Time Series Analysis Project: Analysis of Tesla stock returns

Yandi Chen
Baruch College
December 19, 2019

1. Introduction

In this project, I will keep analyzing Tesla stock and forecast the stock price of Tesla. My original interest in selecting Tesla stock was motivated by its stock price increase in late October. The stock price of Tesla went up by 30% in two days, and it is at $350 per share as of November 14th when I did my first project for this course. As of today, December 18th, Tesla stock is worth $394 per share. I am interested in learning if right now is an appropriate time to continue investing in Tesla's stock by performing a further time series analysis.

The dataset was collected from Yahoo Finance web source, and it contains weekly adjusted closing price from June 06, 2010 to December 09, 2019.

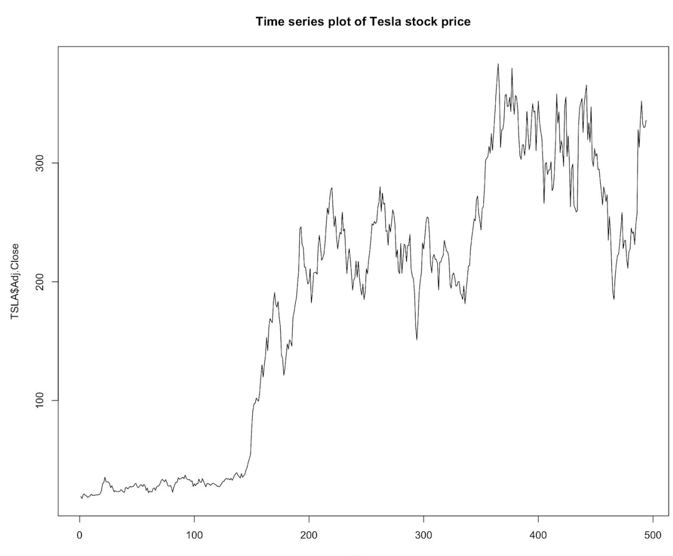


Figure 1: Time series plot of Tesla stock price

From the plot below we can observe that time series is a random walk with drift, and it is obvious that the raw data of stock price is non-stationary because it has non-stable variance and unconstant moving trend.

2. Exploratory Data analysis

First, I plot the log of Tesla stock as shown in Figure 2. Even with log transformation, the series still looks non-stationary. Therefore, instead of using the adjusted closing price, I decided to use weekly stock returns, which is also the same as log of stock price, where X is the value of the stock at time t:

$$Return_t = \frac{X_t - X_{t-1}}{X_{t-1}}$$

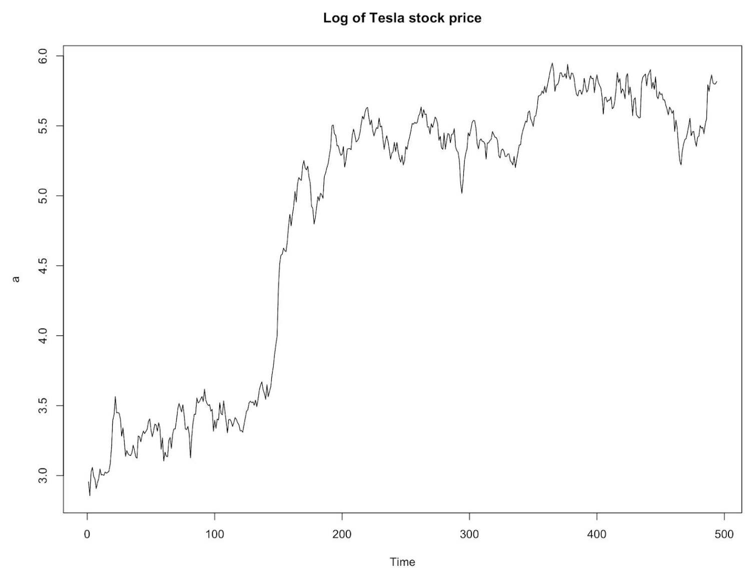


Figure 2: Time series plot of Log of Tesla stock price

Then, after difference the log of Tesla stock price, also known as log returns of the stock, the outcome is shown in figure 3.

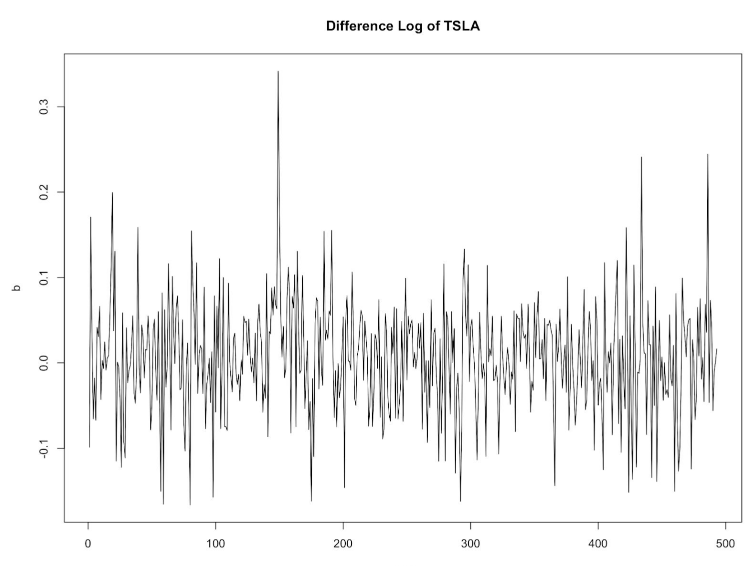


Figure 3: Log-Returns of Tesla

As we can see, difference the log returns shows us a relatively stationary time series. There is no apparent trend and the mean is relatively constant and equal to 0.0058. The Augmented Dickey-Fuller test also confirmed my previous analysis. We reject the null hypothesis and accept the alternative hypothesis that the series is stationary with p-value less than 0.01.

```
        Augmented Dickey-Fuller Test

data:  b
Dickey-Fuller = -7.8643, Lag order = 7, p-value = 0.01
alternative hypothesis: stationary
```

The ACF/ PACF plot shows no discernible pattern and would most likely be considered random or white noise.The returns appear equally likely to be positive or negative from one observation to the next. This implies that the sign of the value of observations is independent of the past, however, the magnitude of change in observations may show correlation.

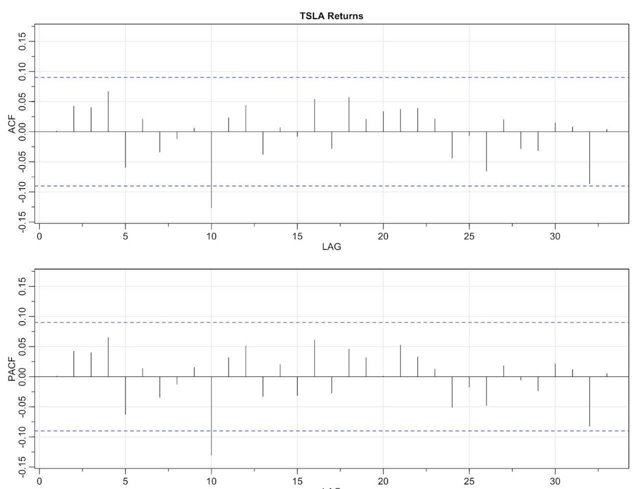


Figure 4: ACF and PACF plots of Tesla log-returns

To verify if there is a correlation, I plot the squared log returns of the series as shown in Figure 5. As we can notice, there is somehow slow decay of the squared log returns, and this might indicate there is some dependence on the variance of the data.

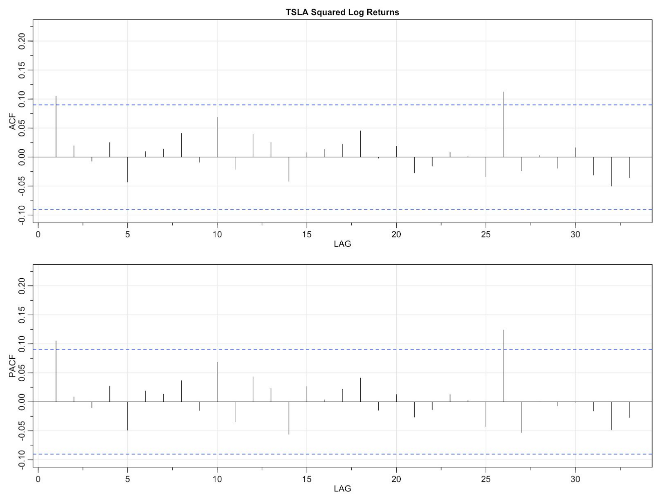


Figure 5: ACF and PACF plots of Tesla Squared log-returns

The ACF and PACF of Tesla squared returns indicate that a Generalized Autoregressive Conditional Heteroskedasticity (GARCH) model could be a good fit for the data.

3. Model selection

I first took eacf of squared log return to test the mean, and the result is shown on the right. R determines that the best fit model is ARMA(1,0) because it has the lowest AIC value.

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

As to modelling the volatility, I calculated the AIC and BIC for different GARCH models in order to select the best fit. After comparing these models through AIC and BIC criteria, which are summarized in Table 1, we can observe that GARCH(1,0) model has the lowest AIC and BIC.

Model	AIC	BIC
GARCH (1,0)	-2.585563	-2.558522
GARCH (1,1)	-2.592059	-2.566498
GARCH (1,2)	-2.588282	-2.554201
GARCH (2,2)	-2.584225	-2.541623

4. Model diagnostic

Using the R-studio we can fit GARCH(1,1) and get the following estimation for the parameters:

```
Coefficient(s):
     omega      alpha1       beta1
0.00040821  0.03645898  0.86957752
```

Since the sum of alpha1 and beta1 is less than 1, the stationarity condition holds. Then I doubled tested to see if GARCH(1,1) is a really good fit. As we can see from the results below, the Jarque Bera Test is significant for the null hypothesis. Ljung-Box Tests and LM ARCH test are not significant, which indicates that the ARCH effects have been accommodated by the model and the chosen GARCH(1,1) model fits the series well.

```
Standardised Residuals Tests:
                                Statistic p-Value
 Jarque-Bera Test   R    Chi^2  67.9477   1.776357e-15
 Shapiro-Wilk Test  R    W      0.9802272 3.070639e-06
 Ljung-Box Test     R    Q(10)  13.68086  0.1880524
 Ljung-Box Test     R    Q(15)  15.1383   0.4415022
 Ljung-Box Test     R    Q(20)  19.35251  0.4990302
 Ljung-Box Test     R^2  Q(10)  4.912305  0.8969585
 Ljung-Box Test     R^2  Q(15)  7.638798  0.9374463
 Ljung-Box Test     R^2  Q(20)  9.112522  0.981567
 LM Arch Test       R    TR^2   6.178869  0.906798

Information Criterion Statistics:
      AIC       BIC       SIC      HQIC
-2.592059 -2.566498 -2.592133 -2.582023
```

Then, I checked the model diagnostics to verify that my model selection is appropriate. As shown below, there are a few outliers in the series as exhibited in the Q-Q plot of residuals and a very heavy left tail, but the observations in the QQ-plot do line up along the 45 degree line, which means the residuals are normally distributed.

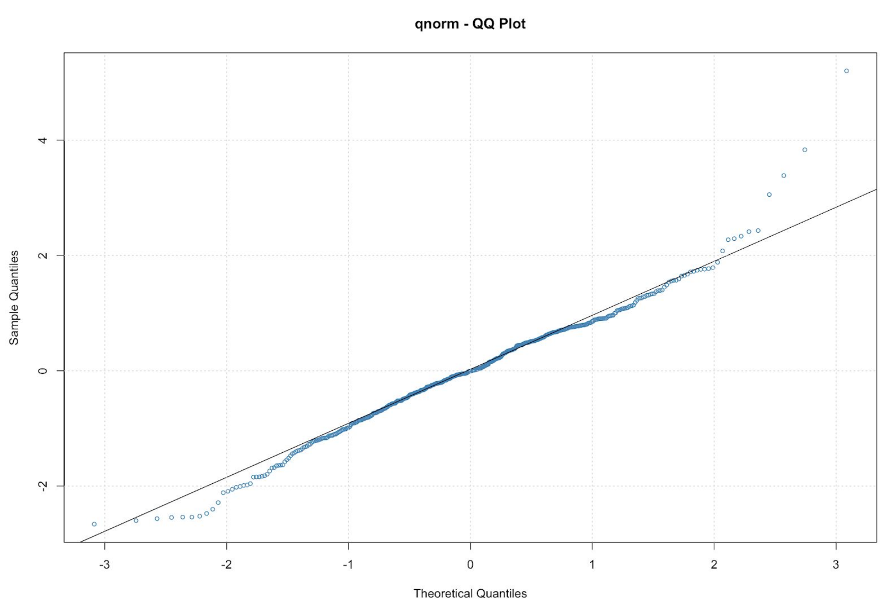
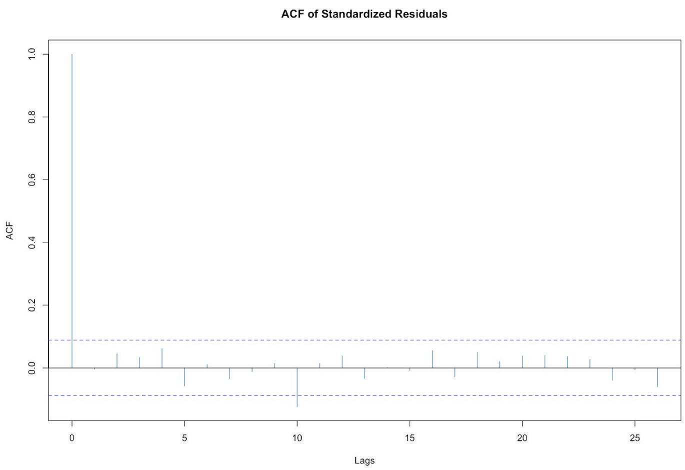


Figure 6: QQ-Plot of Standardized Residuals

The ACF plot of the residuals shown in Figure 7 and ACF of squared residuals demonstrated in Figure 8 shows white noise pattern.



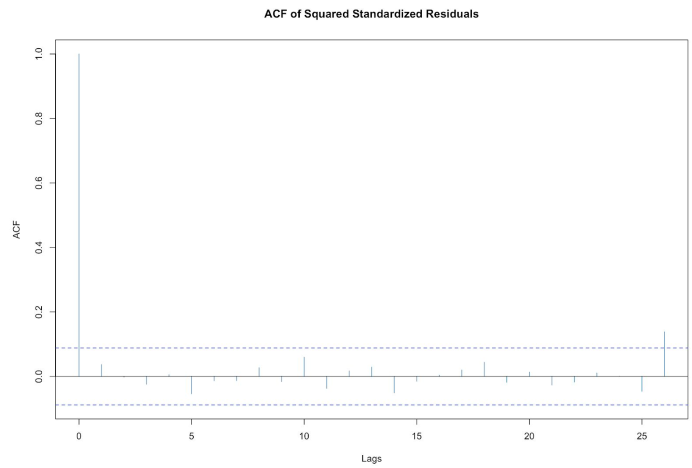
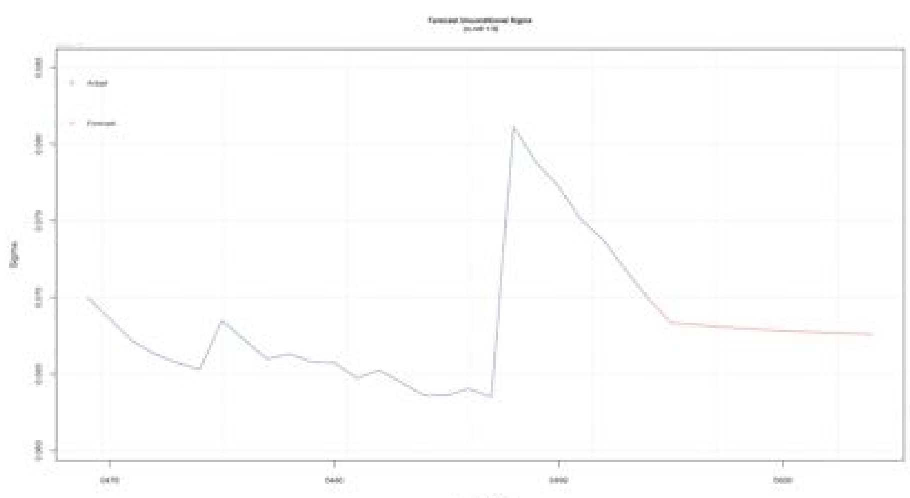


Figure 7:ACF of Standardized Residuals ` Figure 8:ACF of Squared Standardized
Residuals

5. Model forecast

Using the R function ugarchforecast, I forecasted 10 days ahead. Because the stock price kept decreasing in the past few months, the model also fell slightly short.



Appendix:

```
library(gdata)
library(forecast)
library(TSA)
library(tseries)
library(astsa)
library(xts)
install.packages("rugarch")
library(rugarch)
install.packages("fGarch")
library(fGarch)

TSLA2 = read.csv("/Users/yandichen/Downloads/TSLA.csv", header = TRUE)
names(TSLA2)

# plot time series using daily adj.close
tsla = ts(TSLA2$Adj.Close)
plot.ts(TSLA2$Adj.Close, main = "Time series plot of Tesla stock price")

a = log(tsla)
plot.ts(a, main = "Log of Tesla stock price")

# difference the log of TSLA
b = diff(a)
plot.ts(b, main = "Difference Log of TSLA")

# Test on Stationarity
adf.test(b)

sampleACF1 = acf(b, lag.max = 500, main="sample ACF")
samplePACF2 = pacf(b, lag.max = 500,main="sample PACF")

acf2(b, main="TSLA Returns")
acf2(b^2, main="TSLA Squared Log Returns")
eacf(b^2)

ma1 = arima(b^2,c(0,0,1))
ma2 = arima(b^2,c(1,0,0))
ma3 = arima(b^2,c(1,0,1))

# Model selection
y = b - mean(b)
g0.model = garchFit(~garch(0,1),y,include.mean= F, trace=F )
(g0.model@fit)$ics[1]    # AIC
```

```
(g0.model@fit)$ics[2]    # BIC

g1.model = garchFit(~garch(1,1),y, include.mean= F, trace=F)
(g1.model@fit)$ics[1]    # AIC
(g1.model@fit)$ics[2]    # BIC

g2.model = garchFit(~garch(1,0),y, include.mean= F, trace=F)
(g2.model@fit)$ics[1]    # AIC
(g2.model@fit)$ics[2]    # BIC

g3.model = garchFit(~garch(1,2),y, include.mean= F, trace=F)
(g3.model@fit)$ics[1]    # AIC
(g3.model@fit)$ics[2]    # BIC

g4.model = garchFit(~garch(3,1),y, include.mean= F, trace=F)
(g4.model@fit)$ics[1]    # AIC
(g4.model@fit)$ics[2]    # BIC


# GARCH (1,0) ~fGarch
model = garchFit(~garch(1,0),y, include.mean= F, trace=F)
summary(model)
plot(model)

# GARCH (1,0) ~ruGarch
y1 = na.omit(y)

model1 = ugarchspec(variance.model = list(model="sGARCH", garchOrder = c(1,1)),
                    mean.model= list(armaOrder= c(1,0), include.mean= F),
                    distribution.model="norm")
modelfit1 = ugarchfit(spec = model1, data=y1)
coef(modelfit1)
infocriteria(modelfit1)
plot(modelfit1)
show(modelfit1)
summary(modelfit1)

preds = ugarchforecast(modelfit1, n.head=10)
sigma(preds)
plot(preds)

mdl1 = garchFit(~arma(1, 0) + garch(1,1),x_t)
```

```
# 2nd way
#tslart1 = CalculateReturns(tsla)
model2 = ugarchspec(mean.model = list(armaOrder= c(0,0)),
                    variance.model= list(model= "sGARCH"), distribution.model = "sstd")
modelfit2 = ugarchfit(spec = model2, data=y1)
preds2 = ugarchforecast(modelfit2, n.head=10)
plot(preds2)

# use sigma to retrieve the est volatilities
modelvol = sigma(modelfit1)
modelvol

# compute unconditional volatility
sqrt(uncvariance(modelfit1))


# forecast 10 days
modelforecast = ugarchforecast(modelfit1, n.ahead = 10)
```