Volatility Prediction by E-GARCH Model

Takato Watanabe

2019/6/10

Contents

1 Introduction

Virtual and digital crypto-currencies, now termed more frequently as crypto-assets, have become a new paradigm of the digital world. In particular, Bitcoin, the first cryptocurrency, has reached the 10th year since its inception. We will focus specifically on other multi-cryptocurrecies which trades specific currency.

One interesting fact that separates crypto-currencies from stocks is that they do not have intrinsic value. According to Arthur Iinuma (2018), Crypto-currencies do not sell a product, earn revenue or employ thousands of people like what companies do. These differences give people to have least information to know whether it is overbought or oversold, or even if it is a good value or overpriced or not. Those issues make crypt-currency to have a higher volatility, which means higher risk but can have higher return.

Volatility of a price process is a fundamental parameter to many practical problems on financial markets ranging from hedging to pricing assets and consequently also pricing of derivatives of those assets. It has occupied the field of financial econometrics for several decades and various results are accomplished. The current volatility literature is mostly concentrated on realized volatility model, the model that estimates intraday return by using daily volatility.

Historically, volatility of a price processes was a vivid field of exploration. Various parametric methods which represents volatility as a major variable for their estimation. For an example, (Generalized) autoregressive conditional heteroskedasticity (GARCH) models as in Eagle (1982) and further developed by Bollerslev (1986), estimates volatility by studying volatilities implied by options prices in conjunction with specific option pricing models such as Black Scholes (1973), or by studying direct indicators of volatility such as ex-post squared or absolute returns. But all those approaches have distinct weaknesses that may suggest misspecification since there are several competing parametric volatility models with different properties. As a result, one of the models could be correct and surely none of the models be strictly correct.

Volatility prediction, has high difficulty to have higher predicted rate since it is difficult to predict the sudden move on data, which are called as white noise in data-science. Eventhough those prediction models are difficult to identify whether they are surely correct or not, it is still great methods to do volatility prediction. As for the research proposal, we will use GARCH model to predict volatility of stock price datas.

2 Executive Summary

This time, we will use time series anaylsis for stock price data, which to predict the future value of stock price. However, to predict the stock price data is really difficult so in here, we will predict the volatility of stock price, which is squared variable of variance.By checking volatility, we can know how much difference will happen in the future.

The equation first to be used is one model of the times series analysis, which is called ARIMA model. AutoRegressive Integrated Moving Average Model, ARIMA model is the basic model that are mostly used for volatility predicting in time series analysis. ARIMA model can be shown as the following.

$$x_t = \sum_{i=1}^{p} \alpha_i x_{t-i} + e_t + \sum_{i=1}^{q} \beta_i \epsilon_{t-i} + \epsilon_i \tag{1}$$

if this model sets to follow to independent and identically distributed (i.i.d), Residuals of stock price data should be flatterlized since previous result doesn’t influence your current result and probability of residuals going higher or lower should be identical. However, in the real datas, the residuals shown by fitting to the model mostly have a sudden noises and they are not homogeneous. Those sudden noises dreadfully affects to the result since ARIMA model is weak if confusion goes on for long period and residuals does not become i.i.d. The reason is the output of ARIMA model does not considers about Time Series Structure of the data. Thus the model that added the time series structure to its result, is the GARCH model that we will use for this research.GARCH model can be shown as the following.

$$(Variance)\sigma_i^2 = \omega + \sum_{i=1}^{p} \beta_i \sigma_{t-i}^2 + \sum_{j=1}^{q} \alpha_j \epsilon_{t-j}^2 \tag{2}$$

GARCH model is constructed with Variance of time series and squared errors od time series and omega, which is constant term. It depends on with price change. We make some limitations for example omega to be more than 0 in order to prevent sigma squared for not to be less than 0. It might be natural that it is inconvenient when sigma squared is lesser than 0 since it does not satisfies normal distribution which is N(0,1) In order to avoid this, there is a improved example called EGARCH model.EGARCH model can be shown as the following.

$$lm(\sigma_t^2) = \omega + \sum_{i=1}^{p} \beta_i lm(\sigma_{t-i}^2) + \sum_{j=1}^{q} \alpha_j [\theta z_{t-j} + \gamma(|z_{t-j}| - E(|z_{t-j}|)] \tag{3}$$

GARCH model is superior for a volatility change model, but cannot catch the specialty in the stock market. For an example, the volatility of the stock price goes higher on the next day of the stock price gets down than the stock price gets higher. This kind of characteristics in stock price market is difficult to catch. However, this EGARCH comes to be able to catch it. Since left side of E-GARCH model is exponential, even if right side is negative, we can continue calculation.

3 Goals for Project

For our project, we will use Nikkei 225 data taken from Yahoo Finance. Nikkei 225 is the more commonly called the Nikkei, the Nikkei index, or the Nikkei Stock Average, is a stock market index for the Tokyo Stock Exchange (TSE).It has been calculated daily by the Nihon Keizai Shinbun (The Nikkei) newspaper since 1950. For our research, we will use the data from 2018/6/11 to 2019/6/10, removed datas that are not collected, which is in total 256.

There are four steps in this research. First we will check the suitable variables to be used for ARIMA model. Variables that to be input is mostly choosed manually but there are functions to find out the suitable variable to be used. We will check the most suitable variable for ARIMA model, since we will use ARIMA model for GARCH fitting. Second, we will check the suitable variable for GARCH model. In our research, we will use EGARCH model but we will check suitable variable for GARCH model since there are no functions to find out suitable variable for EGARCH model. Third, we will check infoCriteria for GARCH and EGARCH. by using infocriteria function, we can check which model is suitable for prediction. in case for our research, we will use EGARCH model since it recorded higher value than GARCH model. Fourth, we will finally predict volatility by using predicting function. this may take few mninutes but the time to take will be different by two points, number of data and variables to be input for each ARIMA and GARCH.

The goal of this research is to have predicted volatility to be well forecast the move on LogReturn of Nikkei 225. If LogReturn of Nikkei225 is not stationary, it means there will be high volatility rate since the volatility shows how risky the stock price is.

4 Methods and Analysis

First, we will load packages that is needed. I implemented the function that will installs packages that are not installed so that we do not have to type install.packages() manually.

For this project, we will use 5 packages. One is rugarch package. rugarch package is the package includes various GARCH models, for example EGARCH and also some variety of ARIMA functions. xts package is a powerful package that provides an extensible time series class, enabling uniform handling of many R time series classes by extending zoo. In order to fit datas in models used from rugarch, we will use datas that are changed to xts object. forecast package includes methods and tools to analyse time series data. We will use this package to find out the most suitable value to be used for ARIMA and GARCH model.fGarch package is mostly specialized for GARCH model. For this project, we will use this package to find out the suitable variable that to be used for GARCH model. dplyr function is used only to use mutate function.

The data we will use for our analysis is Nikkei 225 from Yahoo Finance. as.xts function makes csv files to xts object. we will use read.zoo function inside of xts since without using this function, errors will occur.

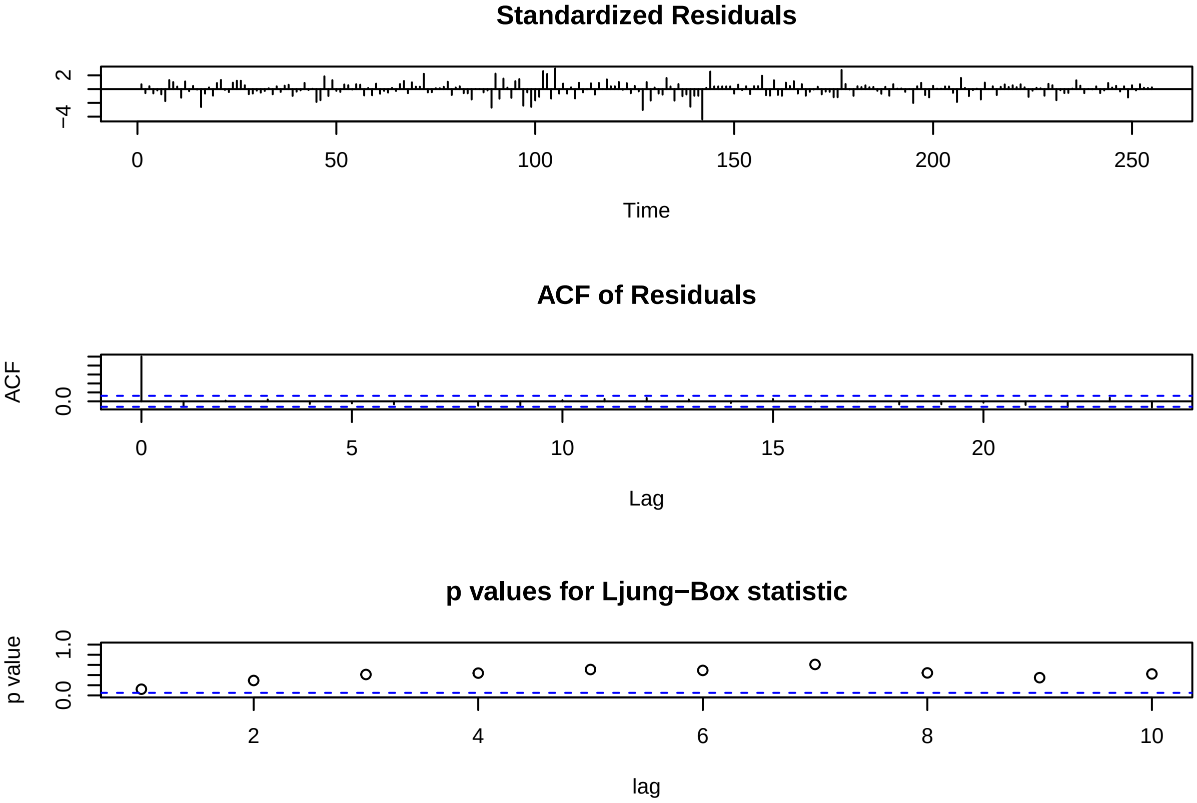
```
##                Open     High      Low    Close Adj.Close Volume
## 2018-06-11 22686.95 22856.08 22667.30 22804.04  22804.04  55700
## 2018-06-12 22977.22 23011.57 22797.73 22878.35  22878.35  63900
## 2018-06-13 22896.17 22993.26 22895.31 22966.38  22966.38  57000
## 2018-06-14 22842.96 22898.39 22738.61 22738.61  22738.61  72100
## 2018-06-15 22883.24 22885.84 22770.38 22851.75  22851.75  88500
## 2018-06-18 22806.57 22806.89 22601.13 22680.33  22680.33  65000
##               LogReturn
## 2018-06-11  0.002235677
## 2018-06-12 -0.001872804
## 2018-06-13  0.001329723
## 2018-06-14 -0.001988496
## 2018-06-15 -0.000598056
## 2018-06-18 -0.002410610
```

By looking the head datas of nikkei 225, we can see some classes. For our project, we will use LogReturn, which are variables that I have added to original csv file taken from Yahoo Finance. The reason why we use LogReturn is simple. We can understand growth and the volatility of the stock by analyzing a return. The average of the one-day LogReturn becomes the standard of the growth, and the standard deviation becomes the standard of the volatility. These datas are important in financial analysis since we can find stocks according with your investment strategy from the performance of a past return and predict trends of stock price.

After we clean up the data, we will fit those datas in auto.arima function. auto.arima function is included in forecast package and forecasts the most suitable variable to be used for ARIMA model with datas you have chosen.

```
##
```

```
##  ARIMA(2,0,2) with non-zero mean : Inf
##  ARIMA(0,0,0) with non-zero mean : -2202.699
##  ARIMA(1,0,0) with non-zero mean : -2203.076
##  ARIMA(0,0,1) with non-zero mean : -2203.012
##  ARIMA(0,0,0) with zero mean     : -2204.404
##  ARIMA(1,0,1) with non-zero mean : -2201.08
##
##  Best model: ARIMA(0,0,0) with zero mean
```



By running the function, we got the result that ARIMA(0,0,0). by running tsdiag function from forecast function, we can see the plot that includes three graphs. Top graph shows the standard residuals of the data fitted to ARIMA model. We can check whether a model is a “good” model to some extent by checking whether a residual error (predicted value - acftual value). ACF is Auto Correlation Function. if there is no auto correlation in datas, it means that the model does not fit well. The smaller the amount of auto correlation shows that the model fits well. for our anlysis, the value of ACF gets lower than the blue line, which is autocorrelation of original data after first prediction. Thus, we may think that the model fits quite well.

Ljung box test is the test that a residual error really becomes “the error (white noise)”. If round dots do not seem to be less than a blue line in a plot. we can consider that the dots are white noise.

Next, we will fit datas to GARCH model. First, we will find out the most suitable value for GARCH. In order to do this, we will use garchFit function from fGarch packages. One thing to consider is that we should input numbers manually to check. for example, “garchfit(˜ garch(1,1),data=sp,trace=FALSE” allows you to check how the model fits well with using only variables (1,1). in order to check every steps from (0$_{5,0}$5) we will create a for loop function.

After we check all result, we will find out the most suitable values from the result. by Reducing results that does not meets condition, we can find out the most suitable values to be used.

```
##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~garch(3, 0), data = sp, trace = FALSE)
##
## Mean and Variance Equation:
##  data ~ garch(3, 0)
## <environment: 0x7f887844f9d8>
##  [data = sp]
##
## Conditional Distribution:
##  norm
##
## Coefficient(s):
##          mu        omega       alpha1       alpha2       alpha3
## -1.2718e-05   6.6027e-06   6.7409e-02   1.0691e-01   1.6063e-01
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##          Estimate  Std. Error  t value Pr(>|t|)
## mu     -1.272e-05   1.845e-04   -0.069   0.9450
## omega   6.603e-06   9.463e-07    6.978    3e-12 ***
## alpha1  6.741e-02   6.589e-02    1.023   0.3063
## alpha2  1.069e-01   7.500e-02    1.425   0.1541
## alpha3  1.606e-01   8.296e-02    1.936   0.0528 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  1113.252    normalized:  4.365694
##
## Description:
##  Sun Jun 16 13:15:22 2019 by user:
```

From the result, we can know that garch(3,0) is the most suitable value to be used.

Next, we will check which models to be used to predict volatility. In case for our project, we will check SGARCH, which is mostly known as basic GARCH model, and EGARCH. We will consider which to be used by seeing InfoCriteria of each fit. We will use infocriteria function to do this.

```
##    aic.sg    aic.eg
## -8.692053 -8.726817
```

From the result, we can know that infoCriteria for EGARCH seems to be better since the lower the result, it is better.

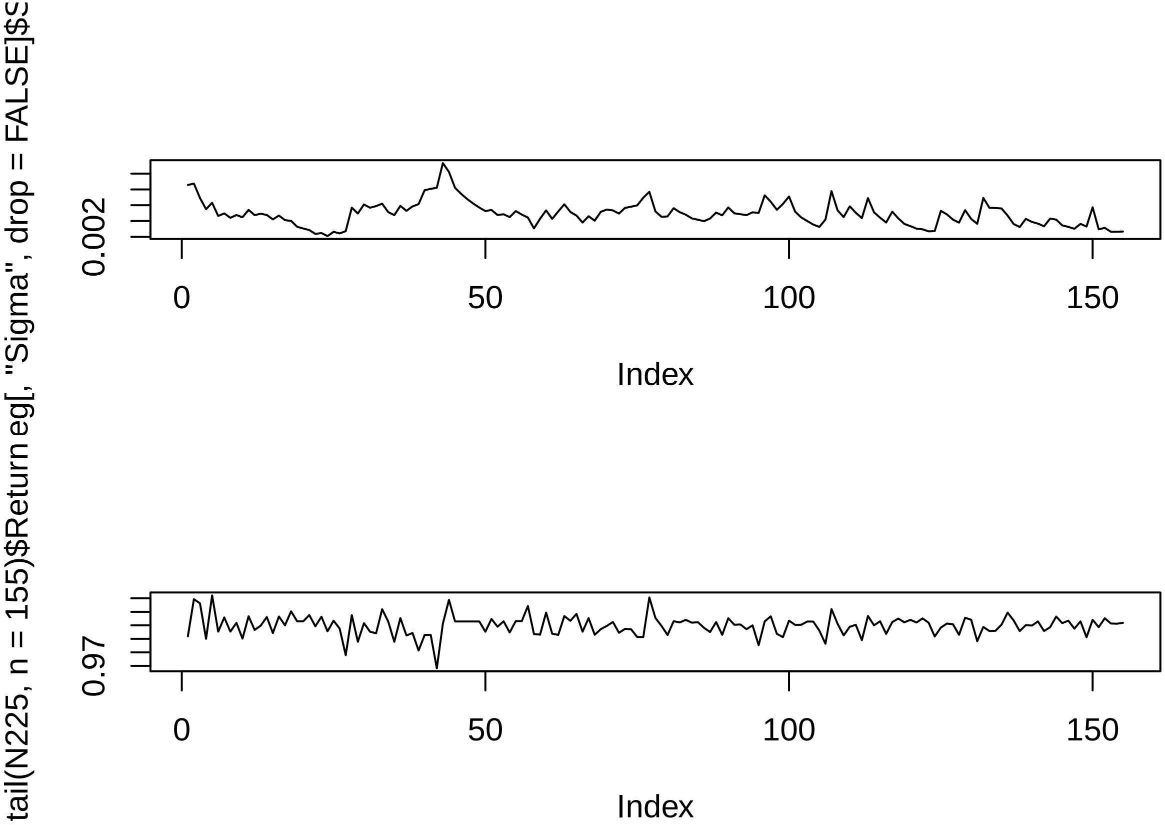
Finally, we will predict the volatility. We will use ugarchroll function in here. ugarchroll function allows you to calculate predicted volatility. In case for our project, we will predict volatility 1 day ahead and calculate again and again.

```
##
## *------------------------------------*
## *              GARCH Roll            *
## *------------------------------------*
## No.Refits        : 4
## Refit Horizon    : 50
## No.Forecasts : 155
## GARCH Model      : eGARCH(1,1)
## Distribution : norm
##
## Forecast Density:
##                Mu  Sigma Skew Shape Shape(GIG) Realized
## 2018-10-29 -3e-04 0.0053    0     0          0  -0.0036
## 2018-10-30 -3e-04 0.0054    0     0          0   0.0083
## 2018-10-31 -3e-04 0.0045    0     0          0   0.0070
## 2018-11-01 -3e-04 0.0037    0     0          0  -0.0044
## 2018-11-02 -3e-04 0.0042    0     0          0   0.0095
## 2018-11-05 -3e-04 0.0033    0     0          0  -0.0020
##
## ..........................
##                Mu  Sigma Skew Shape Shape(GIG) Realized
## 2019-06-03 -2e-04 0.0039    0     0          0   0.0018
## 2019-06-04 -2e-04 0.0025    0     0          0  -0.0006
## 2019-06-05 -2e-04 0.0026    0     0          0   0.0023
## 2019-06-06 -2e-04 0.0023    0     0          0   0.0006
## 2019-06-07 -2e-04 0.0023    0     0          0   0.0005
## 2019-06-10 -2e-04 0.0023    0     0          0   0.0008
##
## Elapsed: 0.58781 secs
```

by showing the result from roll.eg, we can see the amounts that have been predicted. Simga is the values of volatility that are predicted.

5 Results

We will create two plots. One is the predicted volatility and other is Return of nikkei225. We can calculate Return by subtracting previous close value from present close value but in case for our project, we will use (1+(Close-Open)/Open). This equation is nearly equal to the return values that can be get from subtracting close values.



From 2 plots, we can know that the volatility rate gets higher when the wave of return gets not stationary. for example around 25˜60 in predicted volatilty is getting really high compared to other results and also in return plot, the wave gets really rough. from those result, we can know that the prediction of volatility seems to be predicted by comparing to the return of Nikkei 225.

6 Conclusion

By using ARIMA model and GARCH model, we could predict the future volatility of stock price datas. From the result that we got, we could know that prediction of volatility goes pretty well. The prediction of volatility is mainly used in financial analysis worldwide and also by stockholders that earns by selling and buying stocks. However, still prediction of volatility is not enough to predict perfectly since sudden move in stock price is not predictable. Even humans can not predict sudden move of stock price since it depends on human movements. Even though we can predict that stock price will move in the nearer future by using media information, we can not know how much it will move. we can say this on prediction of volatility using models and computers as well. Problems of predicting those sudden moves on stock price is the core part of machine learning methods for financial models and if we could solve this, there will be a huge inovation on the world of Financial Analysis.

Reference

Andersen, Torben G. and Tim Bollerslev, “Answering the Skeptics: Yes, Standard Volatility Models Do Provide Accurate Forecasts,” International Economic Review, 39(4), 1998, pp.885 905.

Arthur Iinuma, “Why Is the Cryptocurrency Market So Volatile: Expert Take” COINTELEGRAPH, 2018 , https://cointelegraph.com/news/why-is-the-cryptocurrency-market-so-volatile-expert-take, Accessed on 2019/5/27

Beran, Jan, “Maximum Likelihood Estimation of the Differencing Parameter for Invertible Short and Long Memory Autoregressive Integrated Moving Average Models,” Journal of the Royal Statistical Society, B57, 1995, pp.659 672.

Black, Fischer and Myron Scholes, “The Pricing of Options and Corporate Liabilities,” Journal of Political Economy, 81(3), 1973, pp.637 654.

Bollerslev, Tim, “Generalized Autoregressive Conditional Heteroskedasticity,” Journal of Econometrics, 31(3), 1986, pp.307 327.

Cryptodata 2019, https://www.cryptodatadownload.com/, Accessed on 2019/5/26.

Engle, Robert. F., “Autoregressive Conditional Heteroskedasticity with Estimates of the Variance of United Kingdom Inflation,” Econometrica, 50(4), 1982, pp.987 1007.

Hochreiter, Sepp and Schmidhuber, Jurgen, Long short-term memory, Neural computation vol. 9, No. 8, pages=1735 1780, 1997.

Johansen, S., Juselius, K. (1990) Maximum likelihood estimation and inferences on cointegration with applications to the demand for money, Oxford Bulletin of Economics and Statistics vol. 52, pp. 169-210.

Tim Falk, “Cryptocurrency arbitrage made easy: A beginner’s guide, https://www.finder.com/cryptocurrency-arbitrage, Accessed on 2019/5/26