Time Series Models on High Frequency Trading Data of SHA:600519

MAFS 5130

QUANTITATIVE ANALYSIS OF FINANCIAL TIME SERIES

EDITED BY

LU YIFAN
No.20305030

HUANG JINGYING
No.20294918

JIN GAOZHENG
No.20295467

WANG YILEI
No.20305250

GAO XIANG
No.09813967

TU JIA
No.09593359

APRIL 2016

Contents

1 Introduction

1.1 Introduction to High Frequency Trading

High-frequency trading (HFT) is a type of algorithmic trading characterized by high speeds, high turnover rates, and high order-to-trade ratios that leverages high-frequency financial data and electronic trading tools. While there is no single definition of HFT, among its key attributes are highly sophisticated algorithms, specialized order types, co-location, very short-term investment horizons, and high cancellation rates of orders. HFT can be viewed as a primary form of algorithmic trading in finance. Specifically, it is the use of sophisticated technological tools and computer algorithms to rapidly trade securities. HFT uses proprietary trading strategies carried out by computers to move in and out of positions in seconds or fractions of a second. It is estimated that as of 2009, HFT accounted for 60-73% of all US equity trading volume, with that number falling to approximately 50% in 2012. High-frequency traders move in and out of short-term positions at high volumes and high speeds aiming to capture sometimes a fraction of a cent in profit on every trade. HFT firms do not consume significant amounts of capital, accumulate positions or hold their portfolios overnight. As a result, HFT has a potential Sharpe ratio (a measure of reward to risk) tens of times higher than traditional buy-and-hold strategies. High-frequency traders typically compete against other HFTs, rather than long-term investors. HFT firms make up the low margins with incredibly high volumes of trades, frequently numbering in the millions.

A substantial body of research argues that HFT and electronic trading pose new types of challenges to the financial system. Algorithmic and high-frequency traders were both found to have contributed to volatility in the Flash Crash of May 6, 2010, when high-frequency liquidity providers rapidly withdrew from the market. Several European countries have proposed curtailing or banning HFT due to concerns about volatility.

1.2 Introduction to High Frequency Trading Data Analysis

In recent years, high-frequency trading is becoming more and more popular. Due to the rapid development of computing capability and storage capacity, people are able to collect and process high frequency data, resulting in a great concern for high frequency data research in the both academic and industry field. For example, Wood [1] described some historical perspective of the high frequency data research; Eric Ghysels [2] reviewed the application of some econometrics methods in the field of high frequency data; LUO Zhongzhou [3] conducted researches on application of high frequency trading in the Chinese market.

The analysis on high frequency financial data is not only important for many issues like research, trading procedures and market microstructure, but also related to many fields of studies like econometrics, finance, statistics and etc. High frequency financial data can be used to compare the effectiveness of the price discovery in different trading systems (for example, open outcry trading system in NYSE and the Nasdaq computer system) and can also be used to study the dynamic changes of trading quotes on a specific stock, see Zhan [4], etc. It will be very helpful to answer the question, such as "who is providing market liquidity", by researching high frequency financial data.

High frequency data has some unique features: various of time intervals, such as that the time interval between each trades of stocks are generally not the same; discrete values of the price, such as that the prices of financial underlying are discrete variables during the transaction; there exists daily cycle, such as that the shape of trading intensity exhibits a U-shaped curve; multiple transactions within one second and multiple transaction prices within very short time (one second). Due to these characteristics, we cannot directly apply the common low frequency data analyzing method here. Therefore, it is a brand new challenge for financial economists and statisticians to bring up solutions of analyzing high frequency data.

1.3 Bid Ask Price

Bid ask Price is a two-way price quotation that indicates the best price at which a security can be sold and bought at a given point in time. The bid price represents the maximum price that a buyer or buyers are willing to pay for a security. The ask price represents the minimum price that a seller or sellers are willing to receive for the security. A trade or transaction occurs when the buyer and seller agree on a price for the security.The difference between the bid and asked prices, or the spread, is a key indicator of the liquidity of the asset - generally speaking, the smaller the spread, the better the liquidity.

Bid one price is the highest among all bid prices and ask one price is the lowest among all ask prices. Mid-quote price is the weighted average price of bid one price and ask one price.

1.4 Role of Market Maker

According to www.sec.gov, a "market maker" is "a firm that stands ready to buy and sell a particular stock on a regular and continuous basis at a publicly quoted price. You'll most often hear about market makers in the context of the Nasdaq or other "over the counter" (OTC) markets. Market makers that stand ready to buy and sell stocks listed on an exchange, such as the New York Stock Exchange, are called "third market makers." Many OTC stocks have more than one market-maker.

Market-makers generally must be ready to buy and sell at least 100 shares of a stock they make a

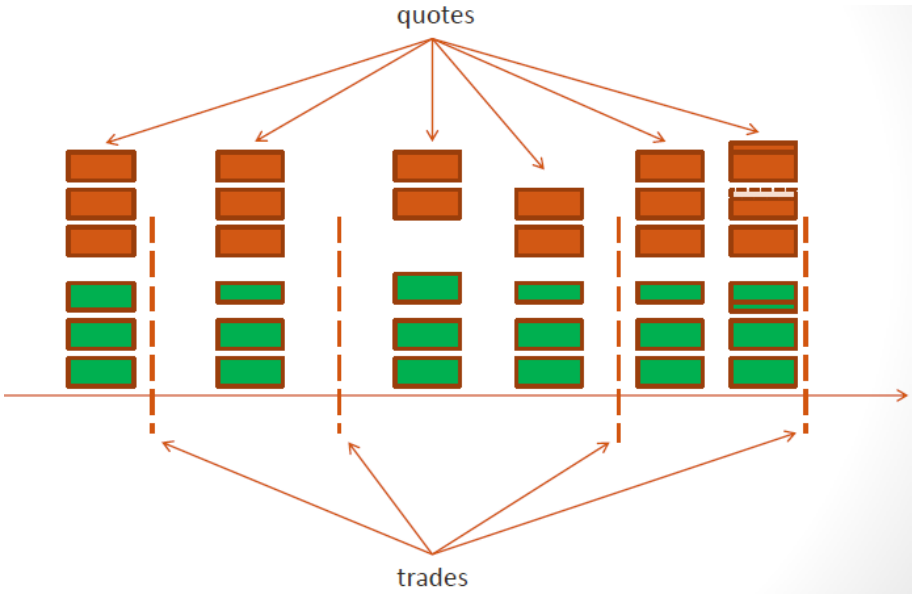


Figure 1: Bid-Ask Spread

market in. As a result, a large order from an investor may have to be filled by a number of market-makers at potentially different prices."

There can be a significant overlap between a 'market maker' and 'HFT firm'. HFT firms characterize their business as "Market making"-a set of high-frequency trading strategies that involve placing a limit order to sell (or offer) or a buy limit order (or bid) in order to earn the bid-ask spread. By doing so, market makers provide counterpart to incoming market orders. Although the role of market maker was traditionally fulfilled by specialist firms, this class of strategy is now implemented by a large range of investors, thanks to wide adoption of direct market access. As pointed out by empirical studies this renewed competition among liquidity providers causes reduced effective market spreads, and therefore reduced indirect costs for final investors." A crucial distinction is that true market makers don't exit the market at their discretion and are committed not to, where HFT firms are under no similar commitment.

Some high-frequency trading firms use market making as their primary strategy. Automated Trading Desk, which was bought by Citigroup in July 2007, has been an active market maker, accounting for about 6% of total volume on both the NASDAQ and the New York Stock Exchange. Building up market making strategies typically involves precise modeling of the target market microstructure together with stochastic control techniques.

These strategies appear intimately related to the entry of new electronic venues. Academic study of Chi-X's entry into the European equity market reveals that its launch coincided with a large HFT that made markets using both the incumbent market, NYSE-Euronext, and the new market, Chi-X. The study

shows that the new market provided ideal conditions for HFT market-making, low fees (i.e., rebates for quotes that led to execution) and a fast system, yet the HFT was equally active in the incumbent market to offload nonzero positions. New market entry and HFT arrival are further shown to coincide with a significant improvement in liquidity supply.

1.5 Motivation of the Project

In our report, we would like to focus on the high frequency trading data of Kweichow Moutai Co Ltd (SHA: 600519) from January 4th, 2013 to February 26th, 2014 from the market makers' perspective. Market makers are interested in two kind of information from the high frequency trading data.

Firstly, market makers are certainly interested in how the mid-quote price would change. Admittedly, market makers are often employed by the order-driven markets to quote prices continuously for the well being of the market, and therefore, market makers do not assume the goal of making money. They always tries to adjust their quoted bids and offer prices in order to facilitate the price discovery process of an exchange. However they really do not want to lose money either. That's why they are interested in whether the price trend is upward or downward.

Secondly, market makers are even more interested in the duration time during which the mid-quote price remains unchanged. In the theory of Market Microstructure, such duration time is called "Market Microstructure Characteristics Time", which is a "time scale" during which the price process moves from a random bid-ask bounce into a random walk process.

Although a large number of high-frequency transactions may occur within one second, the bid one and ask one price stay unchanged most of time. Market makers like this kind of situation, because they can simply apply the strategy of "buy low, sell high". During the time duration that the bid one and ask one price stay unchanged, market makers can automatically buy at bid one price and sell at ask one price, making money at the same time when they facilitate the price discovery process of an exchange.

However, if the mid-quote price tends to change, market makers have to make a decision whether to change the bid ask spread or not. Market makers can remain their bid-ask spreads, losing money due to their duties. They can either change their bid-ask spreads in order to avoid losing money. No matter what decisions market makers make, they cannot simply apply a fixed strategy. As a result, $\triangle t_i$, the duration time during which the transaction prices remain unchanged is of great importance for market makers.

In our report, assuming we are a market maker in the SHA market, we would like to build some time series models on the two important random variables P_{t_i} and $\triangle t_i$ according to the high frequency data analysis. From the market maker's point of view, we are more concerned about the model focusing on

$\triangle t_i$.

Admittedly, the forecast value of $\triangle t_i$ might be quite small but still useful. Since low-latency communication technologies at both the software level and hardware level significantly contributed the development of algorithmic trading as a trading practice. Market makers can write a program to forecast the "Market Microstructure Characteristics Time" and use it to determine the algorithmic trading strategies.

2 Time Series Analysis on Price

2.1 Data Process

We firstly plot the data of price. It can be easily observed from Figure 2 that it is not stationary, so we take the log return and modified the data.

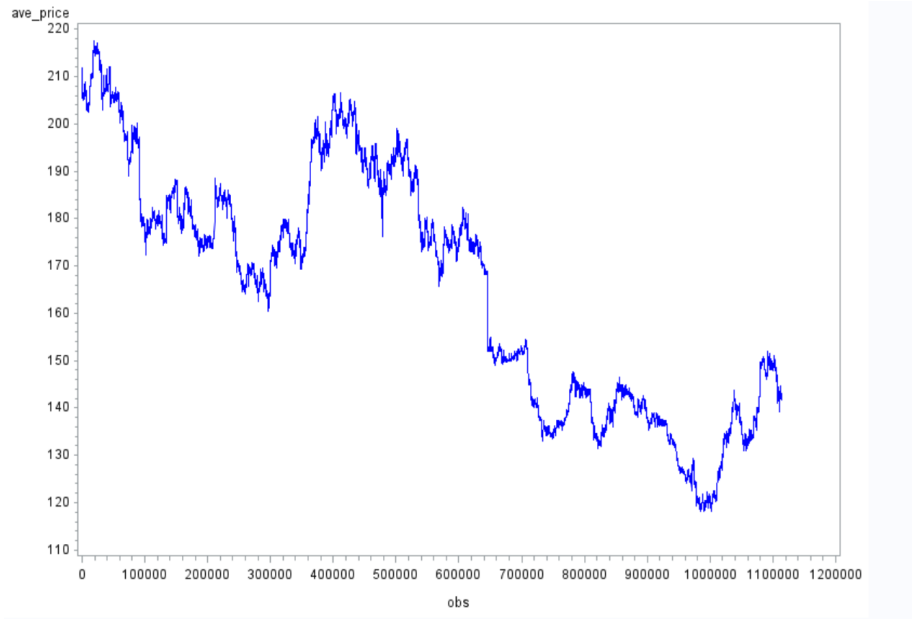


Figure 2: Price-Time

2.1.1 Dickey-Fuller Unit Root Test

After we change the data (seen in Appendix .A Data), we can see from Figure 3 that the log return series look somehow stationary. We then make the Dickey-Fuller Unit Root Test by using SAS, getting the following results in Figure 4. As all P-values are smaller than 0.05, the log return series are stationary.

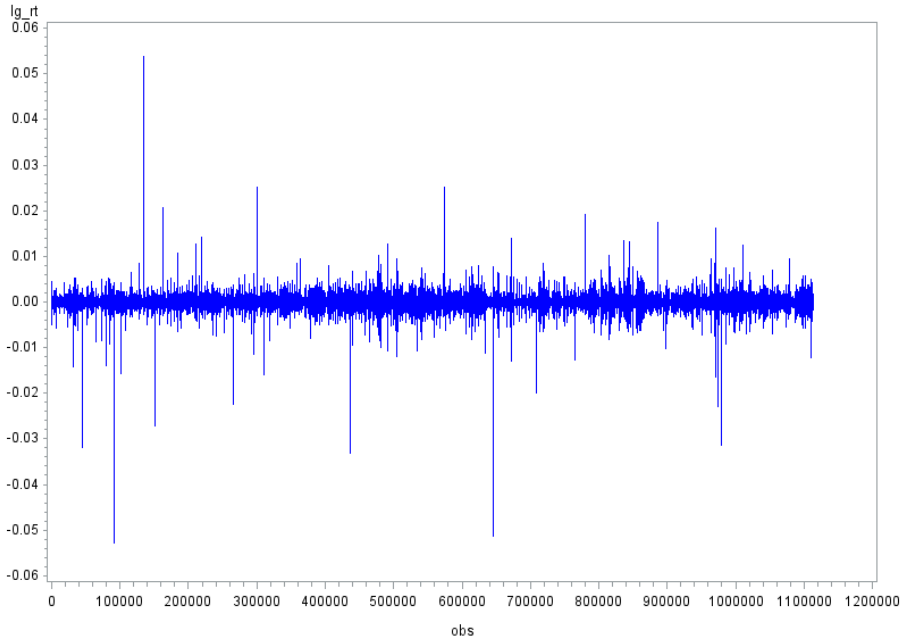


Figure 3: Log Return-Time

Augmented Dickey-Fuller Unit Root Tests							
Type	Lags	Rho	Pr < Rho	Tau	Pr < Tau	F	Pr > F
Zero Mean	0	-1478054	0.0001	-1481.8	0.0001		
	1	-2006011	0.0001	-1001.5	0.0001		
	2	-2600860	0.0001	-774.61	0.0001		
Single Mean	0	-1478055	0.0001	-1481.8	0.0001	1097890	0.0010
	1	-2006018	0.0001	-1001.5	0.0001	501515	0.0010
	2	-2600892	0.0001	-774.61	0.0001	300011	0.0010
Trend	0	-1478055	0.0001	-1481.8	0.0001	1097890	0.0010
	1	-2006020	0.0001	-1001.5	0.0001	501515	0.0010
	2	-2600900	0.0001	-774.61	0.0001	300011	0.0010

Figure 4: Dickey-Fuller Unit Root Test on Log Return

2.1.2 Ljung−Box Test

We then check the serial correlation of log return series by using Ljung−Box Test in SAS. As we can see from Figure 5 that all P-values are smaller than 0.05, which means that we will reject $\mathbb{H}_0$ and that there exists serial correlation.

The SAS System

The ARIMA Procedure

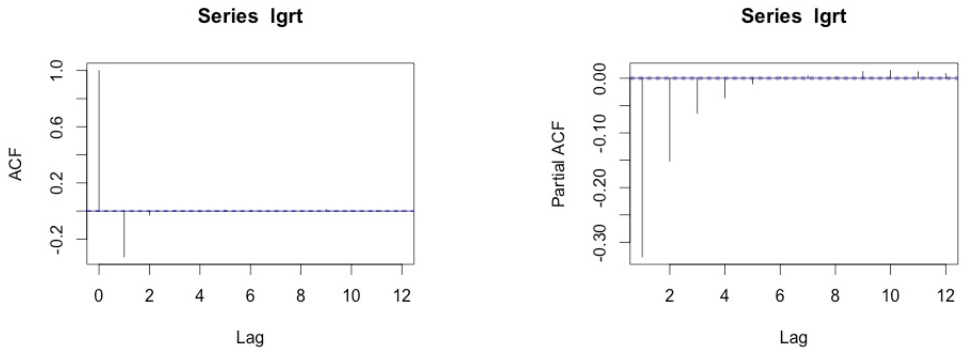
Name of Variable = lg_rt	
Mean of Working Series	-3.53E-7
Standard Deviation	0.00044
Number of Observations	1113950

Autocorrelation Check for White Noise									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	9999.99	6	<.0001	-0.327	-0.029	0.004	-0.007	0.007	0.005
12	9999.99	12	<.0001	0.001	-0.001	0.010	0.004	0.002	0.001
18	9999.99	18	<.0001	0.003	0.003	0.001	0.004	0.002	0.003
24	9999.99	24	<.0001	-0.002	0.003	0.004	-0.003	0.007	0.001
30	9999.99	30	<.0001	0.004	-0.001	0.001	0.001	0.001	-0.000
36	9999.99	36	<.0001	0.006	-0.002	0.004	0.002	-0.002	0.001
42	9999.99	42	<.0001	0.004	-0.000	0.001	-0.001	0.001	-0.001
48	9999.99	48	<.0001	0.005	0.000	-0.003	0.002	-0.003	0.001
54	9999.99	54	<.0001	0.003	-0.001	0.003	-0.002	0.001	-0.001
60	9999.99	60	<.0001	-0.001	0.002	0.000	0.001	0.000	0.000
66	9999.99	66	<.0001	-0.001	0.003	-0.002	0.000	-0.001	0.002
72	9999.99	72	<.0001	-0.003	0.000	0.001	0.001	0.001	-0.002
78	9999.99	78	<.0001	0.000	-0.003	0.001	0.004	-0.002	-0.002
84	9999.99	84	<.0001	0.003	0.002	-0.005	0.004	-0.001	0.000
90	9999.99	90	<.0001	-0.001	0.001	0.001	0.001	-0.001	-0.001
96	9999.99	96	<.0001	0.000	0.003	0.001	-0.003	-0.001	0.000

Figure 5: Ljung Box Test on Log Return

2.2 ARIMA Model

We try ARIMA (2,0,1) model, ARIMA (3,0,1) model and MA (1) model based on the ACF and partial ACF in Figure 6 of the log return series. Unfortunately, under no models the residual can be modeled as white noise. Based on the results of model checking, we can not find fitting model for our data using ARIMA. The result means that ARIMA model is not a good model for the high frequency trading data. We need to build some other models to deal with the problem.



(a) ACF (b) PACF

Figure 6: ACF and PACF

The SAS System

The ARIMA Procedure

Name of Variable = lg_rt	
Mean of Working Series	-3.53E-7
Standard Deviation	0.00044
Number of Observations	1113950

Autocorrelation Check for White Noise									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	9999.99	6	<.0001	-0.327	-0.029	0.004	-0.007	0.007	0.005
12	9999.99	12	<.0001	0.001	-0.001	0.010	0.004	0.002	0.001

Conditional Least Squares Estimation					
Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag
MU	-3.5328E-7	2.27227E-7	-1.55	0.1200	0
MA1,1	0.42949	0.0045960	93.45	<.0001	1
AR1,1	0.04084	0.0046795	8.73	<.0001	1
AR1,2	-0.01560	0.0020518	-7.60	<.0001	2

Constant Estimate	-3.44E-7
Variance Estimate	1.679E-7
Std Error Estimate	0.00041
AIC	-1.422E7
SBC	-1.422E7
Number of Residuals	1113950

Figure 7: ARIMA(2,0,1)

Autocorrelation Check of Residuals									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	231.72	3	<.0001	-0.000	-0.000	-0.001	-0.005	0.009	0.010
12	983.50	9	<.0001	0.007	0.007	0.017	0.013	0.009	0.007
18	1364.02	15	<.0001	0.009	0.008	0.006	0.008	0.007	0.006
24	1703.26	21	<.0001	0.003	0.006	0.008	0.003	0.011	0.008
30	1847.02	27	<.0001	0.008	0.003	0.003	0.003	0.003	0.004
36	2054.46	33	<.0001	0.009	0.004	0.007	0.005	0.001	0.004
42	2128.69	39	<.0001	0.007	0.003	0.003	0.001	0.002	0.002
48	2180.26	45	<.0001	0.006	0.001	-0.002	0.001	-0.002	0.002

Figure 8: Model Checking on ARIMA(2,0,1)

The SAS System

The ARIMA Procedure

Name of Variable = lg_rt	
Mean of Working Series	-3.53E-7
Standard Deviation	0.00044
Number of Observations	1113950

Autocorrelation Check for White Noise									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	9999.99	6	<.0001	-0.327	-0.029	0.004	-0.007	0.007	0.005
12	9999.99	12	<.0001	0.001	-0.001	0.010	0.004	0.002	0.001

Conditional Least Squares Estimation					
Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag
MU	-3.5328E-7	2.27514E-7	-1.55	0.1205	0
MA1,1	0.41177	0.01072	38.42	<.0001	1
AR1,1	0.02310	0.01075	2.15	0.0317	1
AR1,2	-0.02242	0.0042617	-5.26	<.0001	2
AR1,3	-0.0044461	0.0021908	-2.03	0.0424	3

Constant Estimate	-3.55E-7
Variance Estimate	1.679E-7
Std Error Estimate	0.00041
AIC	-1.422E7
SBC	-1.422E7
Number of Residuals	1113950

Figure 9: ARIMA(3,0,1)

Autocorrelation Check of Residuals									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	228.76	2	<.0001	-0.000	-0.000	0.000	-0.006	0.008	0.010
12	976.12	8	<.0001	0.007	0.007	0.016	0.013	0.009	0.007
18	1354.43	14	<.0001	0.009	0.008	0.006	0.008	0.007	0.006
24	1692.55	20	<.0001	0.003	0.006	0.008	0.003	0.011	0.008
30	1835.37	26	<.0001	0.008	0.003	0.003	0.003	0.003	0.004
36	2042.35	32	<.0001	0.009	0.004	0.007	0.005	0.001	0.004
42	2116.16	38	<.0001	0.007	0.003	0.003	0.001	0.002	0.002
48	2167.71	44	<.0001	0.006	0.001	-0.002	0.001	-0.002	0.002

Figure 10: Model Checking on ARIMA(3,0,1)

The SAS System

The ARIMA Procedure

Name of Variable = lg_rt	
Mean of Working Series	-3.53E-7
Standard Deviation	0.00044
Number of Observations	1113950

Autocorrelation Check for White Noise									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	9999.99	6	<.0001	-0.327	-0.029	0.004	-0.007	0.007	0.005
12	9999.99	12	<.0001	0.001	-0.001	0.010	0.004	0.002	0.001

Conditional Least Squares Estimation					
Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag
MU	-3.5323E-7	2.32828E-7	-1.52	0.1292	0
MA1,1	0.40070	0.0008678	461.75	<.0001	1

Constant Estimate	-3.53E-7
Variance Estimate	1.681E-7
Std Error Estimate	0.00041
AIC	-1.422E7
SBC	-1.422E7
Number of Residuals	1113950

Figure 11: MA(1)

Autocorrelation Check of Residuals									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	1234.85	5	<.0001	0.012	-0.027	-0.007	-0.007	0.008	0.010
12	1930.38	11	<.0001	0.006	0.007	0.016	0.013	0.009	0.007
18	2276.62	17	<.0001	0.008	0.008	0.006	0.008	0.007	0.006
24	2591.32	23	<.0001	0.003	0.006	0.007	0.003	0.011	0.008
30	2719.99	29	<.0001	0.008	0.003	0.003	0.003	0.003	0.004
36	2913.39	35	<.0001	0.009	0.003	0.007	0.004	0.001	0.004
42	2983.57	41	<.0001	0.007	0.003	0.002	0.001	0.002	0.002
48	3035.56	47	<.0001	0.006	0.001	-0.002	0.001	-0.002	0.001

Figure 12: Model Checking on MA(1)

2.3 ARIMA-GARCH Model

As ARIMA model is not a good model, we will then try to fit an ARIMA-GARCH model to the data. The R code is appended in Appendix B.

2.3.1 Checking ARCH Effect

As there is a strong correlation in the log returns, we first consider fit an AR (1) model to the log return in order to remove the serial correlation and then do the test of ARCH effect.

According to the test result from R, $\mathbb{Q}(12, a_t^2) = 14794$, with a p-value near zero. Therefore, we reject $\mathbb{H}_0$ that all ACF are zero. As a result, there is ARCH effect.

2.3.2 Model Fitting and Checking

We try different kinds of ARMA GARCH models by using R, finding that the AR(1)GARCH(1,1) model has the smallest AIC and BIC.

As we can see from Table 1 that all parameters have passed the T test.

However, when we check Table 2, we find that even though $\mathbb{Q}(20, R^2) = 28.32919$ with a P-value of 0.101833, all $\mathbb{Q}(l, R)$ have a very small P-value for all $l = 10, 15, 20$.

Therefore, ARMA GARCH model can only remove ARCH effect. It cannot remove serial correlation, which corresponds to the result in Chapter 2.2.

	Estimate	Std. Error	t value	Pr(>|t|)
mu	-3.199e-05	1.143e-05	-2.800	0.00511
ar1	-3.070e-01	4.191e-02	-7.326	2.37e-13
omega	4.945e-09	1.160e-09	4.262	2.03e-05
alpha1	2.142e-01	3.503e-02	6.114	9.69e-10
beta1	7.899e-01	2.484e-02	31.802	< 2e-16

Table 1: Error Analysis

2.4 Conclusion

Both ARIMA model and ARIMA GARCH model cannot fit the data of mid-quote price well. Even though GARCH model can remove the ARCH effect, ARIMA model cannot remove the serial correlation well. The reason might be that the ARIMA model cannot describe the features of high frequency trading price. We have to build some new models.

			Statistic	p-Value
Jarque-Bera Test	R	Chi^2	226.6154	0
Shapiro-Wilk Test	R	W	0.956208	1.386691e-13
Ljung-Box Test	R	Q(10)	28.83545	0.001324794
Ljung-Box Test	R	Q(15)	32.80012	0.005001912
Ljung-Box Test	R	Q(20)	49.91691	0.0002276229
Ljung-Box Test	R^2	Q(10)	19.5223	0.03410873
Ljung-Box Test	R^2	Q(15)	27.39051	0.02571104
Ljung-Box Test	R^2	Q(20)	28.32919	0.101833
LM Arch Test	R	TR^2	24.21535	0.01901148

Table 2: Standardised Residuals Tests

3 Time Series Analysis on Time Duration

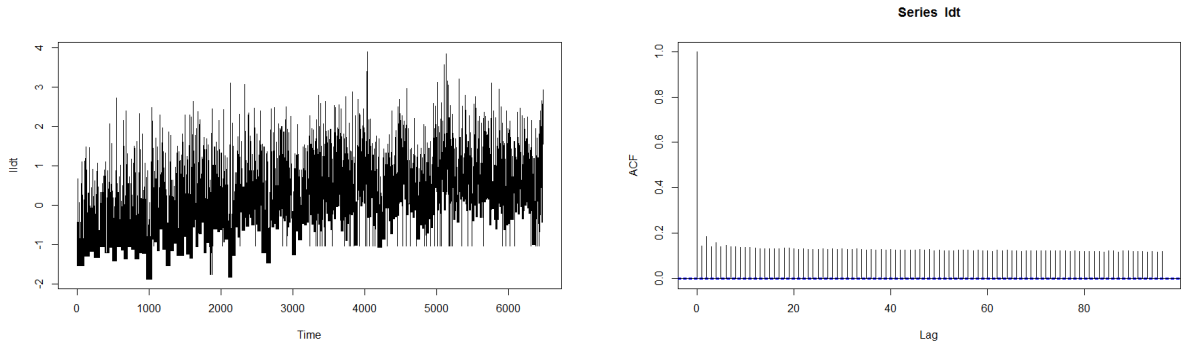
We would like to build a time series model on time duration in this section. We first define these variables as follows. Let t_i , ($i = 1, 2, ..., n$) denote the time of the i^{th} change of transaction price. Let P_{t_i} denote the transaction price of the i^{th} transaction price change, so $\triangle t_i = t_i - t_{i-1}$ is the duration time of price remain unchanged. Let S_i denote the changing amount of i^{th} price change, i.e. $S_i = P_{t_i} - P_{t_{i-1}}$. Let N_i denote the number of transactions happened within time interval (t_{i-1}, t_i) during which the transaction prices remain unchanged. This variable N_i can represent the trading intensity during the time with no transaction price change.

3.1 Long Memory Phenomenon of Time Duration

Engle and Rusel [5] proposed the famous autoregressive conditional duration (ACD) model on the research of time duration $\triangle t_i$. McCuloch and Tsay [6] conducted researches on the nonlinearity of high frequency financial data through nonlinear layer model to describe the relationship between $\triangle t_i$, S_i and N_i. Together, they came up with the price changing and duration model (PCD) to describe the price changing feature and the multiple factors dynamic structure during the time durations.

However, we found the time duration of the Kweichow Moutai Co Ltd (SHA: 600519) has a long memory. The timing chart of its log time duration and the autocorrelation function are shown in Figures 13. We can neither see the up or down trend of its log time duration from the left one, but from the right one we can easily observe the log time duration has a long duration of memory. Therefore we need to build a time series model in order to remove this kind of long memory phenomenon.

The long memory phenomenon was already known by people long before many stochastic model was developed. For example, people observed that the Nile River has long-term behavioral characteristic since ancient time. A long period of drought and a long period of flooding always came one after each



(a) log time duration (b) ACF

Figure 13: Long Memory Phenonmeon

other, so cycled. This phenomenon was recorded in the Bible (Genesis 41, 29-30) and was called Joseph effects by Mandelbrot [7] later. Since the pioneering of Mandelbrot [8] [9] [10], self-similarity and the associated long memory process were introduced to the field of statistics. In fact, long memory processes have a wide range of applications in many fields such as astronomy, geography, physics, chemistry and environmental science.

3.2 ARFIMA Model and Regression

Typically, in the field of time series, people often use the classic ARFIMA (p, d, q) [11] to model the long memory feature of certain financial capital's time duration $(-0.5 \leqslant d \leqslant 0.5)$. However, the classic ARFIMA (p, d, q) model does not take many other factors that affects the time duration into account. Meanwhile, PCD model [6] though considers the effects from other factors, it ignores the long memory feature of the time duration itself. In order to study the relationship between other factors and the time duration of financial assets with long memory, CAO Zhiqiang [12] combined ARFIMA (0, d, 0) and regression model then got the following long memory regression model:

$$(1-B)^d ln(\triangle t_i) = c + \alpha S_{i-1} + \beta N_{i-1} + e_i \,, \tag{1}$$

where B is the delay operator, e_i is a short memory smooth sequence with a mean of zero. α, β, c, and d are the parameters to be estimated, d can be decomposed into $d = m + \delta$, m is an integer greater than or equal to zero with $\delta \in (-0.5, 0.5)$. Here the estimated time duration must be a positive number as there is a log transform on the left side of the model.

However, as is discussed in section 3.3, if the model (1) is not applicable to the high frequency trading data of the Kweichow Moutai Co Ltd (SHA: 600519) from January 4th, 2013 to February 26th, 2014, i.e.,

the parameter α or β is not significant, we have to change the model. In order to adjust the model for our own data, we add cross terms and quadratic terms with regard to S_{i-1}, N_{i-1} that affect the time durations to the right side of the model (1). The new model (2) goes as follows.

$$(1-B)^d ln(\triangle t_i) = c + \alpha_1 S_{i-1} + \alpha_2 N_{i-1} + \beta_1 S_{i-1}^2 + \beta_2 N_{i-1}^2 + \beta_3 S_{i-1} N_{i-1} + e_i \,. \qquad (2)$$

3.3 ARFIMA Model Fitting and Checking

As it is very difficult for us to figure out the maximum likelihood function of estimated parameters, we therefore give up on the maximum likelihood method but use other method instead to estimate the parameter in the model (1) or (2). We can use the profile-least squares method or use the ARFIMA package in R directly.

3.3.1 Profile-Least Squares Method

The idea of this method is similar to which of Jan Beran's [13]. We give the brief introduction as followings: given d, we estimate the remaining parameters using least squares estimation method and calculate of the estimated residual variance $\widehat{\sigma}_e^2(d)$. Then we set the value of d within a range and increase little by little. Therefore the d which gives out the minimum $\widehat{\sigma}_e^2(d)$ is our estimated value of d. This method is very intuitive and easy to implement using R.

As the $lag(n)$ autocorrelation function of $ln(\triangle t_i)$ is not 0 significantly and no obvious trend is observed from its own timing chart, we can conclude that $d = 0 + \delta \in (-0.5, 0.5)$. Besides, as the P-value of the t test on α_1 in model (1)is larger than 0.05, and therefore we consider to fit the data into the following model (2). We take d from -0.5 to 0.5 by the step of 0.02, getting the following result in R.

```
> order(sigma2)
 [1] 37 36 38 35 39 34 40 41 33 42 43 32 44 45 31 46 47 30 48 49 50 29 51
[24] 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10  9  8  7
6
[47]  5  4  3  2  1
```

As a result, the 37^{th} one has the minimum $\widehat{\sigma}_e^2(d)$, and therefore, $d = -0.5 + 0.02 * (37 - 1) = 0.22$. According to the definition of fraction difference proposed by Granger, Joyeux [14] and Hosking [15], we have

$$(1-B)^\delta = \sum_{k=0}^{\infty} b_k(\delta) B^k \,,$$

where

$$b_k(\delta) = (-1)^k \frac{\Gamma(\delta+1)}{\Gamma(k+1)\Gamma(\delta-k+1)}\,.$$

Here $\Gamma(z)$ is the gamma function with $\Gamma(z) = \int_0^\infty \frac{t^z-1}{e^t}dt$. The remaining parameters could now be easily estimated. We estimate them in R, getting the following results as is shown in Table 3.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	0.066833683	0.001378441	48.485	< 2e-16
st1	-0.002949702	0.014511981	-0.203	0.83893
nt	-0.040695953	0.000386792	-105.214	< 2e-16
st2	-0.014885260	0.005129479	-2.902	0.00371
nt2	0.000200261	0.000004421	45.302	< 2e-16
snt	-0.004813683	0.002913033	-1.652	0.09844

Table 3: Parameters in Model2

As is shown in the results, the P-value of the t test on α_1 and β_3 is larger than 0.05, and therefore we consider to fit the data into the following model (3).

$$(1-B)^d ln(\triangle t_i) = c + \alpha_2 N_{i-1} + \beta_1 S_{i-1}^2 + \beta_2 N_{i-1}^2 + e_i\,. \tag{3}$$

The result is shown in Table 4.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	0.06683717	0.00137843	48.488	< 2e-16
nt	-0.04069426	0.00038676	-105.218	< 2e-16
st2	-0.01494172	0.00512219	-2.917	0.00353
nt2	0.00019879	0.00000433	45.906	< 2e-16

Table 4: Parameters in Model3

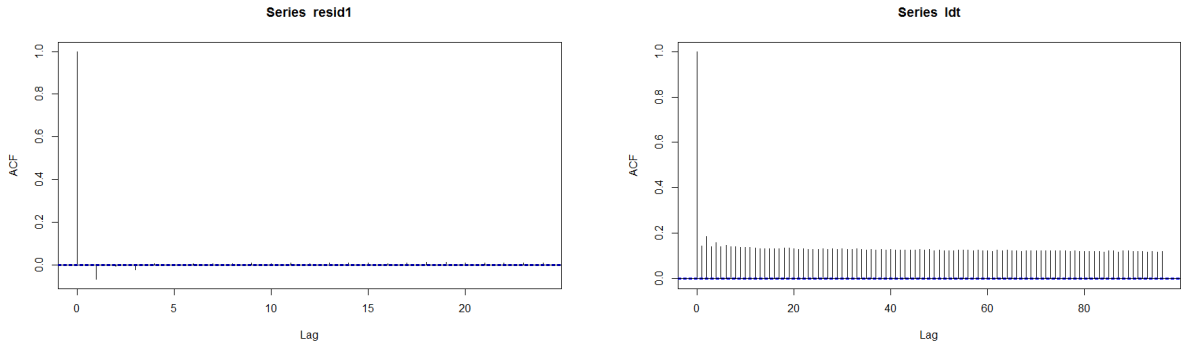
All parameters pass the t-test. The model can be written as

$$(1-B)^{0.22} ln(\triangle t_i) = 0.06683717 - 0.04069426 N_{i-1} - 0.01494172 S_{i-1}^2 + 0.00019879 N_{i-1}^2 + e_i\,, \tag{4}$$

where $(1-B)^{0.22} = \sum_{k=0}^{\infty} (-1)^k \frac{\Gamma(1.22)}{\Gamma(k+1)\Gamma(1.22-k)} B^k$.

Here $\Gamma(z)$ is the gamma function and e_i is a short memory smooth sequence with a mean of zero.

Finally, we need to do the model checking in order to see whether e_i has a long memory phenomenon or not. We can compare the two graphics in Figure 14, finding that e_i has no long memory phenomenon now. Therefore, we can draw a conclusion that the model (4) is somewhat reasonable.



(a) No Long Memory (b) Long Memory

Figure 14: Before and After ARFIMA

3.3.2 Using AFIMA Package in R to Calculate d

The other method is to use the AFIMA Package in R to calculate the value of d in model (1).

The result shows that $d = 0.139462$, which is quite different from the results estimated by Profile-Least Squares Method that $d = 0.22$.

We then estimate the remaining parameters in R, getting the following results as is shown in Table 5.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	0.0534654	0.0015391	34.74	<2e-16
st	-0.4684479	0.0160741	-29.14	<2e-16
nt	-0.0181302	0.0002946	-61.54	<2e-16

Table 5: Parameters in Model1

As is shown in the results, the P-value of the t test on α and β is very small, and therefore we do not need to add any cross terms or quadratic terms with regard to S_{i-1}, N_{i-1}. The model (1) is already adequate.

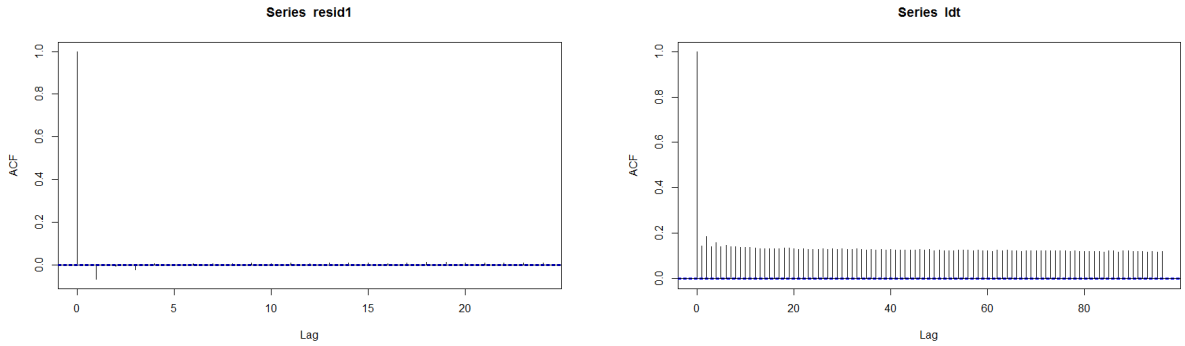
The model can be written as

$$(1-B)^{0.139462} ln(\triangle t_i) = 0.0534654 - 0.4684479 N_{i-1} - 0.0181302 S_{i-1} + e_i \,, \tag{5}$$

where $(1-B)^{0.139462} = \sum_{k=0}^{\infty} (-1)^k \frac{\Gamma(1.139462)}{\Gamma(k+1)\Gamma(1.139462-k)} B^k$.

Here $\Gamma(z)$ is the gamma function and e_i is a short memory smooth sequence with a mean of zero.

Finally, we need to do the model checking in order to see whether e_i has a long memory phenomenon or not. We can compare the two graphics in Figure 15, finding that e_i has no long memory phenomenon now. Therefore, we can draw a conclusion that the model (5) is also somewhat reasonable.



(a) No Long Memory

(b) Long Memory

Figure 15: Before and After ARFIMA

3.3.3 Comments on the Two Methods

We have used two methods in section 3.31 and section 3.32 respectively, getting two different values of d and two different models. Even though both of the models sound reasonable, we finally choose the second one. The reasons go as follows.

In section 3.31, as the parameter α is not significant, we add cross terms and quadratic terms with regard to S_{i-1}, N_{i-1} to the right side of the model (1) in order to show how the price would affect the time durations. However, there exist some problems when using the Least Squares Method to estimate a model with quadratic terms. Therefore, if we use the Profile-Least Squares Method to estimate d, we would either face some problems when estimating or ignore the influence of price on time duration.

Fortunately, if we use the ARFIMA package in R directly, the problem is solved. Both the parameter α and the parameter β are significant. We therefore use the second method to estimate our model. The final model goes as follows.

$$(1-B)^{0.139462} ln(\triangle t_i) = 0.0534654 - 0.4684479 N_{i-1} - 0.0181302 S_{i-1} + e_i \,,$$

where $(1-B)^{0.139462} = \sum_{k=0}^{\infty} (-1)^k \frac{\Gamma(1.139462)}{\Gamma(k+1)\Gamma(1.139462-k)} B^k$.

Here $\Gamma(z)$ is the gamma function and e_i is a short memory smooth sequence with a mean of zero.

3.4 ARIMA GARCH Model on Residuals e_i

Notice that even though the residuals e_i do not have long memory phenomenon, e_i are not white noise series. When we do the Ljung Box Test on e_i and e_i^2 respectively, the results shows that both P-values are near to 0. As a result, there is not only serial correlation but also ARCH effect in the series of e_i. Therefore, we would like to fit an ARIMA GARCH model to the residuals e_i. Note that the mean of e_i should be equal to 0.

We have tried the different ARIMA GARCH models according to the ACF and PACF of e_i, getting the result that the ARMA(2,1)-GARCH(1,0) model with the residuals ϵ follows normal distribution has the least AIC. According to R, the estimated parameters go as Table 6. As all p-values are extremely close to 0, all parameters pass t-test.

	Estimate	Std. Error	t value	Pr(>|t|)
ar1	0.926741	0.000067	13887.851	0
ar2	0.073041	0.000055	1339.097	0
ma1	-0.996010	0.000010	-100686.523	0
omega	0.466424	0.001411	330.518	0
alpha1	0.036881	0.002012	18.332	0

Table 6: ARMA(2,1)-GARCH(1,0)

Therefore, the model of e_i could be written as

$$\begin{aligned} e_i &= 0.926741e_{i-1} + 0.073041e_{i-2} + a_i - 0.996010a_{i-1} \\ a_i &= \epsilon_i\sigma_i \\ \sigma_i^2 &= 0.466424 + 0.036881a_{i-1}^2\,, \end{aligned}$$

where $\epsilon_i \sim whitenoise(0, 1)$.

Fortunately, when we make the model checking by using the Ljung Box test, the results go as Table 7. As all the p-value are larger than 0.05, we cannot reject any of the two $\mathbb{H}_0$. We can conclude that ϵ_i are white noises with no ARCH effect.

variable	X-squared	df	p-value
residepsi	18.613	12	0.09832
residepsi2	15.625	12	0.209

Table 7: Ljung Box Test

The final model of $\triangle t_i$ that we build goes as follows.

$$\begin{aligned}(1-B)^{0.139462}ln(\triangle t_i) &= 0.0534654 - 0.4684479N_{i-1} - 0.0181302S_{i-1} + e_i \\ e_i &= 0.926741e_{i-1} + 0.073041e_{i-2} + a_i - 0.996010a_{i-1} \\ a_i &= \epsilon_i\sigma_i \\ \sigma_i^2 &= 0.466424 + 0.036881a_{i-1}^2\,,\end{aligned}$$

where $\epsilon_i \sim whitenoise(0,1)$, $(1-B)^{0.139462} = \sum_{k=0}^{\infty}(-1)^k\frac{\Gamma(1.139462)}{\Gamma(k+1)\Gamma(1.139462-k)}B^k$, and $\Gamma(z)$ is the gamma function.

3.5 Forecast

According to our final model, we can forecast the $(1-B)^{0.139462}ln(\triangle t_i(1)) = 0.0534654 - 0.4684479N_i - 0.0181302S_i + e_i(1)$, where $e_i(1) = 0.926741e_i + 0.073041e_{i-1} + a_i - 0.996010a_i$.

Besides, the 1-step 95% forecasting interval of $e_i(1)$ is $(e_i(1) - 1.96\sigma_i(1), e_i(1) + 1.96\sigma_i(1))$, where $\sigma_i(1)^2 = 0.466424 + 0.036881a_i^2$.

The forecasting result goes as Table 8.

	e	$(1-B)^d ln(\triangle t)$	$ln(\triangle t)$	$\triangle t$
forecast value	0.4461	0.393037815	-2.3606	0.09436
low boundary	-0.897284	-0.950346185	-3.7041	0.024622
high boundary	1.789484	1.736421815	-1.0171	0.361642

Table 8: Forecast

4 Conclusion

In our report, we focus on the high frequency trading data of Kweichow Moutai Co Ltd (SHA: 600519) from January 4th, 2013 to February 26th, 2014 from the market makers' perspective. Market makers are interested in two kind of information from the high frequency trading data.

Firstly, market makers are certainly interested in how the mid-quote price would change. However, even though the log return series are stationary according to the Dickey-Fuller Unit Root Test, both ARIMA model and ARIMA GARCH model cannot fit the data of mid-quote price well. There exists serial correlation and ARCH effect in log return series, while only ARCH effect can be removed by ARIMA GARCH model. Under no models can the serial correlation be removed. The result means that ARIMA model is not a good model for the high frequency trading data. Fortunately, market makers are often employed by the order-driven markets to quote prices continuously for the well being of the market, and therefore, market makers do not assume the goal of making money.

Secondly, market makers are even more interested in the duration time during which the mid-quote price remains unchanged. Market makers like this kind of situation, because they can simply apply the strategy of "buy low, sell high". During the time duration that the bid one and ask one price stay unchanged, market makers can automatically buy at bid one price and sell at ask one price, making money at the same time when they facilitate the price discovery process of an exchange. If the mid-quote price tends to change, market makers cannot simply apply a fixed strategy. Fortunately, a ARFIMA and regression model can be fitted to the duration time during which the mid-quote price remains unchanged. Market makers can use the model to forecast the duration time.

Admittedly, the forecast value of $\triangle t_i$ might be quite small but still useful. Since low-latency communication technologies at both the software level and hardware level significantly contributed the development of algorithmic trading as a trading practice. Market makers can write a program to forecast the "Market Microstructure Characteristics Time" and use it to determine the algorithmic trading strategies.

References

[1] Wood R A. Market microstructure research databases: history and projections [J]. Journal of Business and Economic Statistics, 2000, 18(2): 140.

[2] Eric Ghysels. Some econometric recipes for high frequency data cooking [J]. Journal of Business and Economic Statistics, 2000, 18(2): 154.

[3] LUO Zhongzhou, YIN Hang, LIU Minglei. HFT market research and its application in China [J]. SSE Joint Research Program Report, 2012(23): 1.

[4] Zhang M Y, Russell J R, Tsay R S. Determinants of bid and ask quotes, and implications for the cost of trading [J]. Journal of Empirical Finance, 2008, 15(4): 656.

[5] Engle R F, Russell J R. Autoregressive conditional duration: a new model for irregularly spaced transaction data [J]. Econometrics, 1998, 66(5): 1127.

[6] McCulloch R E, Tsay R S. Nonlinearity in high frequency financial data and hierarchical models [J]. Studies in Nonlinear Dynamics and Econometrics, 2000, 5(1): 1.

[7] Mandelbrot B B, van Ness JW.Fractional Brownian motions, fractional noises and applications [J]. Society for Industrial and Applied Mathematics Review, 1968, 10(4): 422.

[8] Mandelbrot B B. New methods in statistical economy [J]. Journal of Political Economy, 1963, LXXI(5): 421.

[9] Mandelbrot B B. Self-similar error clusters in communication systems and the concept of conditional stationarity [J]. IEEE Transactions on Communication Technology, 1965, COM(13): 71.

[10] Mandelbrot B B. Sporadic random functions and conditional spectral analysis: self-similar examples and limits [C]. Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 3: Physical Sciences, University of California Press, Berkeley, 1967: 155-179.

[11] Craig Ellis.Estimation of the ARIMA (p, d, q) fractional differencing parameter (d) using the classical rescaled adjusted range technique [J]. International Review of Financial Analysis, 1999,8 (1): 53.

[12] CAO Zhiqiang, LI Hui, TONG Xingwei. Long Memory Regression Model and Investment Strategies of the IF1407 contract. Journal of Beijing Normal University (Natural Science), 2015,8,51 (4): 348.

[13] Jan Beran. Maximum likelihood estimation of the differencing parameter for invertible short and long memory autoregressive integrated moving average models [J]. Journal of the Royal Statistical Society, Series B (Methodological), 1995, 57(4): 659.

[14] Granger C W J, Joyeux R.An introduction to long range time series models and fractional differencing [J]. Journal of Time Series Analysis, 1980,1 (1): 15.

[15] Hosking JRM. Fractional differencing [J]. Biometrika, 1981,68 (1): 165.

Appendices

Appendix .A Data

The following data is part of our raw data.

	A	B	C
1	time	price	
2	930	210.84	
3	930	210.28	
4	930	210.84	
5	930	210.83	
6	930	210.28	
7	930	210.1	
8	930	210.05	
9	930	210	
10	930	210	
11	930	210	
12	930	210.05	
13	930	210.31	
14	930	210.22	
15	930	210.6	
16	930	210.5	
17	930	210.6	
18	930	210.6	
19	930	210.6	
20	930	210.6	
21	930	210.6	
22	930	210.6	
23	930	210.6	
24	930	210.84	
25	930	210.84	
26	930	210.84	
27	930	210.84	

After we treat the raw data, the settled data go as follows. The first table is the data of log return price. The second one is the data of time duration that mid-quote price keeps unchanged.

	newdate	ave_price	obs	lg_rt
1	20130104-93001	210.2625	2	-0.001413898
2	20130104-93002	210.87820513	3	0.0029239898
3	20130104-93003	211.044	4	0.0007859026
4	20130104-93004	210.758	5	-0.001356087
5	20130104-93005	211.725	6	0.004577707
6	20130104-93006	210.71166667	7	-0.004797573
7	20130104-93007	210.42875	8	-0.001343574
8	20130104-93009	211	9	0.0027110175
9	20130104-93011	210.95	10	-0.000236995
10	20130104-93015	210.05	11	-0.004275541
11	20130104-93016	210.05	12	0
12	20130104-93017	210.05	13	0
13	20130104-93019	210.05	14	0
14	20130104-93021	210.84	15	0.0037539544
15	20130104-93022	211.1025	16	0.0012442455
16	20130104-93024	210.05	17	-0.0049982
17	20130104-93026	210.03	18	-0.00009522
18	20130104-93028	210.00090909	19	-0.000138518
19	20130104-93029	209.90333333	20	-0.000464752
20	20130104-93030	209.74666667	21	-0.000746654
21	20130104-93031	209.696	22	-0.00024159
22	20130104-93032	209.62	23	-0.000362495
23	20130104-93033	209.81	24	0.0009059915
24	20130104-93034	209.62	25	-0.000905992
25	20130104-93035	209.8	26	0.0008583282
26	20130104-93036	209.68	27	-0.000572137
27	20130104-93037	209.62	28	-0.000286191
28	20130104-93038	209.81	29	0.0009059915
29	20130104-93039	209.62	30	-0.000905992
30	20130104-93041	209.57	31	-0.000238555
31	20130104-93042	209.5	32	-0.000334073
32	20130104-93044	209.815	33	0.0015024507
33	20130104-93045	209.44333333	34	-0.001772973
34	20130104-93046	209.605	35	0.0007715896
35	20130104-93047	209.62	36	0.0000715606
36	20130104-93048	209.1115	37	-0.002428765
37	20130104-93050	209.006	38	-0.000504643

	A	B	C	D	E
1		d_t	s_t	n_t	delta
2	1	0.005682	0.56	0	-1
3	2	0.005682	0.56	0	1
4	3	0.005682	0.01	0	-1
5	4	0.005682	0.55	0	-1
6	5	0.005682	0.18	0	-1
7	6	0.005682	0.05	0	-1
8	7	0.005682	0.05	0	-1
9	8	0.017047	0.05	2	1
10	9	0.005682	0.26	0	1
11	10	0.005682	0.09	0	-1
12	11	0.005682	0.38	0	1
13	12	0.005682	0.1	0	-1
14	13	0.005682	0.1	0	1
15	14	0.039777	0.24	6	1
16	15	0.051141	0.05	8	1
17	16	0.005682	0.91	0	1
18	17	0.005682	1.58	0	-1
19	18	0.005682	0.66	0	1
20	19	0.011365	0.92	1	1
21	20	0.005682	0.8	0	-1
22	21	0.005682	0.12	0	-1
23	22	0.017047	0.92	2	1
24	23	0.005682	0.92	0	-1
25	24	0.005682	0.3	0	1
26	25	0.017047	0.62	2	1
27	26	0.005682	0.92	0	-1

Appendix .B R Codes in Section 2

```
# ARCH effect
> at=lgrt-mean(lgrt)
> Box.test(at^2,lag=12,type='Ljung')

        Box-Ljung test

data:  at^2
X-squared = 2378.4, df = 12, p-value < 2.2e-16

> Box.test(arima(lgrt,order = c(1,0,0))$resid^2,lag=12,type = 'Ljung')

        Box-Ljung test

data:  arima(lgrt, order = c(1, 0, 0))$resid^2
X-squared = 14794, df = 12, p-value < 2.2e-16

# ARMA-GARCH model

> fitq1=garchFit(lgrt~arma(1,0)+garch(1,1),data=lgrt,trace=F)
> summary(fitq1)

Title:
 GARCH Modelling

Call:
 garchFit(formula = lgrt ~ arma(1, 0) + garch(1, 1), data = lgrt,
    trace = F)

Mean and Variance Equation:
 data ~ arma(1, 0) + garch(1, 1)
<environment: 0x1212f7ca0>
```

```
[data = lgrt]

Conditional Distribution:
 norm

Coefficient(s):
        mu          ar1        omega       alpha1        beta1
-3.1993e-05  -3.0703e-01  4.9455e-09  2.1421e-01  7.8991e-01

Std. Errors:
 based on Hessian

Error Analysis:
         Estimate  Std. Error  t value Pr(>|t|)
mu     -3.199e-05   1.143e-05   -2.800  0.00511 **
ar1    -3.070e-01   4.191e-02   -7.326 2.37e-13 ***
omega   4.945e-09   1.160e-09    4.262 2.03e-05 ***
alpha1  2.142e-01   3.503e-02    6.114 9.69e-10 ***
beta1   7.899e-01   2.484e-02   31.802  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Log Likelihood:
 4387.835    normalized:  6.268335

Description:
 Sun May  1 21:14:33 2016 by user:


Standardised Residuals Tests:
                                Statistic p-Value
 Jarque-Bera Test   R    Chi^2  226.6154  0
```

```
 Shapiro-Wilk Test  R     W       0.956208   1.386691e-13
 Ljung-Box Test     R     Q(10)   28.83545   0.001324794
 Ljung-Box Test     R     Q(15)   32.80012   0.005001912
 Ljung-Box Test     R     Q(20)   49.91691   0.0002276229
 Ljung-Box Test     R^2   Q(10)   19.5223    0.03410873
 Ljung-Box Test     R^2   Q(15)   27.39051   0.02571104
 Ljung-Box Test     R^2   Q(20)   28.32919   0.101833
 LM Arch Test       R     TR^2    24.21535   0.01901148

Information Criterion Statistics:
      AIC       BIC       SIC      HQIC
-12.52238 -12.48988 -12.52249 -12.50982
```

```
# GARCH model
> fitq2=garchFit(~garch(1,1),data=lgrt,trace=F)
> summary(fitq2)

Title:
 GARCH Modelling

Call:
 garchFit(formula = ~garch(1, 1), data = lgrt, trace = F)

Mean and Variance Equation:
 data ~ garch(1, 1)
<environment: 0x1215fa6a8>
 [data = lgrt]

Conditional Distribution:
 norm

Coefficient(s):
```

```
        mu         omega        alpha1         beta1
-2.6592e-05  5.4495e-09  2.2163e-01  7.8535e-01

Std. Errors:
 based on Hessian

Error Analysis:
         Estimate  Std. Error  t value Pr(>|t|)
mu     -2.659e-05   1.151e-05   -2.310   0.0209 *
omega   5.450e-09   1.234e-09    4.416 1.01e-05 ***
alpha1  2.216e-01   3.556e-02    6.233 4.59e-10 ***
beta1   7.853e-01   2.449e-02   32.066  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Log Likelihood:
 4365.56     normalized:  6.236515

Description:
 Sun May  1 21:14:33 2016 by user:


Standardised Residuals Tests:
                                Statistic  p-Value
 Jarque-Bera Test   R     Chi^2  262.0727  0
 Shapiro-Wilk Test  R     W      0.9447482 1.744111e-15
 Ljung-Box Test     R     Q(10)  47.44318  0.0000007841877
 Ljung-Box Test     R     Q(15)  50.55803  0.00000975758
 Ljung-Box Test     R     Q(20)  64.81859  0.000001247344
 Ljung-Box Test     R^2   Q(10)  21.65193  0.01697939
 Ljung-Box Test     R^2   Q(15)  29.73687  0.0129065
 Ljung-Box Test     R^2   Q(20)  31.49438  0.04899305
```

LM Arch Test	**R**	TR^2	28.22129	0.005134042

Information Criterion Statistics:

AIC	BIC	SIC	HQIC
−12.46160	−12.43559	−12.46167	−12.45155

Appendix .C R Codes in Section 3

```
rm(list=ls(all=TRUE))
setwd("C:/Users/Ted/Desktop/TSProject/Rcodes")
xgsj=read.csv("midquote.csv",header=T)
time=xgsj[,1]
un=length(unique(time));un  #different time
sjbh=cbind(time,xgsj[,2])
date=xgsj[,3]
colnames(sjbh)=c("time","price")
#import the origin data
logtime=log(sjbh[,1])
price=sjbh[,2]
plot.ts(price)
#change the time from factor to numeric
t.set=unique(sjbh[,1]) #set of time
nt=numeric(un)
for(i in 1:un){
  nt[i]=sum(t.set[i]==sjbh[,1])
}
plot(ts(nt),xlab="trade minute",ylab="trade times in one minute")

#no difference between morning and afternoon
tt=NULL
for(i in 1:un){
  temp.t=seq((i-1)*60,i*60-0.5,length=nt[i])
  tt=c(tt,temp.t)
}
data=cbind(tt,sjbh[,2])

pt=data[,2]  #trade price
dpt=diff(pt)  #dp[t]=p[t]-p[t-1]
index=which(dpt!=0)
```

```
ct=tt[c(1,index+1)]  #that's ct be the calendar time
# of the ith price change of an asset
d_t=diff(ct)
s_t=abs(dpt[index])  #size of the ith price change measured in ticks
n_t=c(0,diff(index)-1) #the number of trades in the time interval(t[i-1],t[i])
m=length(n_t);m
delta=numeric(m)
i_up=which(dpt[index]>0)
i_down=which(dpt[index]<0)
delta[i_up]=1
delta[i_down]=-1
newdata519=cbind(d_t,s_t,n_t,delta)
write.csv(newdata519,file="C:/Users/Ted/Desktop/TSProject/Rcodes/quote.csv")
setwd("C:/Users/Ted/Desktop/TSProject/Rcodes")
ndata=read.csv("trade600519.csv",header=TRUE)
dt1=ndata[,2]
st1=ndata[,3]
nt1=ndata[,4]
delta1=ndata[,5]
index=dt1!=0
dt=dt1[index]
st=st1[index]
nt=nt1[index]
delta=delta1[index]
n=length(dt)
dt=dt[2:n]
st=st[1:(n-1)]
nt=nt[1:(n-1)]
delta=delta[1:(n-1)]
st1=st*delta
st2=st1^2
nt2=nt^2
```

```
snt=st1*nt
n=n-1
ldt=log(dt)
md=mean(ldt)
lldt=ldt-md
cor(cbind(ldt,st1,nt,st2,nt2,snt))

plot.ts(lldt)
acf(ldt,lag.max = 96,plot = TRUE)

#MLE
require(fracdiff)
arfima1=fracdiff(ldt,nar=0,nma=0,drange = c(0,0.5))
arfima1
d=0.139462
#Granger,Joyeux,Hosking
options(scipen = 3)
nb=(-1)^(i-1)*gamma(d+1)/(gamma(i-1+1)*gamma(d-i+1+1))
for(j in 1:n){
  if(j<=m1) {
    ed[j]=sum(nb[1:j]*rev(ldt[1:j]))-sum(nb[1:j])*md
  }
  else   ed[j]=sum(nb*rev(ldt[(j-m1+1):j]))-sum(nb)*md
}
lm.reg1=lm(ed~st+nt)
summary(lm.reg1)

#LSE
delta=seq(-0.5,0.5,by=0.02)
m=length(delta)
sigma2=numeric(m)
ed=numeric(n)
```

```
m1=171
nb=numeric(m1)
i=1:m1

for(k in 1:m){
  if(k!=26){
    nb=(-1)^(i-1)*gamma(delta[k]+1)/(gamma(i-1+1)*gamma(delta[k]-i+1+1))
    for(j in 1:n){
      if(j<=m1) {
        ed[j]=sum(nb[1:j]*rev(ldt[1:j]))-sum(nb[1:j])*md
      }
      else   ed[j]=sum(nb*rev(ldt[(j-m1+1):j]))-sum(nb)*md
    }
    lm.reg=summary(lm(ed~nt+st2+nt2+snt))
    sigma2[k]=lm.reg$sigma
  }
  else{
    lm.reg=summary(lm(ldt~nt+st2+nt2+snt))
    sigma2[k]=lm.reg$sigma
  }
  cat("iteration=␣",k,"\n")
}
order(sigma2)
#Granger, Joyeux, Hosking
options(scipen = 3)
k=37
nb=(-1)^(i-1)*gamma(delta[k]+1)/(gamma(i-1+1)*gamma(delta[k]-i+1+1))
for(j in 1:n){
  if(j<=m1) {
    ed[j]=sum(nb[1:j]*rev(ldt[1:j]))-sum(nb[1:j])*md
  }
  else   ed[j]=sum(nb*rev(ldt[(j-m1+1):j]))-sum(nb)*md
```

```
}
lm.reg1=lm(ed~nt+st2+nt2)
summary(lm.reg1)

#residuals analysis
resid=resid(lm.reg1)
acf(resid,lag.max = 24,plot = TRUE)
plot(ts(resid))
boxplot(resid)
fivenum(resid)
fu=0.4764604
fl=-0.5965162   #MLE
fu=0.4721599
fl=-0.6038488
df=fu-fl
ol=fl-3*df
ou=fu+3*df
ol1=fl-1.5*df
ou1=fu+1.5*df
resid1=resid[resid<=ou1&resid>=ol1]
length(resid1)/length(resid)

require(FinTS)
AutocorTest(resid1)
ArchTest(resid1)


acf(resid1,lag.max = 24,plot = FALSE)
pacf(resid1,lag.max = 24,plot = FALSE)

#rugarch
require(rugarch)
```

```
spec<-ugarchspec(variance.model=list(model="sGARCH",garchOrder=c(1,0)),
mean.model=list(armaOrder=c(2,1),include.mean=FALSE),
distribution.model = "norm")
garch3<-ugarchfit(spec=spec,data=resid1)
garch3
AutocorTest(garch3@fit$z)
AutocorTest((garch3@fit$z)^2)
ArchTest(garch3@fit$z)
ArchTest((garch3@fit$z)^2)

#fgarch
library(fGarch)
garch4=garchFit(resid1~arma(2,1)+garch(1,0),data=resid1,trace=F)
summary(garch4)

#epsilon
residepsi=residuals(garch3,standardize=TRUE)
boxplot(residepsi)
mean(residepsi)
var(residepsi)
acf(residepsi,lag.max = 24,plot = TRUE)
pacf(residepsi,lag.max = 12,plot = TRUE)
Box.test(residepsi,lag = 12,type = "Ljung")

#MLE predict
predictgarch3=ugarchforecast(garch3,data = NULL,n.ahead = 4,conf=.95)
predictgarch3

d=0.139462
newed=c(ed,0)
newldt=c(ldt,-2.3606)
nb=(-1)^(i-1)*gamma(d+1)/(gamma(i-1+1)*gamma(d-i+1+1))
```

```
for(j in 1:(n+1)){
  if(j<=m1) {
    newed[j]=sum(nb[1:j]*rev(newldt[1:j]))-sum(nb[1:j])*md
  }
  else   newed[j]=sum(nb*rev(newldt[(j-m1+1):j]))-sum(nb)*md
}
rev(newed)[1]

0.393037815
-0.950346185
1.736421815

#LSE predict
predictgarch3=ugarchforecast(garch3,data = NULL,n.ahead = 4,conf=.95)
predictgarch3

newed=c(ed,0)
newldt=c(ldt,-2.31592)
k=37
nb=(-1)^(i-1)*gamma(delta[k]+1)/(gamma(i-1+1)*gamma(delta[k]-i+1+1))
for(j in 1:(n+1)){
  if(j<=m1) {
    newed[j]=sum(nb[1:j]*rev(newldt[1:j]))-sum(nb[1:j])*md
  }
  else   newed[j]=sum(nb*rev(newldt[(j-m1+1):j]))-sum(nb)*md
}
rev(newed)[1]

0.147155676
-1.203088324
1.497399676
```