Category: preliminary communication

Al-Mahmood, Ahmed Wasfi Dhahir [1]
Markovskaya, Natalia [2]

Using GARCH Algorithm to Analyze Data in R Language

Abstract:

One of the challenging aspects of conditional heteroskedasticity series is that if we were to plot the correlogram of a series with volatility we might still see what appears to be a realisation of stationary discrete white noise. That is, the volatility itself is hard to detect purely from the correlogram. This is despite the fact that the series is most definitely non-stationary as its variance is not constant in time. So ARCH and GARCH models have become important tools in the analysis of time series data, particularly in financial applications. These models are especially useful when the goal of the study is to analyze and forecast volatility. This paper gives the motivation behind the simplest GARCH model and illustrates its usefulness in examining portfolio risk. So an ARCH (autoregressive conditionally heteroskedasticity) model is a model for the variance of a time series. ARCH models are used to describe a changing, possibly volatile variance. Although an ARCH model could possibly be used to describe a gradually increasing variance over time, most often it is used in situations in which there may be short periods of increased variation. (Gradually increasing variance connected to a gradually increasing mean level might be better handled by transforming the variable). In this article we will see what is ARCH and GARCH, how it's helpful for analyzing economic and financial data and how to use it in R-Studio.

Keywords:

ARCH; volatility clustering; GARCH; Akaike

Author´s data:

[1] Master Student, Yanka Kupala State University of Grodno (Belarus), Gaspadarchaya St., 23, 230009, Grodno, Belarus
[2] Associate Professor, Yanka Kupala State University of Grodno (Belarus), Gaspadarchaya St., 23, 230009, Grodno, Belarus; e-mail: n.markovskaya@grsu.by

Introduction

The great workhorse of applied econometrics is the least squares model. This is natural because applied econometricians are typically called upon to determine how much one variable will change in response to a change in some other variable. Increasingly however, econometricians are being asked to forecast and analyze the size of the errors of the model. In this case the questions are about volatility and the standard tools have become the ARCH/GARCH models. The basic version of the least squares model assumes that, the expected value of all error terms when squared is the same at any given point. This assumption is called homoscedasticity and it is this assumption that is the focus of ARCH/GARCH models. Data in which the variances of the error terms are not equal, in which the error terms may reasonably be expected to be larger for some points or ranges of the data than for others, are said to suffer from heteroskedasticity. The standard warning is that in the presence of heteroskedasticity, the regression coefficients for an ordinary least squares regression are still unbiased, but the standard errors and confidence intervals estimated by conventional procedures will be too narrow, giving a false sense of precision. Instead of considering this as a problem to be corrected, ARCH and GARCH models treat heteroskedasticity as a variance to be modeled. As a result, not only are the deficiencies of least 3 squares corrected, but a prediction is computed for the variance of each error term. This turns out often to be of interest particularly in finance [1].

Autoregressive conditional heteroskedasticity (ARCH). In econometrics, the autoregressive conditional heteroskedasticity (ARCH) model is a statistical model for time series data that describes the variance of the current error term or innovation as a function of the actual sizes of the previous time periods' error terms, often the variance is related to the squares of the previous innovations. The ARCH model is appropriate when the error variance in a time series follows an autoregressive (AR) model; if an autoregressive moving average (ARMA) model is assumed for the error variance, the model is a generalized autoregressive conditional heteroskedasticity (GARCH) model for forecasting, combining ARIMA and ARCH models could be considered. For instance, a hybrid ARIMA-ARCH model was examined for shipping freight rate forecast [2].

ARCH models are commonly employed in modeling financial time series that exhibit time-varying volatility and volatility clustering, i.e. periods of swings interspersed with periods of relative calm. ARCH-type models are sometimes considered to be in the family of stochastic volatility models, although this is strictly incorrect since at time t the volatility is completely pre-determined (deterministic) given previous values [1].

Volatility clustering. Volatility clustering is the tendency of large changes in prices of financial assets to cluster together, which results in the persistence of these magnitudes of price changes. Another way to describe the phenomenon of volatility clustering is to quote famous scientist-mathematician Benoit Mandelbrot, and define it as the observation that "large changes tend to be followed by large changes and small changes tend to be followed by small changes" when it comes to markets. This phenomenon is observed when there are extended periods of high market volatility or the relative rate at which the price of a financial asset change, followed by a period of "calm" or low volatility [3].

Volatility clusters the phenomenon of there being periods of relative calm and periods of high volatility is a seemingly universal attribute of market data. There is no universally accepted explanation of it. GARCH (Generalized Autoregressive Conditional Heteroskedasticity) models volatility clustering. It does not explain it. Figure 1 is an example of a Garch model of volatility [4].

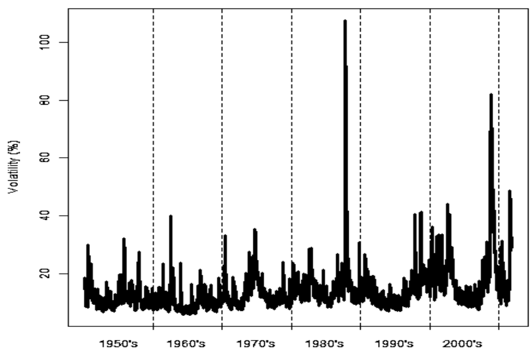

Figure 1- S&P 500 volatility until late 2011 as estimated by a Garch (1, 1) model [4]

The definition of GARCH process. The generalized autoregressive conditional heteroskedasticity (GARCH) process is an econometric term developed in 1982 by Robert F. Engle, an economist and 2003 winner of the Nobel Memorial Prize for Economics, to describe an approach to estimate volatility in financial markets. There are several forms of GARCH modeling. The GARCH process is often preferred by financial modeling professionals because it provides a more real-world context than other forms when trying to predict the prices and rates of financial instruments [5].

Breaking down GARCH Process. Heteroskedasticity describes the irregular pattern of variation of an error term, or variable, in a statistical model. Essentially, where there is heteroskedasticity, observations do not conform to a linear pattern. Instead, they tend to cluster. The result is that the conclusions and predictive value one can draw from the model will not be reliable. GARCH is a statistical model that can be used to analyze a number of different types of financial data, for instance, macroeconomic data. Financial institutions typically use this model to estimate the volatility of returns for stocks, bonds and market indices. They use the resulting information to help determine pricing and judge which assets will potentially provide higher returns, as well as to forecast the returns of current investments to help in their asset allocation, hedging, risk management and portfolio optimization decisions [5].

The general process for a GARCH model involves three steps. The first is to estimate a best-fitting autoregressive model. The second is to compute autocorrelations of the error term. The third step is to test for significance. Two other widely used approaches to estimating and predicting financial volatility are the classic historical volatility (VolSD) method and the exponentially weighted moving average volatility (VolEWMA) method [5].

Example of GARCH Process. GARCH models help to describe financial markets in which volatility can change, becoming more volatile during periods of financial crises or world events and less volatile during periods of relative calm and steady economic growth. On a plot of returns, for example, stock returns may look relatively uniform for the years leading up to a financial crisis such as the one in 2007. In the time period following the onset of a crisis, however, returns may swing wildly from negative to positive territory. Moreover, the increased volatility may be predictive of volatility going forward. Volatility may then return to levels resembling that of pre-crisis levels or be more uniform going forward. A simple regression model does not account for this variation in volatility exhibited in financial markets and is not representative of the "black swan" events that occur more than one would predict [5].

GARCH Models Best for Asset Returns. GARCH processes differ from homoscedastic models, which assume constant volatility and are used in basic ordinary least squares (OLS) analysis. OLS aims to minimize the deviations between data points and a regression line to fit those points. With asset returns, volatility seems to vary during certain periods of time and depend on past variance, making a homoscedastic model not optimal [5].

GARCH processes, being autoregressive, depend on past squared observations and past variances to model for current variance. GARCH processes are widely used in finance due to their effectiveness in modeling asset returns and inflation. GARCH aims to minimize errors in forecasting by accounting for errors in prior forecasting and, thereby, enhancing the accuracy of ongoing predictions [5].

Estimation of Garch. We are staying with a GARCH (1, 1) model; not because it is the best it certainly is not. We are staying with it because it is the most commonly available, the most commonly used, and sometimes good enough [4]. GARCH models are almost always estimated via maximum likelihood. That turns out to be a very difficult optimization problem. That nastiness is just another aspect of us trying to ask a lot of the data. Assuming that you have enough data that it matters, even the best implementations of GARCH bear watching in terms of the optimization of the likelihood [4].

We know that returns do not have a normal distribution, that they have long tails. It is perfectly reasonable to hypothesize that the long tails are due entirely to GARCH effects, in which case using a normal distribution in the GARCH model would be the right thing to do. However, using the likelihood of a longer tailed distribution turns out to give a better fit (almost always). The t distribution seems to do quite well [4].

The Usefulness of Garch model. GARCH or generalized autoregressive conditional heteroskedasticity models are used to model the conditional volatility of a time series.

Financial markets data often exhibit volatility clustering, where time series show periods of high volatility and periods of low volatility. In fact, with economic and financial data, time-varying volatility is more common than constant volatility, and for accurate modeling of time-varying volatility we use GARCH models. A GARCH (1, 1) is in fact equivalent to an ARCH (infinity) model [6].

Using GARCH model in R-Language. The first step to build our GARCH model is we need to install the "quantmod" package by using the command → install.packages ("quantmod") after pressing enter the package will be installed.

This command is so important to build the model to check how it work we use the command → library ("quantmod") and we press enter we get the result:

UDK 330.43; DOI 10.2507/IJVA.5.1.5.59

```
       Loading required package: xts
Loading required package: zoo
Attaching package: 'zoo'
The following objects are masked from 'package:base':

    as.Date, as.Date.numeric

Loading required package: TTR
Version 0.4-0 included new data defaults. See ?getSymbols.
Learn from a quantmod author: https://www.datacamp.com/courses/importing-and-managing-financial-data-in-r
Warning messages:
1: package 'quantmod' was built under R version 3.5.2
2: package 'xts' was built under R version 3.5.2
3: package 'TTR' was built under R version 3.5.2
```

Now we need to use the getSymbols command to bring the data that we will use in GARCH model and it will be Facebook stock data and we will store it (Fb) by using this command → Fb<-getSymbols("FB",auto.assign =F) and press enter and the result will be the stock data of Facebook of the year 2012 and it will be stored in the (Fb):

```
'getSymbols' currently uses auto.assign=TRUE by default, but will
use auto.assign=FALSE in 0.5-0. You will still be able to use
'loadSymbols' to automatically load data. getOption("getSymbols.env")
and
getOption("getSymbols.auto.assign")
will still be checked for
alternate defaults.

This message is shown once per session and may be disabled by setting
options("getSymbols.warning4.0"=FALSE). See ?getSymbols for details.
WARNING: There have been significant changes to Yahoo Finance data.
Please see the Warning section of '?getSymbols.yahoo' for details.

This message is shown once per session and may be disabled by setting
options("getSymbols.yahoo.warning"=FALSE).
```

If we want to see what is this data we use the command → head (Fb) the result will be:

```
           FB.Open   FB.High   FB.Low
FB.Close FB.Volume FB.Adjusted
2012-05-18     42.05     45.00    38.00
38.23 573576400       38.23
2012-05-21     36.53     36.66    33.00
34.03 168192700       34.03
2012-05-22     32.61     33.59    30.94
31.00 101786600       31.00
2012-05-23     31.37     32.50    31.36
32.00  73600000       32.00
2012-05-24     32.95     33.21    31.77
33.03  50237200       33.03
2012-05-25     32.90     32.95    31.11
31.91  37149800       31.91
```

Now we have the stock data of Facebook if we want to see the chart of this data we can do it by using the command □ chart_Series(Fb) it will open the chart for this data and it will be from 2012 until 2019 in the chart we see the stock market for Facebook is growing since 2012 until 2018 it stat going done and we can see it clearly in the Figure 2 that shows the Facebook stocks price:

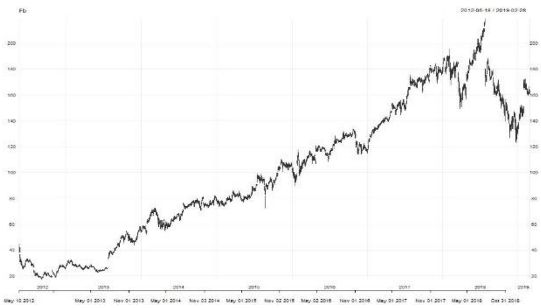

Figure 2 - Facebook stocks price

Now we take the FB.Close to do the GARCH model on it the way to do it by using the command → FbClose<-Fb$FB.Close the $ means that we need to take the FB.Close data and by the way we store it in FbClose to see the data we use the command → head(FbClose) and we will receive the result:

```
             FB.Close
2012-05-18     38.23
2012-05-21     34.03
2012-05-22     31.00
2012-05-23     32.00
2012-05-24     33.03
2012-05-25     31.91
```

Now we need to install the GARCH package and the name of this package in R-language is “rugarch” and we can install it by using the command → install.packages("rugarch") and it will be installed in R-Studio package system.

And if we want to check the package we use the command → library("rugarch").

Now it’s the time to build the GARCH model and we need to build it by using several commands:

The first command will be → Fb0<-ugarchspec(variance.model = list(model="sGARCH",garchOrder=c(1,1)),mean.model = list(armaOrder=c(0,0)),distribution.model = "std").

What we did we put the variance , mean and the distribution in the model and we use (0,0) as the mean because if we used larger number there will be no result at all because the model use the minimum value and we store it in (Fb0).

And the second command will be → FbGarch0<-ugarchfit(spec = Fb0,data = FbClose).

This command will activate the model and give use some results by the way we store it in FbGarch0 after that we see the results by using the command → FbGarch0 and we see what we got :

```
*---------------------------------*
*          GARCH Model Fit        *
*---------------------------------*

Conditional Variance Dynamics
-----------------------------------
GARCH Model       : sGARCH(1,1)
Mean Model        : ARFIMA(0,0,0)
Distribution      : std
Optimal Parameters
------------------------------------
Estimate  Std. Error    t value Pr(>|t|)
mu      117.848477     0.25169 468.230899
0.000000
omega     1.483258     0.33576   4.417569
0.000010
alpha1    0.998993     0.11957   8.354842
0.000000
beta1     0.000007     0.12780   0.000052
0.999958
shape    99.999936    24.40302   4.097851
0.000042

Robust Standard Errors:
Estimate  Std. Error    t value Pr(>|t|)
mu      117.848477     0.62511 188.525699
0.000000
omega     1.483258     1.28268   1.156378
0.247526
alpha1    0.998993     0.23992   4.163916
0.000031
beta1     0.000007     0.24791   0.000027
0.999978
shape    99.999936     1.22662  81.524543
0.000000

LogLikelihood : -8471.395

Information Criteria
------------------------------------
Akaike       9.9430
Bayes        9.9589
Shibata      9.9430
Hannan-Quinn 9.9489

Weighted Ljung-Box Test on Standardized
Residuals
------------------------------------
statistic p-value
Lag[1]                      1591       0
Lag[2*(p+q)+(p+q)-1][2]     2375       0
Lag[4*(p+q)+(p+q)-1][5]     4660       0
d.o.f=0
H0 : No serial correlation

Weighted Ljung-Box Test on Standardized
Squared Residuals
------------------------------------
statistic p-value
Lag[1]                    0.0857 0.76971
Lag[2*(p+q)+(p+q)-1][5]           2.8916
0.42720
Lag[4*(p+q)+(p+q)-1][9]           9.0539
0.07924
d.o.f=2
Weighted ARCH LM Tests
------------------------------------
Statistic Shape Scale P-Value
ARCH  Lag[3]   0.0006206   0.500   2.000
0.98013
ARCH  Lag[5]   2.2798322   1.440   1.667
0.41264
ARCH  Lag[7]   8.3779082   2.315   1.543
0.04321

Nyblom stability test
------------------------------------
Joint Statistic:  469.6939
Individual Statistics:
mu         2.52781
omega      0.04648
alpha1     4.54412
beta1      0.28380
shape    442.01008

Asymptotic Critical Values (10% 5% 1%)
Joint Statistic:             1.28     1.47
1.88
```

```
Individual Statistic:      0.35    0.47
0.75
Sign Bias Test
------------------------------------
t-value   prob sig
Sign Bias           0.4474 0.6546
Negative Sign Bias  1.0337 0.3014
Positive Sign Bias  1.0822 0.2793
Joint Effect        2.6456 0.4496

Adjusted Pearson Goodness-of-Fit Test:
------------------------------------
group statistic p-value(g-1)
1     20      9703            0
2     30     14521            0
3     40     17731            0
4     50     16631            0

Elapsed time : 0.4993329
```

The most important information is the Akaike = 9.9430 the less it is the batter the model will be and that's how to build Garch model and how it works.

Akaike information criterion. The Akaike information criterion (AIC) is an estimator of the relative quality of statistical models for a given set of data. Given a collection of models for the data, AIC estimates the quality of each model, relative to each of the other models. Thus, AIC provides a means for model selection.

AIC is founded on information theory. When a statistical model is used to represent the process that generated the data, the representation will almost never be exact; so some information will be lost by using the model to represent the process. AIC estimates the relative amount of information lost by a given model: the less information a model loses, the higher the quality of that model.

In estimating the amount of information lost by a model, AIC deals with the trade-off between the goodness of fit of the model and the simplicity of the model. In other words, AIC deals with both the risk of over fitting and the risk of under fitting.

The Akaike information criterion is named after the statistician Hirotugu Akaike, who formulated it. It now forms the basis of a paradigm for the foundations of statistics; as well, it is widely used for statistical inference [7]. In the future if we build several Garch models the model with the lowest Akaike value will be the beat model to use.

Now lest try other company like Google and let's see how the model will be we do the same stapes that we did with the Facebook but we need to change the data by using the command → GG<-getSymbols ("GOOG",auto.assign =F) this help us to get the financials data for Google and we use the chart command to see the chart for Google stocks price. After doing the commands we will get both result for our model and the chart Figure 3 that shows the Google stocks pries and how it changes since 2007:

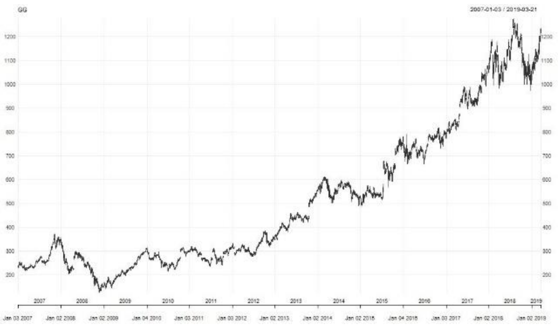

Figure 3 - Google stocks price chart

```
Information Criteria
------------------------------------
-----

Akaike       12.232
Bayes        12.241
Shibata      12.232
Hannan-Quinn 12.235
```

Conclusion

So the Akaike is 12.232 and that means our model not that good and if we compare it with Facebook model we see that Facebook model is better than Google model and as we see the that we can use the model on Facebook stocks but we can't use It on Google stocks because it more efficient to use the model that can help us to find the specific information or details of the stock of any company.

References

[1] GARCH 101: An Introduction to the Use of ARCH/GARCH models in Applied Econometrics *Available from:* http://www.stern.nyu.edu/rengle/GARCH101.PDF *Accessed:* 2019-04-15

[2] Autoregressive conditional heteroskedasticity *Available from:* https://en.wikipedia.org/wiki/Autoregressive_conditional_heteroskedasticity *Accessed:* 2019-04-15

[3] Overview of Volatility Clustering *Available from:* https://www.thoughtco.com/volatility-clustering-in-economics-1147328 *Accessed:* 2019-04-15

[4] A practical introduction to GARCH modeling *Available from:* https://www.r-bloggers.com/a-practical-introduction-to-garch-modeling/ *Accessed:* 2019-04-15

[5] GARCH Process *Available from:* https://www.investopedia.com/terms/g/generalalizedautogregressiveconditionalheteroskedasticity.asp *Accessed:* 2019-04-15

[6] What is a GARCH model? What are they used for? *Available from:* https://www.quora.com/What-is-a-GARCH-model-What-are-they-used-for *Accessed:* 2019-04-15

[7] Akaike information criterion *Available from:* https://en.wikipedia.org/wiki/Akaike_information_criterion *Accessed:* 2019-04-15