Time Series Study

Volatility Estimation of Facebook Stock

Using Garch Modeling

STA 9701 Time Series Course Project 2

Xiaoying Zhao

December 17, 2019

Table of Contents

Introduction

This project explored the stock market volatility using garch model. It was common sense that news, either good or bad, had huge impacts on the stock market. To find a company with substantial changes in its stock, this project emphasized on the companies with big scandals then finally decided to study volatility by using Facebook stock price data.

Exploratory Analysis

Daily Closing Price Observation

The data is daily closing price of Facebook stock from Yahoo Finance, from July 31st 2015 to July 31st 2019, totalling 1007 analytic observations.

From the daily closing price plot, we can see that, before 2018, daily closing price presented an overall increasing trend with some fluctuations, which are mainly due to nature of stock market. At the early 2018, daily closing price of facebook experienced a dramatic drop, which was because data scandal of Facebook and Cambridge Analytica was initially released by the Observer, the guardian, and the Channel 4 New simultaneously. Cambridge Analytica, a British political consulting firm, mined the personal data of an estimated 87 million[1] facebook profiles without user consent and used it for political advertising purposes. This huge drop

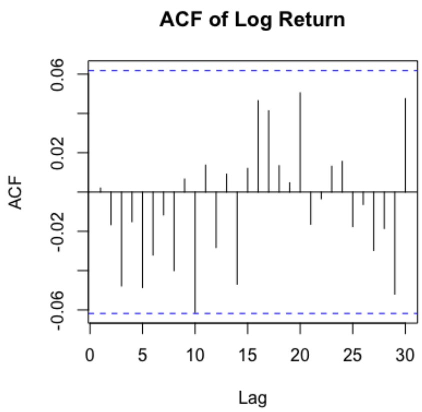
[1] https://www.nytimes.com/2018/04/10/us/politics/mark-zuckerberg-testimony.html

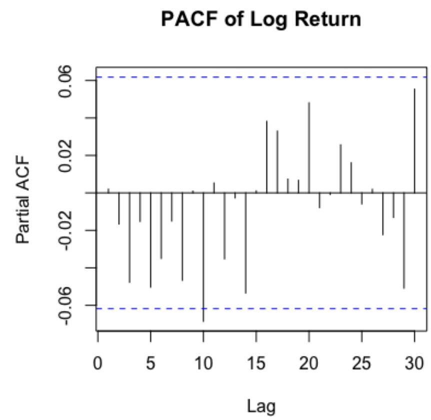
knocked off more than $100 billions from facebook's market capitalization. It eventually brought Facebook CEO, Mark Zuckerberg to testify in front of the United States Congress.

The first Zuckerberg's testimony was held on April 2018, which had a positive influence on the stock. He showed humour during the testimony, helping increase public confidence on social media market. But as investigation continued, more detailed information about data mishandling within facebook and association with Trump election was revealed. The stock continued dropping until end of 2018, which was the time that facebook changed its privacy policies.

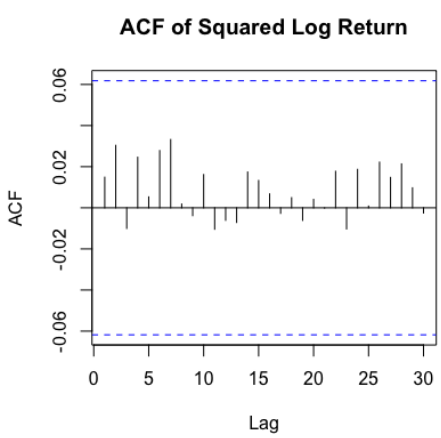
Log Return Transformation

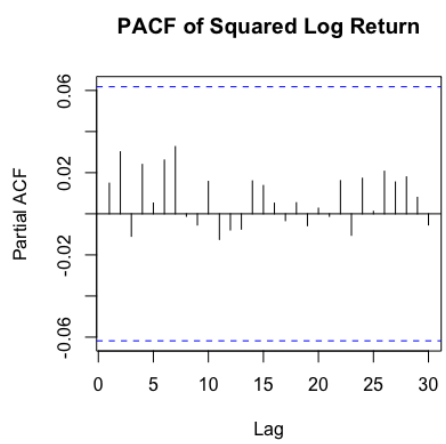
Plots of both Log return and squared log return indicated that facebook daily closing price presented volatility clustering, suggesting changing variance. Also, its mean fluctuated around 0.





Both acf and pacf of log return cut at lag 10, illustrating little serial correlation. Correlation plots below suggested ARMA(p,q) model. ARMA(0,0), white noise model, could be the most possible candidate model, which would be verified in the further discussion.





Plots of both acf and pacf of squared log return did not present any significant serial correlation. Consequently, facebook daily closing price return data are serially uncorrelated and did not admit any higher-order dependence structure.

Garch Modeling

ARMA models were used to capture the conditional expectation of a process given the past, but its conditional variance was constant, meaning it was not able to capture the volatility of a process, such as daily returns. Garch time series models have randomly varying volatility, therefore it was adopted here to study the variance performance of daily returns.

Mean Function Modeling: ARMA(p,q)

Performing the Augmented Dickey-Fuller test to exam stationarity of Facebook price return data. It was confirmed that the return data was stationary time series, meaning ARIMA models can be used to present mean function of daily return model.

```
AR/MA
  0 1 2 3 4 5 6 7 8 9 10 11 12 13
0 o o o o o o o o o o o  o  o  o
1 x o o o o o o o o o o  o  o  o
2 x x o o o o o o o o o  o  o  o
3 x x x o o o o o o o o  o  o  o
4 x x x x o o o o o o o  o  o  o
5 x x x x o o o o o o o  o  o  o
6 x x x x x o o o o o o  o  o  o
7 x x o x x x o o o o o  o  o  o
```

MODEL	AIC	SELECTION
ARIMA(0,0,0)	4108.02	
ARIMA(1,0,1)	4113.84	
ARIMA(2,0,2)	4015.46	
ARIMA(3,0,3)	4111.29	
ARIMA(4,0,4)	4106.56	
ARIMA(5,0,4)	4109.58	
ARIMA(6,0,5)	4113.49	

Auto.arima function in R was used to find the candidate ARMA models, returning candidate model ARMA(2,0,2). Eacf function in R was also performed, also giving some possible candidates. All selected candidate models were fitted into daily returns dataset and examined AIC to select the optimal model. Candidate models with relatively least AIC value were selected out: ARIMA(1,0,1), ARIMA(2,0,2), and ARIMA(3,0,3). Coefficient tests for each model were followed to test coefficient significance. But coefficients test of ARIMA(3,0,3) returned NA.

Models	Z test of coefficients	Selection
ARIMA(0,0,0)	`Estimate Std. Error z value Pr(>|z|)` `intercept 0.072131 0.058716 1.2285 0.2193`	

ARIMA(2,0,2)	`Estimate Std. Error z value Pr(>|z|)`
	`ar1 1.701007 0.197237 8.6242 < 2.2e-16 ***`
	`ar2 -0.769442 0.177648 -4.3313 1.482e-05 ***`
	`ma1 -1.716064 0.210376 -8.1571 3.431e-16 ***`
	`ma2 0.767235 0.196637 3.9018 9.548e-05 ***`
	`intercept 0.072599 0.043742 1.6597 0.09697 .`	

ARIAM(3,0,3)	--	

Based on the results of coefficient tests, ARIMA(2,0,2) presented all significant parameters. Consequently, ARIMA(2,0,2) would be regarded as mean function of daily return model.

Variance Function Modeling: sGarch

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

Conditional Variance Dynamics
-----------------------------------
GARCH Model     : sGARCH(1,1)
Mean Model      : ARFIMA(2,0,2)
Distribution    : std

Optimal Parameters
------------------------------------
        Estimate  Std. Error    t value Pr(>|t|)
mu      0.118001    0.017657     6.6828 0.000000
ar1     1.784860    0.000890  2004.9549 0.000000
ar2    -0.786906    0.000797  -987.7581 0.000000
ma1    -1.843013    0.000046 -40034.9771 0.000000
ma2     0.843799    0.000039  21501.1432 0.000000
omega   0.052646    0.018730     2.8109 0.004941
alpha1  0.061867    0.010976     5.6368 0.000000
beta1   0.930372    0.003011   308.9909 0.000000
shape   3.201682    0.346845     9.2309 0.000000
```

Usually sGarch(1,1) can capture volatility clustering of majority stocks. I fit daily return data into the model ARIMA(2,0,2) + sGARCH(1,1). The result turned out perfect. Coefficient tests of all parameters in the model were statistically significant. Therefore, ARIMA(2,0,2) + sGarch(1,1) was regarded as final model to study volatility estimation of Facebook stock data.

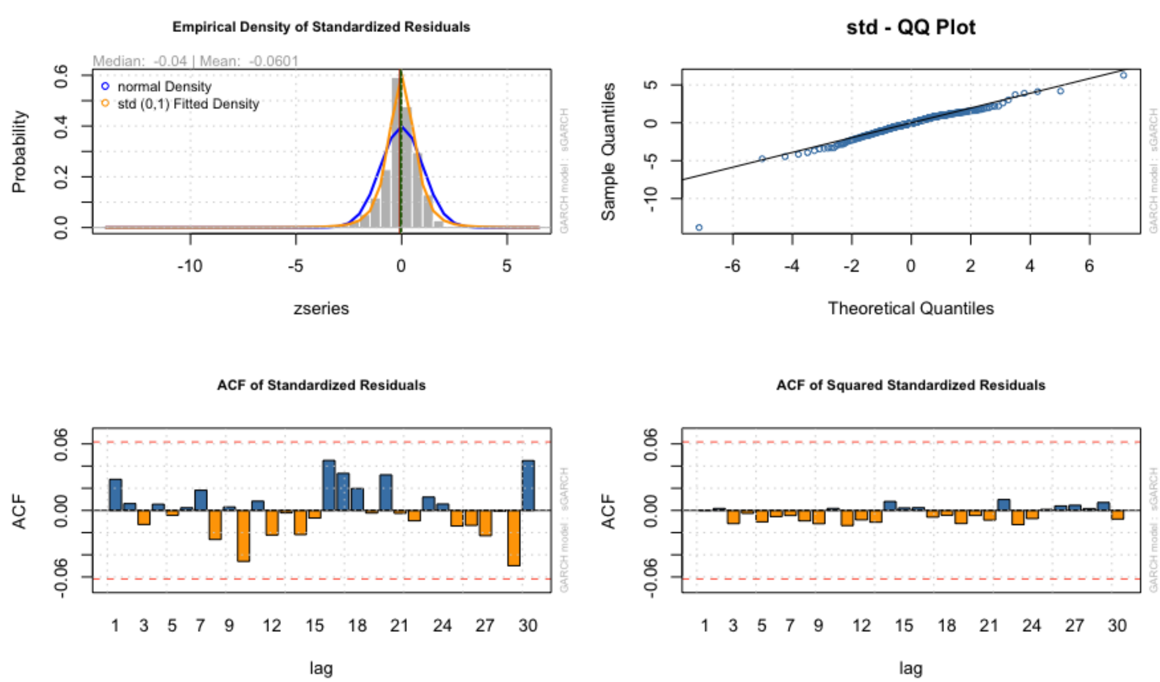
Final Model: ARMA-GARCH

$$Y_t = 0.118001 + 1.784860Y_{t-1} - 0.786906Y_{t-2} + \epsilon_t - 1.843013\epsilon_{t-1} + 0.843799\epsilon_{t-2}$$

$$Y_t = \sigma_t \epsilon_t \ \ , \ \epsilon_t \sim N(0,1)$$

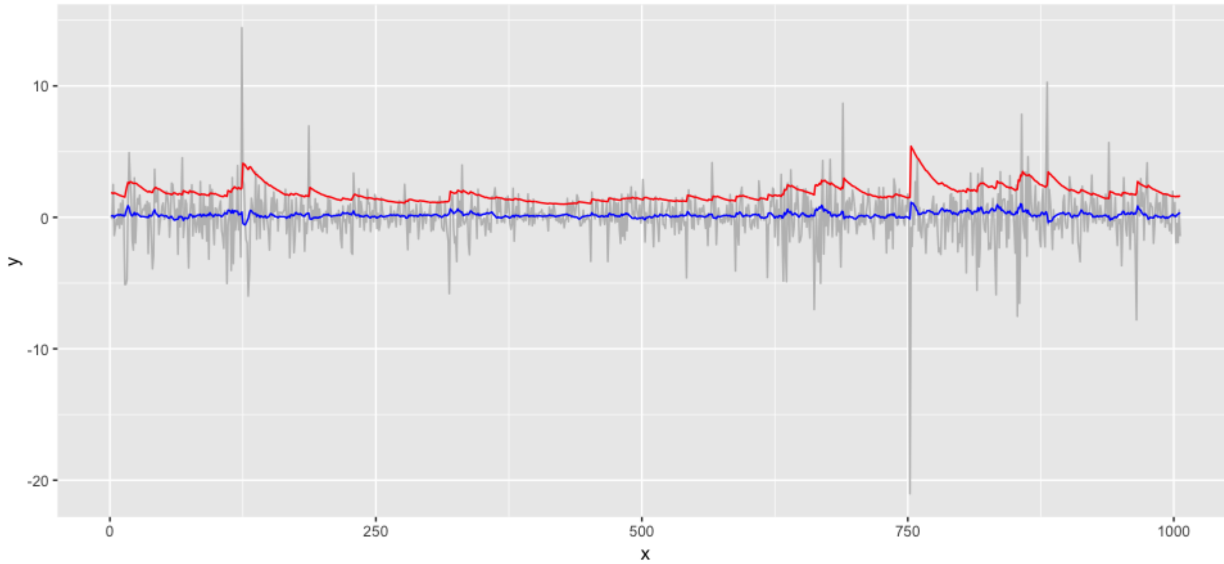
$$\sigma_t^2 = 0.052646 + 0.061867\sigma_{t-1}^2 + 0.0930372Y_{t-1}^2$$

The diagnostic plots were presented below. The residuals of model fitted into normal distribution.



Volatility Analysis

The plot below presented the fitted model ARIMA(2,0,2)+sGarch(1,1). The red line was the conditional volatility over the time, which captured the changing variance of the daily closing price. The blue line was the mean of fitted model, fluctuating around zero.



Further Improvement

Overall, the model well described the volatility of Facebook daily return. All coefficients in the model were statistically significant. No correlation within standardized residuals or standardized squared residuals was found. However, the arch test of log return did not give a significant result, which did not correspond to the model fitness. Transformation or data manipulation mistakes were considered here. It was possible that the data actually did not have arch effect. The component required further study.

Appendix: Code

```
library(quantmod)
library(rugarch)
library(fBasics)
library(lmtest)
library(urca)
library(ggplot2)
library(PerformanceAnalytics)
library(FinTS)
library(forecast)
library(strucchange)
library(zoo)
library(TSA)
library(cold)
library(cubature)
library(lmtest)
library(reshape)
library(magrittr)
library(ggpubr)
library(xts)
library(aTSA)

fb = getSymbols("FB", auto.assign=F)
n = dim(fb)[1]
fb1 = fb[805:(n-96),]
dim(fb1)


chartSeries(fb1)

FBts = ts(fb1$FB.Close)
r.FB=diff(log(FBts))*100
r.fb = r.FB[!is.na(r.FB)]



plot(r.FB,main="Log Return of Facebook Daily Closing Price")
plot(r.FB^2,main="Squared Log Return of Facebook Daily Closing Price")

acf(r.FB, main = "ACF of Log Return")
pacf(r.FB, main = "PACF of Log Return")

acf(r.fb^2, main = "ACF of Squared Log Return")
pacf(r.fb^2, main = "PACF of Squared Log Return")

adf = adf.test(r.FB)
```

```
auto_model = auto.arima(r.fb)
summary(auto_model)
#ARMA(2,2)
coeftest(auto_model)

#selected arma(0,0)
auto_model2 = auto.arima(r.fb,max.q=1)
summary(auto_model2)
coeftest(auto_model2)
#arma(0,0)


eacf(r.fb)
arma00 = arima(r.fb,order=c(0,0,0),method='ML')
arma00
coeftest(arma00)


arma11 = arima(r.fb,order=c(1,0,1),method='ML')
arma11
coeftest(arma11)

#selected ARMR(2,2)

arma202 = arima(r.fb,order=c(2,0,2),method='ML')
arma202
coeftest(arma202)

arma33 = arima(r.fb,order=c(3,0,3),method='ML')
arma33

coeftest(arma33)


#selected ARMA(4,4)
arma44 = arima(r.fb,order=c(4,0,4),method='ML')
arma44
coeftest(arma44)

arma54 = arima(r.fb,order=c(5,0,4),method='ML')
arma54

arma65 = arima(r.fb,order=c(6,0,5),method='ML')
arma65
```

```
#ARCH effect test

model_residuals = residuals(arma11)
ArchTest(model_residuals, lags=23)

summary(pacf(r.fb))
Box.test(r.fb,lag=23,type='Ljung')

var=(r.fb-mean(r.fb))^2
Box.test(var,lag=30,type='Ljung')


plot(density(r.fb))

# ruGARCH
fb_1 = ugarchspec(variance.model = list(model="sGARCH", garchOrder=c(1,1)),
          mean.model = list(armaOrder=c(2,2)),distribution.model = "std")
fbGarch1 = ugarchfit(spec=fb_1,data = r.fb)
fbGarch1

fb_2 = ugarchspec(variance.model = list(model="eGARCH", garchOrder=c(1,1)),
          mean.model = list(armaOrder=c(2,2)),distribution.model = "std")
fbGarch2 = ugarchfit(spec=fb_2,data = r.fb)
fbGarch2

par(mfrow = c(2,2))
plot(fbGarch1, which=8)
plot(fbGarch1, which=9)
plot(fbGarch1, which=10)
plot(fbGarch1, which=11)

par(mfrow=c(1,1))
cond_volatility <- sigma(fbGarch1)
mean_model_fit <- fitted(fbGarch1)

summary(r.fb)
dim(cond_volatility)

df1 = data.frame(x=seq(1,1006), y =cond_volatility)
df2 = data.frame(x=seq(1,1006), y =mean_model_fit )
df3 = data.frame(x=seq(1,1006), y =r.fb)
ggplot(df3, aes(x, y)) +
  geom_line(color="grey") +
  geom_line(data = df1, color = "red")+
  geom_line(data = df2, color = "blue")
```