



HAL
open science

M-quantile estimation for GARCH models

Patrick Ferreira Patrocinio, Valderio A. Reisen, Pascal Bondon, Edson Zambon Monte, Ian Meneghel Danilevicz

► **To cite this version:**

Patrick Ferreira Patrocinio, Valderio A. Reisen, Pascal Bondon, Edson Zambon Monte, Ian Meneghel Danilevicz. M-quantile estimation for GARCH models. Computational Economics, 2023, 10.1007/s10614-023-10398-z . hal-04100547

HAL Id: hal-04100547






<https://universite-paris-saclay.hal.science/hal-04100547>

Submitted on 22 Jul 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

M-quantile estimation for GARCH models

Patrick F. Patrocinio ^{1,3*}, Valderio A. Reisen ^{1,2,3,4†}, Pascal Bondon ^{3†}, Edson Z. Monte ^{1†}
and Ian M. Danilevicz ^{2,3†}

^{1*}PPGEco and Department of Economics, Federal University of Espírito Santo, Av. Fernando Ferrari, Vitória, 29075-910, Espírito Santo, Brazil.

² Department of Statistics, Federal University of Minas Gerais, Av. Pres. Antônio Carlos, Belo Horizonte, 31270-901, Minas Gerais, Brazil.

³Université Paris-Saclay, CNRS, CentraleSupélec, Laboratoire des Signaux et Systèmes, 91190, Gif-sur-Yvette, France.

⁴ Institute of Mathematics and Statistics, Federal University of Bahia, Bahia, Brazil.

*Corresponding author(s). E-mail(s):

patrick.ferreira-patrocinio@centralesupelec.fr;

Contributing authors: valderio.reisen@ufes.br;

pascal.bondon@centralesupelec.fr; edsonzambon@yahoo.com.br;

ian.meneghel-danilevicz@centralesupelec.fr;

†These authors contributed equally to this work.

Abstract

M-regression and quantile methods have been suggested to estimate generalized autoregressive conditionally heteroscedastic (GARCH) models. In this paper, we propose an M-quantile approach, which combines quantile and M-regression to obtain a robust estimator of the conditional volatility when the data have abrupt observations or heavy-tailed distributions. Monte Carlo experiments are conducted to show that the M-quantile approach is more resistant against additive outliers than M-regression and quantile methods. The usefulness of the method is illustrated on two financial datasets.

Keywords: GARCH, M-estimation, quantile, robustness, outliers, abrupt observations

1 Introduction

1 The effects of aberrant observations on model identification, estimation, and
2 forecasting of time series may be significant and depend on the type of obser-
3 vations. Four types of outliers are proposed in the literature on univariate time
4 series: additive outliers, innovation outliers, level shifts, and temporal changes.
5 Additive outliers are quite common in practical problems and are more dan-
6 gerous than the other outlier types in Gaussian and Non-Gaussian processes
7 (Chen & Liu, 1993). For example, in the standard structure of Box-Jenkins
8 models, Ledolter (1989) showed that the predictions in integrated autoregres-
9 sive moving average (ARMA) models are quite sensitive to additive outliers.
10 Chang, Tiao, and Chen (1988) demonstrated that the estimated parameters of
11 ARMA models become biased when the data contains outliers. A similar conclu-
12 sion was obtained for fractionally integrated models (Reisen, Lévy-Leduc, &
13 Taquq, 2017) and periodic ARMA models (Sarnaglia, Reisen, Lévy-Leduc, &
14 Bondon, 2021). In the same way, outliers strongly impact parameter estimation
15 in nonlinear models (Douc, Moulines, & Stoffer, 2014; Fan & Yao, 2003; Tsay,
16 2001; Tsay & Chen, 2018). The effect of additive outliers on the estimation of
17 heteroscedastic models was discussed by Carnero (2003); Carnero, Peña, and
18 Ruiz (2005, 2012); Franses and Ghijsels (1999); Mendes (2000). These authors
19 showed that outliers could substantially bias the parameter estimates of the
20 ARCH and GARCH models.

21 M-estimators are robust alternatives to obtain parameter estimates in
22 time series contaminated by outliers or generated by probability distributions
23 with heavy tails (Bai, Rao, & Wu, 1992; Huber & Ronchetti, 2009; Li, 2008;
24 Maronna, Martin, & Yohai, 2006; Wu, 2007). Quantile regression has vari-
25 ous benefits, such as independence to the conditional distribution, the ability
26 to construct an empirical conditional distribution function, and the possibil-
27 ity to detect series asymmetries (Lee & Noh, 2013; Xiao & Koenker, 2009;
28 Zheng, Qianqian, Li, & Xiao, 2016). M-quantile regression is a generalization
29 of quantile regression, and its use in practical problems can be very fruit-
30 ful. Besides the advantages of quantile regression, M-quantile regression allows
31 trading robustness and efficiency in inference by varying the tuning constant
32 of the M-regression function (Huber & Ronchetti, 2009, Chapter 5). Quantile
33 regression models require linear programming methods, which do not necessar-
34 ily guarantee convergence to a unique solution. M-quantile regression ensures
35 convergence to a unique solution when a continuous monotone function is used
36 (Chambers & Tzavidis, 2006; Jones, 1994). Furthermore, M-quantile regression
37 offers flexibility in choosing the loss function (Huber & Ronchetti, 2009).

38 Since outliers appear in microeconomics and financial time series, robust
39 estimation of heteroscedastic time series models is an essential research topic

from both applied and theoretical points of view (Iqbal, 2013; Mukherjee, 2008; Muler & Yohai, 2008). This paper proposes M-quantile estimators of the parameters of a GARCH model. A simulation study is carried out to compare the method to the traditional quasi-maximum likelihood (QML) method, M-estimation, and quantile regression in the context of non-contaminated and contaminated GARCH processes with additive outliers. As an illustration of our robust method with real data, the daily returns of two financial indices are analyzed.

The paper is organized as follows. In Section 2, the M-quantile estimation method is presented. In Section 3, Monte Carlo simulation results are presented. A real data application is reported in Section 4, and some conclusions are provided in Section 5.

2 M-quantile estimator for GARCH processes

The time series $\{X_t\}$, $t \in \mathbb{Z}$, is called a GARCH(p, q) process if it satisfies the two equations

$$\begin{aligned} X_t &= \sigma_t \varepsilon_t, \\ \sigma_t^2 &= \omega + \sum_{i=1}^p \alpha_i X_{t-i}^2 + \sum_{j=1}^q \beta_j \sigma_{t-j}^2, \end{aligned} \quad (1)$$

where σ_t is the positive square root of σ_t^2 , $\omega > 0$, $\alpha_i \geq 0$, $\alpha_p > 0$, $\beta_j \geq 0$ and $\beta_q > 0$ are constants, and $\{\varepsilon_t\}$ is a sequence of independent and identically distributed (IID) random variables with zero mean and unit variance. If $\sum_{i=1}^p \alpha_i + \sum_{j=1}^q \beta_j < 1$, (1) has a unique strictly stationary solution $\{X_t\}$ and this solution is nonanticipative in the sense that ε_t is independent of X_s for all $s < t$ (Fan & Yao, 2003; Francq & Zakoian, 2019). When $q = 0$, $\{X_t\}$ is the ARCH(p) model introduced by Engle (1982).

Berkes, Hovath, and Kokoszka (2003, Theorem 2.1) established the following representation for σ_t^2 in (1),

$$\sigma_t^2 = c_0 + \sum_{i=1}^{\infty} c_i X_{t-i}^2, \quad (2)$$

where $c_0 = \omega / (1 - \sum_{j=1}^q \beta_j)$ and the coefficients c_i 's are determined by

$$\sum_{i=1}^{\infty} c_i z^i = \frac{\sum_{i=1}^p \alpha_i z^i}{1 - \sum_{j=1}^q \beta_j z^j}, \quad z \in \mathbb{C}, |z| \leq 1. \quad (3)$$

Furthermore, the linear representation (2) of σ_t^2 in terms of X_s^2 for $s \leq t$ is unique when ε_0^2 is a non-degenerate random variable, see Berkes et al. (2003, Theorem 2.3). As an example, consider a GARCH(2, 2) model. Then, the

4 *M-quantile estimation for GARCH models*

71 coefficients c'_i s are given by

$$72 \quad c_0 = \frac{\omega}{1 - \beta_1 - \beta_2}, \quad c_1 = \alpha_1, \quad c_2 = \alpha_2 + \beta_1 \alpha_1$$

73 and, for all $i \geq 3$,

$$74 \quad c_i = \beta_1 c_{i-1} + \beta_2 c_{i-2}.$$

75 Other examples are given in (Hallin, Liu, & Mukherjee, 2022).

76 Let Θ be a compact subset of $(0, \infty) \times D$, where

$$77 \quad D = \left\{ \mathbf{x} = (x_1, \dots, x_{p+q})^T \in (0, 1)^{p+q} \text{ such that } \sum_{i=1}^{p+q} x_i < 1 \right\}$$

78 and \mathbf{x}^T is the transpose of vector \mathbf{x} . We denote by $\boldsymbol{\theta} =$
 79 $(\omega, \alpha_1, \dots, \alpha_p, \beta_1, \dots, \beta_q)^T \in \Theta$ the parameter vector of a GARCH(p, q) pro-
 80 cess. Assume that the process $\{X_t\}$ is generated from (1) with $\boldsymbol{\theta} = \boldsymbol{\theta}_0 \in \Theta$.
 81 For all $\boldsymbol{\theta} \in \Theta$, we define the variance function

$$82 \quad v_t(\boldsymbol{\theta}) = c_0(\boldsymbol{\theta}) + \sum_{i=1}^{\infty} c_i(\boldsymbol{\theta}) X_{t-i}^2, \quad t \in \mathbb{Z},$$

83 where the coefficients $c_i(\boldsymbol{\theta})$'s are defined by (3). We have $v_t(\boldsymbol{\theta}_0) = \text{E}(X_t^2 | \mathcal{F}_{t-1}^X)$
 84 where \mathcal{F}_{t-1}^X is the sigma field generated by the random variables X_s for $s \leq t$.

85 Let X_1, X_2, \dots, X_n be a sample from the process $\{X_t\}$. Based on this
 86 sample, only the following approximation of $v_t(\boldsymbol{\theta})$ can be computed,

$$87 \quad \hat{v}_t(\boldsymbol{\theta}) = c_0(\boldsymbol{\theta}) + \sum_{i=1}^{t-1} c_i(\boldsymbol{\theta}) X_{t-i}^2, \quad 1 \leq t \leq n. \quad (4)$$

88 An M-estimator of $\boldsymbol{\theta}_0$ is any solution

$$89 \quad \hat{\boldsymbol{\theta}}_n = \underset{\boldsymbol{\theta} \in \Theta}{\operatorname{argmin}} \frac{1}{n} \sum_{t=1}^n \left\{ \rho \left[X_t / \hat{v}_t^{1/2}(\boldsymbol{\theta}) \right] + \frac{1}{2} \log \hat{v}_t(\boldsymbol{\theta}) \right\}, \quad (5)$$

90 where $\rho(\cdot) : \mathbb{R} \rightarrow \mathbb{R}$ is a given loss-function. Assuming that $\rho(\cdot)$ has a deriva-
 91 tive $\rho'(\cdot) = \psi(\cdot)$ and $\boldsymbol{\theta}_0$ belongs to an open subset of Θ , (Mukherjee, 2008,
 92 Theorem 3.1) proved under some assumptions on $H(x) = x\psi(x)$ that $\hat{\boldsymbol{\theta}}_n$ con-
 93 verges in probability to $\boldsymbol{\theta}_{0H}$ and is asymptotically normal at the usual \sqrt{n}
 94 rate, where $\boldsymbol{\theta}_{0H} = (c_H \omega_0, c_H \alpha_{01}, \dots, c_H \alpha_{0p}, \beta_{01}, \dots, \beta_{0q})^T$ and c_H is a positive
 95 real number defined by $\text{E} \left[H(\varepsilon_t / c_H^{1/2}) \right] = 1$. Similar results were obtained by
 96 Muler and Yohai (2008) and Boudt and Croux (2010). There are many candi-
 97 date functions for $\rho(\cdot)$. Here, we consider the two classical choices, the Huber

98 and QML functions given respectively by

$$99 \quad \rho_{\text{H}}(x) = \begin{cases} \frac{1}{2}x^2, & \text{if } |x| \leq k, \\ k|x| - \frac{1}{2}k^2, & \text{if } |x| > k, \end{cases}$$

100 where k is a tuning constant, and $\rho_{\text{QML}}(x) = \frac{1}{2}x^2$.

101 We propose an M-quantile estimator of $\boldsymbol{\theta}_0$ defined as any solution

$$102 \quad \widehat{\boldsymbol{\theta}}_{n,\tau} = \underset{\boldsymbol{\theta} \in \Theta}{\operatorname{argmin}} \frac{1}{n} \sum_{t=1}^n \left\{ \rho_{\tau} \left[X_t / \widehat{v}_t^{1/2}(\boldsymbol{\theta}) \right] + \frac{1}{2} \log \widehat{v}_t(\boldsymbol{\theta}) \right\}, \quad (6)$$

103 where

$$104 \quad \rho_{\tau}(x) = \begin{cases} \rho_{\text{H}}[(1-\tau)x] & \text{if } x < 0, \\ \rho_{\text{H}}(\tau x) & \text{if } x \geq 0, \end{cases}$$

105 and τ is a quantile in $(0, 1)$, see [Breckling and Chambers \(1988\)](#).

106 3 Monte Carlo simulations

107 In this section, the performances of $\widehat{\boldsymbol{\theta}}_n$ where $\rho(\cdot) = \rho_{\text{QML}}(\cdot)$ and $\rho(\cdot) = \rho_{\text{H}}(\cdot)$
 108 in (5), $\widehat{\boldsymbol{\theta}}_{n,\tau}$, and the quantile estimator proposed by [Lee and Noh \(2013\)](#) are
 109 compared for finite sample size data generated from a GARCH process. We
 110 take $k = 1.5$ in $\rho_{\text{H}}(\cdot)$. In (5) and (6), $\widehat{v}_t(\boldsymbol{\theta})$ is obtained from (4) where $c_i(\boldsymbol{\theta})$ is
 111 calculated recursively from (3). In the quantile regression method of [Lee and](#)
 112 [Noh \(2013\)](#), (1) is parameterized as follows,

$$113 \quad \begin{aligned} X_t &= \sqrt{h_t} u_t, \\ h_t &= 1 + \sum_{i=1}^p \gamma_i X_{t-i}^2 + \sum_{j=1}^q \beta_j h_{t-j}, \end{aligned}$$

114 where $h_t = \sigma_t^2/\omega$, $u_t = \sqrt{\omega}\varepsilon_t$ and $\gamma_i = \alpha_i/\omega$. Since ω is unknown, [Lee and Noh](#)
 115 [\(2013\)](#) use the consistent estimator $\widehat{\omega}$ based on the QML method. The four
 116 estimation methods are referred in tables below as QML, Huber, Quantile and
 117 M-quantile, respectively.

118 A contaminated version $\{Z_t\}$ of $\{X_t\}$ is obtained by adding outliers
 119 according to

$$120 \quad Z_t = X_t + mI_t,$$

121 where $\{X_t\}$ is defined by (1), m is the magnitude of the outlier, $\{X_t\}$ and
 122 $\{I_t\}$ are independent and $\{I_t\}$ is a sequence of IID random variables with
 123 $\mathbb{P}(I_t = -1) = \mathbb{P}(I_t = 1) = p/2$ and $\mathbb{P}(I_t = 0) = 1 - p$, $p \in (0, 1)$. Here we take
 124 $p = 0.01$ and $m = 7$. The uncontaminated case corresponds to $m = 0$. Other
 125 methods for including outliers in heteroscedastic time series can be found in
 126 [Carnero et al. \(2005\)](#) and [Carnero et al. \(2012\)](#). The sample size of the series
 127 is $n = 500, 1000, 2000$, and the sample mean and sample mean square error

128 (MSE) of each estimator are based on 1000 replicates. The data is generated
129 from a GARCH(1, 1) process with parameters $(\omega_0, \alpha_{01}, \beta_{01}) = (0.1, 0.1, 0.8)$
130 and the innovations $\{\varepsilon_t\}$ follow either a standard normal distribution, denoted
131 by $N(0, 1)$, a Student's t distribution with 3 degrees of freedom (heavy tail),
132 denoted by $t(3)$, and a chi-squared distribution with 4 degrees of freedom
133 (asymmetric), denoted by $\mathcal{X}^2(4)$.

134 The empirical results are displayed in Tables 1-5 and the smallest MSE
135 values are in bold. Table 1 reports the case where $n = 500$ and $\{\varepsilon_t\} \sim N(0, 1)$.
136 When $m = 0$ (no contamination) and $\tau = 0.5$, the methods present similar per-
137 formances. The QML method is the best in terms of MSE, which is an expected
138 result since this method is the most efficient under the normal distribution, see
139 Hallin et al. (2022), who explored M-estimation of GARCH models through
140 extensive Monte Carlo simulations. QML is the most affected estimator in the
141 contaminated data scenario, while Huber and M-quantile methods outperform
142 the other estimators. Even a small amount of outliers affects the QML estima-
143 tor. Tables 2 and 3 correspond to $\{\varepsilon_t\} \sim N(0, 1)$ and sample sizes $n = 1000$
144 and $n = 2000$, respectively. When $m = 0$, all estimators present very close and
145 accurate estimates as the sample size increases. In the contaminated case, the
146 M-quantile method is the best. In Table 4, $\{\varepsilon_t\} \sim t(3)$ (symmetric and heavy
147 tail distribution), we see that the M-quantile method provides the best results
148 in uncontaminated and contaminated cases. In Table 5, $\{\varepsilon_t\} \sim \mathcal{X}^2(4)$ (asym-
149 metric distribution) and the quantile regression method is the most accurate
150 in the absence of outliers, while the M-quantile method is the best in the
151 presence of outliers.

152 As a conclusion, the M-quantile estimators are the most robust against
153 additive outliers under different innovation distributions.

Table 2 Empirical Mean and MSE of GARCH(1, 1) model estimated by QML, Huber, quantile and M-quantile methods, considering $n = 1000$ and $N(0, 1)$ innovation distribution.

	Mean			MSE		
	$\hat{\omega}$	$\hat{\alpha}_1$	$\hat{\beta}_1$	$\hat{\omega}$	$\hat{\alpha}_1$	$\hat{\beta}_1$
	0.1180	0.0833	0.8078	0.0034	0.0007	0.0043
	0.1184	0.1045	0.7757	0.0056	0.0009	0.0093
			QML			
			Huber			
			Quantile			
τ						
0.05	-	0.1295	0.7459	-	0.0168	0.0157
0.25	-	0.1247	0.7563	-	0.0093	0.0132
0.45	-	0.1184	0.7528	-	0.0087	0.0089
0.50	-	0.1101	0.7635	-	0.0051	0.0079
			M-quantile			
τ						
0.05	0.1754	0.1493	0.7307	0.0263	0.0149	0.0181
0.25	0.1134	0.1524	0.8169	0.0024	0.0050	0.0063
0.45	0.1156	0.0900	0.8108	0.0035	0.0009	0.0067
0.50	0.1163	0.0824	0.8105	0.0027	0.0007	0.0058

	Mean			MSE		
	$\hat{\omega}$	$\hat{\alpha}_1$	$\hat{\beta}_1$	$\hat{\omega}$	$\hat{\alpha}_1$	$\hat{\beta}_1$
	0.3952	0.1970	0.6585	0.2259	0.0271	0.0963
	0.1787	0.0901	0.7669	0.0187	0.0146	0.0298
			QML			
			Huber			
			Quantile			
τ						
0.05	-	0.1517	0.7121	-	0.0319	0.0428
0.25	-	0.1497	0.7251	-	0.0431	0.0379
0.45	-	0.1512	0.7288	-	0.0447	0.0503
0.50	-	0.1387	0.7395	-	0.0351	0.0363
			M-quantile			
τ						
0.05	0.2152	0.1809	0.7199	0.0344	0.0362	0.0312
0.25	0.1587	0.1364	0.7105	0.0349	0.0389	0.0322
0.45	0.1580	0.1484	0.7239	0.0295	0.0371	0.0296
0.50	0.1648	0.0897	0.7682	0.0221	0.0169	0.0241

(a) Without outliers

(b) With outliers

Table 4 Empirical Mean and MSE of GARCH(1, 1) model estimated by QML, Huber, quantile and M-quantile methods, considering $n = 2000$ and $t(3)$ innovation distribution.

	Mean			MSE		
	$\hat{\omega}$	$\hat{\alpha}_1$	$\hat{\beta}_1$	$\hat{\omega}$	$\hat{\alpha}_1$	$\hat{\beta}_1$
	0.0893	0.0727	0.7681	0.0121	0.0060	0.0146
	0.0891	0.0912	0.7713	0.0103	0.0049	0.0107
			QML			
			Huber			
			Quantile			
τ			M-quantile			
0.05	-	0.1013	0.7974	-	0.0041	0.0061
0.25	-	0.1086	0.7798	-	0.0054	0.0074
0.45	-	0.1223	0.7691	-	0.0089	0.0089
0.50	-	0.1098	0.7863	-	0.0029	0.0032
			QML			
			Huber			
			Quantile			
			M-quantile			
τ			QML			
0.05	-	0.1256	0.7508	-	0.0224	0.0212
0.25	-	0.1240	0.7657	-	0.0213	0.0243
0.45	-	0.1479	0.7491	-	0.0262	0.0283
0.50	-	0.1189	0.7607	-	0.0196	0.0233
			QML			
			Huber			
			Quantile			
			M-quantile			
τ			QML			
0.05	0.3045	0.1667	0.6324	0.2031	0.0601	0.0994
0.25	0.1258	0.0881	0.7614	0.0278	0.0264	0.0238
0.45			Huber			
0.50			Quantile			
			M-quantile			
τ			QML			
0.05	0.1521	0.1415	0.7621	0.0236	0.0324	0.0316
0.25	0.1382	0.1386	0.7769	0.0219	0.0268	0.0283
0.45	0.1358	0.1173	0.7586	0.0224	0.0216	0.0199
0.50	0.1297	0.1105	0.7817	0.0147	0.0153	0.0178

(a) Without outliers

(b) With outliers

Table 5 Empirical Mean and MSE of GARCH(1, 1) model estimated by QML, Huber, quantile and M-quantile methods, considering $n = 2000$ and centered $\mathcal{N}^2(4)$ innovation distribution.

τ	Mean			MSE		
	$\hat{\omega}$	$\hat{\alpha}_1$	$\hat{\beta}_1$	$\hat{\omega}$	$\hat{\alpha}_1$	$\hat{\beta}_1$
	0.1854	0.0598	0.7103	0.0384	0.0407	0.0547
	0.1382	0.0724	0.7556	0.0231	0.0348	0.0251
			QML			
			Huber			
			Quantile			
0.05	-	0.1071	0.7962	-	0.0058	0.0089
0.25	-	0.1131	0.7874	-	0.0087	0.0093
0.45	-	0.1360	0.7767	-	0.0124	0.0135
0.50	-	0.1251	0.7643	-	0.0142	0.0137
			M-quantile			
0.05	0.1164	0.1113	0.7827	0.0091	0.0058	0.0112
0.25	0.1225	0.1249	0.7752	0.0106	0.0127	0.0125
0.45	0.1288	0.1384	0.7733	0.0113	0.0139	0.0174
0.50	0.1241	0.1305	0.7599	0.0187	0.0194	0.0169

τ	Mean			MSE		
	$\hat{\omega}$	$\hat{\alpha}_1$	$\hat{\beta}_1$	$\hat{\omega}$	$\hat{\alpha}_1$	$\hat{\beta}_1$
	0.3598	0.1926	0.6015	1.0086	0.8753	0.1693
	0.1654	0.0541	0.7105	0.0854	0.0687	0.0927
			QML			
			Huber			
			Quantile			
0.05	-	0.1354	0.7651	-	0.0116	0.0188
0.25	-	0.1457	0.7634	-	0.0151	0.0236
0.45	-	0.1561	0.7569	-	0.0253	0.0223
0.50	-	0.1527	0.7527	-	0.0195	0.0217
			M-quantile			
0.05	0.1289	0.1496	0.7427	0.0301	0.0219	0.0186
0.25	0.1527	0.1615	0.7518	0.0253	0.0194	0.0225
0.45	0.1466	0.1591	0.7602	0.0214	0.0287	0.0211
0.50	0.1301	0.1422	0.7483	0.0239	0.0200	0.0215

(a) Without outliers

(b) With outliers

154 4 Real data application

155 The datasets considered in this paper are asset returns that are widely used by
 156 academics and practitioners and are well modeled by $GARCH(p, q)$ processes.
 157 We analyse the daily returns of two financial indices: the French continu-
 158 ous assisted quotation index (CAC40) and the Brazilian stock exchange index
 159 (Ibovespa). Both series are observed from January 3, 2011, to December 30,
 160 2021, corresponding to 2811 and 2719 observations, respectively.

161 Let p_t be the price of the asset at time t and X_t be the log-return defined
 162 as $X_t = 100 \log \frac{p_t}{p_{t-1}}$. Table 6 reports some statistics of the data. As widely
 163 documented in the literature, both indices exhibit negative skewness and excess
 164 kurtosis. CAC40 index returns on average about 0.0217% per day, slightly
 165 more than the Ibovespa index. In addition, the Ibovespa index displays a much
 166 higher risk than the CAC40 index. Both series were fitted with a $GARCH(1, 1)$
 167 model using the QML, Huber, Quantile and M-quantile methods. Figure 1
 168 shows that both series contain observations that behave as additive outliers.
 169 Thus, differences between the parameter estimates in the four methods are
 170 expected.

Table 6 Descriptive statistics of the daily log-returns of the financial indices CAC40 and Ibovespa

	CAC40	Ibovespa
Mean	0.0217	0.0149
Std. Deviation	1.2672	1.5988
Maximum	8.0561	13.0223
Minimum	-13.0983	-15.9930
Skewness	-0.7031	-0.8471
Kurtosis	8.5321	11.9224

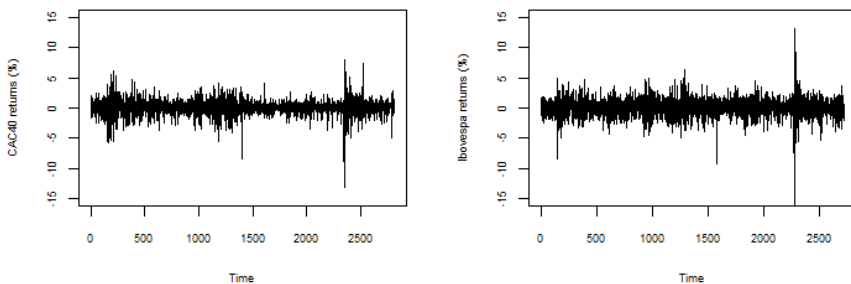


Fig. 1 Plot of the daily log-return series of CAC40 and Ibovespa Indices.

Table 7 QML, Huber, quantile and M-quantile estimates

	CAC40		$\hat{\beta}_1$		$\hat{\omega}$	$\hat{\alpha}_1$	Ibovespa	
	$\hat{\omega}$	$\hat{\alpha}_1$		QML			$\hat{\alpha}_1$	$\hat{\beta}_1$
	0.0545(0.0098)	0.1357(0.0172)	0.8316(0.0202)		0.0993(0.0222)	0.0835(0.0113)	0.8717(0.0177)	
	0.0248(0.0114)	0.0890(0.0127)	0.8602(0.0216)	Huber	0.0619(0.0088)	0.0565(0.0101)	0.8906(0.0215)	
				Quantile				
τ								
0.01	-	0.0722(0.0134)	0.7120(0.0618)	-	-	0.1047(0.0241)	0.8201(0.0221)	
0.50	-	0.1360(0.0181)	0.7374(0.0561)	-	-	0.0839(0.0145)	0.8274(0.0273)	
0.99	-	0.1336(0.0089)	0.8256(0.0158)	-	-	0.0741(0.0100)	0.9028(0.0161)	
				M-quantile				
τ								
0.01	0.0460(0.0111)	0.0762(0.0117)	0.8256(0.0189)	0.0945(0.0156)	0.0620(0.0093)	0.8617(0.0188)		
0.50	0.0246(0.0095)	0.1182(0.0124)	0.8402(0.0103)	0.0222(0.0112)	0.0594(0.0096)	0.8902(0.0174)		
0.99	0.0118(0.0090)	0.1174(0.0168)	0.8645(0.0246)	0.0201(0.0119)	0.0386(0.0113)	0.9391(0.0383)		

(a) Original dataset

	CAC40		$\hat{\beta}_1$		$\hat{\omega}$	$\hat{\alpha}_1$	Ibovespa	
	$\hat{\omega}$	$\hat{\alpha}_1$		M-quantile			$\hat{\alpha}_1$	$\hat{\beta}_1$
τ								
0.50	0.0238(0.0087)	0.1151(0.0104)	0.8478(0.0088)	0.0214(0.0107)	0.0544(0.0083)	0.8941(0.0157)		

(b) Modified dataset

Note: Values in parenthesis denote the standard error of the parameter estimated.

171 The parameter estimates are displayed in Table 7. For the CAC40 series,
 172 we see that the M-quantile and quantile regression estimates of α_1 increase as τ
 173 increases. For the Ibovespa series, the reverse phenomenon appears. This sug-
 174 gests the presence of asymmetry in both series (Lee and Noh (2013)). Moreover,
 175 there is a clear balance between α_1 and β_1 estimates.

176 As pointed out by Mukherjee (2008), the QML, Huber and M-quantile
 177 methods, which are based on M-estimation, should provide a consistent esti-
 178 mate of parameter β_1 in the GARCH(1,1) model. This is observed in Table 7
 179 since these three methods' estimates of this parameter are very close.

180 To identify the effect of the abrupt observations that appear in Figure 1,
 181 we compare the classical sample ACF and the robust sample ACF using $Q_n(\cdot)$
 182 estimator (Lévy-Leduc, Boistard, Moulines, Taqqu, and Reisen (2009)) when
 183 the sample ACF is computed either from the original data or from the modi-
 184 fied data obtained by replacing the abrupt observations with the sample mean.
 185 Figure 2 shows that the classical sample ACFs of the CAC40 index display
 186 different autocorrelation values, which is an expected result when there are
 187 additive outliers in the data, see Reisen et al. (2017). Figure 3 displays the
 188 robust sample ACFs. We see that the values are quite close. This simple exam-
 189 ple shows that the abrupt observations affect the ACFs similarly to additive
 190 outliers. In addition, the modified data was also used to estimate the param-
 191 eters of the GARCH(1,1) model. Table 7 shows that the M-quantile estimates
 192 are very close to those obtained from the original data.

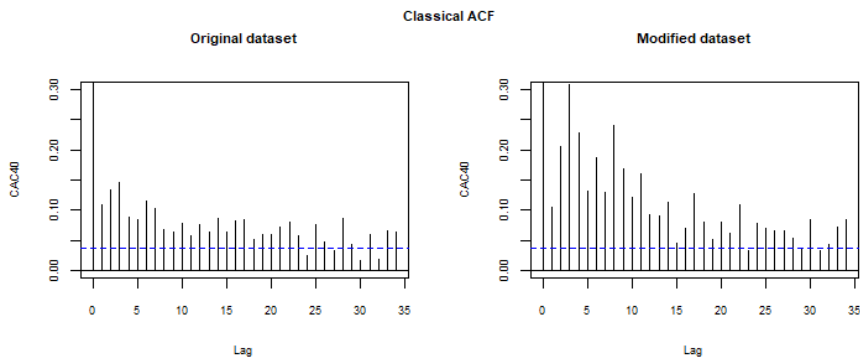


Fig. 2 Classical ACF from the original and modified CAC40 index data.

193 The M-quantile approach also allows to compute the value at risk (VaR),
 194 which is estimated for any $\tau \in (0.0, 1.0)$ (Lee & Noh, 2013; Xiao & Koenker,
 195 2009). This quantity is plotted in Figure 4 for CAC40 and Ibovespa data.

196 The estimated volatilities for $\tau = 0.5$ in the CAC40 and Ibovespa indices
 197 are plotted in Figures 5 and 6, respectively. As can be seen, the M-estimators
 198 accurately estimated the true volatilities, while these are overestimated by
 199 the QML method. This large positive bias leads to a large impact on risk
 200 assessment. Instead of using sample autocorrelations to select the orders of the

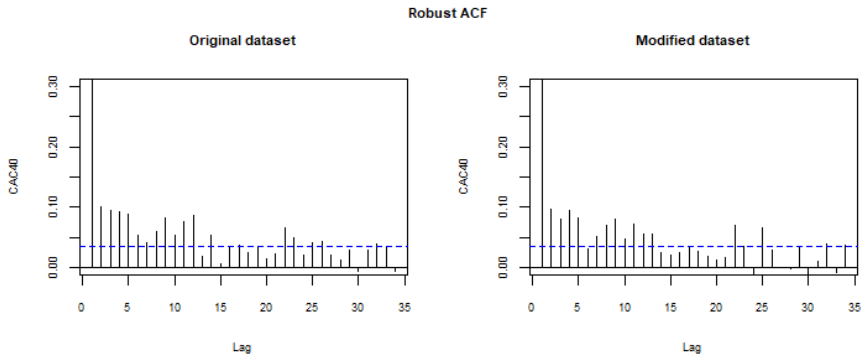


Fig. 3 Robust ACFs from the original and modified CAC40 index data.

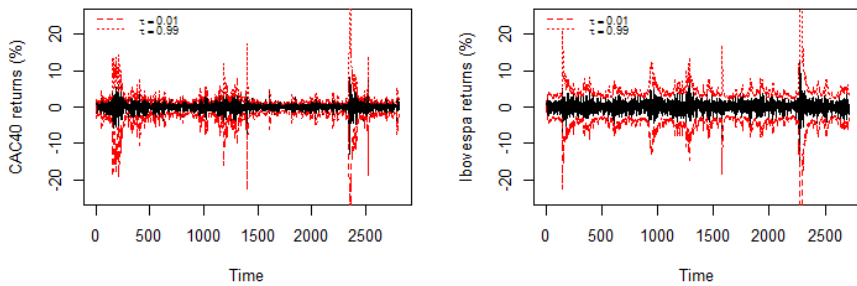


Fig. 4 Estimated conditional quantiles for CAC40 and Ibovespa log-returns.

201 model fitting, the standard criteria AIC, BIC and their variants can be used,
 202 as well as cross validation techniques when enough data are available. Other
 203 GARCH models were fitted to the CAC40 data and their empirical MSEs
 204 were computed. The results are displayed in Table 8. The best fit is obtained
 205 with the GARCH(1,1) model adjusted by the M-quantile method since this
 206 approach gives the smallest MSEs for the one-step ahead prediction error of
 207 the sample volatility. To corroborate the above statements, the AICs of fitted
 models were computed and appear in Table 9.

Table 8 Empirical MSEs of one-step-ahead prediction errors of GARCH models estimated by QML, Huber, quantile and M-quantile methods.

	CAC40			
	ARCH(1)	GARCH(1, 1)	GARCH(1,2)	GARCH(2, 1)
QML	1.895	0.611	1.247	1.043
Huber	1.991	0.857	1.514	1.212
quantile	2.143	1.016	2.516	1.914
<i>M</i> -quantile	1.978	0.519	1.161	1.315

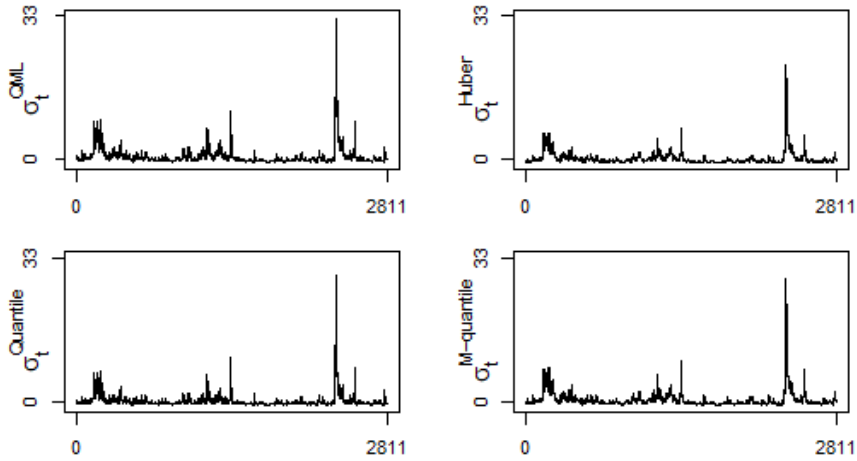


Fig. 5 Estimated volatility for the CAC40 index.

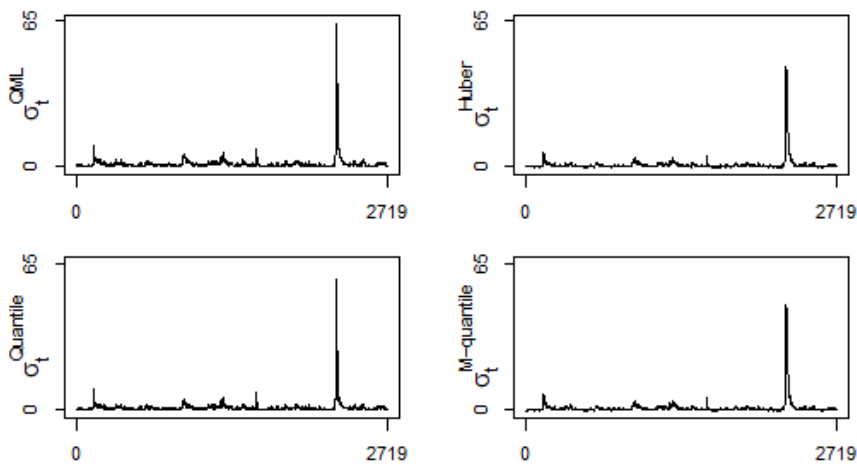


Fig. 6 Estimated volatility for the Ibovespa index.

Table 9 AIC criteria of the fitted GARCH models given in Table 8.

	CAC40			
	ARCH(1)	GARCH(1, 1)	GARCH(1, 2)	GARCH(2, 1)
QML	3.243	3.014	3.160	3.114
Huber	3.261	3.021	3.185	3.037
quantile	3.681	3.036	3.719	3.186
<i>M</i> -quantile	3.281	3.000	3.054	3.193

208 **5 Conclusion**

209 In this paper, we proposed M-quantile estimators for GARCH processes. Monte
210 Carlo simulations showed that these estimators outperform QML, Huber and
211 M-quantile estimators in the presence of additive outliers. The real data analy-
212 sis in the context of financial time series revealed that the estimated volatility
213 using the M-quantile method is not affected by aberrant observations, which is
214 not true when QML estimation is used. Therefore, the M-quantile estimator is
215 recommended in practice, whether or not outliers are present in the data. As
216 a future investigation, it will be interesting to consider M-quantile estimation
217 method in the spectral domain to estimate the parameters of GARCH models.

18 *M-quantile estimation for GARCH models*

218 **A Appendices**

219 **A.1 R Code used to Estimate the MQGARCH(1,1)**

The results below are generated from an R script.

```

#' @title Robust Estimates for GARCH(1,1) Model
#'
#' @name MQGARCH
#' @aliases MQGARCH
#'
#' @description Methods for fitting a GARCH(1,1) model with daily log return
#' time series, using three methods of M-Estimates:
#' (1) QML;
#' (2) Huber;
#' (3) M-quantile.
#'
#' @param data a time series of log returns, need to be numeric value.
#' @param methods robust M-Estimate method used for Garch(1,1) model, "QML",
#' "M" and "M-quantile"
#' @param optimizer optimizer used for optimization, one of "nloptr",
#' "Rsolnp", "nlnmb", default is "Rsolnp".
#' @param optimizer_x0 user-defined starting point for searching the optimum,
#' c(x0_omega, x0_alpha, x0_beta). Default is "FALSE", where the starting point
#' will be calculated instead of being user-defined.
#' @param optimizer_control list of control arguments passed to the optimizer.
#' Default is list(trace=0). If wanting to print out the optimizer result, use
#' list() instead.
#' @param stderr_method method used to calculate standard error, one of
#' "numDeriv", "optim", "sandwich", default is "numDeriv" using hessian from
#' numDeriv
#'
#' @return
#' A rg object(S3), the components of the object are:
#' \item{data}{the log returns data object for the rg model to be fitted}
#' \item{data_name}{the name of data variable input used}
#' \item{methods}{the method called}
#' \item{optimizer}{the optimizer called}
#' \item{optimizer_x0}{user-defined or calculated starting point for
searching the optimum}
#' \item{optimizer_control}{the list of control arguments passed to the
optimizer}
#' \item{optimizer_result}{output of the called optimizer}
#' \item{stderr_method}{the method called to calculate standard error}
#' \item{QML}{logical argument controlling the non-robustness of the fitting
method}
#' \item{fitted_pars}{Garch(1,1) parameter estimations output of the called
method}
#' \item{sigma}{the time series of the conditional standard deviation}
#' \item{yt}{the time series of  $\log(\text{data}^2)$ }
#' \item{observed_I}{observed information matrix}
#' \item{objective}{the optimal likelihood value obtained by the optimizer}
#' \item{time_elapsed}{the time used for the optimization routine}
#' \item{message}{the message of the convergence status produced by the
called solver}
#' \item{standard_error}{standard errors of the fitted parameters using the
method called}
#' \item{t_value}{t-values of omega, alpha and beta}

```

```

#' \item{p_value}{p-values of omega, alpha and beta}
#'
#' @details
#' The MQGARCH function fits a GARCH(1, 1) model to a time series of
#' log return data, using one of the two methods of robust extended M-Estimates
#' with certain parameters specified by the user, with guidance and examples
#' from the vignette. The user can also specify the optimizer used during
#' optimization procedure, and the method used to calculate standard error for
#' the fitted parameters.
#'
#' For details of the list of control arguments, please refer to
#' nloptr::nloptr, Rsolnp::solnp, nlminb.
#'
#' @references Patrocinio, P.F & Reisen, V.A & Bondon, P. & Monte,
#' E. & Danilevich, I. (2023). Heteroscedastic Process: An M-quantile Approach.
#' Computational Economics (preprint).
#'
#'
#' This code is under development
#' @rdname rg-MQGARCH
#' @export
#
library(robustbase)
library(MASS)
MQGARCH <- function(data, methods = c("M", "QML", "M-quantile"),
                    optimizer = c("Rsolnp", "nloptr", "nlminb"),
                    optimizer_x0 = FALSE, optimizer_control = list(trace=0),
                    stdErr_method = c("numDeriv", "optim", "sandwich")){

  if(!is.numeric(data) || length(data)==0)
    stop("Data must be a numeric vector of non-zero length")

  methods = match.arg(methods)
  optimizer = match.arg(optimizer)
  stdErr_method = match.arg(stdErr_method)

  assign("methods", methods, envir = .GlobalEnv)

  assign("div", 1.0, envir = .GlobalEnv)

  fit <- rgFit_local(data, optimizer, optimizer_x0, optimizer_control)

  # std_err calculation
  if(optimizer == "Rsolnp"){
    solution <- fit$optimizer_result$spars
    H <- fit$optimizer_result$hessian
  } else if(optimizer == "nloptr"){
    solution <- fit$optimizer_result$solution
    stop("use Rsolnp optimizer for now")
  } else if(optimizer == "nlminb"){
    solution <- fit$optimizer_result$par
    stop("use Rsolnp optimizer for now")
  }
}

```

```

std_errors <- sqrt(diag(abs(solve(H)/length(data))))
standard_error <- std_errors[2:4]
fit$observed_I <- -H[c(2,3,4), c(2,3,4)]
standard_error[1] <- standard_error[1] * (fit$fitted_pars[1] / solution[1])
t_value <- fit$fitted_pars / standard_error
p_value <- 2*(1-pnorm(abs(t_value)))

names(standard_error) <- c("omega", "alpha", "beta")
names(t_value) <- c("omega", "alpha", "beta")
names(p_value) <- c("omega", "alpha", "beta")

fit$data_name <- noquote(deparse(substitute(data)))
fit$standard_error <- standard_error
fit$t_value <- t_value
fit$p_value <- p_value

structure(fit, class="rg")
}
#' @export
k = 3
MQGARCHDistribution <- function(param = c(8.76e-04, 0.135, 0.686),
                               methods = c("M", "QML", "M-quantile"),
                               fixed_pars = c(0.85, 3.0),
                               optimizer = c("Rsolnp", "nloptr", "nlminb"),
                               optimizer_x0 = FALSE,
                               optimizer_control = list(),
                               stdErr_method = c("numDeriv", "optim", "sandwich"),
                               n = 2000, m = 100, rseed = 42){

  methods <- match.arg(methods)
  optimizer <- match.arg(optimizer)
  stdErr_method <- match.arg(stdErr_method)

  par(mfrow=c(2,2))
  spec <- ugarchspec(mean.model = list(armaOrder = c(0,0),
                                       include.mean = FALSE))

  if(length(param)!=3){stop("the parameters for norm distribution should only be
                           omega, alpha, beta")}

  fixed <- param
  names(fixed) <- c("omega", "alpha", "beta")
  fspec <- spec
  setfixed(fspec) <- fixed
  y <- ugarchpath(fspec, n.sim = n, m.sim = m, rseed = 42)
  y. <- y@path$seriesSim

  qml_res <- matrix(0.0, nrow = m, ncol = 3)
  for( i in 1:m){
    y_ <- y.$[((i-1)*n+1):(i*n)]
    fit <- MQGARCH(y_, methods = methods, optimizer=optimizer,
                  optimizer_x0 = optimizer_x0,
                  optimizer_control = optimizer_control,
                  stdErr_method = stdErr_method)
  }
}

```

```

    qml_res[i,1:3] <- fit$fitted_pars
  }

  d_omega <- density(qml_res[,1])
  d_omega
  plot(d_omega, main=paste("Parameter", expression( alpha ),
                          "0 \n True value: ",fixed[1]), cex=.5)

  d_alpha1 <- density(qml_res[,2])
  d_alpha1
  plot(d_alpha1, main=paste("Parameter", expression( alpha ),
                          "1 \n True value: ", fixed[2]), cex=.5)

  d_beta1 <- density(qml_res[,3])
  d_beta1
  plot(d_beta1, main=paste("Parameter", expression( beta ),
                          "1 \n True value: ", fixed[3]), cex=.5)
}
#' @export
rgFit_local <- function(data, optimizer, optimizer_x0, optimizer_control){

  start_time <- Sys.time()
  # optimizer/optimizer_control

  n <- length(data)
  v_data <- new_var(data)
  data_normalized <- data/sqrt(v_data)
  data_normalized <- data_normalized-median(data_normalized)
  for(i in 1:n){
    if(isTRUE(data_normalized[i] == 0)){
      data_normalized[i] <- 10^(-10)
    }
  }
  vini <- new_var(data_normalized)

  res <- nEst(data_normalized, vini, optimizer, optimizer_x0, optimizer_control)
  optimizer_result <- res

  if(optimizer == "Rsolnp"){
    fitted_pars <- res$pars[1:3]
    names(fitted_pars) <- c("omega", "alpha", "beta")
    fitted_pars[1] <- fitted_pars[1]*v_data
    objective <- res$values[length(res$values)]
    message <- res$convergence
  } else if (optimizer == "nloptr"){
    fitted_pars <- res$solution[1:3]
    names(fitted_pars) <- c("omega", "alpha", "beta")
    fitted_pars[1] <- fitted_pars[1]*v_data
    objective <- res$objective
    message <- res$message
  } else if (optimizer == "nlminb"){
    ifitted_pars <- res$par[1:3]
    names(fitted_pars) <- c("omega", "alpha", "beta")
  }
}

```

```

    fitted_pars[1] <- fitted_pars[1]*v_data
    objective <- res$objective
    message <- res$message
  } else{
    NA
  }
}
sigma <- sigmaCal(fitted_pars, data)

time_elapsed <- Sys.time() - start_time

list(data=data,
      methods = methods,
      optimizer=optimizer,
      optimizer_x0=res$x0,
      optimizer_control=optimizer_control,
      optimizer_result=optimizer_result,
      fitted_pars = fitted_pars,
      objective=objective,
      time_elapsed=time_elapsed,
      message=message,
      sigma=sigma,
      yt=Muestram)
}
#' @export
nEst <- function(y, vini, optimizer, optimizer_x0, optimizer_control){

  suppressWarnings(rm(Muestrac, Muestram))

  std <- FALSE

  n <- prod(length(y))
  yc <- y[1:n]^2
  y2 <- log(yc)

  assign("Muestrac", yc, envir = .GlobalEnv)
  assign("Muestram", y2, envir = .GlobalEnv)

  alfa0min <- 0.1
  alfa0max <- 1
  alfa1min <- 0.0
  alfa1max <- 1.
  beta1min <- 0.
  beta1max <- 1.
  nalfa1 <- 5
  nalfa0 <- 5
  nbeta1 <- 5
  ml <- 10^8
  flag <- 0
  lmalfa1 <- (alfa1max-alfa1min)/nalfa1
  lmalfa0 <- (alfa0max-alfa0min)/nalfa0
  lmbeta1 <- (beta1max-beta1min)/nbeta1
  if(std){

```



```

if(length(optimizer_x0) > 1){
  x0 <- c(optimizer_x0[1:3], vini, optimizer_x0[4])
} else{
  shapemin <- 3.0
  shapemax <- 31.
  nshape <- 30
  lmshape <- (shapemax-shapemin)/nshape

  for(nj in 0:nalfa1){
    alfa1 <- alfa1min+nj*lmalfa1
    for(nk in 0:nbeta1){
      beta1 <- beta1min+nk*lmbeta1
      if(alfa1+beta1 < 1){

        for(ni in 0:nalfa0){
          alfa0 <- alfa0min+ni*lmalfa0
          for(np in 0:nshape){
            shape <- shapemin+np*lmshape

            var <- rep(0.0, n)
            var[1] <- vini
            for(ki in c(k, 20)){
              l <- ki+1
              for(i in 2:n){
                var[i]<-alfa0+(alfa1*rk(yc[i-1]/var[i-1],ki,l)
                  +beta1)*var[i-1]
              }

              nml <- mean(nfun(y2[2:n]-log(var[2:n]), shape))
              if(isTRUE(nml<m1)){
                flag <- 1
                vi <- c(alfa0,alfa1,beta1,shape)
                m1 <- nml
              }
            }
          }
        }
      }
    }
  }
  x0 <- c(vi[1:3], vini, vi[4])
}

lb <- c( 0., 0., 0., 0.0, 3.0)
ub <- c( 1.0, 1., 1., Inf, Inf)
} else{

if(length(optimizer_x0) > 1){
  x0 <- c(optimizer_x0[1:3], vini)
} else{
  for(nj in 0:nalfa1){
    alfa1 <- alfa1min+nj*lmalfa1
    for(nk in 0:nbeta1){

```

```

beta1 <- beta1min+nk*lmbeta1

if(alfa1+beta1 < 1 ){
  for(ni in 0:nalfa0){
    alfa0 <- alfa0min+ni*lmalfa0

    var <- rep(0.0, n)
    var[1] <- vini
    for(ki in c(k, 20)){
      l <- ki+1
      for(i in 2:n){
        var[i] <- alfa0+(alfa1*rk(yc[i-1]/var[i-1],ki,l)+
          beta1)*var[i-1]
      }

      nml <- mean(nfun(y2[2:n]-log(var[2:n])))

      if(isTRUE(nml<ml)){
        flag <- 1
        vi <- c(alfa0,alfa1,beta1)
        ml <- nml
      }
    }
  }
}
x0 <- c(vi, vini)
}
lb <- c( 0., 0., 0., 0.0)
ub <- c( 1.0, 1.0, 1.0, Inf)
}
# Please, see Equations 9, 10 and 11 in Muler and Yaohai (2008)

if (optimizer == "nloptr"){
  res <- nloptr::nloptr(x0 = x0,
    eval_f = Fmue,
    lb = lb,
    ub = ub,
    opts=optimizer_control)

  res$x0 <- x0

  return(res)
} else if (optimizer == "Rsolnp"){

  res <- Rsolnp::solnp(pars = x0,
    fun = Fmue,
    LB = lb,
    UB = ub,
    ineqfun = function(vi){vi[2]+vi[3]},
    ineqLB = 0.0,
    ineqUB = 1.0,

```

```

                                control = optimizer_control)
  res$x0 <- x0
  return(res)
} else if (optimizer == "nlminb"){
  res <- nlminb(start = x0,
                objective = Fmue,
                control = optimizer_control,
                lower = lb,
                upper = ub)
  res$x0 <- x0
  return(res)
} else {
  NA
}
}
#' @export
new_var <- function(x){
  n <- length(x)
  tausq_x <- tau_sq(x)
  x_square <- x^2
  tausq_xsquare <- tau_sq(x_square-1)

  i <- 1
  v <- x[1]^2

  while(isTRUE(v > tausq_x+tausq_xsquare) & isTRUE(i<30)){
    i <- i+1
    v <- x[i]^2
  }

  if(i==30){
    error <- 1
  }
  v <- tausq_x

  v
}
#' @export
tau_sq <- function(x){
  s <- s_est(x)
  r <- rho(x/s)
  t <- mean(r)*s^2/0.4797

  t
}
#' @export

```

```

s_est <- function(x){

  b <- 1.625
  emed <- 0.675
  s <- 1
  eps <- 1.
  n <- 1

  m <- median(abs(x))/emed
  x <- x/m
  rho1 <- rho(x/0.405)
  a <- mean(rho1)/b
  v <- 1-a # Starting value
  si <- a # Starting value
  rho1 <- rho(x/(0.405*si)) # Starting value
  a<- mean(rho1)/b # Starting value
  vi <- 1-a # Starting value
  AUX <- v * vi # Starting value

  while (isTRUE(eps> 0.005) & isTRUE(AUX > 0)){
    n <- n+1
    s <- si
    v <- vi
    si <- a*s
    rho1 <- rho(x/(0.405 * si))
    a <- mean(rho1)/b
    vi <- 1-a
    AUX <- v*vi
    eps <- abs(s-si)/s
  }

  nsec <- 0
  while(isTRUE(eps>0.005)){
    ns <- (s+si)/2
    rho1 <- rho(x/(0.405*ns))
    a <- mean(rho1)/b
    nv <- 1-a
    AUX <- nv * vi
    if(isTRUE(AUX<0)){
      v <- nv
      s <- ns
    }
    else{
      vi <- nv
      si <- ns
    }
  }
  eps<- abs(s-si)/s
  nsec <- nsec + 1
  n <- n+1
}
s <- s*m
if(n>30){
  n <-n
}

```

```

}
s
}
#' @export

tau = 0.5

rho <- function(x){

  if(methods == "QML"){

    ps <- x^2/2

  }

  if(methods == "M-quantile"){

    tau = tau
    c = 1.5
    U <- abs(x)
    i <- (U <= c)
    omega <- (x <= 0)
    h <- t(abs(tau - t(omega)))
    ps <- (((x^2)/2*h)*i + (c*U - (c^2)/2)*h*(1 - i))

  }

  if(methods == "M"){

    ps <- Mpsi(x=x, cc=1.5, psi = "huber", deriv = -1)

  }else{

    NA

  }

  ps
}
#' @export
nfun <- function(x){

  x <- (exp(x)-x +log(2*pi))/2

  ps <- freg(x)

  ps
}
#' @export
sigmaCal <- function(pars, data){

  n <- prod(length(data))

```

```

var <- rep(0.0, n)
var[1] <- pars[1]/(1-pars[3])

if(pars[1] >0 & pars[2] >=0 & pars[3] >=0){
  l <- k+1
  for(i in 2:n){
    var[i]<-pars[1]+(pars[2]*rk(data[i-1]^2/var[i-1],k,l)+pars[3])*var[i-1]
  }
}

var
}
#' @export
Fnue <- function(start_pars){

  y2 <- Muestram
  yc <- Muestrac

  n <- prod(length(y2))

  vi <- start_pars[1:3]
  vini <- start_pars[4]

  var <- rep(0.0, n)
  var[1] <- vini

  nml <- 10^7

  if(vi[1] >0 & vi[2] >=0 & vi[3] >=0){
    for(ki in c(k,20)){

      l <- ki+1
      for(i in 2:n){
        var[i]<-vi[1]+(vi[2]*rk(yc[i-1]/var[i-1],ki,l)+vi[3])*var[i-1]
      }
      ml <- mean(nfun(y2[2:n]-log(var[2:n])))

      if(is.nan(ml) || is.nan(nml)){
        break
      }
      else if(ml<nml){ nml <- ml}
    }
  }

  nml
}
#' @export
freg <- function(x){

  if(methods == "QML"){

    g <- x

```

```

}
if(methods == "M-quantile"){
  tau = tau
  c = 1.5
  U <- sign(x)
  i <- (U <= c)
  omega <- (x <= 0)
  h <- t(abs(tau - t(omega)))
  g <- x*h*i + c*U*h*(1 - i)

} else{
  g <- Mpsi(x=x, cc=1.5, psi = "huber", deriv = -1)
}

g
}
#' @export
rk <- function(x, k, l){

  if(methods == "QML" || k == l){

    g <- x

  }

  if(methods == "M-quantile" || k == l){
    tau = tau
    c = 1.5
    U <- sign(x)
    i <- (U <= c)
    omega <- (x <= 0)
    h <- t(abs(tau - t(omega)))
    g <- x*h*i + c*U*h*(1 - i)
  }

  if(methods=="M"){

    g <- Mpsi(x=x, cc=1.5, psi = "huber", deriv = -1)

  }

  g
}

#knitr::stitch('rg-MQGARCH.R')

```

The R session information (including the OS info, R version and all packages used):

```

sessionInfo()

## R version 4.1.2 (2021-11-01)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 22621)
##

```

```

## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.1252 LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252 LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] stats      graphics  grDevices  utils      datasets  methods   base
##
## other attached packages:
## [1] tsqn_1.0.0      fracdiff_1.5-1  MASS_7.3-56    robustbase_0.93-9
## [5] gridExtra_2.3   quantmod_0.4.20 TTR_0.24.3     xts_0.12.1
## [9] zoo_1.8-9
##
## loaded via a namespace (and not attached):
## [1] lattice_0.20-45 grid_4.1.2      gtable_0.3.0    magrittr_2.0.1  evaluate_0.15
## [6] highr_0.9       stringi_1.7.6  curl_4.3.2      tools_4.1.2     stringr_1.4.0
## [11] DEoptimR_1.0-9  xfun_0.37      compiler_4.1.2 knitr_1.38

Sys.time()

## [1] "2023-03-09 16:25:02 -03"

```


²³³ **A.2 R Code to Summarize the MQGARCH(1,1) Model**

March 9, 2023

The results below are generated from an R script.

```

#' @title Summary for rg class
#'
#' @description Summary for rg S3 class
#'
#' @param fit A RG fit object of class \link{rg}
#' @param digits the number of digits for print and plot, default is 3.
#' @param main_name the title of the plot, default is
#' "Conditional SD (vs returns)"
#' @param estimation_pos string that determines the legend position that
#' specifies omega, alpha, beta estimations. Choice of "bottomright", "bottom",
#' "bottomleft", "left", "topleft", "top", "topright", "right" and "center".
#' Default is "topleft".
#' @param line_name_pos string that determines the legend position that
#' specifies the names of lines in the plot. Choice of "bottomright", "bottom",
#' "bottomleft", "left", "topleft", "top", "topright", "right" and "center".
#' Default is "topright".
#' @param par_graphical graphical parameters that can be set, which is in the form of
#' par(...). The default is par(no.readonly = TRUE).
#' @param original_ a logical argument. If TRUE, the original return will be
#' plotted. Default is FALSE
#' @param pctReturn_ a logical argument. IF TRUE, the plot function will plot
#' the returns in percentage instead of original. Default is TRUE.
#' @param abs_ a logical argument, when TRUE, the plot function will plot
#' abs(returns) with conditional standard deviation instead of returns,
#' default to TRUE.
#'
#' @name rg-summary
#' @aliases summary.rg
#' @aliases print.rg
#' @aliases plot.rg
#' @aliases aef
#'
#' This code is under development
#' @rdname rg-summary
#' @export
summary.rg <- function(fit, digits = 3){
  res <- rbind(round(fit$fitted_pars, digits),
               round(fit$standard_error, digits), round(fit$t_value, digits),
               round(fit$p_value, digits))
  colnames(res) <- names(fit$fitted_pars)
  rownames(res) <- c("Estimates", "Std. Errors", "t-statistic", "p-value")
}

```

```

cat("Model: ", fit$methods, " ")
#cat("\nData: ", fit$data_name, "\n")
cat("Observations: ", length(fit$data), "\n")
cat("\nResult: ", fit$fitted_pars, "\n")
cat("\nStd.Errors", fit$standard_error, "\n")
cat("\np-values", fit$p_value, "\n")
#print(res)
#cat("\nLog-likelihood: ", fit$objective)
cat("\n\nOptimizer: ", fit$optimizer)
cat("\nInitial parameter estimates: ", fit$optimizer_x0[1:3])
cat("\nTime elapsed: ", fit$time_elapsed)
cat("\nConvergence Message: ", fit$message)
}
#' @rdname rg-summary
#' @export
print.rg <- function(fit, digits = 3){

  res <- rbind(round(fit$fitted_pars, digits))
  colnames(res) <- names(fit$fitted_pars)
  rownames(res) <- c("Estimates (Std. Errors)")

  res[1,1] <- gsub(" ", "", paste(res[1,1], '( ', round(fit$standard_error[1],
    digits), ' )'))
  res[1,2] <- gsub(" ", "", paste(res[1,2], '( ', round(fit$standard_error[2],
    digits), ' )'))
  res[1,3] <- gsub(" ", "", paste(res[1,3], '( ', round(fit$standard_error[3],
    digits), ' )'))

  cat("Model: ", fit$methods, "\n")
  cat("Data: ", fit$data_name, "\n")
  cat("Result:\n")
  noquote(res)
}
#' @rdname rg-summary
#' @export
plot.rg <- function(fit, digits = 3, estimation_pos = "topleft",
  line_name_pos = "topright", par_ = par(no.readonly = TRUE),
  pctReturn_ = TRUE, abs_ = TRUE, original_ = FALSE,
  main_name = "Conditional Volatility (vs |pctReturns%|)"){

  .plot.garchsim(fit, digits, estimation_pos, line_name_pos, par_, pctReturn_,
    abs_, original_, main_name)
}
#' @rdname rg-summary
#' @export
aef <- function(fit, v=5){

  aTop <- .aValue(fit, TRUE, v)
  aBottom <- .aValue(fit, FALSE, v)

  aef <- aTop/aBottom

  cat("The AEF of the estimate with respect to QML is: ", aef, "\n")
}

```

```

    aef
  }

#knitr::stitch('rg-summary.R')

```

The R session information (including the OS info, R version and all packages used):

```

sessionInfo()

## R version 4.1.2 (2021-11-01)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 22621)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.1252 LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252 LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] stats      graphics  grDevices  utils      datasets  methods   base
##
## other attached packages:
## [1] tsqn_1.0.0      fracdiff_1.5-1  MASS_7.3-56   robustbase_0.93-9
## [5] gridExtra_2.3  quantmod_0.4.20 TTR_0.24.3    xts_0.12.1
## [9] zoo_1.8-9
##
## loaded via a namespace (and not attached):
## [1] lattice_0.20-45 grid_4.1.2      gtable_0.3.0   magrittr_2.0.1 evaluate_0.15
## [6] highr_0.9       stringi_1.7.6  curl_4.3.2     tools_4.1.2    stringr_1.4.0
## [11] tinytex_0.44    DEoptimR_1.0-9 xfun_0.37      compiler_4.1.2 knitr_1.38

Sys.time()

## [1] "2023-03-09 16:25:30 -03"

```

²³⁷ **A.3 Example of Use**

*

March 9, 2023

The results below are generated from an R script.

```

library(quantmod)
library(xts)
library(gridExtra)
library(tsqm)
graphics.off()
rm(list=ls())
setwd("C:/Users/Patrick Ferreira/Desktop/MQGARCH/R")
source("rg-MQGARCH.R")
source("rg-summary.R")

# Data Range year/month/day
sdate <- as.Date("2011-01-01")
edate <- as.Date("2021-12-31")

# CAC40 (^FCHI), Ibovespa (^BVSP)
ca_stock=getSymbols("^FCHI", from=sdate,to=edate,auto.assign = F)

## Warning: ^FCHI contains missing values. Some functions will not work if objects contain
missing values in the middle of the series. Consider using na.omit(), na.approx(), na.fill(),
etc to remove or replace them.

#bv_stock=getSymbols("^BVSP", from=sdate,to=edate,auto.assign = F)

# Typically use previous value for NA
no.na <- which(is.na(ca_stock[,6])) # no for NA
ca_price <- ca_stock[-no.na,6]
#no.na <- which(is.na(bv_stock[,6]))
#bv_price <- bv_stock[-no.na,6]

# log return using adjusted stock price
ca_rtn <- diff(log(ca_price)*100, 1)
#bv_rtn <- diff(log(bv_price)*100, 1)

ts.plot(ca_rtn, main="", ylab="CAC40 returns (%)", ylim = c(-15, 15), xlab="Time")
#ts.plot(bv_rtn, main="", ylab="Ibovespa returns (%)", ylim = c(-15, 15), xlab="Time")

fit <- MQGARCH(ca_rtn[-1], methods = "M-quantile") #at tau = 0.5
summary.rg(fit)

## Model: M-quantile Observations: 2810

```

*This report is automatically generated with the R package knitr (version 1.38).

```
##
## Result: 0.02462506 0.1182234 0.840262
##
## Std.Errors 0.009640252 0.01238386 0.02458654
##
##
## Optimizer: Rsolnp
## Initial parameter estimates: 0.64 0.2 0.6
## Time elapsed: 8.067856
## Convergence Message: 0

## Warning in file(con, "r"): cannot open file 'rg-referee_application.R': No such file or
directory
```

The R session information (including the OS info, R version and all packages used):

```
sessionInfo()

## R version 4.1.2 (2021-11-01)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 22621)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.1252 LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252 LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] stats graphics grDevices utils datasets methods base
##
## other attached packages:
## [1] tsqn_1.0.0 fracdiff_1.5-1 MASS_7.3-56 robustbase_0.93-9
## [5] gridExtra_2.3 quantmod_0.4.20 TTR_0.24.3 xts_0.12.1
## [9] zoo_1.8-9
##
## loaded via a namespace (and not attached):
## [1] knitr_1.38 magrittr_2.0.1 lattice_0.20-45 Rsolnp_1.16 stringr_1.4.0
## [6] highr_0.9 tools_4.1.2 parallel_4.1.2 grid_4.1.2 gtable_0.3.0
## [11] xfun_0.37 tinytex_0.44 curl_4.3.2 evaluate_0.15 stringi_1.7.6
## [16] compiler_4.1.2 DEoptimR_1.0-9 truncnorm_1.0-8

Sys.time()

## [1] "2023-03-09 16:38:36 -03"
```

References

240

241 Bai, Z.D., Rao, C.R., Wu, Y. (1992). M-estimation of multivariate linear
242 regression parameters under a convex discrepancy function. *Statistica*
243 *Sinica*, 2(1), 237–254.

244

245 Berkes, I., Hovath, L., Kokoszka, P. (2003). GARCH process: Structure and
246 estimation. *Bernoulli*, 9(2), 201–227.

247

248 Boudt, K., & Croux, C. (2010). Robust M-estimation of multivariate GARCH
249 models. *Computational Statistics and Data Analysis*, 54, 2459–2469.

250

251 Breckling, J., & Chambers, R. (1988). M-quantiles. *Biometrika*, 75(4), 761–
252 771.

253

254 Carnero, M.A. (2003). *Heterocedasticidad condicional, atípicos y cam-*
255 *bios de nivel en series temporales financieras* (Unpublished doctoral
256 dissertation). Universidad Carlos III de Madrid.

257 Carnero, M.A., Peña, D., Ruiz, E. (2005). Effects of outliers on the iden-
258 tification and estimation of GARCH models. *Journal of Time Series*
259 *Analysis*, 28, 471–497.

260

261 Carnero, M.A., Peña, D., Ruiz, E. (2012). Estimating GARCH volatility in
262 the presence of outliers. *Econometrics Letters*, 114, 86–90.

263

264 Chambers, R., & Tzavidis, N. (2006, 06). M-quantile models for small area
265 estimation. *Biometrika*, 93, 255–268.

266

267 Chang, I., Tiao, G.C., Chen, C. (1988). Estimation of time series parameters
268 in presence of outliers. *Technometrics*, 30, 1936–204.

269

270 Chen, C., & Liu, L. (1993). Joint estimation of model parameters and outlier
271 effects in time series. *Journal of the American Statistical Association*,
272 89, 284–297.

273

274 Douc, R., Moulines, E., Stoffer, D. (2014). *Nonlinear time series: Theory,*
275 *methods and applications with r examples*. Chapman and Hall.

- 276 Engle, R.F. (1982). Autoregressive conditional heteroscedasticity with esti-
277 mates of the variance of united kingdom inflation. *Econometrica*, *50*,
278 987-1007.
279
- 280 Fan, J., & Yao, Q. (2003). *Nonlinear time series*. New York: Springer-Verlag.
281 (Nonparametric and parametric methods)
- 282 Francq, C., & Zakoian, J.-M. (2019). *GARCH models: Structure, statistical*
283 *inference and financial applications*. Wiley.
- 284 Franses, P.H., & Ghijsels, H. (1999). Additive outliers, GARCH and
285 forecasting volatility. *International Journal Forecasting*, *15*, 1–9.
286
- 287 Hallin, M., Liu, H., Mukherjee, K. (2022). *M-estimation in GARCH models*
288 *in the absence of higher-order moments*.
- 289 Huber, P.J., & Ronchetti, E.M. (2009). *Robust statistics* (2nd ed.). Hoboken,
290 NJ: Wiley-Blackwell.
- 291 Iqbal, F. (2013). Robust estimation for the orthogonal GARCH model. *The*
292 *Manchester school*, *81*, 904–924.
293
- 294 Jones, M.C. (1994). Expectiles and m-quantiles are quantiles. *Statistics &*
295 *Probability Letters*, *20*(2), 149-153.
296
- 297 Ledolter, J. (1989). The effect of additive outliers on the forecast from ARIMA
298 models. *International Journal of Forecasting*, *5*, 231-240.
299
- 300 Lee, S., & Noh, J. (2013). Quantile regression estimator for GARCH models.
301 *Scandinavian Journal of Statistics*, *40*(1), 2–20.
302
- 303 Li, T.H. (2008). Laplace periodogram for time series analysis. *Journal of the*
304 *American Statistical Association*, *103*, 757-768.
305
- 306 Lévy-Leduc, C., Boistard, H., Moulines, E., Taqqu, M., Reisen, V. (2009,
307 12). Robust estimation of the scale and of the autocovariance function
308 of gaussian short and long-range dependent processes. *Journal of Time*
309 *Series Analysis*, *32*.
310

- 311 Maronna, R.A., Martin, R.D., Yohai, V.J. (2006). *Robust Statistics: Theory*
312 *and Methods*. New Jersey, USA: John Wiley & Sons.
- 313 Mendes, S.V.M. (2000). Assessing the bias of maximum likelihood estimates of
314 contaminated GARCH models. *Journal of Statistical Computation and*
315 *Simulation*, 67, 359-376.
- 316
- 317 Mukherjee, K. (2008). *M*-estimation in GARCH models. *Econometric Theory*,
318 24, 1530–1553.
- 319
- 320 Muler, N., & Yohai, V.J. (2008). Robust estimates for GARCH models.
321 *Journal of Statistical Planning and Inference*, 138, 2918–2940.
- 322
- 323 Reisen, V.A., Lévy-Leduc, C., Taqqu, M. (2017). An *M*-estimator for the
324 long-memory parameter. *Journal of Statistical Planning and Inference*,
325 187, 44–55.
- 326
- 327 Sarnaglia, A.J.Q., Reisen, V.A., Lévy-Leduc, C., Bondon, P. (2021). *M*-
328 regression spectral estimator for periodic ARMA models. an empirical
329 investigation. *Stochastic Environmental Research and Risk Assessment*
330 *volume*, 35, 653–664.
- 331
- 332 Tsay, R.S. (2001). *Analysis of financial time series*. Wiley-Interscience, New
333 York.
- 334 Tsay, R.S., & Chen, R. (2018). *Nonlinear time series analysis*. Nashville, TN:
335 John Wiley & Sons.
- 336 Wu, W.B. (2007). *M*-estimation of linear models with dependent errors. *The*
337 *Annals of Statistics*, 35(2), 495 – 521.
- 338
- 339 Xiao, Z., & Koenker, R. (2009). Conditional quantile estimation for general-
340 ized autoregressive conditional heteroscedasticity models. *Journal of the*
341 *American Statistical Association*, 104(488), 1696–1712.
- 342
- 343 Zheng, Y., Qianqian, Z., Li, G., Xiao, Z. (2016, 10). Hybrid quantile regres-
344 sion estimation for time series models with conditional heteroscedasticity.
345 *SSRN Electronic Journal*.
- 346

347 **Statements and Declarations**

348 **Funding.** This work was supported by Espirito Santo State Research
349 Foundation - FAPES (grant no. 012/2022-1) and Université Paris-Saclay -
350 CentraleSupélec (grant no. 1941199416F01).

351 **Acknowledge.** Part of this paper was revised when Prof. Valdério Reisen
352 was visiting Université Paris-Saclay - CentraleSupélec (from November 2021
353 to April 2022). This author is indebted to Université Paris-Saclay - Centrale-
354 Supélec for its financial support. The authors also thanks FAPES and CNPq,
355 Brazil. The authors are grateful to the three referees for the time and efforts
356 in providing very constructive and helpful comments.

357 **Non-financial interests.** The authors have no competing interests to
358 declare that are relevant to the content of this article.