Solving Economics and Finance Problems with Matlab

Autumn Semester 2022

GARCH and Index Returns

Niklas Leander Kampe
16-611-618

Charalampos Elenidis
21-607-643

Prof. PhD. Peter Gruber
Professor of Matlab
University of St. Gallen

18 December 2022

Contents

List of Tables

List of Figures

1 Introduction

This paper aims to investigate whether the GARCH model provides accurate estimates for index returns. In particular, a GARCH(1,1) model will be deployed on the basis of the S&P 500 daily returns, to estimate the parameters, standard errors and the filtered variance process. Afterwards, we simulate a 95% confidence interval for a 30-day prediction period, with the use of the estimated parameters and filtered volatility, which is then tested against the 30-day realizations in the sample.

2 Data Description

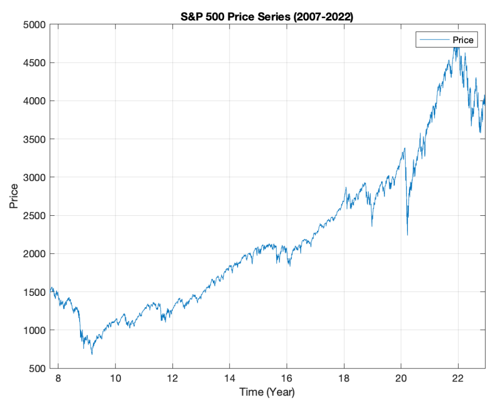

Figure 1: S&P 500 Prices

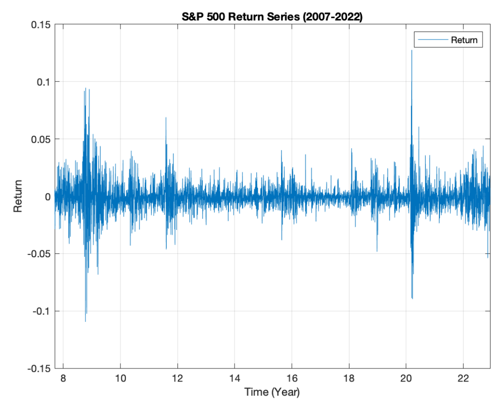

Figure 2: S&P 500 Returns

Figures 1 and 2 depict the developments in the S&P 500 prices and returns, for a fifteen-year period between 2007 and 2022. Given that the index under consideration represents, to a substantial extent, the stock market activity in the United States one would have to closely monitor those plots, especially for the period following the Big Financial Crisis of 2007-08 and the recent pandemic crisis in 2020-21. Indeed, we notice that the returns of the index were more volatile in 2009 and in 2011/12, a fact that could probably be attributed to the Eurodebt crisis. Accordingly, the impact of COVID-19 is clearly evident in 2020, when there was the global outbreak of the pandemic.

3 GARCH model

Dealing with time series presents the substantial challenge of heteroskedasticity, meaning that the variance of the process under review is not constant in our sample. To tackle this issue, Engle (1982) introduced the Autoregressive Conditional Heteroskedasticity model (ARCH), while later Bollerslev (1986) expanded the previously developed design and introduced the General Autoregressive Conditional Heteroskedasticity (GARCH), which allows the researcher to investigate the existence of features in the time series such as the volatility clustering and the volatility dependence. Prior to both of those discoveries, Mandelbrot (1963) noticed the tendency of big changes to be followed by big changes, and vice versa for small changes, which implies the deviation from the average variance for longer than time horizons than just one observation. This observation raised the interest in the volatility modeling, thus triggered the later focus on that research topic. In general, the most frequently used model to test for this phenomenon is the GARCH(1,1), which is expressed as

$$\sigma_t^2 = \alpha_0 + \alpha_1(\epsilon_{t-1})^2 + \beta_1(\sigma_{t-1})^2 \tag{1}$$

The volatility captured in 1 is derived from the returns on an asset, R_t, as indicated by equation 2, where $\epsilon_t = \sigma_t \zeta_t$ and ζ_t following standard normal distribution.

$$R_t = \mu + \epsilon_t \tag{2}$$

The reason why this model has been more commonly deployed in finance than continuous-time models is the simplicity in its estimation, given that financial data are usually discrete, not continuous. In its most general form, GARCH(p,q) is written as follows:

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^{q} \alpha_i(\epsilon_{t-i})^2 + \sum_{i=1}^{p} \beta_j(\sigma_{t-i})^2 \tag{3}$$

It is noteworthy to point out that, for β equal to 0, equation 3 collapses to an ARCH model. The addition of this second part in the GARCH model is significant because it captures the volatility dependence, or alternatively, the claim that the past does matter. However, a flaw of the standard GARCH model is that it does not account for the leverage effect, which is an empirical finding noticed in the real markets, where volatility is more responsive to negative shocks than to positive shocks of equal size.

3.1 Stationarity

In our attempt to find the parameters in equation 1, we need to ensure that the process generated has finite moments. Hence, a stationarity test needs to be conducted to check whether our time series fulfills this condition, and if not adjust it accordingly. In theory, we know that a GARCH(1,1), with α_0 greater than zero and α_1, β_1 greater or equal to zero, has a stationary solution with finite first moment if and only if $\alpha_1 + \beta_1 < 1$, and, in turn, $\mathbb{E}\left[\sigma_t^2\right] = \frac{\alpha_0}{1-\alpha_1-\beta_1}$, according to Williams (2011). However, with the use of Matlab this control is facilitated and one can implement an Augmented Dickey Fuller test, to check for stationarity in the process under consideration.

3.2 Autocorrelation

In time series analysis, there has been observed a certain set of stylized facts. In this context, one would anticipate to capture a small autocorrelation in the price variations, whereas a strong autocorrelation is expected to be found in the squared and absolute returns. The existence of autocorrelations in a dataset can be checked, both graphically and by implementing a test such as the Ljung-Box, the Breusch-Godfrey or the Durbin-Watson test.

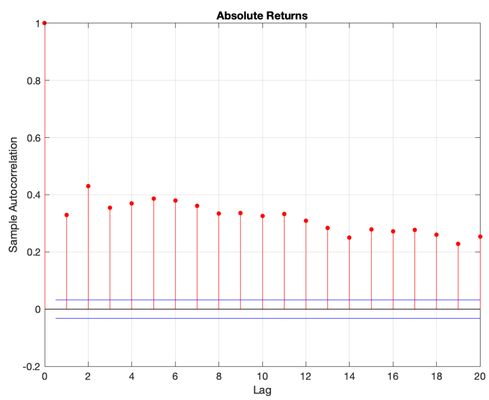

Figure 3: Absolute Returns Autocorrelations

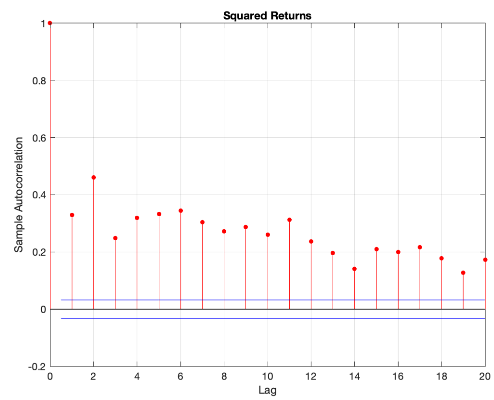

Figure 4: Squared Returns Autocorrelations

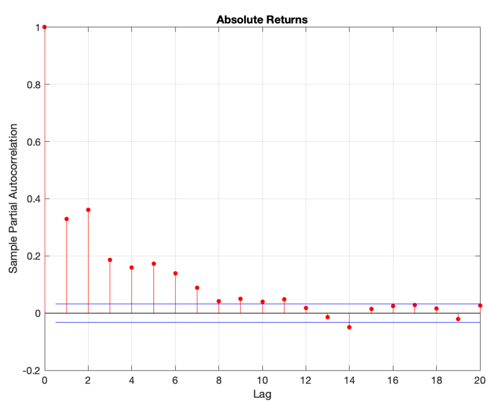

Figure 5: Absolute Returns Partial Autocorrelations

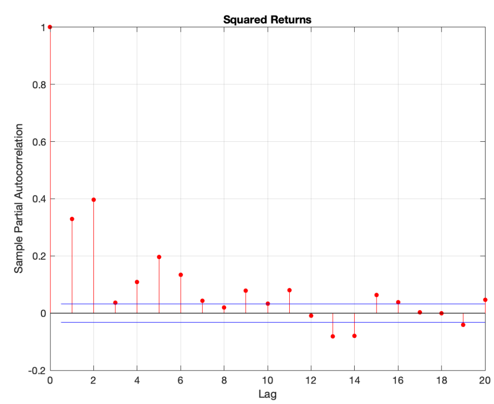

Figure 6: Squared Returns Partial Autocorrelations

In order to monitor the existence of autocorrelation we plot the autocorrelation and the partial autocorrelation functions for the S&P 500 time series, where we observe that both the absolute and the squared returns present significant autocorrelation for all the lags tested. Given that partial autocorrelation produces the unique correlation between two observations after partialling out the intervening correlations, we note that the importance of the partial autocorrelations diminish with lags. Those results indicate that further analysis of the residuals is required to obtain more accurate insights regarding their behavior. Nevertheless, the results underline the foundation of the class of GARCH models while the moving-average and autoregressive terms are verified by the (partial) autocorrelations.

4 Application: GARCH(1,1) for S&P 500 Returns

In order the assess to performance and accuracy of GARCH-models on index returns, this application focuses on the benchmark model GARCH(1,1) on daily returns of the S&P 500 index for a historical period of 15 years until December 2022. In a first step, the model, and in particular its parameters, associated standard errors as well as the filtered variance process, is estimated in-sample across the full data set using Maximum-Likelihood Estimation. In a second step, the estimated parameters and the filtered volatility are used to construct a rolling 95% confidence interval for 30-day ahead return predictions, which is then assessed by using the rolling out-of-sample 30-day ahead realizations. The full application is carried out in MATLAB, while core functions for GARCH estimations are built customly. Hence, any of the following algorithmic structures or explanations might be uniquely tailored to MATLAB syntax and functionalities.

4.1 GARCH(1,1) Estimation

The estimation of the GARCH(1,1) parameters, its standard errors and the resulting filtered variance process follows from applying the Maximum-Likelihood Estimation approach. Given the sample sequence of S&P 500 daily returns $r_1, ..., r_T$ with a vector of parameters θ to be estimated, under the assumption of univariate Gaussian distributed returns, the probability density function $f(r_t;\theta)$ is defined as follows:

$$f(r_t;\theta) = (2\pi\sigma^2)^{-0.5} * \exp(-\frac{1}{2\sigma^2} * (r_t - \mu)^2) \tag{4}$$

Given the univariate probability density function, the joint probability density function of the return series $f(x_1, ..., x_T;\theta)$ under the assumption of Gaussian distributed returns, and hence independence, is defined as follows:

$$f(r_1, ..., r_T;\theta) = f(r_i;\theta) * ... * f(r_T;\theta) = \prod_{i=1}^{T} f(r_i;\theta) \tag{5}$$

Following the definition of the joint probability density function, the Likelihood function can be set up:

$$L(\theta \mid \mathbf{r}) = \prod_{i=1}^{n} \left(2\pi\sigma_t^2\right)^{-1/2} \exp\left(-\frac{1}{2\sigma_t^2}\left(r_t - \mu\right)^2\right) \tag{6}$$

where θ implies the GARCH(1,1) parameters α_0, α_1 and β_1 needed to estimate the unknown variance σ_t^2. The Maximum Likelihood Estimation approach then allows to take and maximize the logarithm of equation 6:

$$LL(\theta \mid \mathbf{r}) = -\frac{N}{2}\log\left(2\pi\sigma^2\right) - \frac{1}{2\sigma^2}\sum_{i=1}^{N}(r_i - \mu)^2 \tag{7}$$

Since the GARCH(1,1) model assumes a zero-mean return process, the log-likelihood function 7 reduces as follows:

$$LL(\theta \mid \mathbf{r}) = -\frac{N}{2}\log\left(2\pi\sigma^2\right) - \frac{1}{2\sigma^2}\sum_{i=1}^{N} r_i^2 \tag{8}$$

As a last step, the log-likelihood function 8 needs to be maximized w.r.t. to the vector of unknown parameters $\theta = (\alpha_0, \alpha_1, \beta_1)'$:

$$\begin{aligned} \alpha_0^* &= \arg\max_{\alpha_0} LL(\theta \mid \mathbf{r}) \\ \alpha_1^* &= \arg\max_{\alpha_1} LL(\theta \mid \mathbf{r}) \\ \beta_1^* &= \arg\max_{\beta_1} LL(\theta \mid \mathbf{r}) \end{aligned} \tag{9}$$

Given the full setup of the maximization of the Log-Likelihood function to derive the optimal GARCH(1,1) parameters $\alpha_0, \alpha_1, \beta_1$, the following data-driven methodology is applied in in order to set up the objective function:

Algorithm 1 GARCH(1,1) - Objective Function

Require: $r_{N,1}, \alpha_0, \alpha_1, \beta_1$
  $N \leftarrow len(r_{N,1})$ ▷ number of observations
  $\hat{\sigma} \leftarrow []_{N,1}$ ▷ initialize vec of volatilities
  $\hat{\sigma}_1 \leftarrow \sqrt{\frac{1}{N}\sum r_i^2}$ ▷ initialize first volatilitiy
  $llh \leftarrow log(\frac{1}{\sqrt{2\pi}\hat{\sigma}_1} - \frac{r_1^2}{2\hat{\sigma}_1^2})$ ▷ initialize log-likelihood
  for i *in* $[2:N]$ **do**
    $\hat{\sigma}_i \leftarrow \alpha_0 + \alpha_1 r_{i-1}^2 + \alpha_1 \hat{\sigma}_{i-1}^2$ ▷ volatility by GARCH(1,1)
    $llh \leftarrow llh + log(\frac{1}{\sqrt{2\pi}\hat{\sigma}_i} - \frac{r_1^2}{2\hat{\sigma}_i^2})$ ▷ update log-likelihood
  end for
  return $-llh$ ▷ minimization to maximization

Given the above alogorithmic structure for the objective function, the volatility is estimated given a set of parameters $\alpha_0, \alpha_1, \beta_1$ by means of GARCH(1,1), which is then used to calculate the log-likelihood statistic iteratively for every day across the sample. Since the objective must be maximized in the case of Maximum-Likelihood Estimation and a minimization algorithm is used for optimization afterwards, the negative log-likelihood value is returned. Hence, the target is to find the optimal parameters which maximizes the above objective methodology.

The optimization problem of maximizing the log-likelihood statistic is carried out through the MATLAB function "fmincon", which finds minimum of constrained nonlinear multivariable function under the following generalized problem:

$$\min_x f(x) \;\; s.t. \begin{cases} c(x) \leq 0 \\ ceq(x) = 0 \\ A * x \leq b \\ Aeq * x = beq \\ lb \leq x \leq ub \end{cases} \tag{10}$$

In the case of the GARCH(1,1) application, the vector of parameters x to be optimized is defined as $\alpha_0, \alpha_1, \beta_1$, the objective function $f(x)$ is taken from the above introduction of the objective function and the constraints involve $A = (0, 1, 1)$, $b = 1$ and $lb = (0, 0, 0)$, while the other constraint parameters are not set. In comparison to chapter 3.1, these constraints ensure that $\alpha_1 + \beta_1 < 1$ and hence that the resulting GARCH process implies stationarity. In addition to the optimal parameters, their standard errors are computed as the square root of the diagonal matrix of the inverse of the associated Hessian. Lastly, the filtered variance process can be replicated by applying the GARCH(1,1) model with the optimal parameters across the sample by means of 1-day ahead predictions. Finally, the following results have been achieved:

	Value	Std Error
Const (α_0)	2.2097e-06	3.5263e-07
Alpha (α_1)	0.15174	0.012051
Beta (β_1)	0.83964	0.010921

Table 1: GARCH(1,1) - Estimated Parameters

	Custom Function	Native Function	Diff
Const (α_0)	2.2097e-06	2.2044e-06	5.3703e-09
Alpha (α_1)	0.15174	0.15183	-8.9401e-05
Beta (β_1)	0.83964	0.83967	-2.9165e-05

Table 2: GARCH(1,1) - Parameter Comparison

Referring to Table 1, one can observe that the constant parameter α_0 is close to zero, while the autoregressive and moving-average parameters α_1 and β_1 are estimated at 0.15 and 0.84, while being significantly larger than zero given the small standard errors. Since the GARCH(1,1) parameters are constructed through Maximum-Likelihood Estimation, while not relying on the built-in native MATLAB functions, the same estimation is carried out through the bult-in native function to check whether the estimation is correct and not significantly different. Given the difference in the third column in Table 2, one can conclude that the custom estimation is not significantly different and hence correctly applied. Based on these result, the following filtered variance process is calculated from the estimated parameters by means of GARCH(1,1):

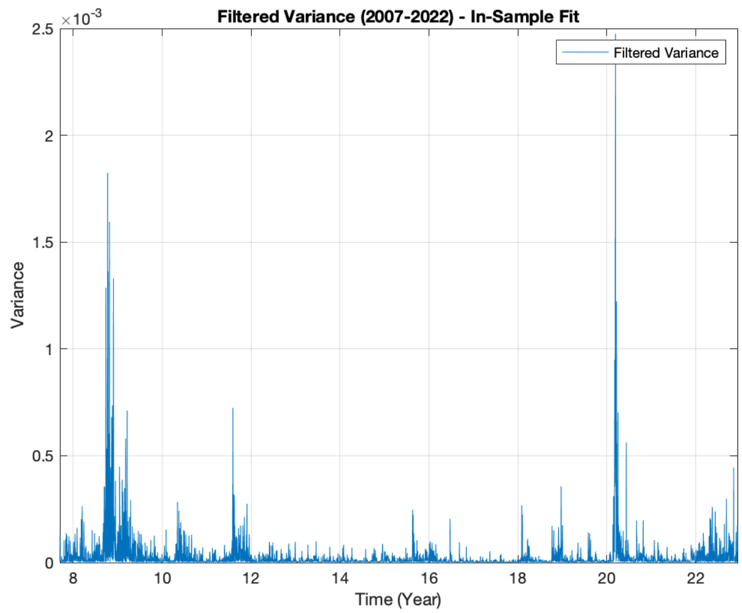

Figure 7: GARCH(1,1) - In-Sample Filtered Variance Process

Based on Figure 7 of the in-sample filtered variance process, one can observe that the highly volatile market period during the financials crisis in 2008 and the Covid-19 crisis in 2020 are well captured. Nevertheless, since the process is modeled as rolling one-day ahead predictions given the today's realizations, the good capture of the volatility regimes is not surprising and does not lead to any conclusions about the model accuracy out-of-sample.

4.2 GARCH(1,1) Prediction Confidence Intervals

In order to test the estimated parameters and hence the model accuracy on a longer prediction horizon, the 95% confidence intervals for 30-day ahead daily returns are predicted. Hence, a perfectly fitted model should deliver violations by the realized 30-day ahead returns in the sample of 5%, whereby a lower (higher) violation rate is considered as over- (under-) estimation of the volatility. In order to calculate the 30-day ahead 95% confidence intervals in a data-driven way, under assumption of Gaussian distributed returns and priorly introduced GARCH(1,1) mechanics, the following methodology is applied:

Algorithm 2 GARCH(1,1) - 30-Day ahead 95% Confidence Intervals

Require: $\hat{\sigma}_{N-1,1}, r_{N,1}$
  $N \leftarrow length(r)$ ▷ number of observations
  $N_{pred} \leftarrow 30$ ▷ length of prediction period
  $\sigma_{pred,30d} \leftarrow []_{N-N_{pred},1}$ ▷ initialize vector of 30d ahead volatilities
  $ci_{pred,30d} \leftarrow []_{N-N_{pred},1}$ ▷ initialize matrix of confidence bounds
  $\epsilon \leftarrow N(0,1)_{N-N_{pred},N_{pred}}$ ▷ random Gaussian realizations
  for i *in* $[2 : N - n_{pred}]$ **do**
    $\sigma_{pred,1d} \leftarrow []_{N_{pred},1}$ ▷ initialize vector of 1d ahead volatilities
    $r_{pred,1d} \leftarrow []_{N_{pred},1}$ ▷ initialize vector of 1d ahead return predictions
    $\sigma_{pred,1d,1} \leftarrow \hat{\sigma}_i$ ▷ initialize first volatility element
    $r_{pred,1d,1} \leftarrow r_i$ ▷ initialize first return element
    for j *in* $[2 : N_{pred}]$ **do**
      $\sigma_{pred,1d,j} \leftarrow \sqrt{\alpha_0 + \alpha_1 * r^2_{pred,1d,j-1} + \beta_1 * \sigma^2_{pred,1d,j-1}}$ ▷ 1d ahead volatility prediction by GARCH(1,1)
      $r_{pred,1d,j} \leftarrow \sigma_{pred,1d,j} * \epsilon_{i,j}$ ▷ 1d ahead return prediction by GARCH(1,1)
    end for
    $\sigma_{pred,30d,i} \leftarrow \sigma_{pred,1d,n_pred}$ ▷ append 30d ahead volatility
    $ci_{pred,30d,i,1} \leftarrow \frac{1}{N} * \sum_{g=1}^{N_{pred}} r_{pred,1d,g} - 1.96 * \sigma_{pred,30d,i}$ ▷ append 30d ahead 95% lower bound
    $ci_{pred,30d,i,2} \leftarrow \frac{1}{N} * \sum_{g=1}^{N_{pred}} r_{pred,1d,g} + 1.96 * \sigma_{pred,30d,i}$ ▷ append 30d ahead 95% upper bound
  end for
  return ci ▷ vector of rolling 30d ahead confidence interval

After the data-driven construction of the 30-day ahead 95% confidence interval according to Algorithm 2, the accuracy of the lower and upper confidence bounds are tested using the 30-day ahead return realizations from the original sample. This methodology is equivalent to constructing a 30-day ahead confidence interval for the return in 30 days, while testing the interval 30 days later as soon as the 30-day ahead return is realized.

According to Figure 8, the upper and lower bound of the 95% confidence interval for 30-day ahead returns seem to capture the realizations well. In addition, it can be seen that violations of the bounds are especially low in low volatility regimes between 2012 and 2016 as well as 2017 and 2020. When it comes to extreme volatility due to black swan events, the GARCH(1,1) is not able to capture the first market impact as well as the highly volatility days after based on volatility clusters, which makes sense by nature of the GARCH model since it only relies on moving-average and autoregressive parts, and hence the past. In order to get a closer insight on the number and the ratio of violations, the following statistics can be reported:

	Statistic
Number of Violations	555
Ratio of Violations	14.57%

Table 3: GARCH(1,1) - 95% Confidence Interval Violations

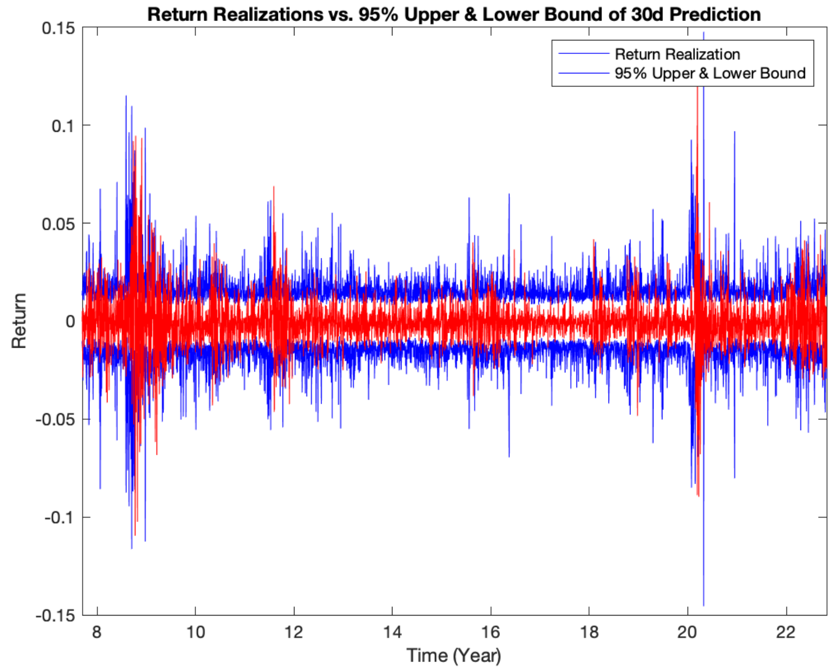

Figure 8: GARCH(1,1) - 30d Confidence Interval Violations

According to Table 3, the 30-day ahead sample realizations exceed the predicted 95% confidence interval 555 times, which relates to a sample probability of 14.57%. By definition, a perfect prediction model for a 95% would lead to 5% violations across the testing period. Hence, the model underestimates the confidence interval across the test set, which is mostly rooted in high volatile market regimes in 2008, 2012, 2016 and 2020. Based on the model assumptions and definition, the stylized facts of returns and the sample, following reasons for the underestimation can be identified:

- **Gaussian Distribution**: The standard GARCH model assumes the return to be Gaussian estimated while being scaled by the volatility component. As the stylized facts have shown, return distributions are more fat tailed, hence the Gaussian assumption fails to capture the fat tails which leads to a lower probability of high positive or negative returns and hence to a underestimation of volatility. Furthermore, stocks seem to deliver a long-term positive average return, which for case to case might violate the zero-mean assumption. Since the Gaussian assumption is applied to the confidence interval construction as well through the 95% quantile, it leads to a underestimation of the upper and lower bounds compared to the sample.

- **30-Day Prediction Horizon**: The GARCH(1,1) model assumes the volatility to have an autoregressive and an moving-average part of order one. Hence, the prediction of 30-day ahead volatilities implies that 30-day estimates are based on 29 rolling estimates, which leads to a biased estimation.

- **GARCH Orders**: According to the (partial) autocorrelation plot of squared returns, the sample includes significant autoregressive lags larger than one. Hence, the specification of the GARCH(1,1) might be insufficient, even though the empirics show that the benchmark GARCH(1,1) leads to robust and most accurate estimations across many data sets and time horizons.

References

Bollerslev, T. (1986, April). Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, *31*(3), 307-327. Retrieved from https://ideas.repec.org/a/eee/econom/v31y1986i3p307-327.html

Engle, R. F. (1982). Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. *Econometrica*, *50*(4), 987–1007. Retrieved 2022-12-09, from http://www.jstor.org/stable/1912773

Mandelbrot, B. (1963). The variation of certain speculative prices. *The Journal of Business*, *36*. Retrieved from https://EconPapers.repec.org/RePEc:ucp:jnlbus:v:36:y:1963:p:394

Williams, B. (2011). *Garch(1,1)-models.* Retrieved from https://math.berkeley.edu/~btw/thesis4.pdf

Author’s Declaration

We hereby certify that:

– We have written the program ourselves except for clearly marked pieces of code

– We have tested the program and it ran without crashing

Charalampos Elenidis, Niklas Kampe