Time Series Analysis of Microsoft Stock Price Return

Yu Liu
Zhaolong Lu

Content

1. Goal and Methodology

Acquisitions has been a common method for tech companies of huge scale to absorb new ideas and enrich market shares. However, it is not a sustainable supportive factor for generating profits. Big tech companies also have higher beta and are sensitive to market elements. So we want to research the risk in terms of their stock price behaviors and our target company is Microsoft.

The purpose of this project is to build a statistical model of stock returns of Microsoft, and to use the predicted mean and volatility to measure its investment risk. With R software, we collect the data of weekly adjusted closing price of Microsoft from finance.yahoo.com and use the log return to smooth dataset. Then we explore the descriptive statistics and analyze the ACF, PACF and ARCH effects of log returns. We try and test several ARMA and GARCH models in order to find the one with significant coefficients (significant level is 0.05) and the lowest AIC and BIC values. By checking its standard residuals and squared standard residuals and refitting models with out-of-sample method, we make one steps (one week) ahead forecast and get VaR for the next week with different confidence levels.

2. Exploratory Data Analysis

2.1 Introduction of Data

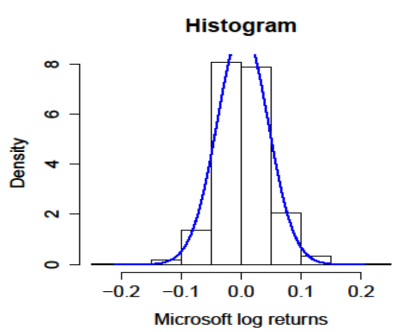
With an interest to explore the overall time trend of Microsoft, we cite data from finance.yahoo.com and choose the weekly adjusted closing price from 1994 to 2014. The adjusted closing price can efficiently reflect the true value of stocks which excludes the effect of dividends and stock split. During the period of internet bubble the data shows great volatilities, but we still keep an eye on this part because of the outliers also communicate the important return information.

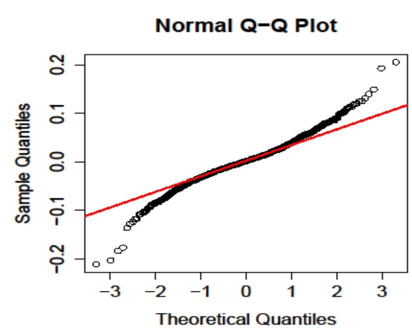
2.2 Exploratory Data Analysis

2.1.1 Descriptive statistic of the Data(Original and Log return)

The plot given on the left presenting a genral trend of the stock price of Microsoft during the time period from 1994 to 2014. In order to make the data smootheier, we decided to use the log data. On the right, we present the plot of the INRETURN with the time range from 1994 to 2014.

```
                     x
nobs       1042.000000
NAs           0.000000
Minimum      -0.211123
Maximum       0.205199
1. Quartile  -0.019045
3. Quartile   0.024365
Mean          0.002668
Median        0.001551
Sum           2.780487
SE Mean       0.001311
LCL Mean      0.000096
UCL Mean      0.005241
Variance      0.001791
Stdev         0.042319
Skewness     -0.074312
Kurtosis      2.760867
```

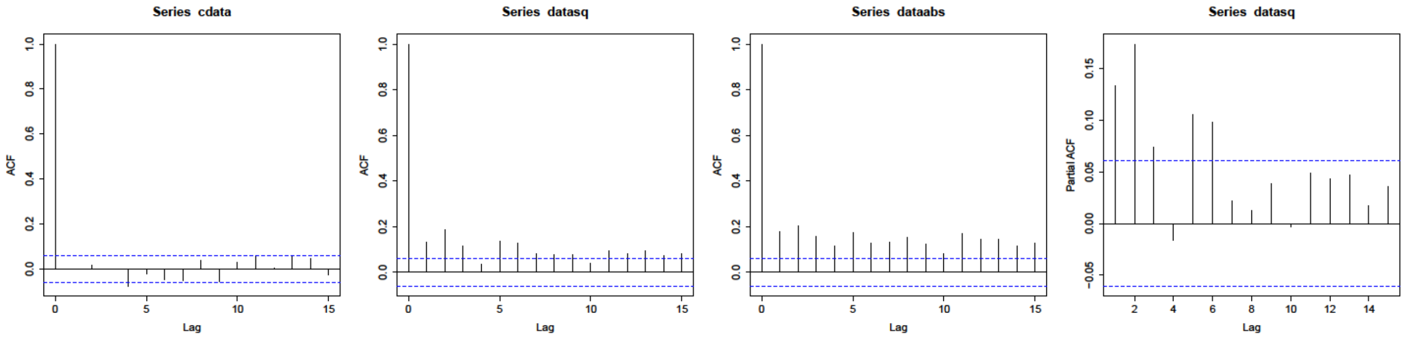




	Skewness	*Kurtosis*	*JB*
Value	**-0.98**	**18.19**	**Xsquared:334.56**
p-value	**0.327**	**0**	**2.2e-16**

The parameters of the Microsoft stock price Log return present the conclusion as below: the data has fat tail, it is distributed symmetrically and it is not normal distribution. (The kurtosis =18.19, the skewness=-0.98, Asymptotic p value: 2.2e-16).

2.1.2 Autocorrelation and partial autocorrelation analysis



The ACF analysis is quite intuitive as the R outputs showed above. Orderly from the left to right, there are ACF of original data, the ACF of squared data, the ACF of the absolute value of the data, the PACF of the squared original data. The Original data presents a small ACF value but both the squared data and the absolute value of the data shows non zero acf values indicating a significant ARCH effect. The PACF of squared data plot give the hint that the AR model would begin with Order (2,0) as the lag2 presents a significance.

<u>As the analysis above, this dataset has ARCH effect, which can be explained by the GARCH model. Also, the PACF plot of log returns has serial correlation order of 2, so we started to fit an ARMA(2,0)-GARCH(1,1) model.</u>

3. Model fitting

As for many models we would selected, I want to clarify our mind flow here:
We would like to start from the AR model selection using the GARCH(1,1) model as a constant variable, keep subtracting the parameters from the model for reasons such as inadequate and no-significant.

3.1 Fitting a Best ARMA Model

ARMA(2,0) ~ GARCH(1,1) with normal distribution

Started with an ARMA(2,0)~GARCH(0,0) model, we eliminated one insignificant coefficient each time (Details please refer to Appendix 2) until all coefficients left in the model are significant. After eliminated ar2, ar1, we got this model as:

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

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

Optimal Parameters
------------------------------------
        Estimate  Std. Error   t value Pr(>|t|)
mu      0.003143    0.001086  2.892626 0.003820
ar1    -0.024565    0.032490 -0.756073 0.449605
ar2     0.000631    0.032331  0.019512 0.984432
omega   0.000032    0.000012  2.599145 0.009346
alpha1  0.064001    0.014945  4.282359 0.000018
beta1   0.917722    0.019216 47.759029 0.000000
```

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

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

Optimal Parameters
------------------------------------
        Estimate  Std. Error  t value Pr(>|t|)
mu      0.003142    0.001086  2.89424 0.003801
ar1    -0.024580    0.032482 -0.75672 0.449218
ar2     0.000000          NA       NA       NA
omega   0.000032    0.000012  2.60417 0.009210
alpha1  0.063990    0.014913  4.29092 0.000018
beta1   0.917748    0.019162 47.89340 0.000000
```

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

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

Optimal Parameters
------------------------------------
        Estimate  Std. Error  t value Pr(>|t|)
mu      0.003152    0.001111   2.8372 0.004551
ar1     0.000000          NA       NA       NA
ar2     0.000000          NA       NA       NA
omega   0.000032    0.000012   2.6136 0.008960
alpha1  0.064057    0.014869   4.3081 0.000016
beta1   0.917688    0.019065  48.1355 0.000000
```

(1)parameter estimate		mu	omega	alpha1	beta1
	estimate	0.003152	0.0.000032	0.064057	0.0.917688
(2)adjusted pearson goodness-of-fit test	group	20	30	40	50
	p value	0.012666	0.002448	0.006625	0.049909

Current conclusion on the ARMA model:

<u>So far, we can make the conclusion that the ARMA(2,0) model seems not very good to continue with further analysis with the GARCH since the insignificant parameters. With a better ARMA(0,0) model, we would fit relatively best GARCH model with the ARMA(2,0) model for the best result of the prediction and forecast in the following part.</u>

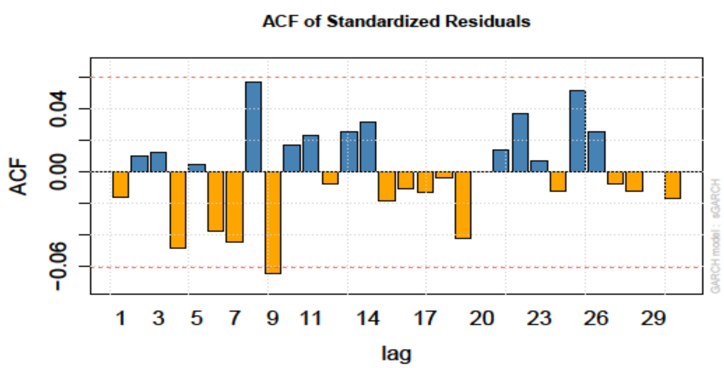
3.2 Fitting a Best GARCH Model

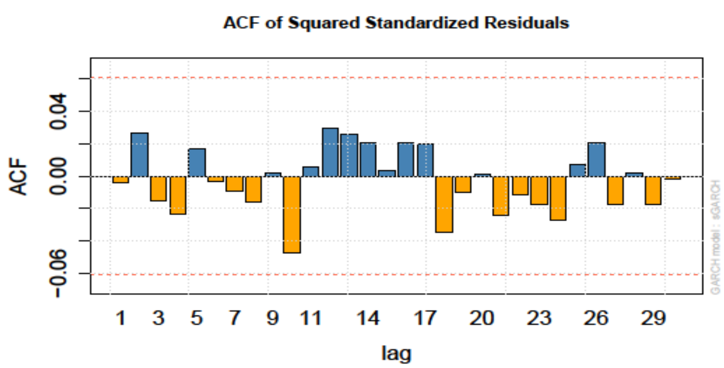
3.2.1 ARMA(0,0)~ GARCH(1,1) with normal distribution

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

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

Optimal Parameters
------------------------------------
        Estimate  Std. Error  t value Pr(>|t|)
mu      0.003161    0.001111   2.8446 0.004447
omega   0.000032    0.000012   2.6150 0.008924
alpha1  0.064022    0.014844   4.3130 0.000016
beta1   0.917676    0.019060  48.1471 0.000000
```

```
Weighted Ljung-Box Test on Standardized Residuals
------------------------------------
                        statistic p-value
Lag[1]                     0.2601  0.6100
Lag[2*(p+q)+(p+q)-1][2]    0.3122  0.7890
Lag[4*(p+q)+(p+q)-1][5]    1.4286  0.7575
d.o.f=0
H0 : No serial correlation

Weighted Ljung-Box Test on Standardized Squared Residuals
------------------------------------
                        statistic p-value
Lag[1]                    0.01414  0.9053
Lag[2*(p+q)+(p+q)-1][5]   1.04469  0.8495
Lag[4*(p+q)+(p+q)-1][9]   1.51214  0.9546
d.o.f=2
```

```
Adjusted Pearson Goodness-of-Fit Test:
------------------------------------
  group statistic p-value(g-1)
1    20     35.31     0.012803
2    30     54.87     0.002565
3    40     64.03     0.006982
4    50     67.21     0.043006
```






```
Weighted Ljung-Box Test on Standardized Residuals
------------------------------------
                        statistic p-value
Lag[1]                     0.2076  0.6487
Lag[2*(p+q)+(p+q)-1][2]    0.2658  0.8151
Lag[4*(p+q)+(p+q)-1][5]    1.2224  0.8076
d.o.f=0
H0 : No serial correlation

Weighted Ljung-Box Test on Standardized Squared Residuals
------------------------------------
                        statistic p-value
Lag[1]                     0.2356  0.6274
Lag[2*(p+q)+(p+q)-1][5]    1.0083  0.8578
Lag[4*(p+q)+(p+q)-1][9]    1.6868  0.9389
d.o.f=2
```

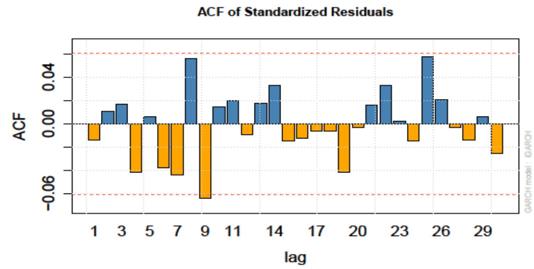
<u>As we can see that the residuals of this model is fine as both the ACF of the residual and the squared residuals present to be white noise. Here we can also capture an information that $\alpha_1 + \beta_1 = 0.9817$, which is very much close to 1. However, we still have to check the residuals to make the conclusion. However, what we are sure about is that the residual is not normal distributed since the p-value is relatively small.</u>

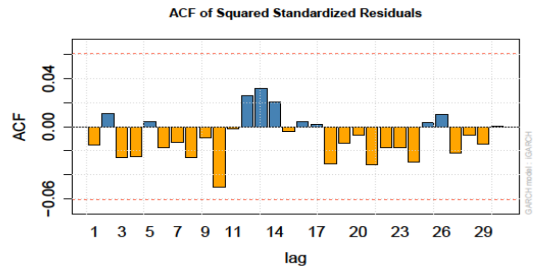
3.2.2 Following IGARCH models are going to be compared

ARMA(0,0)~ IGARCH(1,1) with normal distribution
ARMA(0,0)~ IGARCH(1,1) with t-distribution
ARMA(0,0)~ IGARCH(1,1) with skewed t-distribution

<u>SKEWED t-DISTRIBUTION</u>:

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

Conditional Variance Dynamics
-----------------------------------
GARCH Model : iGARCH(1,1)
Mean Model : ARFIMA(0,0,0)
Distribution : sstd

Optimal Parameters
------------------------------------
        Estimate  Std. Error  t value Pr(>|t|)
mu      0.003152    0.001075   2.9309 0.003380
omega   0.000015    0.000007   2.0755 0.037936
alpha1  0.085151    0.018582   4.5824 0.000005
beta1   0.914849          NA       NA       NA
skew    1.072920    0.045713  23.4710 0.000000
shape   5.121739    0.707781   7.2363 0.000000
```





<u>As the goodness-of-fit increase significantly, the IGARCH with the skewed t-distribution would be the best model within the IGARCH area so far.</u>

```
Adjusted Pearson Goodness-of-Fit Test:
------------------------------------
  group statistic p-value(g-1)
1    20     15.35       0.7000
2    30     31.26       0.3531
3    40     33.24       0.7294
4    50     48.88       0.4778
```

3.2.3 ARMA(0,0)~ eGARCH(1,1) with normal distribution

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

Conditional Variance Dynamics
-----------------------------------
GARCH Model : eGARCH(1,1)
Mean Model : ARFIMA(0,0,0)
Distribution : norm

Optimal Parameters
------------------------------------
        Estimate  Std. Error  t value Pr(>|t|)
mu      0.002655    0.001038   2.5577 0.010535
omega  -0.144484    0.022234  -6.4983 0.000000
alpha1 -0.024598    0.016290  -1.5100 0.131046
beta1   0.976676    0.003454 282.7713 0.000000
gamma1  0.135684    0.021428   6.3322 0.000000
```

```
Weighted Ljung-Box Test on Standardized Residuals
------------------------------------
                        statistic p-value
Lag[1]                     0.1002  0.7516
Lag[2*(p+q)+(p+q)-1][2]    0.1707  0.8725
Lag[4*(p+q)+(p+q)-1][5]    1.3131  0.7857
d.o.f=0
H0 : No serial correlation

Weighted Ljung-Box Test on Standardized Squared Residuals
------------------------------------
                        statistic p-value
Lag[1]                    0.07795  0.7801
Lag[2*(p+q)+(p+q)-1][5]   2.40104  0.5272
Lag[4*(p+q)+(p+q)-1][9]   3.01962  0.7557
d.o.f=2
```

```
Adjusted Pearson Goodness-of-Fit Test:
------------------------------------
  group statistic p-value(g-1)
1    20     33.70     0.019929
2    30     53.89     0.003329
3    40     50.98     0.094846
4    50     72.88     0.015050
```

ARMA(0,0)~ eGARCH(1,1) with t-distribution

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

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

Optimal Parameters
------------------------------------
        Estimate  Std. Error  t value Pr(>|t|)
mu      0.002145    0.001033   2.0769 0.037813
omega  -0.103188    0.025231  -4.0898 0.000043
alpha1 -0.028779    0.021168  -1.3596 0.173971
beta1   0.984024    0.003882 253.4560 0.000000
gamma1  0.176431    0.034823   5.0665 0.000000
shape   5.358070    0.898449   5.9637 0.000000

Adjusted Pearson Goodness-of-Fit Test:
------------------------------------
  group statistic p-value(g-1)
1    20     12.43       0.8663
2    30     31.03       0.3639
3    40     28.33       0.8967
4    50     54.45       0.2749
```

```
Weighted Ljung-Box Test on Standardized Residuals
------------------------------------
                        statistic p-value
Lag[1]                    0.05399  0.8163
Lag[2*(p+q)+(p+q)-1][2]   0.13195  0.8978
Lag[4*(p+q)+(p+q)-1][5]   1.11070  0.8341
d.o.f=0
H0 : No serial correlation

Weighted Ljung-Box Test on Standardized Squared Residuals
------------------------------------
                        statistic p-value
Lag[1]                   0.005468  0.9411
Lag[2*(p+q)+(p+q)-1][5]  1.454859  0.7510
Lag[4*(p+q)+(p+q)-1][9]  2.169578  0.8838
d.o.f=2
```

Exponential GARCH model is used to capture leverage or asymmetric behavior. After running both ARMA(0,0)~eGARCH(1,1) with t-distribution and normal distribution we can easily find out that both model present an insignificant alpha1 which indicates as a matter of fact, there is no asymmetric effect of the Microsoft stock return data. So we don't need the EGARCH to help us here.

3.3 Model Comparison

Though taking many models into consideration, we still want to select a few of the models to make our life easier. Based on the analysis we have done before:

The Log return data of the Microsoft doesn't show the asymmetric distribution so that we don't need use the eGarch or TGARCH of GJRGARCH to capture the asymmetric effects. The GARCH model with t-distribution might be a good choice, however the constraints on the GARCH model are not satisfied as the GARCH(1,1) with the $\boldsymbol{\alpha_1 + \beta_1 = 0.9962.}$ So we say that volatility has non-stationary behavior. Then we move our attention on the IGARCH model. At the IGARCH part, we compare the three IGARCH model: ARMA(0,0)~ IGARCH(1,1) with normal distribution; ARMA(0,0)~ IGARCH(1,1) with t-distribution; ARMA(0,0)~ IGARCH(1,1) with skewed t-distribution. As a conclusion that we select the ARMA(0,0)~ IGARCH(1,1) with skewed t-distribution because the model presents the best goodness-of-fit, significant parameters and the relatively good effect of the residuals(White noise).

The last thing we would like to do is illustrating a comparison of the Akaike information criterion and Bayesian Information criterion. All of the IGARCH give very good parameters on both BIC and AIC, though the IGARCH t-distribution shows a better parameters on than IGARCH skewed-t-distribution, but the residual of the skewed-t-distribution are more adequate for the model. So at last we decide to use the IGARCH(1,1) with skewed-t-distribution to do the forecast.

	BIC	AIC
GARCH std	-3.6792	-3.703
IGARCH norm	-3.6187	-3.6329
IGARCH t-distribution	-3.6858	-3.7048
IGARCH skewed-t-distribution	-3.6817	-3.7054

3.4 Interpretation of final model

The model can be displayed as:

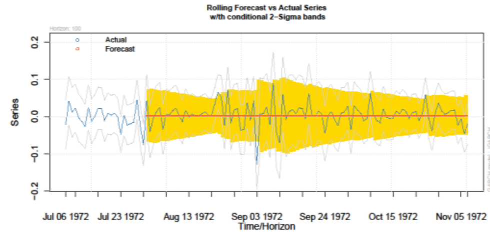
$$\begin{cases} r_t = 0.002754 + a_t, \quad a_t = \sigma_t e_t \\ \sigma_t^2 = 0.075084 a_{t-1}^2 + 0.924916 \sigma_{t-1}^2 \end{cases}$$, error term has a skewed t-distribution with 6 d.f.

All the coefficients are significant. The estimated constant parameter in the GARCH model is almost zero and omitted in the final model. The high beta value means the effect of a shock would be persistent to the return movement. The residual is not perfect white noise, which is basically uncorrelated but shows a slight interdependence in higher order (lag 9).

4. Prediction

The plot of ARMA(0,0)~IGARCH(1,1) with errors of skewed t-distribution model utilizes the out of sample method. We make a 1-step (one week) ahead forecast and the predicted mean and volatility are 0.002754 and 0.03473. So we can get VaR for holding a $10 million position of Microsoft for 1% and 5% significance levels during the next week. For example, there is 5% chance that the potential loss for holding this position in the next week is $ 646,000 or more.

Significance level	VaR for $10,000000 long position	
	Quantile	Dollar
1%	0.1063	$ 1,063,000
5%	0.0646	$ 646,000



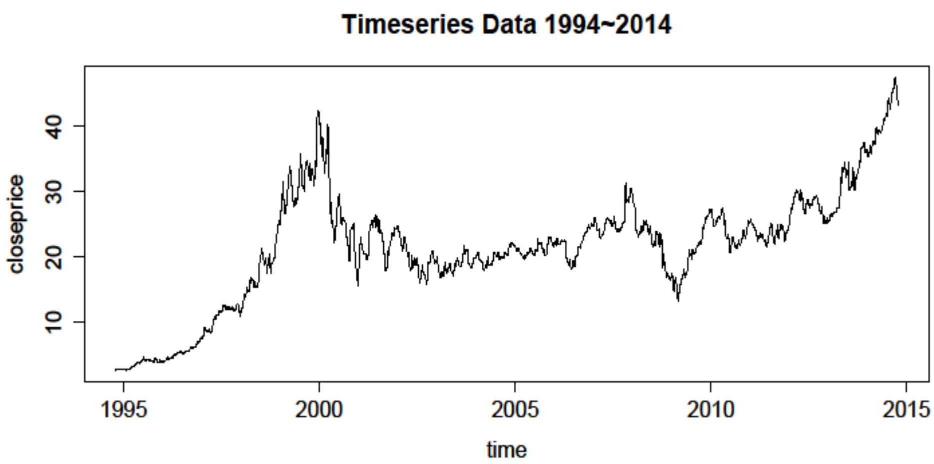
The straight red line is the mean of predicted log return and the 95% confidence level shown as the yellow area includes most of the actual movement, which is shown as the blue line. So our model can capture most of the volatility in the future and has a good prediction ability.

5. Conclusion

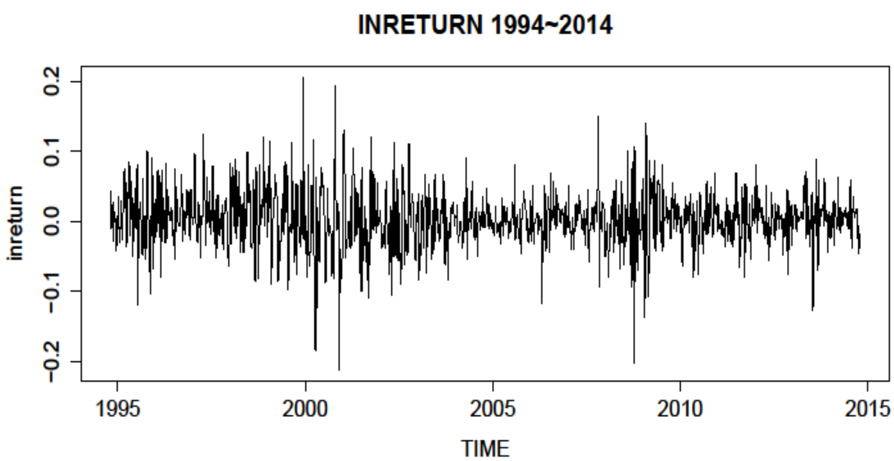
Our ARMA(0,0)~IGARCH(1,1) with errors of skewed t-distribution model presents the best performance in predicting and forecasting the log return of the Microsoft data. Though our model still have some weakness such as the residuals still show a little of interdependency caused by the large size of the data , the relatively small alpha value indicates that the model may not be good at capturing the sudden big shocks. But our model has the best overall parameters and long persistency of predicting the mode.

APPENDIX1

Original stock price data plot:



Log return plot:



Statistic diagnosis with parameters:

```
 Box-Ljung test

data:  cdata
X-squared = 9.522, df = 6, p-value = 0.1463

> Box.test(cdata,lag=12,type='Ljung')

 Box-Ljung test

data:  cdata
X-squared = 21.3319, df = 12, p-value = 0.04573
```

```
 Box-Ljung test

data:  dataabs
X-squared = 166.6564, df = 6, p-value < 2.2e-16

> Box.test(dataabs,lag=12,type='Ljung')

 Box-Ljung test

data:  dataabs
X-squared = 285.9178, df = 12, p-value < 2.2e-16
```

```
 Box-Ljung test

data:  datasq
X-squared = 106.6853, df = 6, p-value < 2.2e-16

> Box.test(datasq,lag=12,type='Ljung')

 Box-Ljung test

data:  datasq
X-squared = 144.0421, df = 12, p-value < 2.2e-16

> #LB tests abs
> Box.test(dataabs,lag=6,type='Ljung')
```

	Original Data	Squared Data	Absolute Data
p-value(lag-6)	0.146	2.2e-16	2.2e-16
p-value(lag-12)	0.046	2.2e-16	2.2e-16

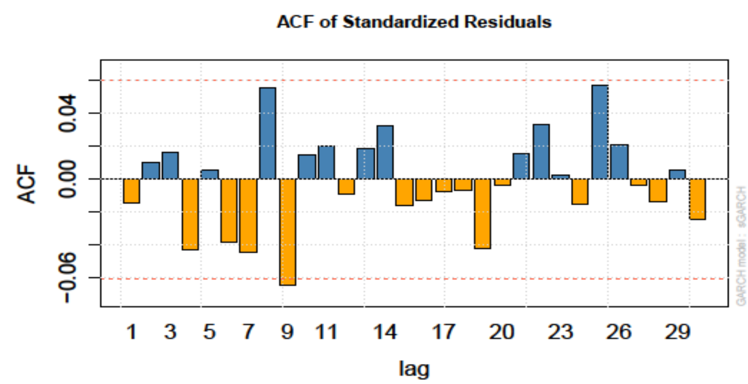
APPENDIX2

ARMA(0,0) ~ GARCH(1,1) with t-distribution:

```
*---------------------------------*
*          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      0.002465    0.000990   2.4909 0.012742
omega   0.000017    0.000010   1.6474 0.099474
alpha1  0.080459    0.019306   4.1676 0.000031
beta1   0.915771    0.018527  49.4295 0.000000
shape   5.350390    0.898688   5.9536 0.000000
```

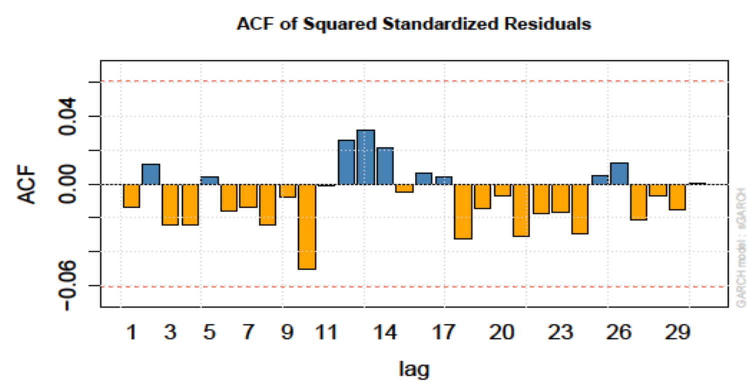
```
Weighted Ljung-Box Test on Standardized Residuals
------------------------------------
                        statistic p-value
Lag[1]                     0.2243  0.6358
Lag[2*(p+q)+(p+q)-1][2]    0.2789  0.8075
Lag[4*(p+q)+(p+q)-1][5]    1.2537  0.8001
d.o.f=0
H0 : No serial correlation

Weighted Ljung-Box Test on Standardized Squared Residuals
------------------------------------
                        statistic p-value
Lag[1]                     0.2073  0.6489
Lag[2*(p+q)+(p+q)-1][5]    0.9353  0.8741
Lag[4*(p+q)+(p+q)-1][9]    1.5591  0.9506
d.o.f=2
```

```
Adjusted Pearson Goodness-of-Fit Test:
------------------------------------
  group statistic p-value(g-1)
1    20     19.61       0.4182
2    30     34.26       0.2300
3    40     38.54       0.4908
4    50     51.47       0.3772
```



<u>This time, the R output presents a 'pretty' fit of the model both for the parameters and the residual analysis. For the GARCH model, we can draw the conclusion that the ARMA(0,0) ~ GARCH(1,1) with t-distribution is better than normal distribution . So we can so far choose the ARMA(0,0) ~ GARCH(1,1) t-distribution within the GARCH area.</u>

<u>However, what we need to pay attention here is the sum of alpha and the beta $\alpha_1 + \beta_1 = 0.9962$ which is pretty much close to 1 leading us to consider the IGARCH model.</u>



APPENDIX3

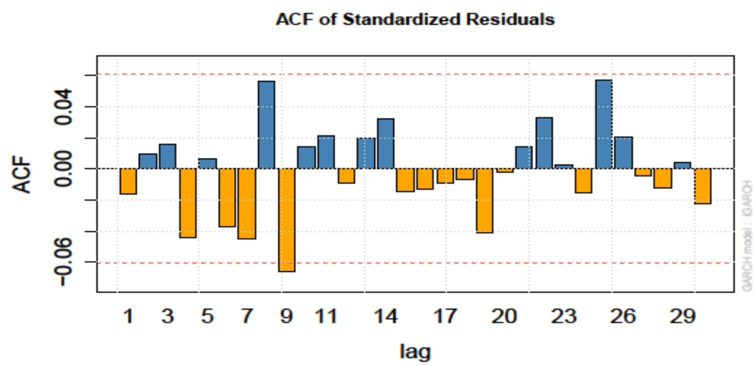
ARMA(0,0) ~ IGARCH(1,1)~normal distribution:

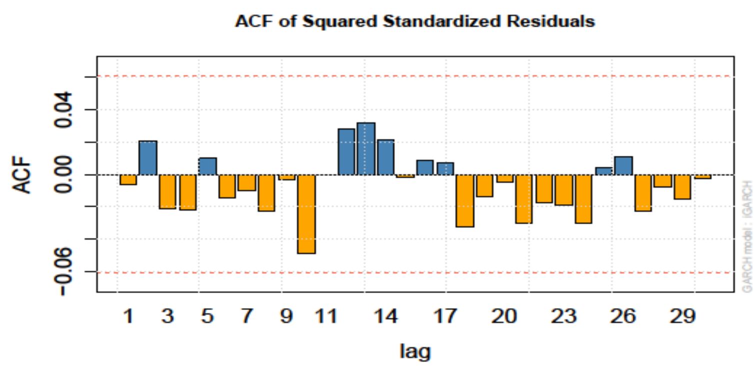
```
*---------------------------------*
*          GARCH Model Fit        *
*---------------------------------*

Conditional Variance Dynamics
-----------------------------------
GARCH Model : iGARCH(1,1)
Mean Model  : ARFIMA(0,0,0)
Distribution : norm

Optimal Parameters
------------------------------------
        Estimate  Std. Error  t value Pr(>|t|)
mu      0.003137    0.001117   2.8075 0.004993
omega   0.000015    0.000005   3.2285 0.001244
alpha1  0.072027    0.013306   5.4130 0.000000
beta1   0.927973          NA       NA       NA
```

```
Adjusted Pearson Goodness-of-Fit Test:
------------------------------------
  group statistic p-value(g-1)
1    20     51.13    8.924e-05
2    30     68.63    4.653e-05
3    40     96.12    1.002e-06
4    50     84.58    1.198e-03
```

<u>The residual analysis plots show that the original residual and the squared residual is both white noise which leads the model adequate. However, the goodness-of –fit is disaster… The residual is obviously not normal distributed! We will try other way to measure the residual.</u>

```
Weighted Ljung-Box Test on Standardized Residuals
------------------------------------
                        statistic p-value
Lag[1]                     0.2714  0.6024
Lag[2*(p+q)+(p+q)-1][2]    0.3207  0.7843
Lag[4*(p+q)+(p+q)-1][5]    1.3467  0.7775
d.o.f=0
H0 : No serial correlation

Weighted Ljung-Box Test on Standardized Squared Residuals
------------------------------------
                        statistic p-value
Lag[1]                    0.04405  0.8338
Lag[2*(p+q)+(p+q)-1][5]   0.91491  0.8786
Lag[4*(p+q)+(p+q)-1][9]   1.47291  0.9578
d.o.f=2
```





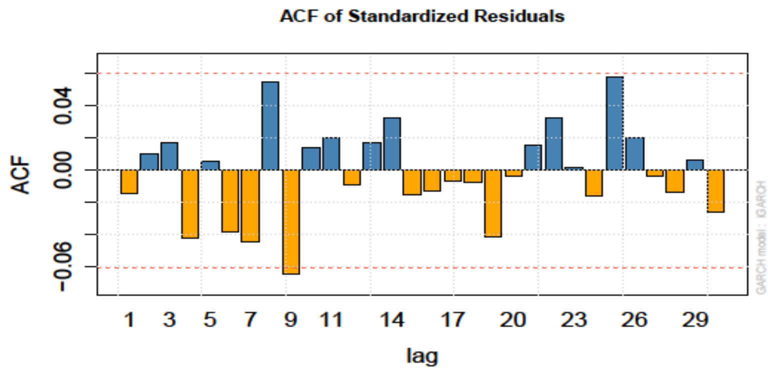
APPENDIX4

ARMA(0,0) ~IGARCH(1,1) with t-distribution:

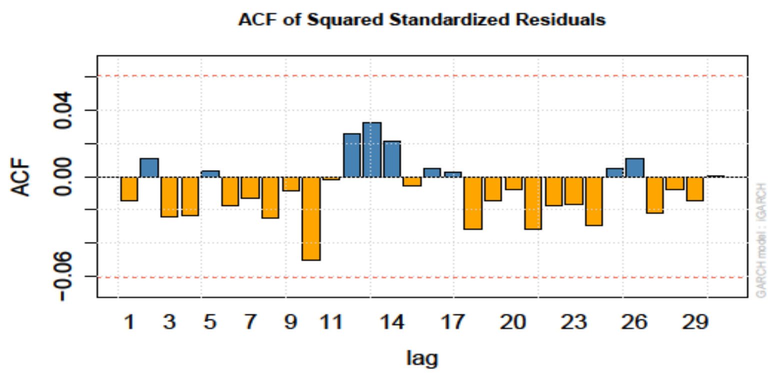
```
*---------------------------------*
*          GARCH Model Fit        *
*---------------------------------*

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

Optimal Parameters
------------------------------------
        Estimate  Std. Error  t value Pr(>|t|)
mu      0.002446    0.000986   2.4818 0.013071
omega   0.000014    0.000007   2.0822 0.037324
alpha1  0.082971    0.017835   4.6521 0.000003
beta1   0.917029          NA       NA       NA
shape   5.181157    0.721492   7.1812 0.000000
```

```
Weighted Ljung-Box Test on Standardized Residuals
------------------------------------
                        statistic p-value
Lag[1]                     0.2224  0.6372
Lag[2*(p+q)+(p+q)-1][2]    0.2765  0.8089
Lag[4*(p+q)+(p+q)-1][5]    1.2370  0.8041
d.o.f=0
H0 : No serial correlation

Weighted Ljung-Box Test on Standardized Squared Residuals
------------------------------------
                        statistic p-value
Lag[1]                     0.2168  0.6415
Lag[2*(p+q)+(p+q)-1][5]    0.9267  0.8760
Lag[4*(p+q)+(p+q)-1][9]    1.5711  0.9496
d.o.f=2
```

```
Adjusted Pearson Goodness-of-Fit Test:
------------------------------------
  group statistic p-value(g-1)
1    20     19.84       0.4041
2    30     37.77       0.1275
3    40     40.23       0.4157
4    50     59.54       0.1440
```



<u>This time, the goodness-of-fit is somewhat increased. However, we think it could do better so we will try the skewed-t-distribution to check whether the parameter will fit better.</u>



APPENDIX5

Rcode:

```
install.packages('fGarch')
install.packages('rugarch')

#Libraries#
library(tseries)
library(fBasics)
library(zoo)
library(fGarch)
library(rugarch)


################################## General Review
##########################################
#get data#
setwd("C:/Users/LZHAOLON/Desktop")
data=read.table("1.csv",head=T,sep=',')
head(data)
summary(data)


#create the ts data#
datats=zoo(data$AdjClose,as.Date(as.character(data$Date), format = "%d/%m/%Y"))
plot(datats,xlab='time',ylab='closeprice',main='Timeseries Data 1994~2014')

#sort data in chronological order#
data$Date=as.Date(as.character(data$Date), format = "%d/%m/%Y")
data=data[order(data$Date),]
#create lag#
datapricelag=lag(datats, k=-1)
inreturn=log(datats/datapricelag)
#Analysis of all data
basicStats(inreturn)
par(mfcol=c(1,1))
plot(inreturn,xlab='TIME',main='INRETURN 1994~2014')
#plots the data for analyszing
par(mfcol=c(1,2))
hist(inreturn,xlab="Microsoft log returns",prob=TRUE, main="Histogram")
xfit<- seq(min(inreturn,na.rm=TRUE),max(inreturn,na.rm=TRUE),length=40)
```

```
yfit<- dnorm(xfit,mean=mean(inreturn,na.rm=TRUE),sd=sd(inreturn,na.rm=TRUE))
lines(xfit,yfit,col='blue',lwd=2)
qqnorm(inreturn)
qqline(inreturn,col='red',lwd=2)
#(we need to talk about the outlier a little bit. Max(2),Min(4) those 6 are outliers)



######################### Normal JB test# ##############################################
normalTest(inreturn,method=c('jb'))
#kurtosis and skewness test#
kstat=kurtosis(inreturn)/sqrt(24/length(inreturn))
print("Kurtosis test statistic")
kstat
print("P-value = ")
2*(1-pnorm(abs(kstat)))

skewtest = skewness(inreturn)/sqrt(6/length(inreturn))
skewtest
#P-value
2* (1-pnorm(abs(skewtest)))




#################### ACFs (inreturn,squared,absolute) ########################


par(mfcol=c(4,1))
cdata=coredata(inreturn)
acf(cdata, plot=T, lag=15)
#LB tests
Box.test(cdata,lag=6,type='Ljung')
Box.test(cdata,lag=12,type='Ljung')

#ACF datasquare#
datasq=cdata^2
acf(datasq,plot=T,lag=15)
dataabs=abs(cdata)
acf(dataabs,plot=T,lag=15)
#LB tests square
Box.test(datasq,lag=6,type='Ljung')
Box.test(datasq,lag=12,type='Ljung')
#LB tests abs
```

```
Box.test(dataabs,lag=6,type='Ljung')
Box.test(dataabs,lag=12,type='Ljung')


#plots PACF of squared returns to identify order of AR model
pacf(datasq,lag=15)




#################### Models #########################


################## AR ###################

# Fitting an AR(0)-ARCH(2) model with normal distribution
#ARMA(2,0) - GARCH(1,1) model
garch211.spec=ugarchspec(variance.model=list(garchOrder=c(1,1)), mean.model=list(armaOrder=c(2,0)))
#estimate model
garch211.fit=ugarchfit(spec=garch211.spec, data=cdata)
garch211.fit

#ARMA(2,0) without lag 2 - GARCH(1,1) model
garch211.spec2=ugarchspec(variance.model=list(garchOrder=c(1,1)),
mean.model=list(armaOrder=c(2,0)),fixed.pars=list(ar2=0))
garch211.fit2=ugarchfit(spec=garch211.spec2, data=cdata)
garch211.fit2

#ARMA(2,0) without lag 2 and lag 1 - GARCH(1,1) model
garch211.spec1=ugarchspec(variance.model=list(garchOrder=c(1,1)),
mean.model=list(armaOrder=c(2,0)),fixed.pars=list(ar2=0,ar1=0))
garch211.fit1=ugarchfit(spec=garch211.spec1, data=cdata)
garch211.fit1



############### SELECTING GARCH ##################



###### GARCH ######

#fitiing ARMA(0,0) - GARCH(1,1) model
```

```
#use normal distribution#
garch11.spec=ugarchspec(variance.model=list(garchOrder=c(1,1)), mean.model=list(armaOrder=c(0,0)))
garch11.fit=ugarchfit(spec=garch11.spec,data=cdata)
garch11.fit

#estimated coefficients#
#using parameters to write function#
coef(garch11.fit)

#persistence = alpha1+beta1
persistence(garch11.fit)
#half-life: ln(0.5)/ln(alpha1+beta1)
halflife(garch11.fit)
par(mfcol=c(2,1))
plot(garch11.fit,which='all')

# use Student-t innovations
#specify model using functions in rugarch package
#Fit ARMA(0,0)-GARCH(1,1) model with t-distribution
garch11.t.spec=ugarchspec(variance.model=list(garchOrder=c(1,1)), mean.model=list(armaOrder=c(0,0)),
distribution.model = "std")
#estimate model
garch11.t.fit=ugarchfit(spec=garch11.t.spec, data=cdata)
garch11.t.fit

#persistence = alpha1+beta1
persistence(garch11.t.fit)
plot(garch11.t.fit,which='all')



##### IGARCH ######

#Fit ARMA(0,0)-IGARCH(1,1) model with  normal distribution
igarch11.spec=ugarchspec(variance.model=list(model = "iGARCH", garchOrder=c(1,1)),
mean.model=list(armaOrder=c(0,0)))
#estimate model
igarch11.fit=ugarchfit(spec=igarch11.spec, data=cdata)
igarch11.fit
plot(igarch11.fit,which='all')

#Fit ARMA(0,0)-IGARCH(1,1) model with t-distribution
igarch11.t.spec=ugarchspec(variance.model=list(model = "iGARCH", garchOrder=c(1,1)),
mean.model=list(armaOrder=c(0,0)), distribution.model = "std")
#estimate model
```

```
igarch11.t.fit=ugarchfit(spec=igarch11.t.spec, data=cdata)
igarch11.t.fit
plot(igarch11.t.fit,which='all')

#Fit ARMA(0,0)-IGARCH(1,1) model with skewed t-distribution
igarch11.skt.spec=ugarchspec(variance.model=list(model = "iGARCH", garchOrder=c(1,1)), mean.model=list(armaOrder=c(0,0)), distribution.model = "sstd")
#estimate model
igarch11.skt.fit=ugarchfit(spec=igarch11.skt.spec, data=cdata)
igarch11.skt.fit
plot(igarch11.skt.fit,which='all')


######## eGARCH ##########

#Fit ARMA(0,0)-eGARCH(1,1) model with  Normal distribution
egarch11.spec=ugarchspec(variance.model=list(model = "eGARCH", garchOrder=c(1,1)), mean.model=list(armaOrder=c(0,0)))
#estimate model
egarch11.fit=ugarchfit(spec=egarch11.spec, data=cdata)
egarch11.fit
plot(egarch11.fit,which='all')



# compute expected value E(|e|)
shape=coef(egarch11.fit)[6]
exp.abse=(2*sqrt(shape-2)*gamma((shape+1)/2))/((shape-1)*gamma(shape/2)*sqrt(pi))



############## MODEL COMPARISON #################

# compare information criteria
# MODEL COMPARISON
# compare information criteria
model.list = list(garch11.t = garch11.t.fit,
            igarch11 = igarch11.fit,
            igarch11.t = igarch11.t.fit,
            igarch11.skt = igarch11.skt.fit,
            egarch11 = egarch11.fit)
```

```
info.mat = sapply(model.list, infocriteria)
rownames(info.mat) = rownames(infocriteria(garch11.fit))
info.mat


# re-fit models leaving 100 out-of-sample observations for forecast
# evaluation statistics
garch11.t.fit = ugarchfit(spec=garch11.t.spec, data=cdata, out.sample=100)

igarch11.skt.fit = ugarchfit(spec=igarch11.skt.spec, data=cdata, out.sample=100)


# compute 100 1-step ahead rolling forecasts w/o re-estimating
par(mfrow=c(1,1))
igarch11.skt.fcst = ugarchforecast(igarch11.skt.fit, n.roll=100, n.ahead=1)
garch11.t.fcst = ugarchforecast(garch11.t.fit, n.roll=100, n.ahead=1)

igarch11.skt.fcst
plot(igarch11.skt.fcst)

fcst.list = list(igarch = igarch11.skt.fcst,garch = garch11.t.fcst)
fpm.mat = sapply(fcst.list, fpm)
fpm.mat
```

APPENDIX6

	igarch	garch
MSE	0.0008914853	0.0008961665
MAE	0.0208329	0.0209877
DAC	0.65	0.65