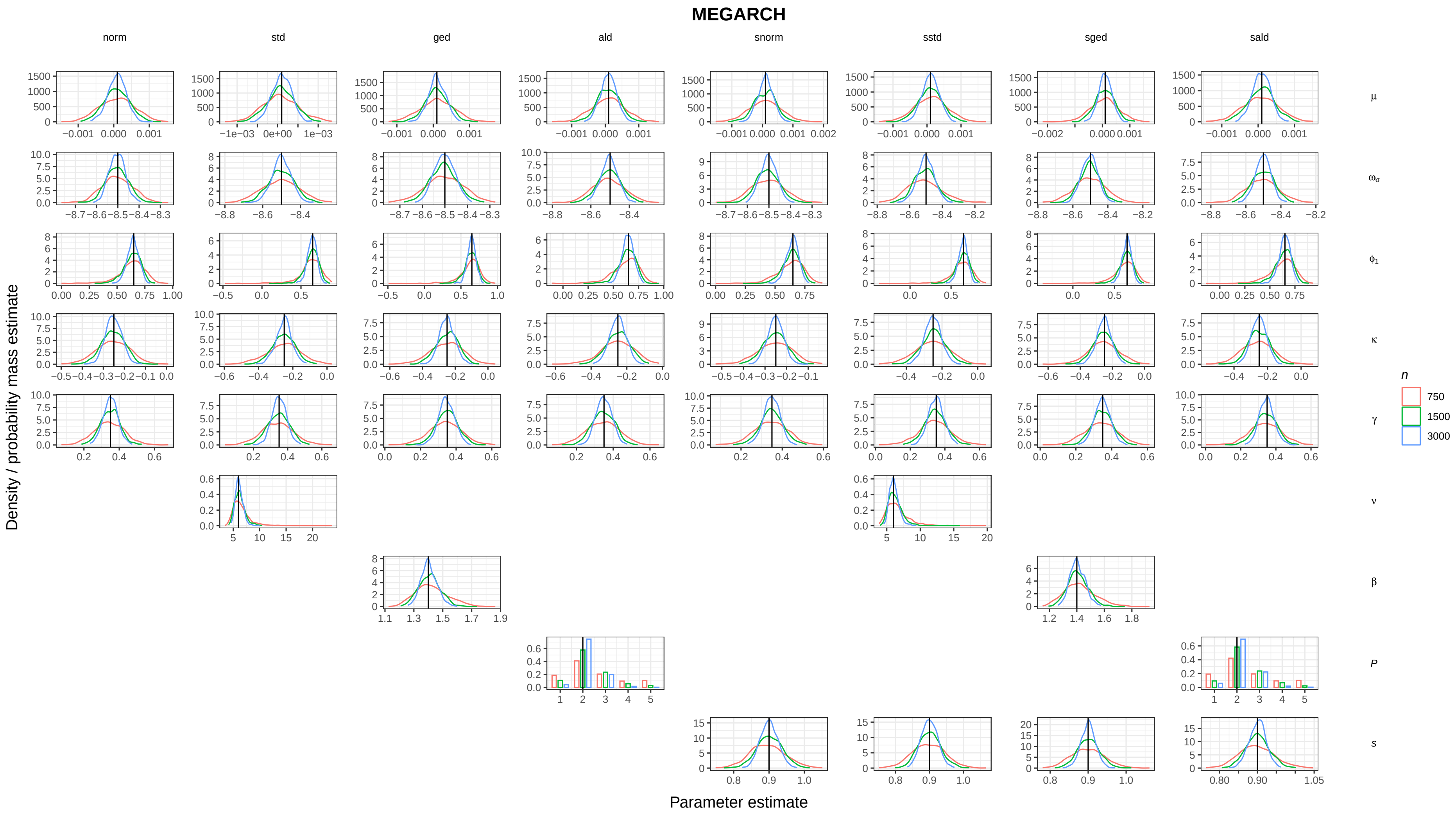

Figure 9: Simulated distributions of the MEGARCH$(1, 1)$-parameter-estimators.

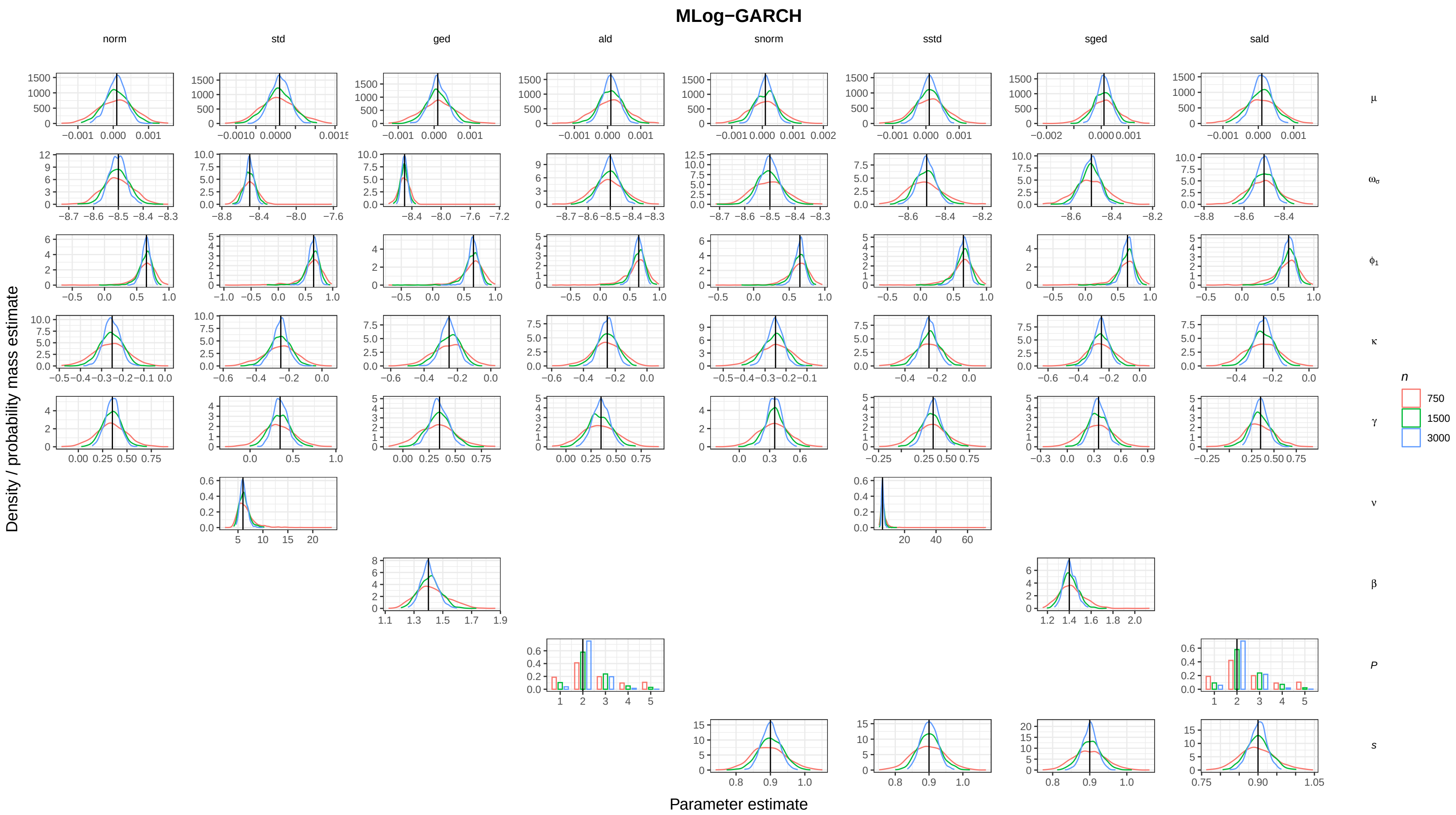

Figure 10: Simulated distributions of the MLog-GARCH$(1,1)$-parameter-estimators.

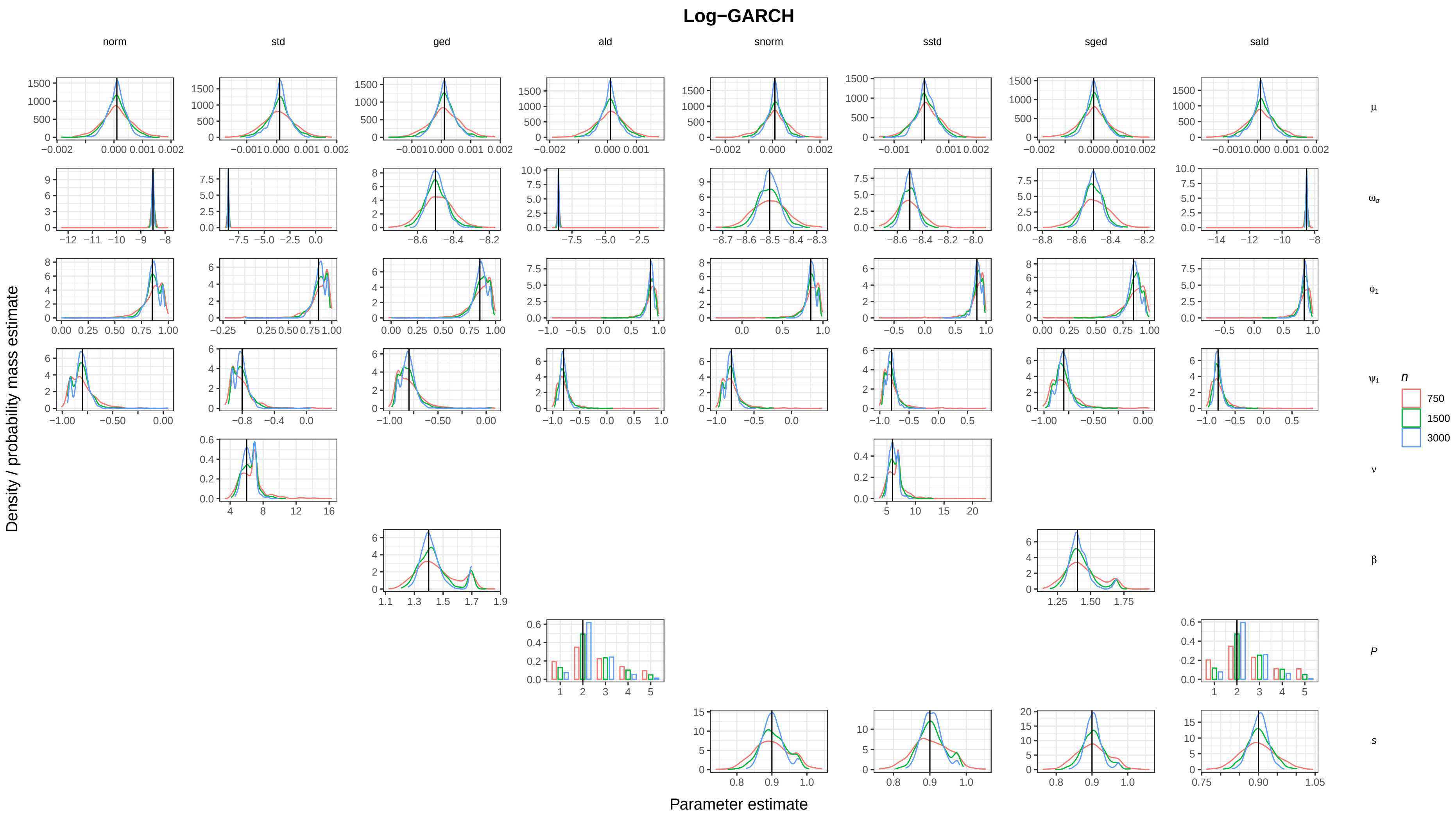

Figure 11: Simulated distributions of the Log-GARCH(1, 1)-parameter-estimators.

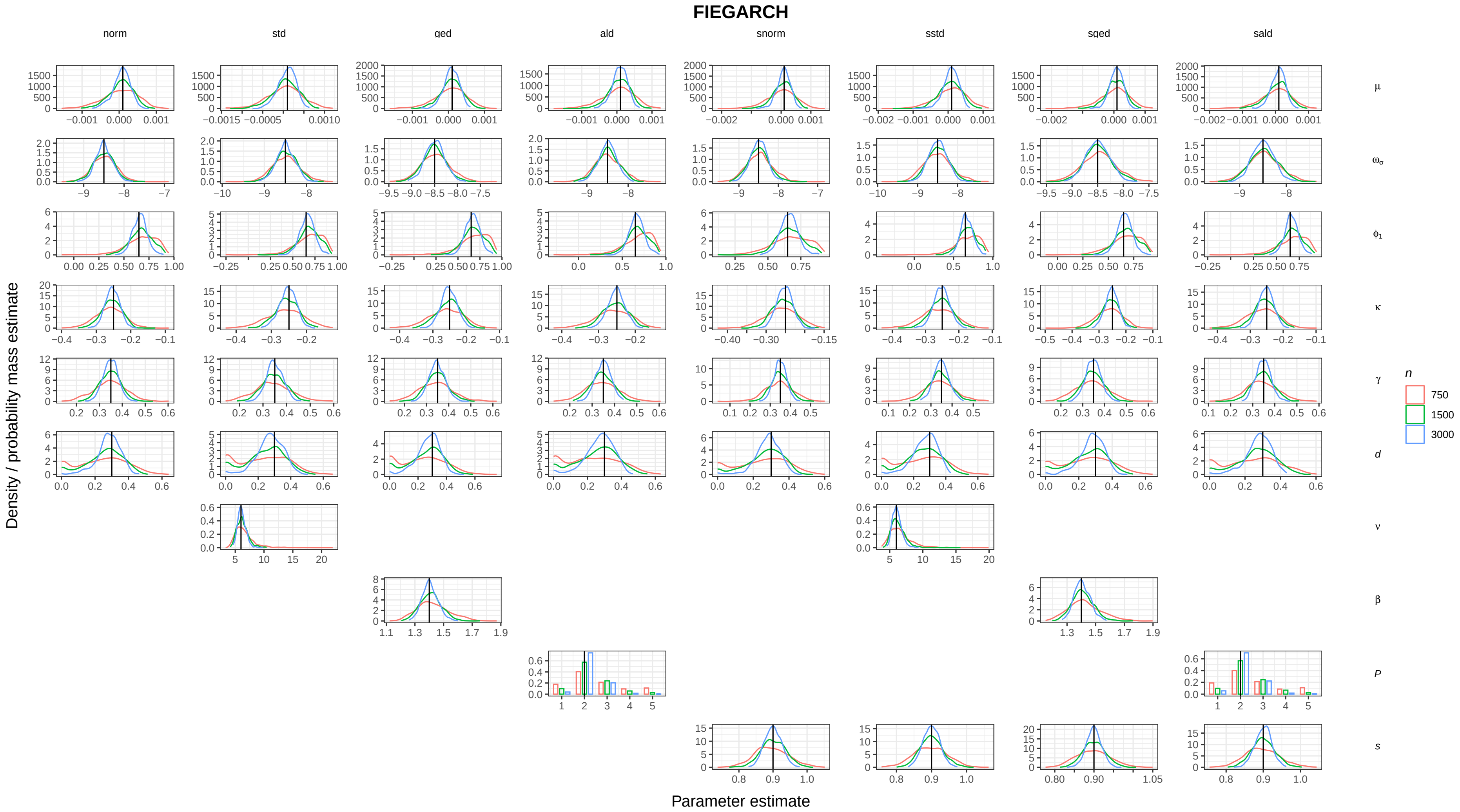

Figure 12: Simulated distributions of the FIEGARCH$(1, d, 1)$-parameter-estimators.

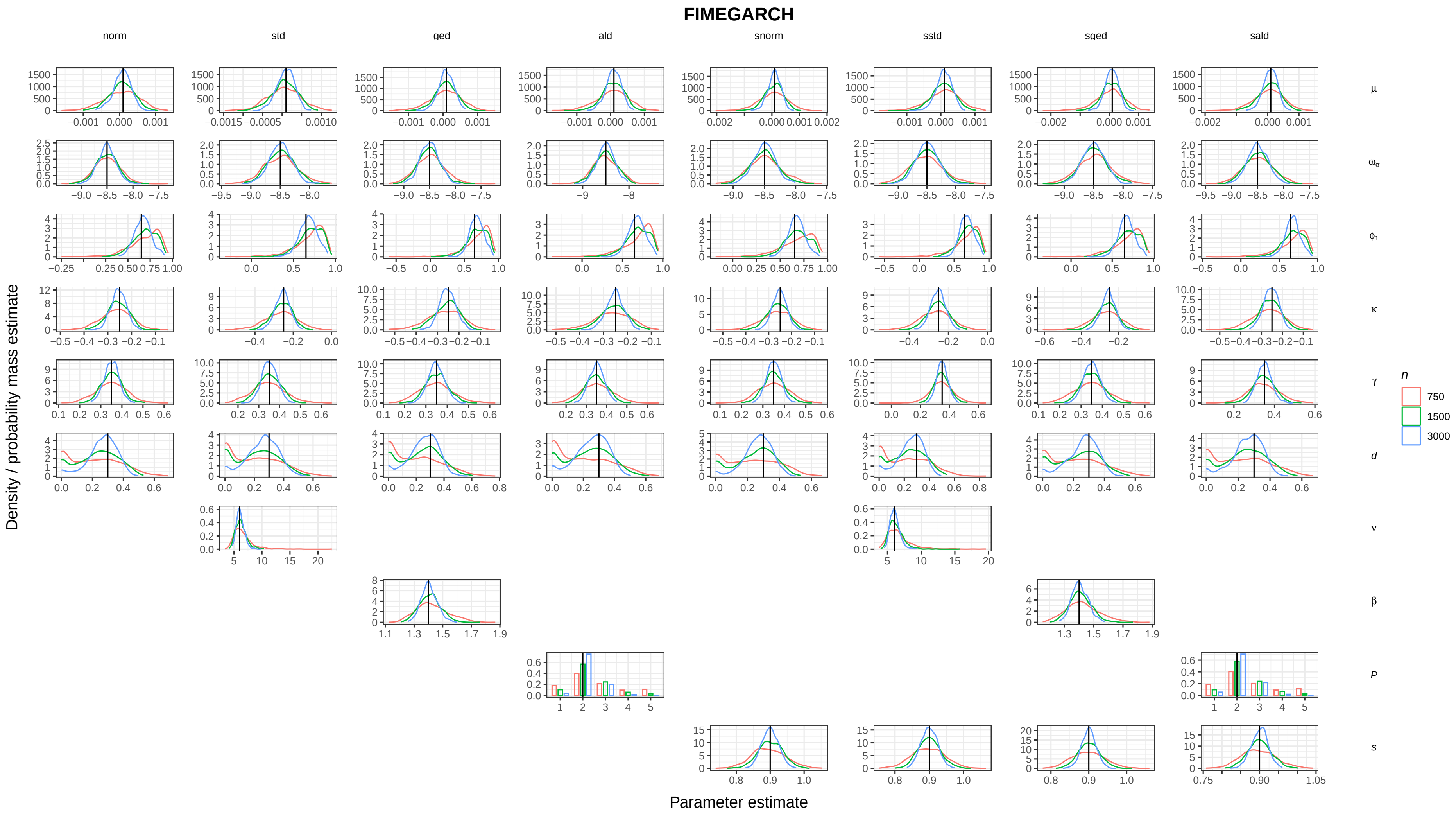

Figure 13: Simulated distributions of the FIMEGARCH$(1, d, 1)$-parameter-estimators.

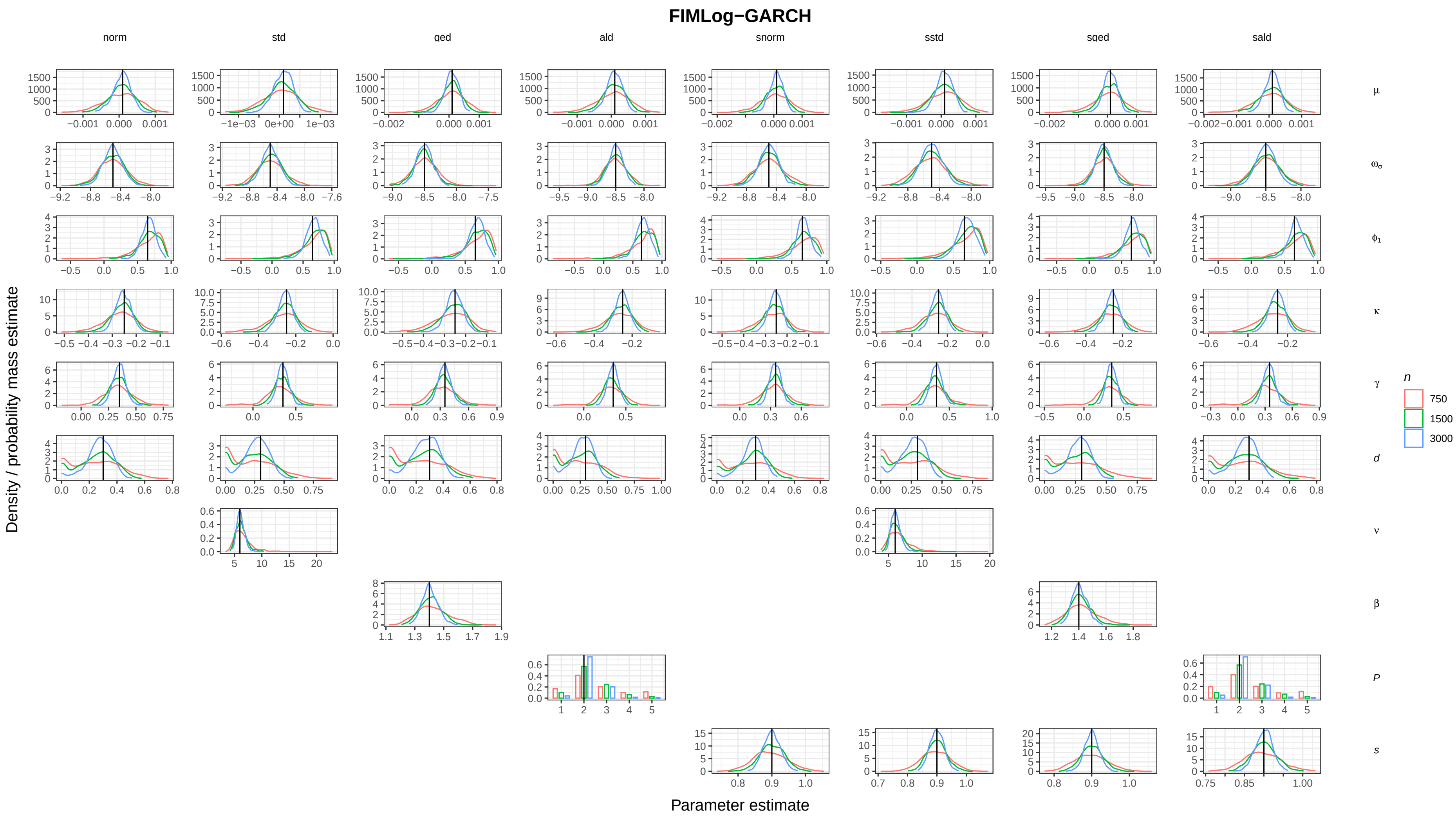

Figure 14: Simulated distributions of the FIMLog-GARCH$(1, d, 1)$-parameter-estimators.

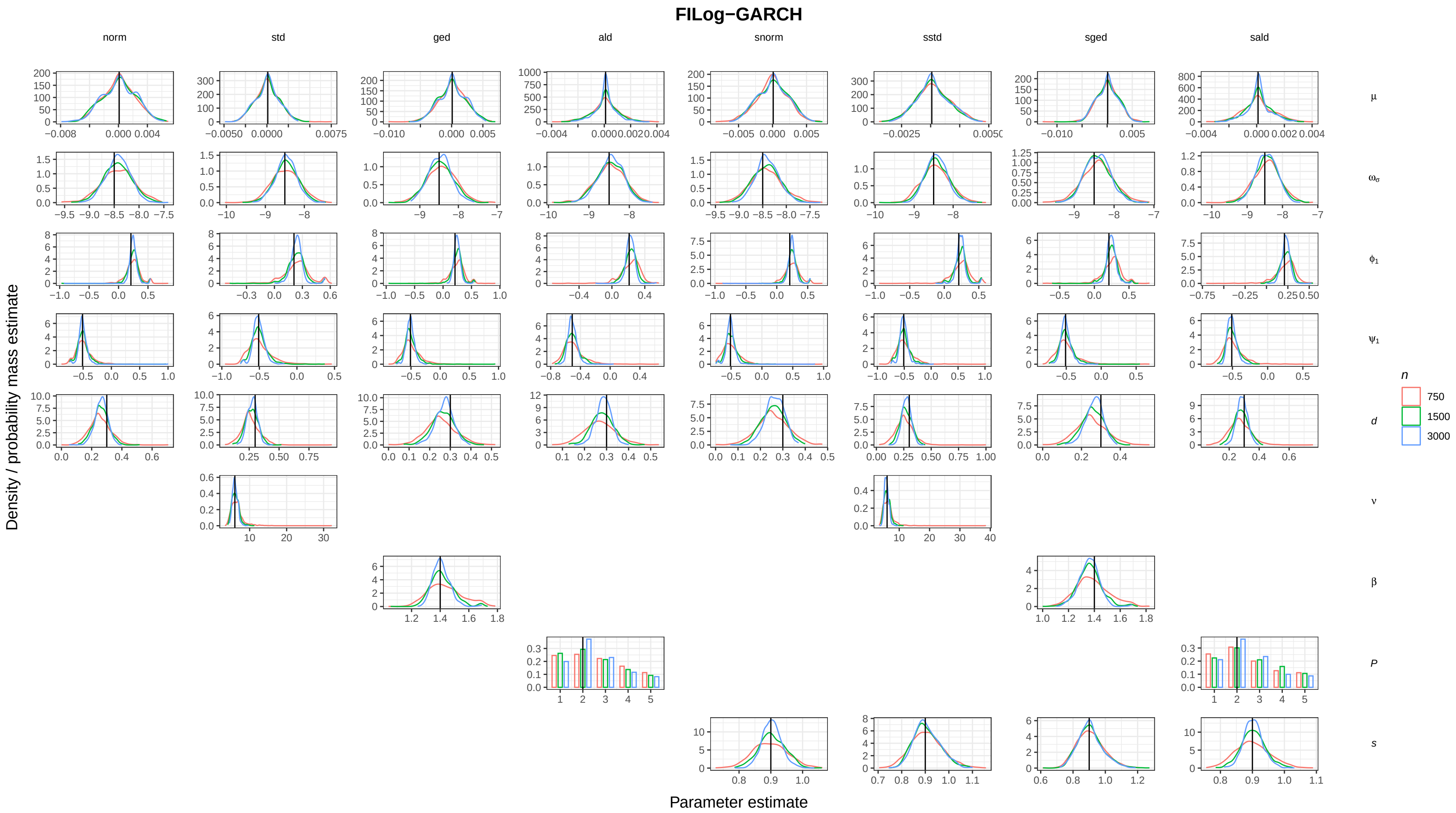

Figure 15: Simulated distributions of the FILog-GARCH$(1, d, 1)$-parameter-estimators.

C Further theoretical information on fEGarch

C.1 Available conditional distributions in fEGarch

Symmetric innovation distributions

In the fEGarch package, the four available basic conditional innovation distributions are the normal distribution, the Student t-distribution, the generalized error GED, and the ALD, all scaled to have variance one and with mean set to zero. Skewed versions of these conditional distributions will be discussed further below.

Assume that η is a continuous random variable with mean zero and variance one - for simplicity without time index - following any of the four distributions here denoted with probability density function (pdf) $f_\eta(\cdot)$. Define then $y = \mu + \sigma\eta$ with $\mu \in \mathbb{R}$ and $\sigma \in \{x \in \mathbb{R} : x > 0\}$ as two constants. y must have density $f_y(x, \mu, \sigma) = \sigma^{-1} f_\eta\left(\frac{x-\mu}{\sigma}\right)$, which corresponds to

$$f_y^{\text{norm}}(x, \mu, \sigma) = \frac{\sigma^{-1}}{\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \tag{28}$$

for a conditional normal distribution, to

$$f_y^{\text{std}}(x, \mu, \sigma) = \frac{\sigma^{-1}\Gamma\left(\frac{\nu+1}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)\sqrt{\pi(\nu-2)}} \left[1 + \frac{1}{\nu-2}\left(\frac{x-\mu}{\sigma}\right)^2\right]^{-\frac{\nu+1}{2}}, \tag{29}$$

where $\Gamma(\cdot)$ is the gamma function and $\nu > 2$ are the degrees of freedom, in case of a conditional t-distribution, for a conditional GED to

$$f_y^{\text{ged}}(x, \mu, \sigma) = \frac{\sigma^{-1}\beta}{2}\sqrt{\frac{C_{\Gamma,3}}{C_{\Gamma,1}^3}} e^{-\left|\frac{x-\mu}{\sigma}\right|^\beta \left(\frac{C_{\Gamma,3}}{C_{\Gamma,1}}\right)^{\frac{\beta}{2}}} \tag{30}$$

with constants $C_{\Gamma,i} = \Gamma\left(\frac{i}{\beta}\right)$, $i \in \{1, 3\}$, and with shape $\beta > 0$, and to

$$f_y^{\text{ald}}(x, \mu, \sigma) = \frac{\sigma^{-1}\iota B}{2} e^{-\iota\left|\frac{x-\mu}{\sigma}\right|} \sum_{j=0}^{P} c_j \left(\iota\left|\frac{x-\mu}{\sigma}\right|\right)^j \tag{31}$$

for an ALD with scale $\iota = \sqrt{2(P+1)}$ for unit-variance of η,

$$B = 2^{-2P}\binom{2P}{P}, \quad P \geq 0, \quad \text{and} \tag{32}$$

$$c_j = \frac{2(P-j+1)}{j(2P-j+1)} c_{j-1}, \quad j = 2, 3, \ldots, P, \tag{33}$$

alongside $c_0 = c_1 = 1$. In the `fEGarch` package, the case $P = 0$ is usually not considered. As described in Missiakoulis (1983) and Tse (1987), therein however denoted as Sargan densities, the ALD can be understood as the distribution of the arithmetic mean of $P + 1$ independent and identically Laplace distributed random variables.

The K-th absolute moment $E(|\eta|^K)$, $K \in \{x \in \mathbb{R} : x > -1\}$, of a random variable η following a normal distribution, a t-distribution, a GED and an ALD, all with mean zero and variance one, can be derived to be

$$a^{\text{norm}}(K) = \frac{2^{K/2}}{\sqrt{\pi}}\Gamma\left(\frac{K+1}{2}\right), \tag{34}$$

$$a^{\text{std}}(K) = \frac{(\nu-2)^{K/2}\Gamma\left(\frac{K+1}{2}\right)\Gamma\left(\frac{\nu-K}{2}\right)}{\sqrt{\pi}\Gamma\left(\frac{\nu}{2}\right)}, \tag{35}$$

$$a^{\text{ged}}(K) = \frac{\Gamma\left(\frac{K+1}{\beta}\right)}{\left[\Gamma\left(\frac{3}{\beta}\right)\right]^{K/2}\left[\Gamma\left(\frac{1}{\beta}\right)\right]^{(2-K)/2}} \quad \text{and} \tag{36}$$

$$a^{\text{ald}}(K) = \frac{B}{[2(P+1)]^{K/2}}\sum_{j=0}^{P} c_j\Gamma(j+K+1) \tag{37}$$

under usage of the definitions of the commonly known gamma and beta functions. In general, these functions are relevant in constructing the standardized skewed innovation distributions following Fernández and Steel (1998) as will be shown below and for deriving some characteristics of the EGF as explained in Appendix C.2.

Skewed innovation distributions

Fernández and Steel (1998) introduced a unified approach to create skewed versions of otherwise symmetric distributions with one mode. Given some skew parameter $s \in \mathbb{R}^+$ and the symmetric density f_η of a standardized continuous random variable with mean zero and variance one, the corresponding skewed density is

$$f_s(x) = \frac{2s}{s^2+1}\left[f_\eta(sx)\mathbb{I}_{(-\infty,0]}(x) + f_\eta\left(\frac{x}{s}\right)\mathbb{I}_{(0,\infty)}(x)\right], \tag{38}$$

where $\mathbb{I}_{(-\infty,0]}(x)$ and $\mathbb{I}_{(0,\infty)}(x)$ are equal to one, if x lies within the interval in the corresponding subscript, and zero otherwise. Note that (38) is however usually not yet for a random variable with mean zero and variance one.

In fact, given that the underlying $f_\eta(\cdot)$ reflects mean zero and variance one, the theoretical mean C_E and variance C_V of the corresponding skewed random variable that are required for standardization of $f_s(\cdot)$ are

$$C_E = a_j(1)\frac{s^2-1}{s} \qquad \text{and} \tag{39}$$

$$C_V = (1-[a_j(1)]^2)\frac{s^4+1}{s^2} + 2[a_j(1)]^2 - 1 \tag{40}$$

following Fernández and Steel (1998), where $a_j(1)$ is one of the functions (34) to (37) evaluated at $K = 1$.

The pdf relevant in context of GARCH fitting under a skewed conditional distribution is then

$$f_y^{\text{skew}}(x,\mu,\sigma) = \sigma^{-1}\sqrt{C_V} \times f_s\left[\sqrt{C_V}\left(\frac{x-\mu}{\sigma}\right) + C_E\right]. \tag{41}$$

C.2 Characteristics of selected EGARCH-family members

Given a process from the EGF, its conditional variance equation is

$$\sigma_t^2 = \exp\{\omega_\sigma\}\prod_{i=0}^{\infty}\exp\{\theta_i g(\eta_{t-i-1})\}. \tag{42}$$

The result (42) is identical to that of the basic EGARCH (Francq and Zakoïan, 2019, p. 78). (42) highlights the inherently multiplicative nature of the EGF models, because the conditional variances are products of other quantities. $(\ln(\sigma_t^2))$ and (r_t) following an EGF are strictly stationary, if and only if the roots of $\phi(B)$ are outside the unit circle and at the same time $0 \leq d < 0.5$. This result may be obtained by noting the following aspects: firstly, the log-conditional variance follows a FARIMA (Granger and Joyeux, 1980; Hosking, 1981) process with innovations from $(g\,(\eta_t))$. Secondly, both $g(\eta_t)$ and $\exp\{g(\eta_t)\}$ are iid. Thirdly, the same reasoning as in Nelson (1991) can be applied. For the weak stationarity of $(\ln{(\sigma_t^2)})$ we additionally require the existence of the variance of $g\,(\eta_t)$.

For the existence of moments $E(r_t^{2m})$, $m = 1, 2, \ldots$, sets of conditions can be derived for special cases of the EGF from the fact that by calculating expectations in (42), $E\left[\exp\left\{m\theta_i g\left(\eta_t\right)\right\}\right]$ for all i in the infinite product are values of the moment-generating function (MGF) $M_g(u) = E\left[\exp\left\{ug(\eta_t)\right\}\right]$ of the random variable $g(\eta_t)$ at $m\theta_i$, whose existence for $m\theta_i$ depends on the corresponding distribution of η_t. Presuming that the conditions for the existence of the corresponding moments are fulfilled, then

$$E(r_t^{2m}) = \exp(m\omega_\sigma) \times E\left(\eta_t^{2m}\right) \times \prod_{i=0}^{\infty} M_g(m\theta_i) \qquad \text{and} \tag{43}$$

$$E(r_t^{2m} r_{t+k}^{2m}) = \exp(2m\omega_\sigma) \times E(\eta_t^{2m}) \times \left\{ \prod_{j=0}^{k-2} M_g(m\theta_j) \right\}$$
$$\times \left\{ \prod_{i=0}^{\infty} M_g(m(\theta_i + \theta_{i+k})) \right\} \times E\left(\eta_t^{2m} \exp\left\{ m\theta_{k-1} g(\eta_t) \right\} \right), \quad (44)$$

$k = 1, 2, \ldots$, where $\left\{ \prod_{j=0}^{k-2} M_g(m\theta_j) \right\}$ in (44) is replaced by 1 for $k = 1$ (see also Karanasos and Kim, 2003, for the autocorrelation function of a simple EGARCH to the power of $2m$, which is strongly related to the result here). Taking (43) and (44) into account, we have

$$\text{cov}(r_t^2, r_{t+k}^2) = \exp(2\omega_\sigma) \Bigg\{ \left\{ \prod_{j=0}^{k-2} M_g(\theta_j) \right\} \times \left\{ \prod_{i=0}^{\infty} M_g(\theta_i + \theta_{i+k}) \right\}$$
$$\times E\left(\eta_t^2 \exp\left\{ \theta_{k-1} g(\eta_t) \right\} \right) - \prod_{l=0}^{\infty} \left[M_g(\theta_l) \right]^2 \Bigg\}, \quad (45)$$

$k = 1, 2, \ldots$, which aligns with the results for EGARCH models in Francq and Zakoïan (2019, p. 80). Moreover,

$$\text{var}(r_t^2) = \exp(2\omega_\sigma) \times \left\{ E\left(\eta_t^4 \right) \times \left\{ \prod_{i=0}^{\infty} M_g(2\theta_i) \right\} - \prod_{j=0}^{\infty} \left\{ M_g(\theta_j) \right\}^2 \right\}. \quad (46)$$

Considering (43), the unconditional moment of order $2m$ of r_t, when (r_t) follows an EGF-process, exists exactly then, whenever $E\left(\eta_t^{2m}\right) < \infty$ together with $\prod_{i=0}^{\infty} M_g(m\theta_i) < \infty$, where the second part is equivalent to stating that $\sum_{i=0}^{\infty} C_g(m\theta_i) < \infty$ with $C_g(u) = \ln\left\{M_g(u)\right\}$ being the cumulant-generating function (CGF) of $g(\eta_t)$. This insight leads to the first conclusion that $E(r_t^{2m})$ cannot exist, if $C_g(u)$ of $g(\eta_t)$ does not exist for any $u \in \{m\theta_0, m\theta_1, \ldots\}$.

In Section 4.3 in Peitz et al. (2026), explicit conditions for the existence of $E(r_t^{2m})$ are provided for cases, where η_t is a continuous random variable with mean zero and variance one, whose density is symmetric, bounded above and bounded away from zero on its domain $(-\infty, \infty)$. Additionally, they assume that $g(\cdot)$ is a measurable function with $E[g(\eta_t)] = 0$ and $\text{var}[g(\eta_t)] < \infty$ and that κ and γ are not both equal to zero. They define $\theta_{\max} = \sup_{i \geq 0}\{\theta_i\}$ and $\theta_{\min} = \inf_{i \geq 0}\{\theta_i\}$ for Type-I EGF models, $\gamma_{\max} = \sup_{i \geq 1}\{\gamma_i\}$ and $\gamma_{\min} = \inf_{i \geq 1}\{\gamma_i\}$ for Type-II EGF models, and $M_\eta(u)$ as the moment-generating function of η_1 itself. Peitz et al. (2026) describe that $E(r_t^{2m})$ exists

(i) for a (FI)EGARCH process, if $M_\eta(u)$ exists for $u \in (-m\delta_{\max}, m\delta_{\max})$ with $\delta_{\max} = \max\{|\theta_{\max}||\kappa + \gamma|, |\theta_{\max}||\kappa - \gamma|, |\theta_{\min}||\kappa + \gamma|, |\theta_{\min}||\kappa - \gamma|\}$,

(ii) for a (FI)MEGARCH process, if $M_\eta(u)$ exists for $u \in (-m\delta_{\max}, m\delta_{\max})$ with $\delta_{\max} = \max\{|\theta_{\max}||\gamma|, |\theta_{\min}||\gamma|\}$,

(iii) for a (FI)MLog-GARCH process, if $E(|\eta_t|^K) < \infty$ for $K \geq \max\{m\delta_{\max}, 2m\}$ with $\delta_{\max} = \theta_{\max}(\gamma + |\kappa|)$, and

(iv) for a (FI)Log-GARCH process, if $E(|\eta_t|^K) < \infty$ for $K \geq \max\{m\gamma_{\max}, 2m\}$ with $\gamma_{\min} > -1/(2m)$,

if additionally $\sum_{i=0}^{\infty} \theta_i^2 < \infty$ (Type-I models) or $\sum_{i=1}^{\infty} \gamma_i^2 < \infty$ (Type-II models). Following (iii) and (34) through (37), more specific conditions can be derived for a (FI)MLog-GARCH based on the corresponding conditional innovation distribution. It is apparent that for the normal distribution, the GED with $\beta > 0$ and the ALD with $P \geq 0$ the moments $E(|\eta_t|^K)$ exist for all $K > -1$ and thus the moments $E(r_t^{2m})$ exist for all integer-valued $m \geq 1$ in context of a (FI)MLog-GARCH process. However, (35) implies that $E(|\eta_t|^K)$ only exists for a t-distribution with shape ν, if $\nu > K > -1$ under the convention that $0^0 \equiv 1$ in case of $\nu = 2$ with $u = 0$. Thus, under a conditional t-distribution, the moment $E(r_t^{2m})$ of a (FI)MLog-GARCH process exists, if $\nu > \max\{m\theta_{\max}(\gamma + |\kappa|), 2m\}$.

Note that it is well-established that under a conditional normal distribution and a conditional GED with shape $\beta > 1$, all unconditional moments of r_t exist for a simple EGARCH and that no unconditional moments of r_t with positive order exist usually under a conditional t-distribution for a finite ν (Nelson, 1991). Due to the same tail-behavior of $g(\eta_t)$, this result can be easily extended to FIEGARCH, MEGARCH and FIMEGARCH processes. However, Peitz et al. (2026) introduced the conditional ALD in context of GARCH-modelling, which has the advantageous property that $M_\eta(u)$ only exists for $u \in \left(-\sqrt{2(P+1)}, \sqrt{2(P+1)}\right)$, which implies in the context of (FI)EGARCH and (FI)MEGARCH processes that under a conditional ALD those processes will have finite unconditional moments up until a certain order and infinite unconditional moments for larger orders. Hence, for a (FI)EGARCH or a (FI)MEGARCH, $E(r_t^{2m}) < \infty$ up until $m = \left[\sqrt{2(P+1)}/\delta_{\max}\right]$ with $\delta_{\max}$ as defined for (FI)EGARCH and (FI)MEGARCH, respectively, where $[\cdot]$ denotes the integer part only.

We extend the previous approach regarding conditions for the existence of $E(r_t^{2m})$ to Type-I EGF models with $M_{\text{asy}}, M_{\text{mag}} \in \{0, 1\}$ and $p_{\max} = \max\{p_{\text{asy}}, p_{\text{mag}}\} > 0$ in general under the symmetric conditional innovation distributions in `fEGarch`. We also exclude degenerate cases by assuming that κ and γ are not equal to zero simultaneously.

Once again, we assume $\sum_{i=0}^{\infty} \theta_i^2 < \infty$ and $\text{var}\,[g(\eta_t)] < \infty$, which together ensure the finiteness of the infinite product in (43), if all the individual $M_g(m\theta_i)$ exist (see a similar reasoning in Feng et al., 2023a). Define $\kappa^* = \kappa/p_{\text{asy}}$, $\gamma^* = \gamma/p_{\text{mag}}$, and $p_{\max} = \max\{p_{\text{asy}}, p_{\text{mag}}\}$. It is easy to see, under the specified setting but with $M_{\text{asy}} = M_{\text{mag}} = 0$, that

$$
\begin{aligned}
M_g(u) &= C_M E_{\eta_1}\left\{\exp[u\kappa^* \text{sgn}(\eta_1) z^{p_{\text{asy}}} + u\gamma^* z^{p_{\text{mag}}}]\right\} && (47)\\
&= \frac{C_M}{2}\left\{E_z\left\{\exp[-u\kappa^* z^{p_{\text{asy}}} + u\gamma^* z^{p_{\text{mag}}}]\right\} + E_z\left\{\exp[u\kappa^* z^{p_{\text{asy}}} + u\gamma^* z^{p_{\text{mag}}}]\right\}\right\} && (48)
\end{aligned}
$$

holds, where C_M is a finite constant and $z = |\eta_1|$. $E_{\eta_1}(\cdot)$ and $E_z(\cdot)$ are expectations with respect to the random variables η_1 and z, correspondingly. The exponential terms in (48) are dominated by $z^{p_{\max}}$ and finiteness of the integral (47) is hence a question of whether $M_{z,p_{\max}}(u) = E\{\exp[uz^{p_{\max}}]\}$ as the MGF of $z^{p_{\max}}$ exists for a specific range of u. Define

$$
a_- = \left\{\begin{array}{ll} \gamma^* - \kappa^*, & \text{for } p_{\text{asy}} = p_{\text{mag}}, \\ -\kappa^*, & \text{for } p_{\text{asy}} > p_{\text{mag}}, \\ \gamma^*, & \text{for } p_{\text{asy}} < p_{\text{mag}}, \end{array}\right\} \text{ and } a_+ = \left\{\begin{array}{ll} \gamma^* + \kappa^*, & \text{for } p_{\text{asy}} = p_{\text{mag}}, \\ \kappa^*, & \text{for } p_{\text{asy}} > p_{\text{mag}}, \\ \gamma^*, & \text{for } p_{\text{asy}} < p_{\text{mag}}. \end{array}\right\} \quad (49)
$$

Subsequently, define $s_- = a_-\theta_{\max}$, for $a_- \geq 0$, and $s_- = a_-\theta_{\min}$, for $a_- < 0$, and $s_+ = a_+\theta_{\max}$, for $a_+ \geq 0$, and $s_+ = a_+\theta_{\min}$, for $a_+ < 0$. It can be seen that $M_g(u_i)$, $u_i = m\theta_i$, exists for all $i = 0, 1, \ldots$, if $M_{z,p_{\max}}(u)$ exists for $u \in (-\infty, m\delta^*_{\max})$, where $\delta^*_{\max} = \max\{s_-, s_+\}$. This criterion (given also previous assumptions) already implies $E(|\eta_1|^{2m}) < \infty$, if $m\delta^*_{\max} \geq 0 + \varepsilon$ with some arbitrary real $\varepsilon > 0$, and thus both $E(\sigma_t^{2m}) < \infty$ and $E(r_t^{2m}) < \infty$. Note that the density of z can be easily determined for the four densities (28), (29), (30) and (31), with $\mu = 0$ and $\sigma = 1$, by only considering $[0, \infty)$ as the domain and by also multiplying the density functions by 2 due to their symmetry property. For any or both of $M_{\text{asy}} = 1$ and $M_{\text{mag}} = 1$, the same results hold, because $((z+1)^{p_j} - 1)$, with $p_j = p_{\text{asy}}$ or $p_j = p_{\text{mag}}$, is dominated by z^{p_j} for $z \to \infty$. Detailed proof of these results is, however, omitted here. For the selected symmetric distributions in this article, the following results can be obtained. It can be deduced under a conditional normal distribution that $M_{z,p_{\max}}(u)$ exists for all $u \in \mathbb{R}$, if $p_{\max} < 2$, for all $u < 1/2$, if $p_{\max} = 2$, or for all $u \leq 0$, if $p_{\max} > 2$. Furthermore, under a conditional t-distribution with finite shape, $M_{z,p_{\max}}(u)$ only exists for all $u \leq 0$ and it does not exist otherwise. Under a conditional GED, however, $M_{z,p_{\max}}(u)$ exists for all $u \in \mathbb{R}$, if $p_{\max} < \beta$, for all $u < (C_{\Gamma,3}/C_{\Gamma,1})^{\beta/2}$, if $p_{\max} = \beta$, or for all $u \leq 0$, if

$p_{\max} > \beta$. Under a conditional ALD, $M_{z,p_{\max}}(u)$ exists for all $u \in \mathbb{R}$, if $p_{\max} < 1$, for all $u < \sqrt{2(P+1)}$, if $p_{\max} = 1$, or for all $u \leq 0$, if $p_{\max} > 1$. All cases where $M_{z,p_{\max}}(u)$ exists for all $u \leq 0$ are usually irrelevant in the context of the EGF models. Note that these results all imply $p_{\max} > 0$ and they follow from the formulas of $M_{z,p_{\max}}(u)$ under the four symmetric conditional innovation distributions by comparing the growths of the various factors under the integral. Consequently, there are special submodels of the Type-I EGF with certain conditional distributions and certain settings of $p_{\max}$, for which $E(r_t^{2m})$ will exist up to an arbitrary order. In contrast, further information on Type-II EGF models is omitted here. For further details on those models, we refer the reader to Feng et al. (2020a).

The previous idea can also be extended to cases with asymmetric conditional distributions; nonetheless, now the tails of $M_{z,p_{\max}}$ need to be investigated separately. It can be shown that $E(r_t^{2m}) < \infty$, if $M^+_{z,p_{\max}}(u) = \int_0^\infty \exp\{u|x|^{p_{\max}}\} f_\eta(x) \mathrm{d}x$ exists for $u \in (-\infty, ms_+)$ and $M^-_{z,p_{\max}}(u) = \int_0^\infty \exp\{u|x|^{p_{\max}}\} f_\eta(-x) \mathrm{d}x$ exists for $u \in (-\infty, ms_-)$ (while requiring all other conditions as in the symmetric case). $f_\eta(\cdot)$ is then the density of the (potentially) skewed innovation.

C.3 QMLE for parametric GARCH-type models

Given the underlying conditional distribution, a vector of model parameters Δ, and by conditioning on suitable starting values for unknown quantities, σ_t can be computed at all observation time points as well as μ_t if required. For short-memory EGF models, Equations (1) with (5) under corresponding definition of $g(\eta_t)$ in accordance with (7) to (9), and (1) with (13) can be used directly without the need for truncation of the coefficient polynomials. In contrast, for a Type-I long-memory EGF model,

$$\ln\left(\sigma_t^2\right) \approx \tilde{\omega}_\sigma + \sum_{i=0}^{L-1} \theta_i g(\eta_{t-i-1}) \tag{50}$$

with some positive integer L is an appropriate approximation of (6) with finite-length coefficient series. Due to $E\left[g\left(\eta_t\right)\right] = 0$ we have $\tilde{\omega}_\sigma = \omega_\sigma + E\left[g\left(\eta_t\right)\right] \sum_{i=L}^\infty \theta_i = \omega_\sigma$, i.e. the truncation of $\theta(B)$ does not introduce any additional shift in the log-volatility intercept. Once $\ln\left(\sigma_t^2\right)$ has been obtained using (50), $g(\eta_t) = g\left(r_t / \exp\left\{\left[\ln\left(\sigma_t^2\right)\right]/2\right\}\right)$ can be computed as the basis for applying (50) to subsequent time points. Analogously, for Type-II long-memory EGF models, the approximation

$$\ln\left(\sigma_t^2\right) \approx \tilde{\omega}_\sigma + \sum_{i=1}^{L} \gamma_i \xi_{t-i} \tag{51}$$

with $\tilde{\omega}_\sigma = \omega_\sigma + E\left(\xi_t\right) \sum_{i=L+1}^{\infty} \gamma_i = \omega_\sigma$ can be considered. In the `fEGarch` package, the default for long-memory EGF models is $L = n-1$ with pre-sample values $g(\eta_t) = \xi_t = 0$ for $t = \ldots, -1, 0$, so that the infinite-length coefficient series are always practically truncated as far back as needed so that the first observation time point is included. Moreover, for efficient computation of the coefficients of the truncated infinite-length coefficient series, the `fEGarch` package employs parts of the algorithm by Nielsen and Noël (2021), which is based on the fast fourier transform (FFT). For short-memory models, the same pre-sample values for $g(\eta_t)$ and ξ_t are considered as in the long-memory case. As a crude approximation that has no notable effect on the initialization, pre-sample values of $\ln(\sigma_t^2)$ are set to the logarithm of the sample variance of $(r_t)_{t=1}^{n}$.

If a dual model in the mean and in the variance is fitted in accordance with the description in Section 2.2.2, the same concept applies in the mean step. Descriptions under short-memory should use representation (22) with finite-length polynomials, so that $\mu_t = \mu + \sum_{i=1}^{P} \beta_i (y_{t-i} - \mu) + \sum_{j=1}^{Q} \alpha_j r_{t-j}$, however, detailed descriptions are omitted here. Under long memory in the mean equation, the approximation

$$\mu_t - \tilde{\mu} \approx \sum_{i=1}^{L} \delta_i z_{t-i} \tag{52}$$

is considered, where $z_t = y_t - \tilde{\mu}$ for all t, where for $t < 0$, values of (z_t) are replaced by 0. Note that $\tilde{\mu} = \mu + E(y_t - \mu) \sum_{i=L+1}^{\infty} \delta_i = \mu$ due to $E(y_t - \mu) = 0$. Once μ_t has been obtained for all $t \in \{1, \ldots, n\}$, σ_t for all $t \in \{1, \ldots, n\}$ can be computed from $r_t = y_t - \mu_t$ as in the pure volatility models. However, for short-memory volatility models from the EGF, the pre-sample values of $\ln(\sigma_t^2)$ are set differently: they are the log-sample variance of the residuals from a preliminary estimation of a pure ARMA or FARIMA model, respectively.

The (conditional) quasi-log-likelihood is generally defined to be

$$l(\Delta) = \sum_{t=1}^{n} \ln\left[f_y(y_t, \mu_t, \sigma_t)\right], \tag{53}$$

where $f_y(\cdot)$ is one of the four from (28) to (31). Alternative representations also have a scaling factor of n^{-1} to adjust for the sample size. (53) is to be maximized with respect to Δ. Closed-form quasi-log-likelihood functions under available conditional distributions in `fEGarch` are

$$l_{\text{norm}}(\Delta) = -\frac{1}{2}\left[\sum_{t=1}^{n}\ln\left(\sigma_t^2\right) + n\ln(2\pi) + \sum_{t=1}^{n}\left(\frac{y_t-\mu_t}{\sigma_t}\right)^2\right], \tag{54}$$

$$\begin{aligned} l_{\text{std}}(\Delta) = -\frac{1}{2}\Bigg\{&\sum_{t=1}^{n}\ln\left(\sigma_t^2\right) + n\ln[\pi(\nu-2)] - 2n\ln\left[\frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)}\right] \\ &+ (\nu+1)\sum_{t=1}^{n}\ln\left[1+(\nu-2)^{-1}\left(\frac{y_t-\mu_t}{\sigma_t}\right)^2\right]\Bigg\}, \end{aligned} \tag{55}$$

$$\begin{aligned} l_{\text{ged}}(\Delta) = -\frac{1}{2}\Bigg[&\sum_{t=1}^{n}\ln\left(\sigma_t^2\right) - 2n\ln\left(\frac{\beta}{2}\right) - n\ln\left(\frac{C_{\Gamma,3}}{C_{\Gamma,1}^3}\right) \\ &+ 2\left(\frac{C_{\Gamma,3}}{C_{\Gamma,1}}\right)^{\frac{\beta}{2}}\sum_{t=1}^{n}\left|\frac{y_t-\mu_t}{\sigma_t}\right|^{\beta}\Bigg], \end{aligned} \tag{56}$$

$$\begin{aligned} l_{\text{ald}}(\Delta) = -\frac{1}{2}\Bigg\{&\sum_{t=1}^{n}\ln\left(\sigma_t^2\right) - 2n\ln\left(\frac{B}{2}\right) - n\ln\left[2(P+1)\right] \\ &+ 2\sqrt{2(P+1)}\sum_{t=1}^{n}\left|\frac{y_t-\mu_t}{\sigma_t}\right| \\ &- 2\sum_{t=1}^{n}\ln\left[\sum_{j=0}^{P}c_j\left(\sqrt{2(P+1)}\left|\frac{y_t-\mu_t}{\sigma_t}\right|\right)^j\right]\Bigg\} \qquad \text{and} \end{aligned} \tag{57}$$

$$\begin{aligned} l_{\text{skew}}(\Delta) = -\frac{1}{2}\Bigg\{&\sum_{t=1}^{n}\ln\left(\sigma_t^2\right) - n\ln\left(C_V\right) - 2n\ln\left(\frac{2s}{s^2+1}\right) \\ &- 2\sum_{t=1}^{n}\mathbb{I}_{(-\infty,0]}(\check{y}_t)\ln\left[f_\eta\left(s\check{y}_t\right)\right] - 2\sum_{t=1}^{n}\mathbb{I}_{(0,\infty)}(\check{y}_t)\ln\left[f_\eta\left(\frac{\check{y}_t}{s}\right)\right]\Bigg\} \end{aligned} \tag{58}$$

with $\check{y}_t = \sqrt{C_V}\left(\frac{y_t-\mu_t}{\sigma_t}\right) + C_E$ in (58) and where $\sum_{t=1}^{n}\mathbb{I}_{(-\infty,0]}(\check{y}_t)\ln\left[f_\eta(s\check{y}_t)\right]$ and $\sum_{t=1}^{n}\mathbb{I}_{(0,\infty)}(\check{y}_t)\ln\left[f_\eta(\check{y}_t/s)\right]$ in (58) are partial sums of one of (54) to (57) with setting $\mu_t = 0$ and $\sigma_t = 1$ in (54) to (57) due to the standardization implied by $f_\eta(\cdot)$ and with replacing y_t in (54) to (57) by either $s\check{y}_t$ or by $\check{y}_t/s$ depending on the indicator functions. A general algorithm for the maximization of the log-likelihood then reads as follows given a parameter vector Δ:

1. Compute μ_t for all $t = 1, \ldots, n$ from the observed (y_t) following (52) (or the corresponding short-memory formula) using Δ.
2. Calculate $r_t = y_t - \mu_t$ for all $t = 1, \ldots, n$.
3. Compute $\sigma_t = \exp\{\ln(\sigma_t^2)/2\}$ for all $t = 1, \ldots, n$ from (r_t), where $\ln(\sigma_t^2)$ is calculated in accordance with either (50) or (51) (or the corresponding short-memory formulas) using Δ.

4. Plug (y_t), (μ_t) and (σ_t) in the correct log-likelihood formula from (54) to (58) to obtain the corresponding value of the log-likelihood.

After Step 4, the vector Δ will be adjusted using some criterion, and the Steps 1 to 4 will be repeated again. This procedure is repeated until convergence of the log-likelihood is reached or until a maximum number of iterations. The final parameter vector is then the estimator $\hat{\Delta}$. In `fEGarch`, the underlying solver that affects the adjustment of the parameter vector between iterations is the SOLNP algorithm (Ye, 1989) implemented through the R package `Rsolnp` (Galanos and Theussl, 2015).

In order to fit an EGF model with either conditional ALD or skew-ALD, the profile-quasi-log-likelihood is climbed (see also Chapter 3.6 in Millar, 2011, for further information on profile likelihood). For each element P from a set of $\{1, 2, \ldots, 5\}$, an estimate $\hat{\Delta}_P$ of $\Delta_P \equiv \Delta \mid P$ is obtained by maximizing the quasi-log-likelihood function $l(\Delta_P)$. The profile-quasi-log-likelihood is then the collection of $\left(l\left(\hat{\Delta}_P\right)\right)$ for $P = 1, \ldots, 5$. Consequently,

$$\hat{\Delta} = \underset{\Delta \in \{\hat{\Delta}_1, \ldots, \hat{\Delta}_5\}}{\arg\max} \; l(\Delta) \tag{59}$$

is the QMLE under the ALD or skew-ALD. This comes at the cost that for fitting an EGF with conditional ALD or skew-ALD, five models need to be fitted internally. For maximum efficiency, this is done in parallel, however, under these conditional distributions the computation time is larger than for other conditional distributions nonetheless.

By standard theory (see also Chapter 3.2 in Millar, 2011), the Hessian matrix in context of maximum-likelihood estimation is commonly defined as

$$\mathbf{H}\left(\hat{\Delta}\right) = \left.\frac{\partial^2 l(\Delta)}{\partial \Delta \partial \Delta'}\right|_{\Delta = \hat{\Delta}}, \tag{60}$$

i.e. it is the matrix of partial second derivatives of the quasi-log-likelihood function evaluated at the identified estimator $\hat{\Delta}$. Commonly, an estimator of $\mathbf{H}\left(\hat{\Delta}\right)$, denoted by $\hat{\mathbf{H}}\left(\hat{\Delta}\right)$, is implemented numerically. In the `fEGarch` package, the package `numDeriv` (Gilbert and Varadhan, 2019) is called, which implements Richardson's extrapolation (see e.g. Chapter I in Sidi, 2003, for more details) to obtain $\hat{\mathbf{H}}\left(\hat{\Delta}\right)$, after $\hat{\Delta}$ has been identified. An estimator of the variance-covariance matrix $\mathbf{V}$ of $\hat{\Delta}$ is then commonly given by $\hat{\mathbf{V}} = \left[-\hat{\mathbf{H}}\left(\hat{\Delta}\right)\right]^{-1}$, whose main diagonal contains the collection of variance estimates for the elements of $\hat{\Delta}$, if the log-likelihood is correctly specified. While yet to be proven for models of the EGF-class, it is assumed, given the correct specification of the log-likelihood, that $\sqrt{n}(\hat{\Delta} - \Delta_0) \xrightarrow{d} N\left(\mathbf{0}, \mathbf{I}^{-1}(\Delta_0)\right)$, where $\mathbf{0}$ is a column vector of suitable length with zeros only and $\mathbf{I}(\mathbf{x}) = E\left[-\frac{\partial^2}{\partial\Delta\partial\Delta'} l_t(\Delta)\Big|_{\Delta=\mathbf{x}}\right]$, with $l_t(\cdot)$ as the log-likelihood at one time point t only, is the per-observation Fisher information matrix, holds under regularity conditions, due to the fact that asymptotic results of the QMLE in context of exponential GARCH models have not been solved yet (Lopes and Prass, 2014). However, the finite sample properties of the EGF are analyzed through a simulation study in Section 6. Moreover, under conditional ALD or skew-ALD, we assume after the identification of $\hat{\Delta}$ that $P = \hat{P}$ is fixed and obtain $\hat{\mathbf{V}}$ for the remaining parameters only, because standard approaches do not apply for a mixture of real-valued parameters and the discrete $\hat{P}$. The same simulation study in Section 6, however, also shows that $\hat{P}$ is consistent for P.

C.4 Forecasting VaR and ES through EGF models and their extensions

We assume we have rolling one-step conditional standard deviation forecasts $\hat{\sigma}_t$ and mean forecasts $\hat{\mu}_t$, each for $t = n+1, \ldots, n+n_{\text{te}}$, where n and n_{te} are the numbers of training and test observations, respectively. In addition, further parameter estimates like estimated shape and skew parameters are presumed to be known. Then these values can be used to compute corresponding rolling one-step forecasts of VaR and ES, here denoted by $\widehat{\text{VaR}}_\alpha^y(t)$ and $\widehat{\text{ES}}_\alpha^y(t)$. Then generally, we have

$$\widehat{\text{VaR}}_\alpha^y(t) = \hat{\mu}_t + \hat{\sigma}_t \times \widehat{\text{VaR}}_\alpha^\eta \qquad \text{and} \tag{61}$$

$$\widehat{\text{ES}}_\alpha^y(t) = \hat{\mu}_t + \hat{\sigma}_t \times \widehat{\text{ES}}_\alpha^\eta, \tag{62}$$

where $\widehat{\text{VaR}}_\alpha^\eta$ and $\widehat{\text{ES}}_\alpha^\eta$ are the (estimated) constant VaR and ES for η_1 at level α. It is important to note that (61) and (62) are only valid in context of one-step rolling forecasts. Otherwise, the conditional distribution assumption usually does not hold in the context of multi-step forecasts.

Conceptually, $\widehat{\text{VaR}}_\alpha^\eta$ is the estimated quantile function of η_1, under consideration of possibly estimated shape and skew parameters, for cumulative probability $1-\alpha$. In R, we may define a function of the estimated pdf of η_1 and, based upon this R function, a cdf R function that makes use of `integrate` of the `stats` package by following the basic definition of a cdf

$$F_\eta(x) = \int_{-\infty}^{x} f_\eta(x)\mathrm{d}x. \tag{63}$$

To obtain an approximate quantile function, one can then leverage `uniroot` of the `stats` package to find, for which x the previously created cdf R function equals $1 - \alpha$. This approach works universally for continuously distributed random variables and produces suitable approximations of quantiles of distributions commonly considered in volatility modelling.

In addition, one of the general definitions of the ES for a continuous η_1 is

$$\mathrm{ES}_\alpha^\eta = (1-\alpha)^{-1} \int_{-\infty}^{\mathrm{VaR}_\alpha^\eta} y f_\eta(y) \mathrm{d}y. \tag{64}$$

Given the corresponding R function for the estimated pdf of η_1 as well as the previously estimated $\widehat{\mathrm{VaR}}_\alpha^\eta$, one can implement `integrate` of the `stats` package to obtain a suitable approximation of $\widehat{\mathrm{ES}}_\alpha^\eta$ from (64). Once again, this approach works universally for any continuous η_1 and can be implemented for even more distributions than those considered in `fEGarch`. Using the formal definition (see for example McNeil et al., 2015, pp. 65–66) of VaR_α^η under a normal distribution with $\alpha = 0.975, 0.99$, we obtain -1.959964 and -2.326348 rounded to six decimals. Using the previously described numerical approach, we obtain -1.959965 and -2.326355. Using the formal definition of ES_α^η (see for example McNeil et al., 2015, p. 71) under a normal distribution, we compute -2.337803 and -2.665214 for $\alpha = 0.975, 0.99$. Using the described numerical approach, the values are -2.337800 and -2.665171. Therefore, the approximations are almost identical to the theoretical values. Further tests also show robustness of this idea in comparison to the formal definition for common α under a heavy-tailed t-distribution with small degrees of freedom such as 3.

Forecasting of conditional volatility with fitted EGF models is done differently for short- and long-memory models. In the following, denote by $h \in \mathbb{N}^+$ the forecasting horizon. For a short-memory EGF model of Type I, define a multi-step point forecast of $\ln(\sigma_t^2)$ by

$$\widehat{\ln(\sigma_{n+h}^2)} = \hat{\omega}_\sigma \left(1 - \sum_{i=1}^{p} \hat{\phi}_i\right) + \sum_{i=1}^{p} \hat{\phi}_i \widehat{\ln(\sigma_{n+h-i}^2)} + \sum_{j=1}^{q-1} \hat{\psi}_j \hat{g}(\hat{\eta}_{n+h-1-j}) + \hat{g}(\hat{\eta}_{n+h-1}) \tag{65}$$

with

$$\hat{g}(\hat{\eta}_t) = \hat{\kappa} \left\{ g_{\mathrm{asy}}(\hat{\eta}_t) - \widehat{E}\left[g_{\mathrm{asy}}(\hat{\eta}_t)\right] \right\} + \hat{\gamma} \left\{ g_{\mathrm{mag}}(\hat{\eta}_t) - \widehat{E}\left[g_{\mathrm{mag}}(\hat{\eta}_t)\right] \right\}, \tag{66}$$

where $\widehat{E}\left[g_{\mathrm{asy}}(\hat{\eta}_t)\right]$ and $\widehat{E}\left[g_{\mathrm{mag}}(\hat{\eta}_t)\right]$ are obtained through numerical integration given the corresponding transformation settings implied by $g(\cdot)$ and potential estimated distribution parameters. $\hat{\omega}_\sigma$, $\hat{\phi}_i$, for $i = 1, \ldots, p$, $\hat{\psi}_j$, for $j = 1, \ldots, q-1$, $\hat{\kappa}$ and $\hat{\gamma}$ are estimated model parameters, whereas $\hat{\eta}_t$, $t = 1, \ldots, n$, are model residuals. Forecasts are obtained iteratively starting with $h = 1$ and then by increasing h by 1 in a step-wise manner. Note that $\hat{g}(\hat{\eta}_t)$ are replaced by 0 for $t > n$, so that $\widehat{\ln(\sigma^2_{n+h})} \to \hat{\omega}_\sigma$ for $h \to \infty$.

For a long-memory EGF model of Type I, multi-step point forecasts of $\ln(\sigma_t^2)$ are approximated iteratively by

$$\widehat{\ln(\sigma^2_{n+h})} = \hat{\omega}_\sigma + \sum_{i=1}^{n+h-2} \hat{\theta}_i \hat{g}(\hat{\eta}_{n+h-1-i}) + \hat{g}(\hat{\eta}_{n+h-1}), \tag{67}$$

where $\hat{\theta}_i$, $i = 1, \ldots, n+h-2$, are the first $n+h-2$ estimated coefficients of the polynomial $\theta(B)$ obtained from the estimates of ϕ_i, $i = 1, \ldots, p$, ψ_j, $j = 1, \ldots, q-1$, and d.

In contrast, for a Type II short-memory EGF model, consider

$$\widehat{\ln\left(\sigma^2_{n+h}\right)} = \hat{\omega}_\sigma \left(1 - \sum_{i=1}^{p} \hat{\phi}_i\right) + \sum_{i=1}^{p} \hat{\phi}_i \widehat{\ln\left(\sigma^2_{n+h-i}\right)} + \sum_{j=1}^{\max(p,q)} \left(\hat{\psi}_j + \hat{\phi}_j\right) \hat{\xi}_{n+h-j}, \tag{68}$$

where $\hat{\xi}_t = \ln\left(\hat{\eta}_t^2\right) - \hat{E}\left[\ln\left(\hat{\eta}_t^2\right)\right]$ for $t \leq n$ and zero otherwise with $\hat{E}\left[\ln\left(\hat{\eta}_t^2\right)\right]$ being an estimate of $E\left[\ln\left(\eta_t^2\right)\right]$ given estimated distribution parameters. Furthermore, $\hat{\phi}_j = 0$ for $j > p$ and $\hat{\psi}_j = 0$ for $j > q$. For a long-memory Type-II EGF model, however, we employ

$$\widehat{\ln\left(\sigma^2_{n+h}\right)} = \hat{\omega}_\sigma + \sum_{i=1}^{n+h-1} \hat{\gamma}_i \hat{\xi}_{n+h-i}, \tag{69}$$

where $\hat{\gamma}_i$ are the estimated coefficients of $\gamma(B)$ in (10).

Ultimately, multi-step point forecasts of conditional volatility for both short- and long-memory Type-I and Type-II EGF models are obtained through $\hat{\sigma}_{n+h} = \exp\left(\widehat{\ln(\sigma^2_{n+h})}/2\right)$, which is common in practice and also implemented analogously for EGARCH models in the popular `rugarch` package (Galanos, 2024), but it is usually a biased forecast of the future volatility for $h \geq 2$ (see also Lopes and Prass, 2014). Bias-corrections in multi-step volatility forecasts for $h \geq 2$ are, however, left for future versions of `fEGarch`. A theoretical approach for short-memory EGARCH models under conditional normal distribution is discussed for example in Tsay (2005, p. 129) and could be expanded to various conditional distributions by identifying the unknown quantity $E\left\{\exp\left[g\left(\eta_t\right)\right]\right\}$ (for model order $q = 1$) through numerical integration. Although this may be applicable in context of short-memory models, long-memory EGF model approximations to the volatility forecasts make use of long products over such unknown quantities for large h, making numerical integration potentially infeasible to solve the bias issue in this context.

One-step-ahead rolling conditional volatility forecasts are produced in a similar fashion, usually on a test set. Due to the one-step-ahead design, all previous information for each corresponding time point is seen as deterministic and one-step volatility forecasts are therefore unbiased ignoring potential bias effects through parameter estimators (Lopes and Prass, 2014). Assume a Type-I EGF model is fitted to the first n observations of an observed return series and that the last n_{te} observations are reserved for testing. Then iteratively for $h = 1, \ldots, n_{\text{te}}$ consider (65) for short-memory and (67) for long-memory models given that $\left(\widehat{\ln(\sigma_t^2)}\right)_{t=1}^{n+h-1}$ and $(\hat{g}(\hat{\eta}_t))_{t=1}^{n+h-1}$ are known when calculating $\ln\widehat{\left(\sigma_{n+h}^2\right)}$. Once $\ln\widehat{\left(\sigma_{n+h}^2\right)}$ and thus $\hat{\sigma}_{n+h} = \exp\left(\widehat{\ln(\sigma_{n+h}^2)}/2\right)$ are known for the current value of h, compute $\hat{\eta}_{n+h} = r_{n+h}/\hat{\sigma}_{n+h}$ and subsequently update $\hat{g}(\hat{\eta}_{n+h})$ following (66). Increase h by one and repeat the procedure until the end of the test set. Analogously, one-step-ahead rolling volatility forecasts for Type-II EGF models can be obtained using (68) or (69).

The mean part in dual-models as discussed in Section 2.2.2 can be forecasted in accordance with the approach by Box et al. (2008, p. 143) for $D = 0$ or following the mean model's infinite-order AR approximation as discussed in context of FARIMA models with $D > 0$ by Feng and Zhou (2015). Thus, under $D = 0$, we employ

$$\hat{\mu}_{n+h} = \hat{y}_{n+h} = \hat{\mu} + \sum_{i=1}^{P} \hat{\beta}_i (y_{n+h-i} - \hat{\mu}) + \sum_{j=1}^{Q} \hat{\alpha}_j \tilde{r}_{n+h-j} \tag{70}$$

for producing multi-step point forecasts, where $(\tilde{r}_t)$ are the estimates of (r_t) for the sample time points. Under $D > 0$, on the other hand,

$$\hat{\mu}_{n+h} = \hat{y}_{n+h} = \hat{\mu} + \sum_{i=1}^{n+h-1} (y_{n+h-i} - \hat{\mu}) \tag{71}$$

is applied. In both scenarios, for unknown future y_{n+h-i} the previously obtained point forecasts are implemented instead, whereas unknown future $\tilde{r}_{n+h-j}$ are replaced simply by zero. For one-step rolling forecasts, analogously to the volatility forecasts, all previous information at time point $n + h$ is assumed to be known. In the short-memory case, once $\hat{\mu}_{n+h}$ has been computed, $\tilde{r}_{n+h} = y_{n+h} - \hat{\mu}_{n+h}$ can be obtained, which can be used in producing the one-step forecast at the next time point. Both multi-step forecasts and one-step rolling forecasts for the volatility component are calculated based on $(\tilde{r}_t)$ following the previously described approaches for pure EGF-volatility models. In case of multi-step forecasts, both components μ_t and σ_t are therefore forecasted in two separate procedures. In one-step-ahead rolling forecasts, we first require the one-step-ahead forecasts of μ_t to then obtain $\tilde{r}_t$ for all out-of-sample time points before the volatility forecasting procedure can be applied.

For semiparametric extensions of the EGF models (16), the nonparametric and the parametric model parts are forecasted separately. Firstly, for multi-step forecasts, $\hat{s}(x_t)$ is extrapolated constantly into the future leading to forecasts $\hat{s}(x_{n+h}) = \hat{s}(x_n)$, whereas the conditional volatility $\tilde{\sigma}_{n+h}$ is forecasted based on the parametric model fitted to $(y_t - \bar{y})/\hat{s}(x_t)$, $t = 1, \ldots, n$, as explained in Section 2.2.1. The total volatility forecast is then simply the product of both forecasted components $\hat{\Omega}_{n+h} = \hat{s}(x_{n+h})\tilde{\sigma}_{n+h}$. Secondly, one-step-ahead rolling forecasts are implemented by extrapolating $\hat{s}(x_t)$ constantly into the future. Subsequently, substitute future standardized returns are obtained as $\tilde{r}_{n+h} = (y_{n+h} - \bar{y})/\hat{s}(x_{n+h})$, $h = 1, \ldots, n_{\mathrm{te}}$. The rolling point forecasts of the parametric part are then obtained using the in-sample standardized returns and the out-of-sample substitute standardized returns as in the purely parametric case. Once again, $\hat{\Omega}_{n+h} = \hat{s}(x_{n+h})\tilde{\sigma}_{n+h}$ (see also Letmathe et al., 2022). Note that $\hat{\sigma}_t$ in (61) and (62) has to be replaced by the corresponding forecast of Ω_t and that $\hat{\mu}_t = \bar{y}$ for semiparametric models.

In practice, it is both common and advantageous to re-estimate a model regularly when progressing through the test period to compute one-step-ahead forecasts of VaR and ES, imitating how one would apply a model realistically over time and allowing models to adapt to changes in the structure of the observed series. In the package `fEGarch`, this feature is also implemented. Both parametric and semiparametric models can be re-estimated; however for semiparametric models, the bandwidth itself in the nonparametric step is not being re-estimated. Instead, when a semiparametric model is refitted through `fEGarch`, the initially obtained bandwidth for the original training period is reused by rescaling it in accordance with the length of the updated training period.

Recent discussion papers

2026-04	Dominik Schulz Yuanhua Feng Christian Peitz Oliver Kojo Ayensu	Estimating, Forecasting and Backtesting a Family of Exponential and Other GARCH Models Using the fEGarch Package
2026-03	Dominik Schulz Thi Thu Huong Do Yuanhua Feng	A semiparametric spatial FARIMA applied in the presence of spatial seasonality
2026-02	Dominik Schulz	The R Package deseats for Data-Driven Trend and Seasonality Estimation in Time Series
2026-01	Dominik Schulz Yuanhua Feng Thomas Gries Marlon Fritz Sebastian Letmathe	Diagnosing the trend and bootstrapping the forecasting intervals using a semiparametric ARMA
2025-06	Sarah Kühn Nadja Stroh-Maraun	Community Costs in Neighborhood Help Problems
2025-05	Sarah Kühn	Child Care Allocation Mechanisms: Navigating Incomplete Preference Elicitation
2025-04	Sarah Kühn Papatya Duman Britta Hoyer Thomas Streck Nadja Stroh-Maraun	Non-induced Preferences in Matching Experiments
2025-03	Marlon Fritz Thomas Gries Lukas Wiechers	Growth with Mismatch: Theory and Evidence from TFP Estimates
2025-02	Lukas Wiechers	A Real-Time Analysis of Fundamentals and Bubbles in the S&P 500
2025-01	Claus-Jochen Haake Martin R. Schneider	An Axiomatization of the Banzhaf Index to Measure Influence in Qualitative Comparative Analysis
2024-05	Iuliia Myroshnychenko Thomas Gries	Application of a dynamic welfare-centered energy transition model
2024-04	Claus-Jochen Haake Thomas Streck	A Measure for Contestedness of a Two-Person Bargaining Problem
2024-03	Youssouf Sangare Edem Adotey	Institutions and challenges to development: An Analysis of Economic Growth, Governance, and Development in Sub-Saharan Africa
2024-02	Papatya Duman Claus-Jochen Haake	Size Reduction Reform in German Parliament: a game theoretic analysis of power indices in the Bundestag
2024-01	Iuliia Myroshnychenko Thomas Gries	Developing a dynamic welfare-centered energy transition model: methodological challenges and perspectives
2023-03	Yuanhua Feng Thomas Gries Sebastian Letmathe	FIEGARCH, modulus asymmetric FILog-GARCH and trend-stationary dual long memory time series
2023-02	Claus-Jochen Haake Thomas Streck	Distortion through modeling asymmetric bargaining power
2023-01	Carina Burs	A Model of Cycles and Bubbles under Heterogeneous Beliefs in Financial Markets
2022-06	Marlon Fritz Thomas Gries Lukas Wiechers	An Early Indicator for Anomalous Stock Market Performance