UPPSALA
UNIVERSITET

Uppsala University

A closer look at S&P 500 Index using GARCH

Analysis and forecast

Reza Dadfar
5-12-2022

Contents

Introduction

Understanding and analysis of the underlying mechanism of stock price was always one of the major objectives among traders and hedge-funds companies. The non-constant volatility of the stock returns demands for special models to predict the stock price characteristics. In this report, we analyse the underlying price evolution of the S&P 500 index and forecast the volatility and the return. In particular, we discuss the performance of a special model called Generalized AutoRegressive Conditional Heteroskedasticity (GARCH).

Price evolution, the first look

The S&P500 ticker symbol (GSCP) for a period between 2015-01 and 2020-06 is depicted in Figure 1. We can observe a trend on the top of stochastic behaviour of the price.

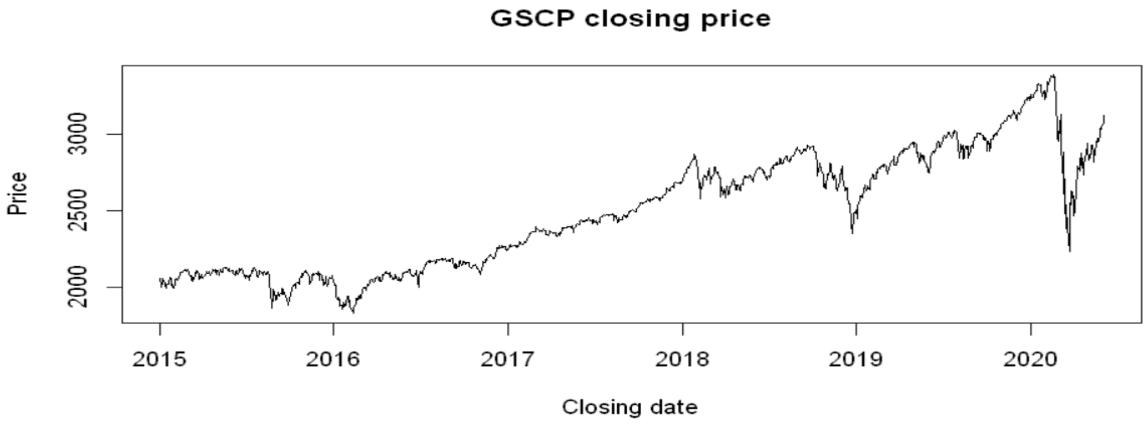


Figure 1 –S&P 500 price from 2015 to 2020 (i.e. GSCP is the ticker symbol for the S&P 500)

The auto autocorrelation function (ACF) and partial autocorrelation function (PACF) of the price are shown in Figure 2. The ACF reduces gradually and the PACF indicates an AR(1) or AR(2) underlying process. However, we observed that the process is not stationary since the mean is not constant. To make the price stationary, we can remove the trend from the price using a model. However, we can instead calculate the index log-return $r_t = \ln P_t - \ln P_{t-1}$, where P_t is the price of the index at time t and model the log-return to have a stationary time series. The log-return is shown in Figure 3a from 2015-01 to 2020-06. From the plot, we observe that the trend is almost disappeared. However, the volatility is not constant with time. The volatility bursts are shown in Figure 3a with red circles. This violates the assumption of ARMA models where they are used to estimate the conditional mean of a process when the conditional variance is constant. To see the volatility bursts more clearly, we report the log-return squared in Figure 3b. The same burst in volatility is clearly noticeable. As a remedy to capture this dynamic, we model the log-return using a GARCH model.

$$r_t = \mu + \sigma_t \epsilon_t$$

$$\sigma_t^2 = \omega + \sum_{j=1}^{p} \alpha_j r_{t-j}^2 + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^2$$

here $\epsilon_t = N(0,1)$ and μ can be a constant or an ARMA model.

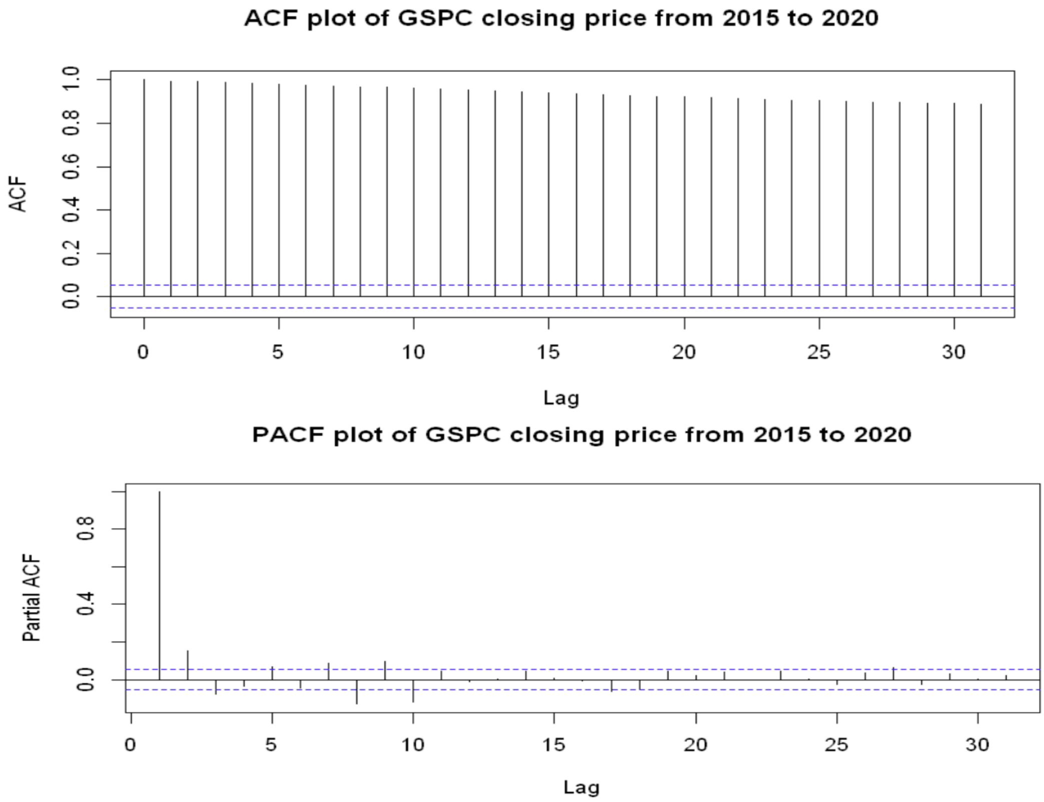
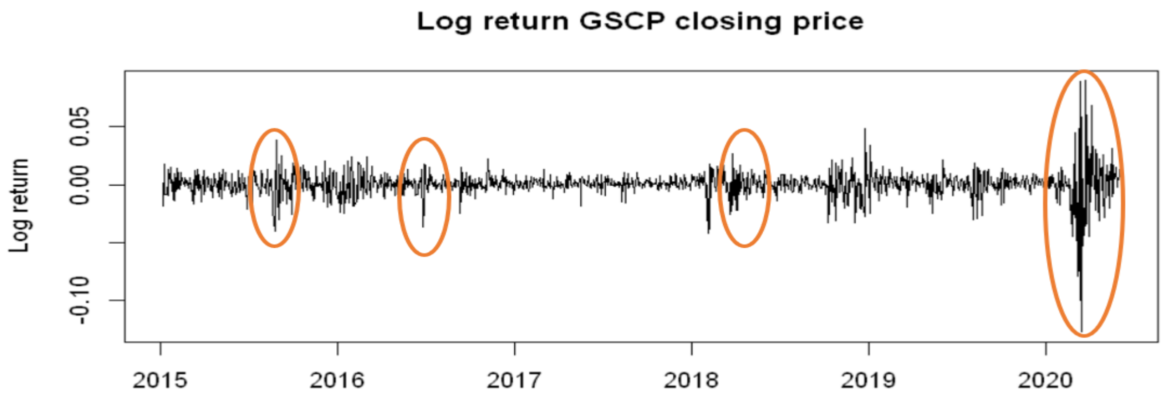


Figure 2 ACF and PACF of the original price Index

(a)



(b)

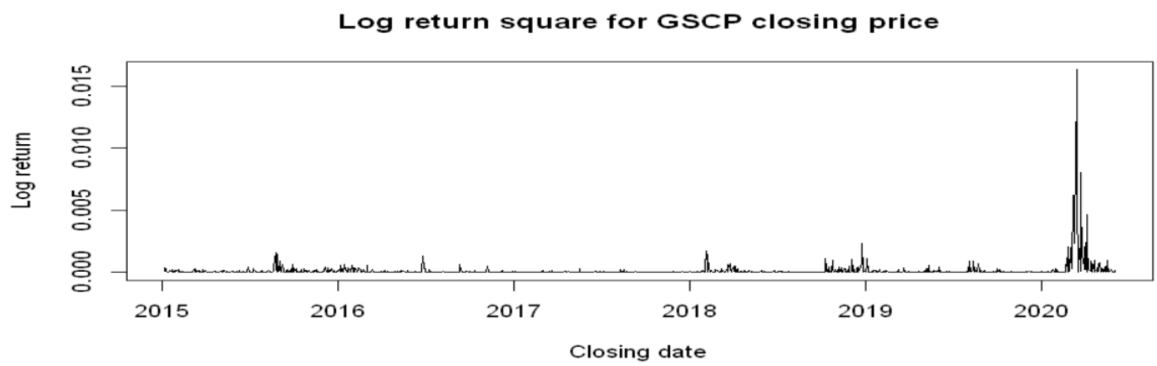


Figure 3 Log return and log return square for S&P 500 from 2015 to 2020

To see if this model is appropriate for our time series, we report the ACF and PACF of the log-return and log-return square in Figure 4.

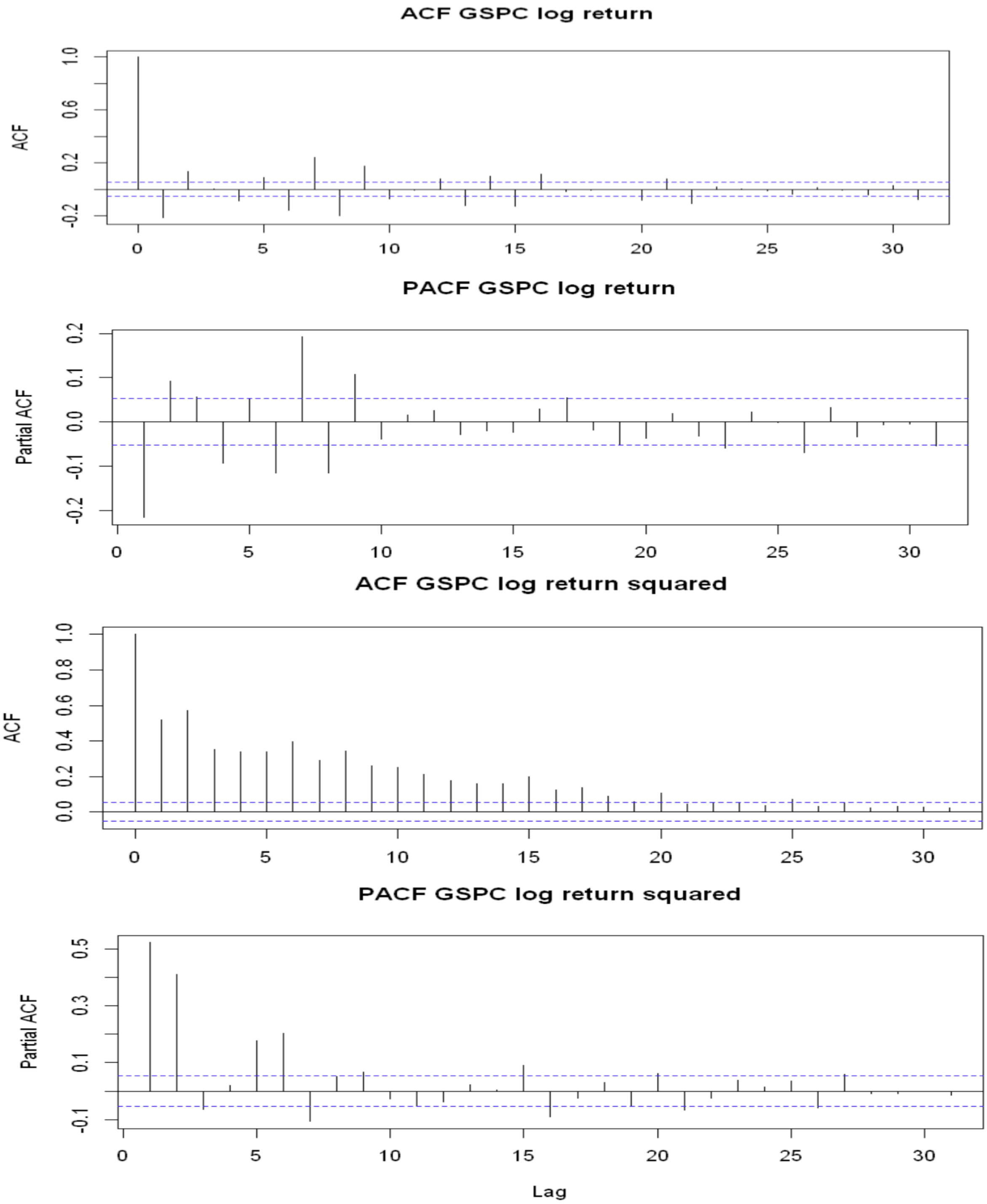


Figure 4 ACF and PACF of log return and log return square for S&P 500

We expect to see a white noise kind of behaviour for log-return and a significant autocorrelation for log-return squared. From Figure 4, it is observed that the “log-return squared” has a much stronger auto correlation compared to “log-return”. However, the white noise assumption of log-return is questionable! To investigate and analyse the autocorrelation of the time series, we perform Box-Ljung test in Table 1. In this test the null hypothesis is that whether the signal is uncorrelated. We test both log-return and log-return squared for different lags. Although the p-value for all the tests is small, indicating that we can reject the null hypothesis, the rejection is much stronger in the case of log-return squared. Hence, we see the existence of a stronger correlation for log-return squared.

Table 1 Box Ljung test on return and return squared using different lags.

Box-Ljung test		
lag	return	Return**2
1	X-squared = 64.131 p-value = 1.11e-15	X-squared = 371.64 p-value < 2.2e-16
2	X-squared = 89.191 p-value < 2.2e-16	X-squared = 817.53 p-value < 2.2e-16
3	X-squared = 89.24 p-value < 2.2e-16	X-squared = 986.19 p-value < 2.2e-16
6	X-squared = 144.33 p-value < 2.2e-16	X-squared = 1519.9 p-value < 2.2e-16

We use the "fGarch" library in R software to fit different Garch and ARMA models on the data. The results are shown in Table 2. The best model corresponds to ARMA(1,0) for the mean and GARCH(1,1) for the residuals with t error distribution. Please note that if we increase Garch(1,1) to Garch(2,1) the extra parameter, α_2 becomes insignificant. On the other hand, we can in principle decrease the Garch order since the coefficient of ar1 in ARMA(1,0)-Garch(1,1) is close to zero ($Ar_1 = -7.53 * 10^{-2}$), hence, we could choose the lighter model "ARMA(0,0)-Garch(1,1)-t-distribution" without loss of accuracy. Nevertheless, we proceed with ARMA(1,0)-Garch(1,1) in this report.

Table 2 - Using different ARMA model for the mean and Garch model for the residual

Model	Error distribution	Log-Likelihood	AIC	Coefficient
ARMA(0,0), Garch(1,0)	normal	4458.822	-6.538257	significant
ARMA (0,0), Garch(1,1)	normal	4667.197	-6.842549	significant
ARMA (0,0), Garch(1,1)	t-distribution	4724.502	-6.925167	significant
ARMA (1,0), Garch(1,1)	t-distribution	4729.616	-6.931205	significant
ARMA (1,0), Garch(2,1)	t-distribution	4729.849	-6.930080	α_2 not significant

The detail summary of the model is shown in Figure 5. We can see that all the parameters are significant at 5%. If we look at the residual statistics, we see that the residuals are not normal as shown in the JB or SW test results.

The Q-Q plot for residuals is shown in Figure 6. Although, it is not clearly normal, it is close enough, hence we accept the model, but we keep in mind that the model can be improved further.

The log-return price and the estimated volatility using all the data (from 2015-01 to 2020-06) is shown in Figure 7. From this result, we believe that the model can predict the volatility quite good.

```
Title:
 GARCH Modelling

Call:
 garchFit(formula = ~arma(1, 0) + garch(1, 1), data = y, cond.dist = "std")

Mean and Variance Equation:
 data ~ arma(1, 0) + garch(1, 1)
<environment: 0x0000000009789380>
 [data = y]

Conditional Distribution:
 std

Coefficient(s):
         mu          ar1        omega       alpha1        beta1        shape
 8.6399e-04  -7.5632e-02   2.2915e-06   2.1347e-01   7.8509e-01   4.8348e+00

Std. Errors:
 based on Hessian

Error Analysis:
         Estimate  Std. Error  t value Pr(>|t|)
mu      8.640e-04   1.544e-04    5.598 2.17e-08 ***
ar1    -7.563e-02   2.817e-02   -2.685 0.007264 **
omega   2.291e-06   6.837e-07    3.352 0.000804 ***
alpha1  2.135e-01   3.632e-02    5.877 4.17e-09 ***
beta1   7.851e-01   3.008e-02   26.104  < 2e-16 ***
shape   4.835e+00   6.520e-01    7.415 1.22e-13 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

 Log Likelihood:
  4729.616    normalized:  3.470005

 Description:
  Thu May 12 15:17:52 2022 by user: rdadf


 Standardised Residuals Tests:
                                Statistic p-Value
  Jarque-Bera Test   R    Chi^2  859.6962  0
  Shapiro-Wilk Test  R    W      0.9575732 0
  Ljung-Box Test     R    Q(10)  7.14788   0.7114143
  Ljung-Box Test     R    Q(15)  12.67103  0.6276906
  Ljung-Box Test     R    Q(20)  20.72693  0.413359
  Ljung-Box Test     R^2  Q(10)  12.13954  0.2758263
  Ljung-Box Test     R^2  Q(15)  14.26014  0.5058978
  Ljung-Box Test     R^2  Q(20)  16.87009  0.6613924
  LM Arch Test       R    TR^2   13.39989  0.3406569

 Information Criterion Statistics:
      AIC       BIC       SIC      HQIC
 -6.931205 -6.908238 -6.931244 -6.922608
```

Figure 5 The results of AR(1,0) Garch(1,1) t distribution

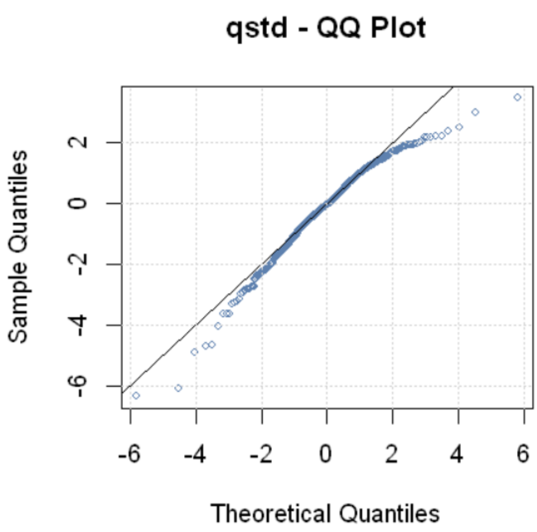


Figure 6 Q-Q plot of the residuals

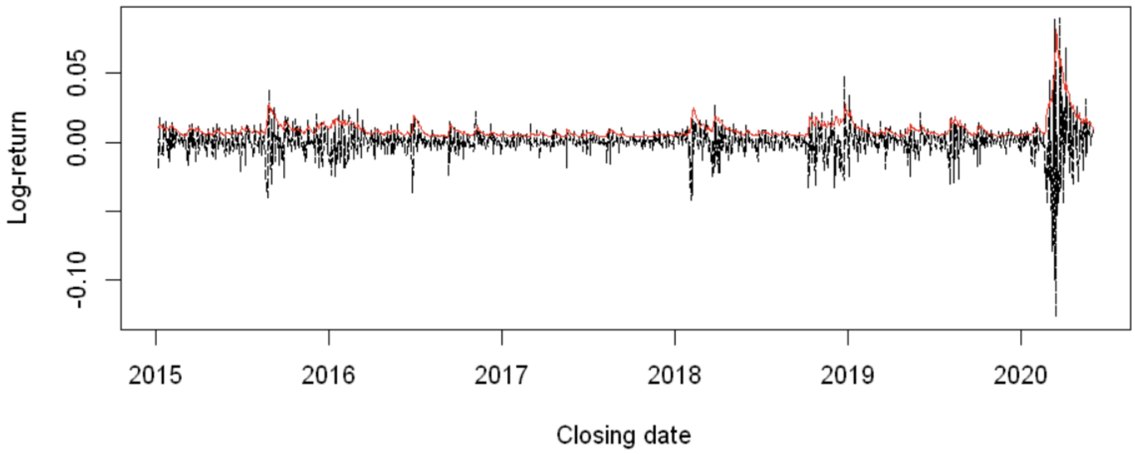


Figure 7 - The log return and the estimated volatility are shown on one plot

Forecasting

To perform the forecasting, we need to change the R library and use “rugarch” instead of “fGarch”. Hence the estimated parameters are a bit different. The estimated parameters for the two libraries are shown in Table 3. One reason for the difference is that using “rugarch” library, we train the model only on half of the data and estimate the parameters. In the next step we test the model on the other half of the original data (test data).

Table 3 different results in the parameter estimation of ARMA-Garch using two different library in R

ARMA(1,0) Garch(1,1) using fGarch lib		ARMA(1,0) Garch(1,1) using rugarch	
mu	0.000863993757579402	**mu**	0.00050655589863897
ar1	-0.0756324189675663	**ar1**	-0.0958356677524878
omega	2.29153212740106e-06	**omega**	2.63723600351457e-06
alpha1	0.213471011452821	**alpha1**	0.198003285224109
beta1	0.785089421731975	**beta1**	0.784570915394028
shape	4.83484597016329	**shape**	4.45493846931359

The estimated volatility and log-return time series for S&P500 are shown in Figure 8. The volatility estimation is surprisingly good. The black line shows the log-return data. The blue line is the fitted value while the red lines are the estimated values. Although both the estimated and forecasted volatility look very good, the log return forecast does not seem to capture the amplitude of the original signals. In the future, more investigation is needed to fix this discrepancy.

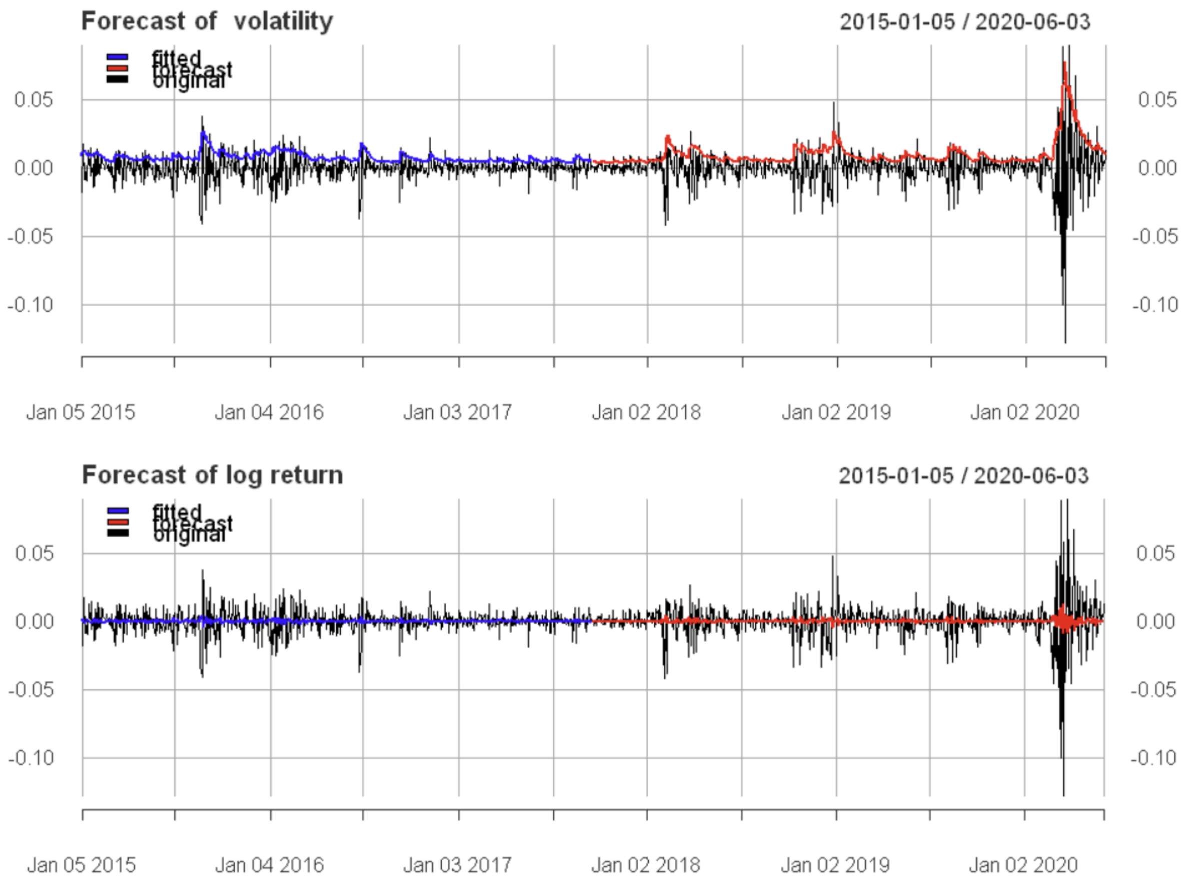


Figure 8 Forecast of the volatility and log return using ARMA(1,0)-Garch(1,1) with rugarch library in R

Conclusion

The main goal of this report was to find an appropriate model to capture the dynamics of the financial index price, S&P500. To this end a Garch model is used. The model is tuned, and the results are presented. The estimated model could capture the volatility of the time series surprisingly good however, the amplitude of the log return was not predicted as expected. More investigation should be done to fix this issue in the future.

Appendix