Time Series Analysis with ARIMA – ARCH/GARCH model in R

I. Introduction:

Time series analysis is a major branch in statistics that mainly focuses on analyzing data set to study the characteristics of the data and extract meaningful statistics in order to predict future values of the series. There are two methods in time series analysis, namely: frequency-domain and time-domain. The former is based mostly on Fourier Transform while the latter closely investigates the autocorrelation of the series and is of great use of Box-Jenkins and ARCH/GARCH methods to perform forecast of the series.

This paper will provide the procedure to analyze and model financial times series in R environment using the time-domain method. The first part covers the stationary and differencing in time series. The second and third parts are the core of the paper and provide a guide to ARIMA and ARCH/GARCH. Next, it will look at the combined model as well as its performance and effectiveness in modeling and forecasting the time series. Finally, summary of time series analysis method will be discussed.

II. Stationarity and differencing of time series data set:

1. Stationarity:

The first step in modeling time index data is to convert the non-stationary time series to stationary one. This is important for the fact that a lot of statistical and econometric methods are based on this assumption and can only be applied to stationary time series. Non-stationary time series are erratic and unpredictable while stationary process is mean-reverting, i.e, it fluctuates around a constant mean with constant variance. In addition, stationarity and independence of random variables are closely related because many theories that hold for independent random variables also hold for stationary time series in which independence is a required condition. The majority of these methods assume the random variables are independent (or uncorrelated); the noise is independent (or uncorrelated); and the variable and noise are independent (or uncorrelated) of each other. So what is a stationary time series?

Roughly speaking, stationary time series shows no long-term trend, has constant mean and variance. More specifically, there are two definitions for stationarity: weak stationarity and strict stationarity.

a. Weak stationarity: *The time series $\{Xt, t \in Z\}$ (where Z is the integer set) is said to be stationary if*
   i. $E(X_t^2)<\infty \ \forall \ t \in Z$.
   ii. $EX_t = \mu \ \forall \ t \in Z$.
   iii. $\gamma_X(s,t)=\gamma_X(s+h,t+h) \ \forall \ s,t,h \in Z$.

b. Strict stationarity: *The time series $\{Xt, t \in Z\}$ is said to be strict stationary if the joint distribution of $(X_{t1}, X_{t2}, \ldots, X_{tk})$ is the same as that of $(X_{t1+h}, X_{t2+h}, \ldots, X_{tk+h})$.*

Mostly in statistics literature, stationarity refers to weak stationarity where the stationary time series satisfies three conditions: constant mean, constant variance, and autocovariance function only depends on (t-s) (not t or s). On the other hand, strict stationarity implies that the probability distribution of time series does not change over time.

For example, white noise is stationary and implies that the random variables are uncorrelated, not necessarily independent. However, a strict white noise indicates the independence among variables. Additionally, a Guassian white noise is strictly stationary since Guassian distribution is characterized by the first two moments, and therefore, uncorrelation also implies independence of random variables.

In a strict white noise, the noise term $\{e_t\}$ cannot be predicted either linearly or non-linearly and it reflects the true innovation in the series; in general white noise, this term might not be predicted linearly yet is probably predicted non-linearly by ARCH/GARCH model that will be discussed later. It is noted that, the noise term is only true innovation of a time series when it is strictly stationary and cannot be predicted; otherwise this term is referred to as errors.
There are three points necessary to mention:

- Strict stationarity does not imply weak stationary because it does not require finite variance
- Weak stationarity does not imply strict stationarity because higher moments might depend on time while strict stationarity requires probability distribution does not change over time
- Nonlinear function of strict stationary series is also strictly stationary, this does not apply to weak stationary

2. Differencing:

In order to convert non-stationary series to stationary, differencing method can be used in which the series is lagged 1 step and subtracted from original series:

For example: $Y_t = Y_{t-1} + e_t \Rightarrow e_t = Y_t - Y_{t-1}$

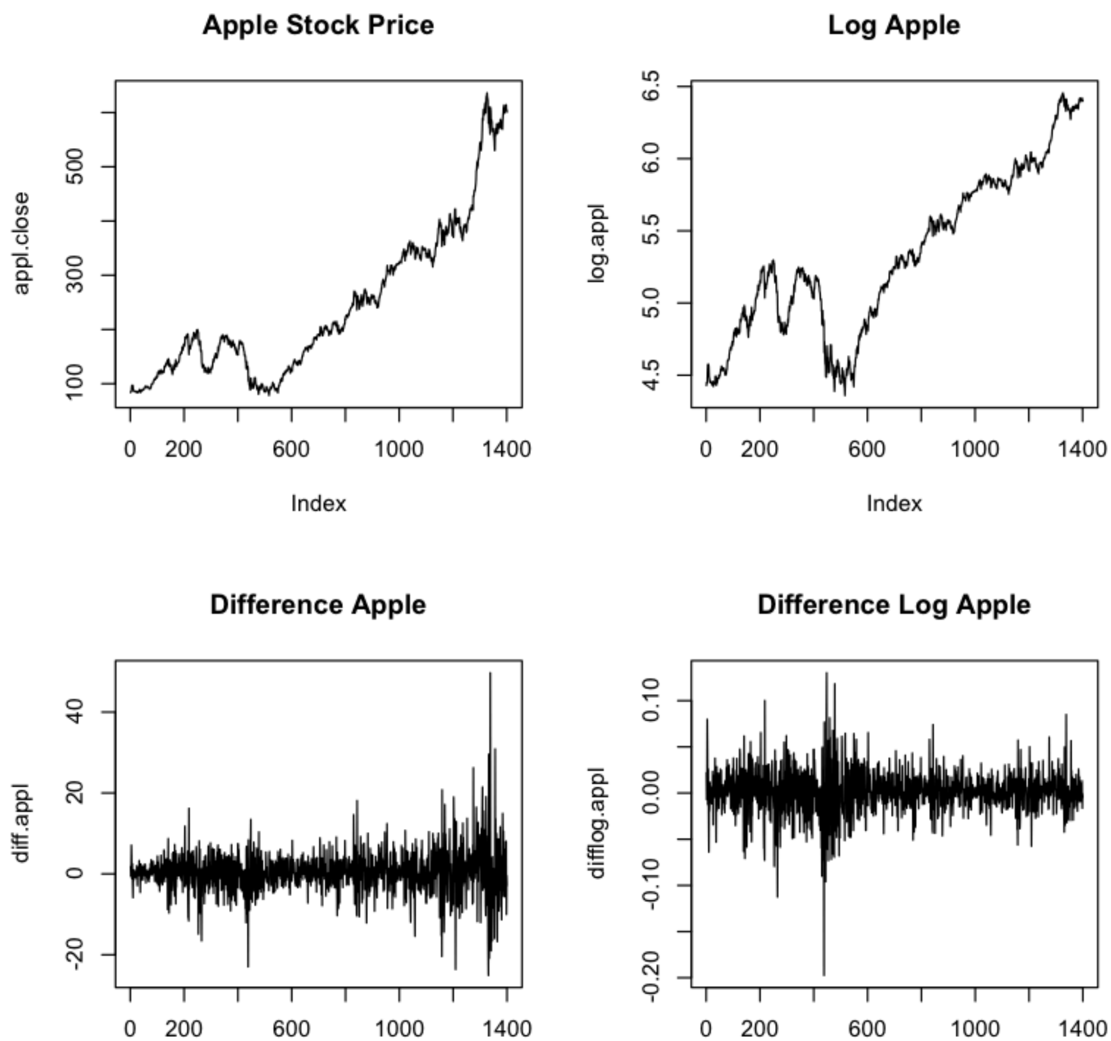
In financial time series, it is often that the series is transformed by logging and then the differencing is performed. This is because financial time series is usually exposed to exponential growth, and thus log transformation can smooth out (linearize) the series and differencing will help stabilize the variance of the time series. Following is an example of Apple stock price:

- The upper left graph is the original time series of Apple stock price from 01/01/2007 to 07/24/2012, showing exponential growth
- The lower left graph shows the differences of apple stock prices. It can be seen that the series is price-dependent; in other words, the variance of the series increases as the level of original series increases, and therefore, is not stationary
- The upper right corner show the graph of log price of Apple. The series is more linear compared to the original one.
- The lower right graphs the differences of log price of Apple. The series seems more mean-reverting, and variance is constant and does not significantly change as level of original series changes

To perform the differencing in R, follow these steps:

- Read the data file in R and store it in a variable
```
appl=read.csv('Apple.csv')
appl.close=appl$Adjclose #read and store adj close price in original file
```
- Plot original stock price
```
plot(appl.close,type='l',main='Apple Stock Price')
```
- Differencing the original series
```
diff.appl=diff(appl.close)
```

- Plot differences of original series
```
plot(diff.appl,type='l',main='Difference Apple')
```
- Take log of original series and plot the log price
```
log.appl=log(appl.close)
plot(log.appl,type='l',main='Log Apple')
```
- Differencing log price and plot
```
difflog.appl=diff(log.appl)
plot(difflog.appl,type='l',main='Difference Log Apple')
```



Another point that makes the differences of time series is more of interest than price series is that people often look at the returns of the stock rather than the its prices. Differences of log prices represent the returns and are similar to percentage changes of stock prices.

III. ARIMA model:

a. Model identification:

Time domain method is established and implemented by observing the autocorrelation of the time series. Therefore, autocorrelation and partial autocorrelation are the core of ARIMA model. Box-Jenkins method provides a way to identify ARIMA model according to autocorrelation and partial autocorrelation graph of the series. The parameters of ARIMA consist of three components: p (autoregressive parameter), d (number of differencing), and q (moving average parameters).

There are three rules to identify ARIMA model:

- If ACF (autocorrelation graph) cut off after lag n, PACF (partial autocorrelation graph) dies down: ARIMA(0, d, n) ➔ identify MA(q)
- If ACF dies down, PACF cut off after lag n: ARIMA(n, d, 0) ➔ identify AR(p)
- If ACF and PACF die down: mixed ARIMA model, need differencing

It is noted that the number of difference in ARIMA is written differently even though referring to the same model. For example, ARIMA(1,1,0) of the original series can be written as ARIMA(1,0,0) of the differenced series. Also, it is necessary to check for overdifferencing in which lag-1 autocorrelation is negative (usually less than -0.5). Overdifferencing can cause the standard deviation to increase.

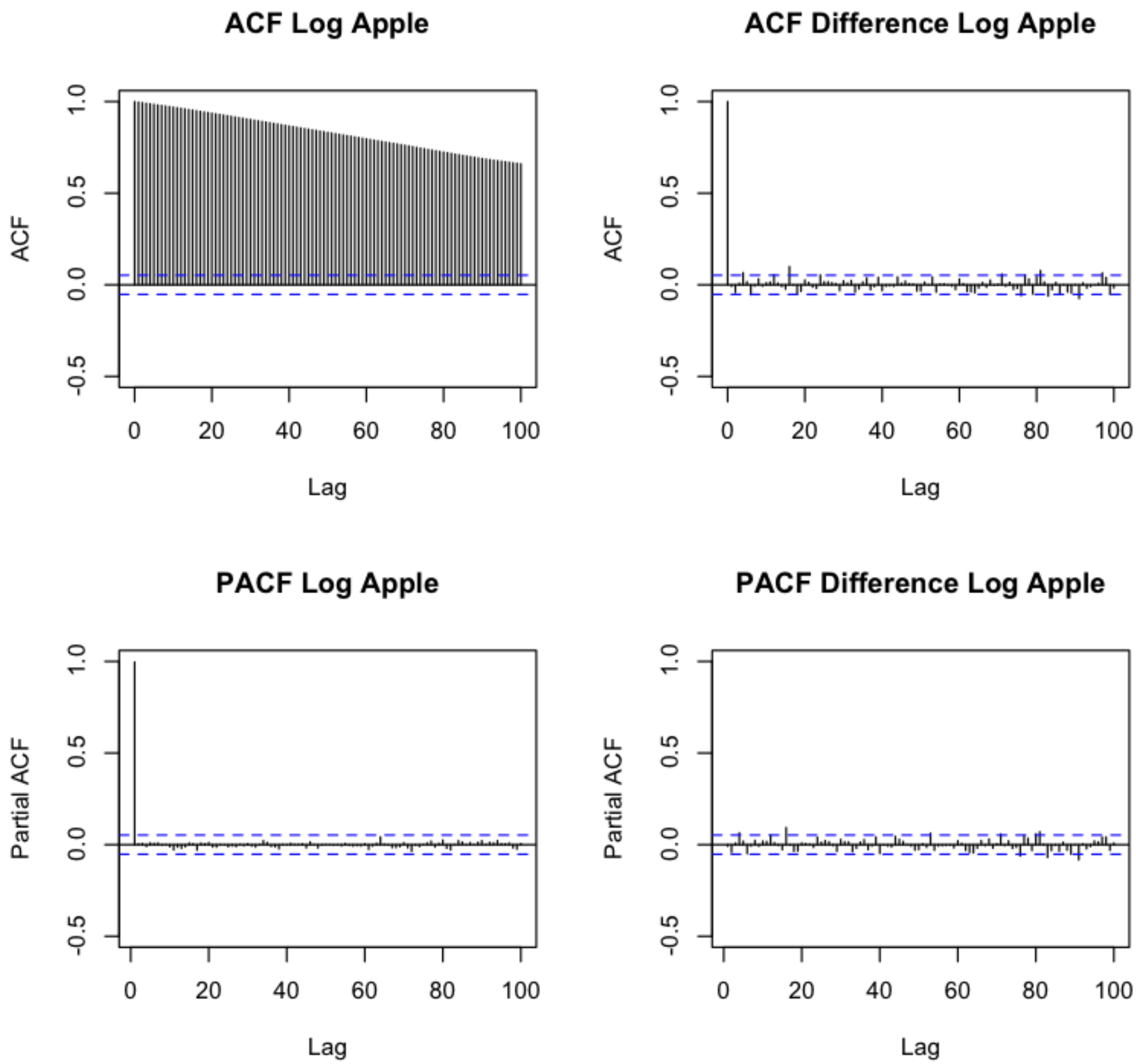
Following is an example from Apple time series:

- The upper left graphs the ACF of Log Apple stock price, showing the ACF slowly decreases (not dies down). It is probably that the model needs differencing.
- The lower left is PACF of Log Apple, indicating significant value at lag 1 and then PACF cuts off. Therefore, The model for Log Apple stock price might be ARIMA(1,0,0)
- The upper right shows ACF of differences of log Apple with no significant lags (do not take into account lag 0)
- The lower right is PACF of differences of log Apple, reflecting no significant lags. The model for differenced log Apple series is thus a white noise, and the original model resembles random walk model ARIMA(0,1,0)

In fitting ARIMA model, the idea of parsimony is important in which the model should have as small parameters as possible yet still be capable of explaining the series (p and q should be 2 or less, or the total number of parameters should be less than 3 in view of Box-Jenkins method[4]). The more parameters the greater noise that can be introduced into the model and hence standard deviation.

Therefore, when checking AICc for the model, one can check for model with p and q are 2 or less. To perform ACF and PACF in R, follow these codes:

- ACF and PACF of Log Apple
```
acf.appl=acf(log.appl,main='ACF Apple',lag.max=100,ylim=c(-0.5,1))
pacf.appl=pacf(log.appl,main='PACF Apple',lag.max=100,ylim=c(-0.5,1))
```
- ACF and PACF of Differenced log Apple
```
acf.appl=acf(difflog.appl,main='ACF Difference Log Apple',lag.max=100,ylim=c(-0.5,1))
pacf.appl=pacf(difflog.appl,main='PACF Difference Log Apple',lag.max=100,ylim=c(-0.5,1))
```



In addition to Box-Jenkins method, AICc provides another way to check and identify the model. AICc is corrected Akaike Information Criterion and can be calculated by the formula:

$$AIC_C = N*\log(SS/N) + 2(p + q + 1)*N/(N - p - q - 2), \text{ if no constant term in model}$$

$$AIC_C = N*\log(SS/N) + 2(p + q + 2)*N/(N - p - q - 3), \text{ if constant term in model}$$

N : the number of items after differencing ($N = n - d$)

SS : sum of squares of differences

p & q : the order of autoregressive and moving average model, respectively

According to this method, the model with lowest AICc will be selected. When perform time series analysis in R, the program will provide AICc as part of the result. However, in other software, it might be necessary to figure out the number manually by calculating the sum of squares and follow the above formula. The figures might be slightly different when different softwares are used yet the effect is small and the selected model can be reliable.

Model	AICc
0 1 0	-6493
1 1 0	-6491.02
0 1 1	-6493.02
1 1 1	-6489.01
0 1 2	-6492.84
1 1 2	-6488.89
2 1 0	-6491.1
2 1 1	-6489.14
2 1 2	-6501.86

Based on AICc, we should select ARIMA(2,1,2). The two methods might sometimes give different results, and therefore, it is necessary to check and test the model once we get all the estimates. Following is the code to perform ARIMA in R:

```
arima212=arima(log.appl,order=c(2,1,2))
summary(arima212)
```

Some notes about whether to include constant in ARIMA model[1]:

- d = 0: stationary model, constant should always be included and constant is the man of series
- d = 1: model has constant average trend, constant is included if the series shows any growth or deterministic trend. According to Box-Jenkins, when d > 0, constant should not be included except for series showing significant trend
- d = 2: model has time-varying trend, constant should not be included

b. Parameters estimation:

To estimate the parameters, implement the same code as previously shown. The result will provide the estimate of each element of the model.
Using ARIMA(2,1,2) as selected model, the result is as follows:

```
arima212
Series: log.appl
ARIMA(2,1,2)

Coefficients:
          ar1      ar2     ma1     ma2
      -0.0015  -0.9231  0.0032  0.8803
s.e.   0.0532   0.0400  0.0661  0.0488

sigma^2 estimated as 0.000559:  log likelihood=3255.95
             AIC=-6501.9   AICc=-6501.86   BIC=-6475.68
```

The full model:

$$(Y_t - Y_{t-1}) = -0.0015(Y_{t-1} - Y_{t-2}) - 0.9231(Y_{t-2} - Y_{t-3}) + 0.0032\varepsilon_{t-1} + 0.8803\varepsilon_{t-2} + \varepsilon_t$$

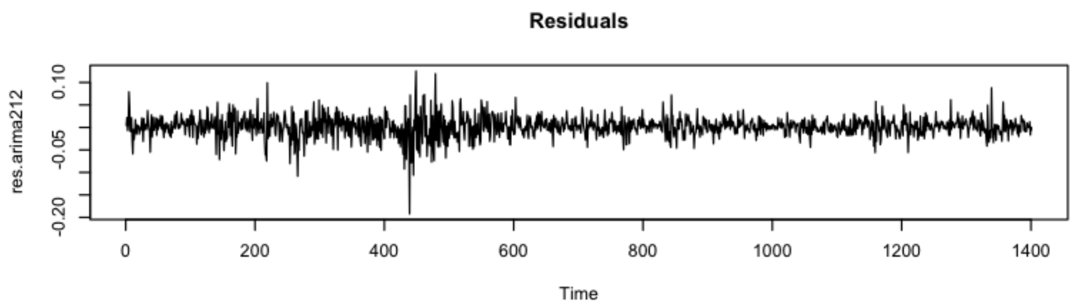
Noted that R will ignore the mean when performing ARIMA model with differencing. This might be one disadvantage to use R compared to Minitab. Following is the output from Minitab for the same model:

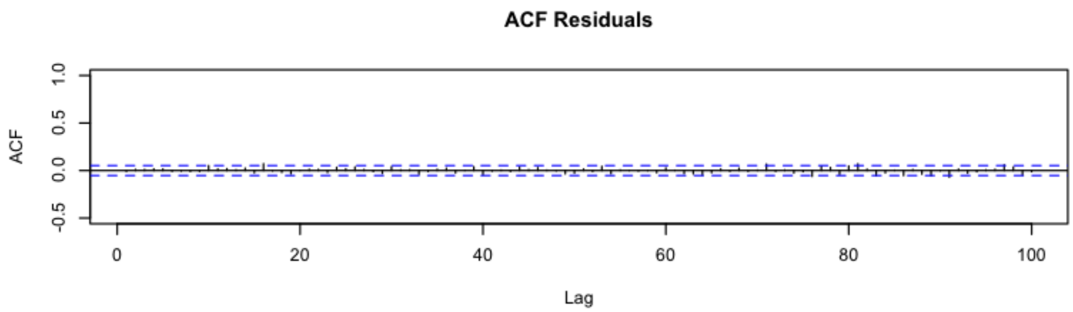
```
Final Estimates of Parameters

Type          Coef   SE Coef       T      P
AR   1      0.0007    0.0430    0.02  0.988
AR   2     -0.9259    0.0640  -14.47  0.000
MA   1      0.0002    0.0534    0.00  0.998
MA   2     -0.8829    0.0768  -11.50  0.000
Constant  0.002721  0.001189    2.29  0.022


Differencing: 1 regular difference
Number of observations:  Original series 1401, after differencing 1400
Residuals:    SS =  0.779616 (backforecasts excluded)
              MS =  0.000559  DF = 1395


Modified Box-Pierce (Ljung-Box) Chi-Square statistic

Lag            12     24     36     48
Chi-Square    6.8   21.2   31.9   42.0
DF              7     19     31     43
P-Value     0.452  0.328  0.419  0.516
```

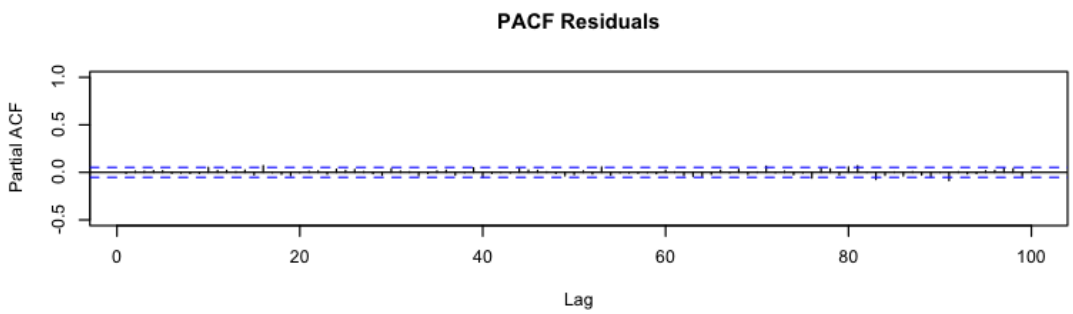
Note that R will give different estimates for the same model depending on how we write the code. For example:
arima(log.appl,order=c(2,1,2))
arima(difflog.appl,order=c(2,0,2))
The resulting parameter estimates of ARIMA(2,1,2) from those two code lines will be different in R even though it refers to the same model. However, in Minitab, the result is similar, and thus less confusing to users.

c. Diagnostic checking:

The procedure includes observing residual plot and its ACF & PACF diagram, and check Ljung-Box result.
If ACF & PACF of the model residuals show no significant lags, the selected model is appropriate.







The residual plot, ACF and PACF do not have any significant lag, indicating ARIMA(2,1,2) is a good model to represent the series.
In addition, Ljung-Box test also provides a different way to double check the model. Basically, Ljung-Box is a test of autocorrelation in which it verifies whether the autocorrelations of a time series are different from 0. In other words, if the result rejects the hypothesis, this means the data is independent and uncorrelated; otherwise, there still remains serial correlation in the series and the model needs modification.

```
Modified Box-Pierce (Ljung-Box) Chi-Square statistic

Lag             12     24     36     48
Chi-Square     6.8   21.2   31.9   42.0
DF               7     19     31     43
P-Value      0.452  0.328  0.419  0.516
```

The output from Minitab show that p-values are all greater than 0.05, so we cannot reject the hypothesis that the autocorrelation is different from 0. Therefore, the selected model is an appropriate one of Apple stock price.

IV. ARCH/GARCH model:

Although ACF & PACF of residuals have no significant lags, the time series plot of residuals shows some cluster of volatility. It is important to remember that ARIMA is a method to linearly model the data and the forecast width remains constant because the model does not reflect recent changes or incorporate new information. In other words, it provides best linear forecast for the series, and thus plays little role in forecasting model nonlinearly. In order to model volatility, ARCH/GARCH method comes into play. How do we know if ARCH/GARCH is necessary for the times series in concern?

Firstly, check if residual plot displays any cluster of volatility. Next, observe the squared residual plot. If there are clusters of volatility, ARCH/GARCH should be used to model the volatility of the series to reflect more recent changes and fluctuations in the series. Finally, ACF & PACF of squared residuals will help confirm if the residuals (noise term) are not independent and can be predicted. As mentioned earlier, a strict white noise cannot be predicted either linearly or nonlinearly while general white noise might not be predicted linearly yet done so nonlinearly. If the residuals are strict white noise, they are independent with zero mean, normally distributed, and ACF & PACF of squared residuals displays no significant lags.

Followings are the plots of squared residuals:

- Squared residuals plot shows cluster of volatility at some points in time
- ACF seems to die down
- PACF cuts off after lag 10 even though some remaining lags are significant

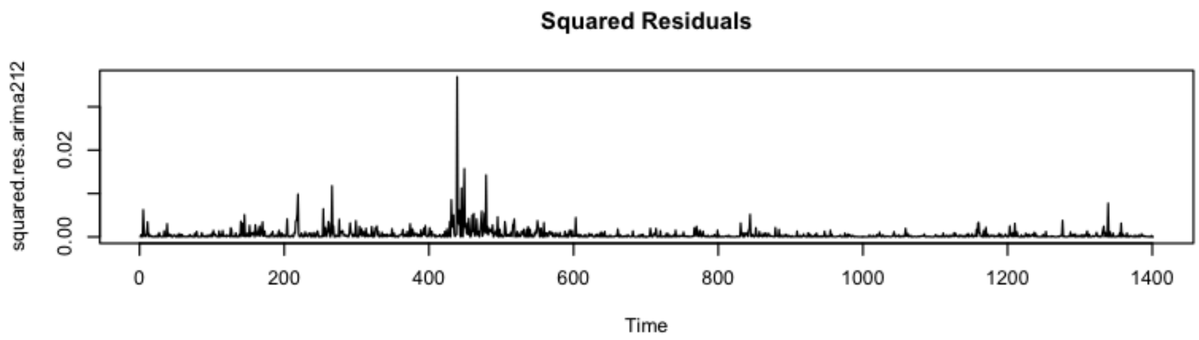
The residuals therefore show some patterns that might be modeled.
ARCH/GARCH is necessary to model the volatility of the series. As indicated by its name, this method concerns with the conditional variance of the series.
The general form of ARCH(q):

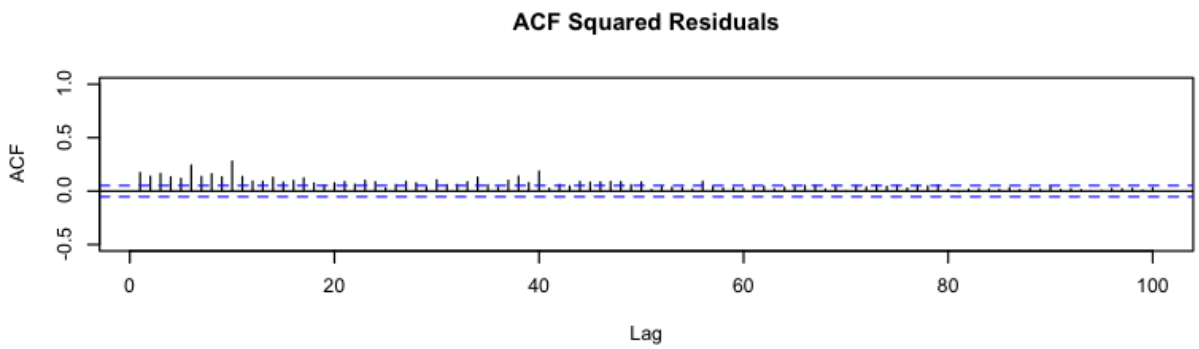
$\varepsilon_t \mid \psi_{t-1} \sim N(0,h_t)$

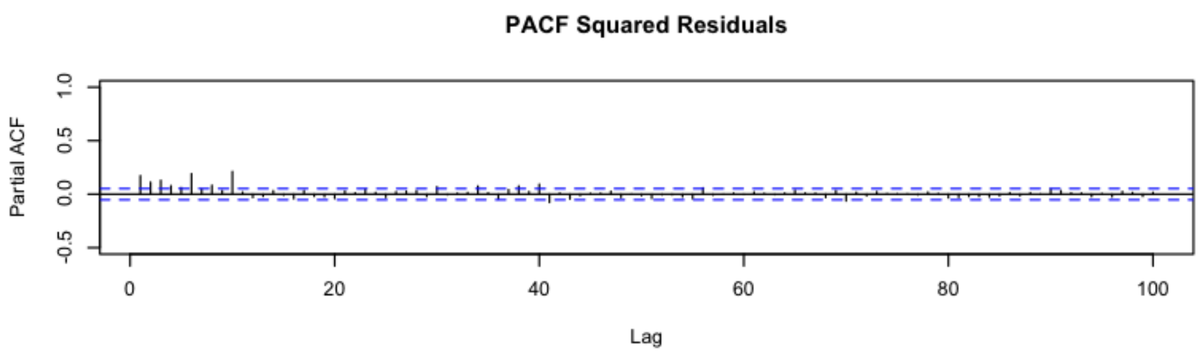
$h_t = \omega + \Sigma\alpha_i\varepsilon^2_{t-i}$

Codes to generate result for Squared Residuals:

```
res.arima212=arima212$res
squared.res.arima212=res.arima212^2
par(mfcol=c(3,1))
plot(squared.res.arima212,main='Squared Residuals')

acf.squared212=acf(squared.res.arima212,main='ACF Squared
Residuals',lag.max=100,ylim=c(-0.5,1))
pacf.squared212=pacf(squared.res.arima212,main='PACF Squared
Residuals',lag.max=100,ylim=c(-0.5,1))
```







ARCH/GARCH orders and parameters are selected based on AICc as follows:

AIC_C = -2*Log likelihood + 2*(q+1)*(N/(N-q-2), no constant

AIC_C = -2*Log likelihood + 2*(q+2)*(N/(N-q-3), with constant

N : the sample size after differencing

q : order of autoregressive

To compute AICc, we need to fit ARCH/GARCH model to the residuals and then calculate the log likelihood using logLik() function in R and follow the formula above. Noted that we fit ARCH to the residuals from ARIMA model selected previously, not to the original series or log or differenced log series because we only want to model the noise of ARIMA model.

Model	N	q	Log likelihood	AICc no const	AICc const
ARCH(0)	1400	0	3256.488	-6510.973139	-6508.96741
ARCH(1)	1400	1	3314.55	-6625.09141	-6623.082808
ARCH(2)	1400	2	3331.168	-6656.318808	-6654.307326
ARCH(3)	1400	3	3355.06	-6702.091326	-6700.076958

ARCH(4)	1400	4	3370.881	-6731.718958	-6729.701698
ARCH(5)	1400	5	3394.885	-6777.709698	-6775.68954
ARCH(6)	1400	6	3396.683	-6779.28554	-6777.262477
ARCH(7)	1400	7	3403.227	-6790.350477	-6788.324504
ARCH(8)	**1400**	**8**	**3410.242**	**-6802.354504**	**-6800.325613**
ARCH(9)	1400	9	3405.803	-6791.447613	-6789.415798
ARCH(10)	1400	10	3409.187	-6796.183798	-6794.149054
GARCH(1, 1)	1400	2	3425.365	-6844.712808	-6842.701326

The table of AICc is provided above for both constant and non-constant cases. Note the decrease in AICc from ARCH(1) to ARCH(8) and then AICc increases in ARCH(9) and ARCH(10). Why does it happen? This is a signal that we need to check for the convergence of our model. In the first 7 cases, the output in R gives "Relative Function Convergence" while ARCH(9) and ARCH(10) have "False Convergence." When the output contains False convergence, the predictive capability of the model is doubted, and we should exclude these model from our selection. Although GARCH(1,1) also has the lowest AICc, the model is falsely converged, and thus excluded. Therefore, ARCH(8) is the selected model.

Moreover, we also include ARCH(0) in the analysis because it can serve as a check to see if there are any ARCH effects or the residuals are independent[6].

R codes to perform ARCH/GARCH model:

```
arch08=garch(res.arima212,order=c(0,8),trace=F)
loglik08=logLik(arch08)
summary(arch08)
```

Note that R will not allow the order of q = 0, and so we cannot get the log likelihood for ARCH(0) from R; yet we need to compute it by the formula[7]:

–.5*N*(1+log(2*pi*mean(x^2)))

N: number of observations after differencing N = n – d

X: the data set in consideration (in this case, the residuals)

The output for ARCH(8):

```
summary(arch08)
Call:
garch(x = res.arima212, order = c(0, 8), trace = F)

Model:
GARCH(0,8)

Residuals:
     Min       1Q   Median       3Q      Max
-4.40329 -0.48569  0.08897  0.69723  4.07181

Coefficient(s):
    Estimate  Std. Error  t value Pr(>|t|)
```

```
a0 1.472e-04   1.432e-05   10.282  < 2e-16 ***
a1 1.284e-01   3.532e-02    3.636 0.000277 ***
a2 1.335e-01   2.839e-02    4.701 2.59e-06 ***
a3 9.388e-02   3.688e-02    2.545 0.010917 *
a4 8.678e-02   2.824e-02    3.073 0.002116 **
a5 5.667e-02   2.431e-02    2.331 0.019732 *
a6 3.972e-02   2.383e-02    1.667 0.095550 .
a7 9.034e-02   2.907e-02    3.108 0.001885 **
a8 1.126e-01   2.072e-02    5.437 5.41e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Diagnostic Tests:
 Jarque Bera Test

data:  Residuals
X-squared = 75.0928, df = 2, p-value < 2.2e-16


 Box-Ljung test

data:  Squared.Residuals
     X-squared = 0.1124, df = 1, p-value = 0.7374
```

The p-values for all parameters are less than 0.05 (except for 6th parameter), indicating that they are statistically significant. In addition, p-value of Box-Ljung test is greater than 0.05, and so we cannot reject the hypothesis that the autocorrelation of residuals is different from 0. The model thus adequately represents the residuals.

Full ARCH(8) model:

$$h_t = 1.472e\text{-}04 + 1.284e\text{-}01\varepsilon^2_{t-1} + 1.335e\text{-}01\varepsilon^2_{t-2} + 9.388e\text{-}02\varepsilon^2_{t-3} + 8.678e\text{-}02\varepsilon^2_{t-4} + 5.667e\text{-}02\varepsilon^2_{t-5} + 3.972e\text{-}02\varepsilon^2_{t-6} + 9.034e\text{-}02\varepsilon^2_{t-7} + 1.126e\text{-}01\varepsilon^2_{t-8}$$

V. ARIMA-ARCH/GARCH performance:

In this section, we will compare the results from ARIMA model and the combined ARIMA-ARCH/GARCH model. As selected earlier, ARIMA and ARCH model for Apple Log price series are ARIMA(2,1,2) and ARCH(8), respectively. Moreover, we will also look at the result from Minitab and compare it with that from R.

Remember R will exclude constant when fitting ARIMA for series needed differencing. Therefore, our result as previously generated from R is ARIMA(2,1,2) without constant. Using forecast() function, the 1-step forecast for the series under ARIMA(2,1,2)

```
Point       Forecast    Lo 95    Hi 95
1402        6.399541 6.353201 6.445882
```

Full model of ARIMA(2,1,2) – ARCH(8):

$(Y_t - Y_{t-1}) = -0.0015(Y_{t-1} - Y_{t-2}) - 0.9231(Y_{t-2} - Y_{t-3}) + 0.0032\varepsilon_{t-1} + 0.8803\varepsilon_{t-2} + 1.472e\text{-}04 + 1.284e\text{-}01\varepsilon^2_{t-1} + 1.335e\text{-}01\varepsilon^2_{t-2} + 9.388e\text{-}02\varepsilon^2_{t-3} + 8.678e\text{-}02\varepsilon^2_{t-4} + 5.667e\text{-}02\varepsilon^2_{t-5} + 3.972e\text{-}02\varepsilon^2_{t-6} + 9.034e\text{-}02\varepsilon^2_{t-7} + 1.126e\text{-}01\varepsilon^2_{t-8}$

Following is the table summarizing all models with their point forecast and forecast interval edited and computed in Excel:

		95% Confident interval		
Model	**Forecast**	**Lower**	**Upper**	**Actual**
ARIMA(2,1,2) in R	6.399541	6.353201	6.445882	6.354317866
ARIMA(2,1,2) in Minitab (constant)	6.40099	6.35465	6.44734	
ARIMA(2,1,2) in Minitab (no constant)	6.39956	6.35314	6.44597	
ARIMA(2,1,2) + ARCH(8) in R	**6.39974330**	**6.35340330**	**6.44608430**	
ARIMA(2,1,2) in Minitab (constant) +ARCH(8)	6.40119230	6.35485230	6.44754230	
ARIMA(2,1,2) in Minitab (no constant) +ARCH(8)	6.39976230	6.35334230	6.44617230	

Converting Log Price to Price, we obtain the forecast for original series:

		95% Confident interval		
Model	**Forecast**	**Lower**	**Upper**	**Actual**
ARIMA(2,1,2) in R	601.5688544	574.3281943	630.1021821	574.9700003
ARIMA(2,1,2) in Minitab (constant)	602.4411595	575.1609991	631.0215411	
ARIMA(2,1,2) in Minitab (no constant)	601.5802843	574.2931614	630.1576335	
ARIMA(2,1,2) + ARCH(8) in R	**601.6905666**	**574.4443951**	**630.2296673**	
ARIMA(2,1,2) in Minitab (constant) +ARCH(8)	602.5630482	575.2773683	631.1492123	
ARIMA(2,1,2) in Minitab (no constant) +ARCH(8)	601.7019989	574.409355	630.28513	

The actual price was obtained on 07/25/2012 when Apple released its earnings report which was lower than estimate, and this announcement has affected the company stock price, causing the stock to drop from $600.92 on 07/24/2012 to $574.97 on the next day. This is an unexpected risk that often happens when companies release positive or negative news. However, our model seems to successfully forecast that risk since the actual price is within our 95% confident interval and very close to the lower limit.

It is noted that the 95% confident interval of ARIMA(2,1,2) is wider than that of the combined model ARIMA(2,1,2) – ARCH(8). This is because the latter reflects and incorporate recent changes and volatility of stock prices by analyzing the residuals and its conditional variances (the variances affected as new information comes in).

So how to compute conditional variance ht of ARCH(8)? The easiest way is to use Excel and put all number we have from R. Before discussing how to perform in Excel, it is necessary to generate codes that will provide the information we need:

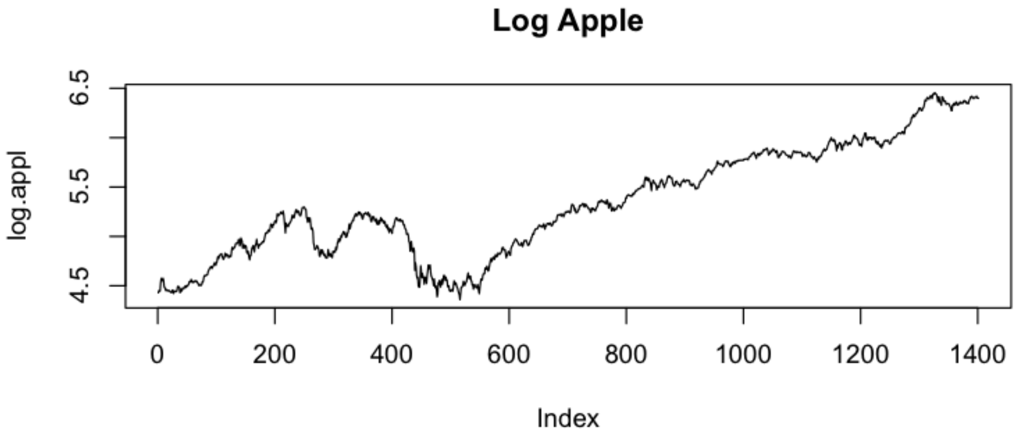
- Generate 1-step forecast, 100-step forecast, and plot of forecast:
```
forecast212step1=forecast(arima212,1,level=95)
forecast212=forecast(arima212,100,level=95) plot(forecast212)
```
- Compute ht, conditional variance:
```
ht.arch08=arch08$fit[,1]^2 #use 1st column of fit
plot(ht.arch08,main='Conditional variances')
```
- Generate plot of Log Price, 95% Upper and Lower limit
```
fit212=fitted.values(arima212)
low=fit212-1.96*sqrt(ht.arch08)
high=fit212+1.96*sqrt(ht.arch08)
plot(log.appl,type='l',main='Log Apple,Low,High')
lines(low,col='red')
lines(high,col='blue')
```

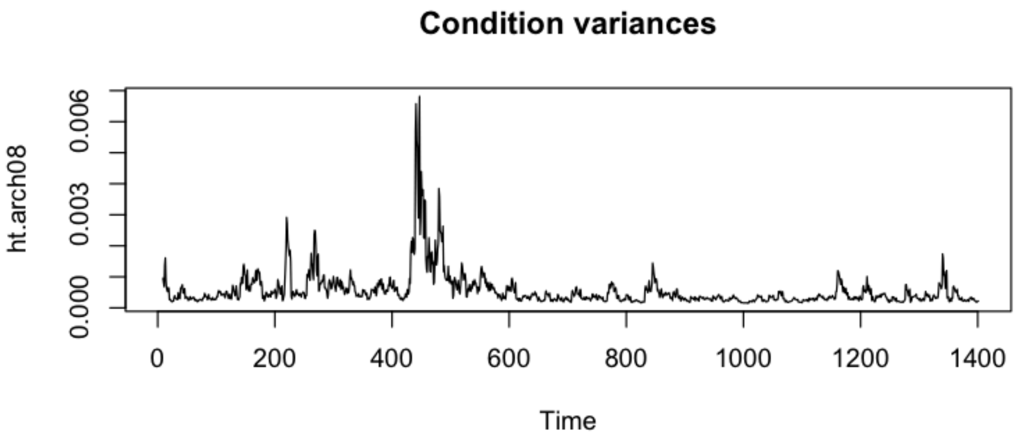
To compute ht, we first list all parameters of the model in one column, and then look for the residuals that are associated with these coefficients, square these residuals, multiply ht coefficients by squared residuals, and sum up those figures to get ht. For example, to estimate point 1402 (our data set has 1401 observations), we need residuals for the last 8 days because our model is ARCH(8). Following is the generated table from Excel:

	ht coeff	res	squared res	ht components
const	1.47E-04			1.47E-04
a1	1.28E-01	-5.18E-03	2.69E-05	3.45E-06
a2	1.34E-01	4.21E-04	1.77E-07	2.37E-08
a3	9.39E-02	-1.68E-02	2.84E-04	2.66E-05
a4	8.68E-02	1.25E-02	1.57E-04	1.36E-05
a5	5.67E-02	-7.41E-04	5.49E-07	3.11E-08
a6	3.97E-02	8.33E-04	6.93E-07	2.75E-08
a7	9.03E-02	2.92E-03	8.54E-06	7.72E-07
a8	1.13E-01	9.68E-03	9.37E-05	1.05E-05
			ht	2.02E-04

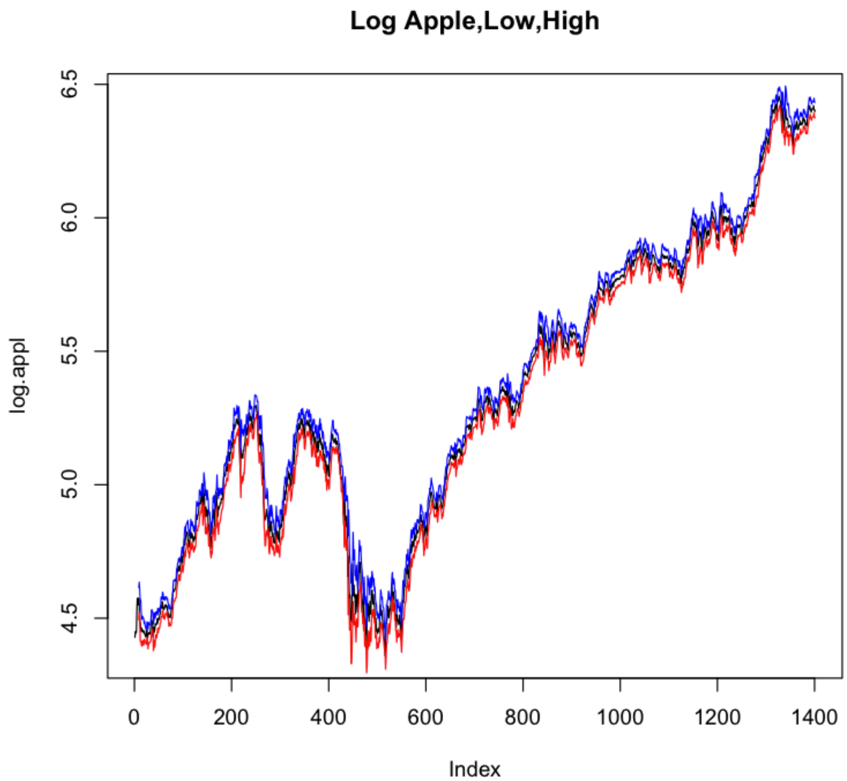
To estimate the 1—step forecast and 95% confident interval of the mixed model as aforementioned, we use the ARIMA forecast obtained from R or Minitab, and then add ht to ARIMA forecast.
The Log Price and condition variances are plotted:

- The conditional variances plot successfully reflects the volatility of the time series over the entire period
- High volatility is closely related to period where stock price tumbled



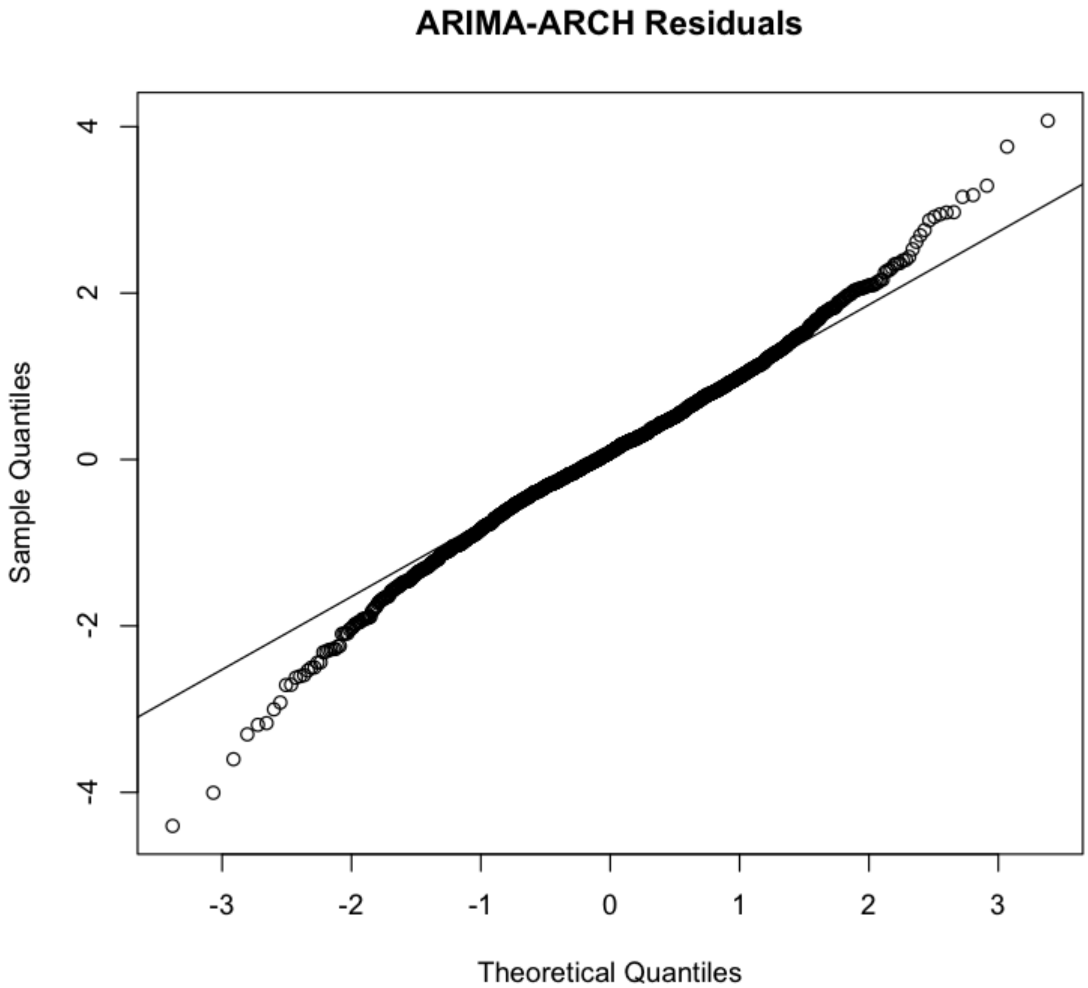


The 95% forecast interval of Log price:

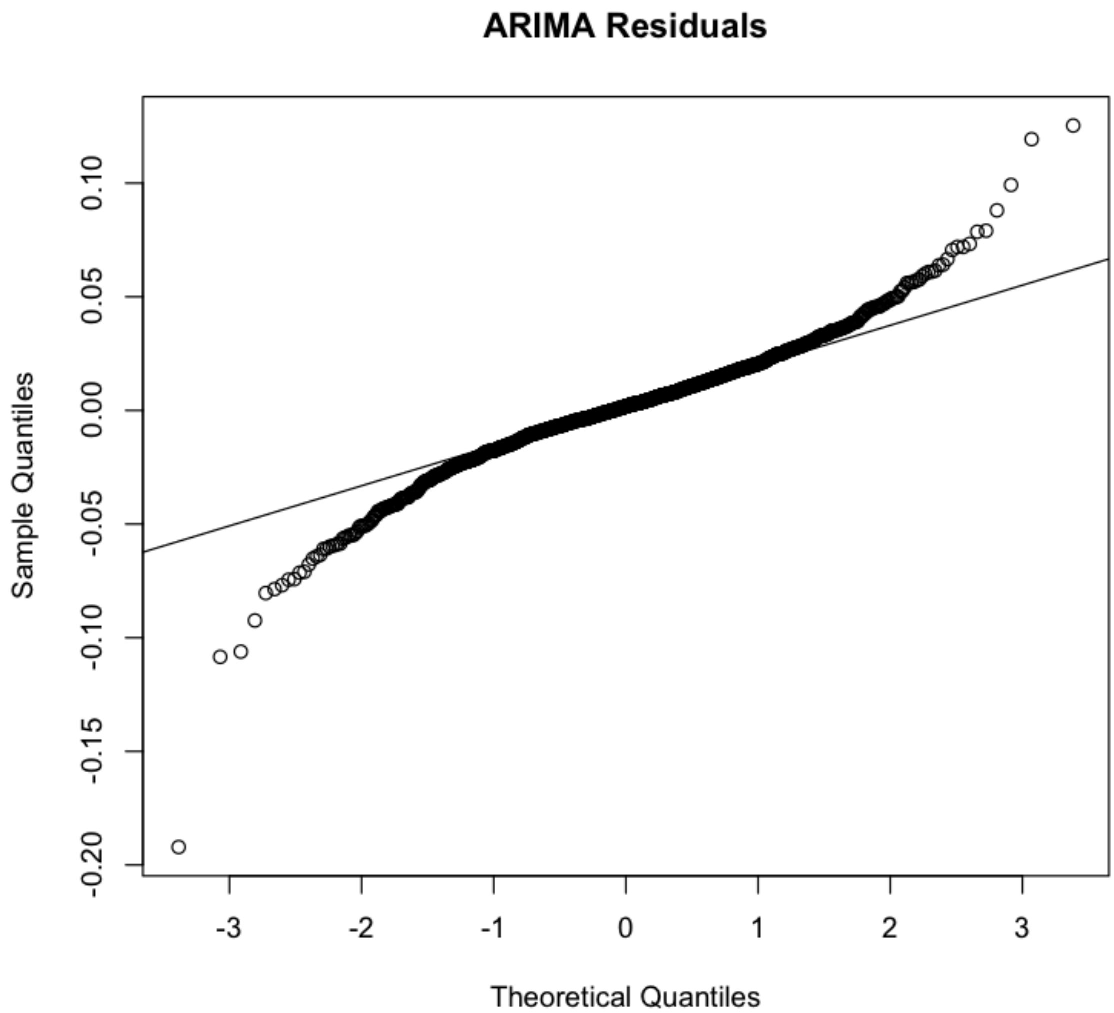


The final check on the model is to look at Q-Q Plot of residuals of ARIMA-ARCH model, which is $e_t = \varepsilon_t/\text{sqrt}(h_t)$ =Residuals/sqrt(Conditional variance). We can compute it directly from R, and plot the Q-Q graph to check the normality of the residuals. Followings are the codes and Q-Q plot:

```
archres=res.arima212/sqrt(ht.arch08)
qqnorm(archres,main='ARIMA-ARCH Residuals')
qqline(archres)
```



The plot shows that residuals seem to be roughly normally distributed although some points remain off the line. However, compared to residuals of ARIMA model, those of mixed model are more normally distributed.



VI. Conclusion:

Time domain method is a useful way to analyze the financial time series. There are some points in forecasting based on ARIM-ARCH/GARCH model that need to take into account.

Firstly, ARIMA model focuses on analyzing time series linearly and it does not reflect recent changes as new information is available. Therefore, in order to update the model, users need to incorporate new data and estimate parameters again. The variance in ARIMA model is unconditional variance and remains constant. ARIMA is applied for stationary series and therefore, non-stationary series should be transformed (such as log transformation).

Additionally, ARIMA is often used together with ARCH/GARCH model. ARCH/GARCH is a method to measure volatility of the series, or more specifically, to model the noise term of ARIMA model. ARCH/GARCH incorporates new information and analyzes the series based on conditional variances where users can forecast future values with up-to-date information. The forecast interval for the mixed model is closer than that of ARIMA-only model.

REFERENCES

Hurvich, Clifford. “Best Linear Forecast vs. Best Optimal Forecast.” New York University. New York, NY. n.d.

Nau, Bob. “Stationary and Differencing.” Duke University. Durham, NC. n.d.

Jong, Robert. “Stationary Time Series.” Ohio State University. Columbus, OH. n.d.