5-2016

Microstructure Noise: The Use of Two Scales Realized Volatility for the Noisy High-Frequency Data and its Implications for Market Efficiency and Financial Forecasting

Aristides Romero
Utah State University

MICROSTRUCTURE NOISE: THE USE OF TWO SCALES REALIZED VOLATILITY FOR NOISY HIGH-FREQUENCY DATA AND ITS IMPLICATIONS FOR MARKET EFFICIENCY AND FINANCIAL FORECASTING

by

Aristides A. Romero Moreno

A thesis submitted in partial fulfillment
of the requirements for the degree

of

MASTER OF ARTS

in

Economics

Approved:

_____	_____
Dr. Tyler Brough Major Professor	Dr. Devon Gorry Committee Member

_____
Dr. James Feigenbaum
Committee Member

_____
Dr. Mark McLellan
Vice President for Research and
Dean of the School of Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2016

ABSTRACT

As a basic principle in statistics, a larger sample size is preferred whenever possible. Nonetheless, in the financial world, especially equities and currencies trading, including all available data poses great challenges due to the noise present in the volatility estimation. In his paper I examine the Two Time Scales Realized Volatility estimator by Zhang, Mykland, and Ait-Sahalia (2005b) and I find that it not only provides a more efficient estimator than a basic estimator of the integrated volatility of returns, but it also consistently estimates the microstructure noise present in the latent efficient return process. I find that by using this approach, it is possible to compare the efficiency of the prices of securities with lower transaction costs traded against those with higher transactions costs.

by

Aristides A. Romero Moreno, Master of Arts

Utah State University, 2016

Major Professor: Dr. Tyler Brough
Department: Economics and Finance

(36 pages)

PUBLIC ABSTRACT

Microstructure Noise: The Use of Two Scales Realized Volatility for Noisy High-Frequency Data and its Implications for Market Efficiency and Financial Forecasting

This paper examines a proposed approach for integrated volatility and its implications for the informational efficiency in particular stocks and the use of the estimator for financial forecasting and market efficiency.

ACKNOWLEDGMENTS

I would like to thank Dr. Tyler Brough for his sharing my excitement about such a fascinating topic and for making available to me the initial computer code used for the simulations. I would specially like to thank my committee members: Dr. Devon Gorry, for her helpful comments and support as well as her passion in teaching Econometrics inspired me into pursuing the field further along with other statistical techniques and computer programs; Dr. James Feigenbaum, I always enjoyed a challenge, and I saw my limits tested in your class. It was most enjoyable and instructive. I am honored to have them as committee members. Thank you to their support and assistance through-out the entire process.

I would also like thank Dr. Yacine Aït-Sahalia, Princeton University, for graciously providing me his code for the TSRV estimator.

I give special thanks to my family, friends (near and far), and colleagues for their encouragement, moral support, and patience as I worked my way from the initial proposal writing to this final document. I could not have done it without all of you.

Aristides A. Romero Moreno

LIST OF TABLES

CONTENTS

Page

I. INTRODUCTION

As a basic principle in statistics, sampling more frequently is preferred whenever possible. Most fields of study, however, are constrained by the amount and frequency of the data in question, in finance this is less of a problem. Trading data, particularly stocks and currencies, are abundant in the financial world. Data is generated in a sub-second time series, thus making it possible to sample at higher frequencies. One obvious application is to estimate the variance (realized volatility), or integrated volatility given the frequency of the data, of stock returns using the sum of log squared returns. Nonetheless, the problem with this approach is that when trying to estimate the variance, using all available high-frequency data, the data are noisy. This is known in the literature[1] as market microstructure noise; a deviation from the fundamental value of a security. Microstructure noise arises from the trading process. The bid-ask spread is an example.

In this paper I focus on an estimator that would be consistent and unbiased when estimating volatility, in the presence of the market microstructure noise, using high-frequency data generated every second. Particularly, I focus focuses on equity returns.

Understanding volatility is of utmost importance in the financial markets because it has broad economic and financial implications. Asset managers, traders, investment advisors, banks and policy makers pay special attention to volatility in the financial markets and its repercussions to the general economy. Since volatility is widely used as a measure of risk, participants in the financial markets need to estimate the volatility of returns for investment decisions or transacting

[1] Andersen, Tobern G.; Benzoni, Luca (2008) "Realized Volatility." Federal Reserve Bank of Chicago.

in particular stocks or sectors of the market. Similarly, policy makers look at market volatility before implementing policies, such as monetary policy, that would create instability in the financial markets and broad economy, such as tightening or loosening of monetary policy.

While there are many estimators proposed, I focus on the Two Scales Realized Volatility (TSRV) estimator due to its novel approach of using all available data.[2] It is widely known in the literature, that the sum of squared log returns, $[X,X]_T \triangleq \sum_{t_i}(X_{t_{i+1}} - X_{t_i})^2$ (where the X_{t_i}'s represent all the observations of the return process in a single time interval from 0 to T), for high-frequency data, in the absence of market microstructure noise, should consistently estimate the integrated volatility of the return process. The integrated volatility, also known as the continuous quadratic variation, is the cumulative volatility over successive time periods. The continuous quadratic variation is expressed as $\langle X,X\rangle_T = \int_0^T \sigma^2$, where σ^2 is the instantaneous variance of the returns process X_t for time period [0,T]. In the next section, however, I explain why this approach fails to work in the presence of market microstructure noise.

This paper explores five volatility estimators to address the market microstructure noise problem and tests them empirically using a Monte Carlo simulation using the Heston model as the data generating process to simulate stock returns.

The main two areas of interest of this paper are: the use of the TSRV as a consistent and unbiased estimator of volatility with high-frequency data and the use of a by product of the "fifth-

2 Li et al. (2014); Camponovo et al. (2015); Ysusi et al. (2008); Yu (2014); Misaki (2015); Zhang et. al (2006). See references page for details.

best" estimator as a proxy for market microstructure noise.

To accomplish these tasks, before discussing a review of the literature and the motivation behind the estimators in section 2, the market microstructure noise is defined. Then, section 3 explains the mechanics of the model; section 4 examines the practical applications and implications; section 5 tests the model with a Monte Carlo study to check the robustness and consistency of the estimators. Section 6 addresses the use of an important ramification of the "fifth-best" estimator to measure the market microstructure noise, using the bid-ask spread as a measure for efficiency, in the components of the Dow Jones Industrial Average (DJIA); and section 7 concludes the study.

A. THE MICROSTRUCTURE NOISE

The market microstructure noise is a deviation of the fundamental price of a security. In the market microstructure literature, it known that high-frequency data are contaminated with this type of noise. The data contains, as most econometricians refer to, an "observation error"[3] component. In a very simple form, Zhang, Mykland, and Ait-Sahalia (2005b) model the return process and the microstructure noise as,

$$Y_{t_i} = X_{t_i} + \epsilon_{t_i} \tag{1}$$

where $Y_{t_i}$is the observed return process, X_{t_i} represents the fundamental or efficient price of the return process, and $\epsilon_{t_i}$is the independent error term capturing the noise of the true return. Many

[3] Zhang, Lan; Mykland, Per A.; Aït-Sahalia, Yacine (2005b). "A Tale of Two Time Scales: Determining Integrated Volatility With Noise High-Frequency Data."

researchers such as Ait-sahalia and Yu (2009) and Hasbrouck (1993) conclude that it is the source of noise is attributable to the trading process, while Roll (1984) argues it is due to the bid-ask spread. Security dealers may have access to distinct sources of information or be motivated to buy or sell securities for diverse factors. This behavior adds noise to the latent, true price of a security. Stocks with wider bid-ask spreads, then, should exhibit more market microstructure noise than stocks with narrower bid-ask spreads. To test this hypothesis, I use the Two Time Scale Realized Volatility (TSRV) estimator, an estimator developed by Zhang, Mykland, and Ait-Sahalia (2005b) aimed at using high-frequency data to estimate the realized volatility of the return process. Particularly, in assessing the market microstructure noise, a by product of one of the estimators, the "fifth-best" estimator, is used to estimate the noise contained in the the constituents of the Dow Jones Industrial Average (DJIA).

II. LITERATURE REVIEW

As financial data becomes more abundant due to high-frequency trading, more researchers have made realized volatility a center of focus. For instance, Campnovo et al. (2015) developed a nonparametric estimator for realized volatility using high-frequency data using conventional statistics and a Pearson's chi-square for special cases. Additionally, Ysusi et al. (2008) highlight the issues when using the central limit theorem and high-frequency intra-day data and propose an absolute high-frequency return estimator.

More pertinent to this study, Misaki et al. (2015) suggest the use of a Separating Information Maximum Likelihood (SMIL) estimator. They show that after accounting for market microstructure noise and through random sample of high-frequency data, their estimator "is

consistent and has a stable convergence."[4] Others such as Yu et al.(2014) suggested a threshold kernel estimator that estimates the spot volatility when it is time-dependent and the latent price has been contaminated by finite activity jumps. While Yu et al. (2014) propose the use of this estimator in analyzing intra-day volatility patterns, they do not address the market microstructure noise component.

Zhang, Mykland, and Ait-Sahalia (2005b) construct five estimators to estimate volatility using high-frequency data. After a sample adjustment, they zero in one consistent and bias-adjusted estimator, "the first-best", to estimate the variance of returns. These five nonparametric estimators are the basis of this paper and will be discussed in detail in the next section.

Zhang et al. (2005a) propose a closed-form optimal sampling frequency, even when the noise distribution is misspecified, for high-frequency data by explicitly modelling the noise. Their estimator, they argue, possesses the same asymptotical properties as if it were correctly specified. Therefore, they recommend to sample as often as possible. Zhang (2006) proposed a generalized version of the TSRV, the MSRV[5] which expands the TSRV from a two period time scale to a multiple time scale approach.

Relying on the TSRV estimator, Aït-sahalia and Yu (2009) studied the fundamental and noise component of stock prices. Using different measures of liquidity, they found that more liquid stocks were priced more adequately; that is, their price contained a lower pricing error than stocks

[4] Misaki et al. (2015). "On robust properties of the SIML estimation of volatility under micro-market noise and random sampling." International Review of Economics and Finance.

[5] Zhang, Lan (2006). Efficient estimation of stochastic volatility using noisy observations: a multi-scale approach. Bernoulli Volume 12, Number 6 (2006), 1019-1043.

traded less often. Aït-Sahalia and Mykland (2009)[6] provide a didactic overview of the main properties and uses of the TSRV and MTSRV (Multi Time Scale Realized Volatility) estimators. Their paper lays out a summarized version of the two estimators aimed at explaining the estimators in its simplest form. This paper follows a similar approach by summarizing the five estimators developed by Zhang, Mykland, and Ait-Sahalia (2005b).

A. OVERVIEW OF THE ESTIMATORS

The Two Time Scale realized Volatility (TSRV) estimator reconciles the use of a continuous-time estimator, the sum of squared log returns, with a discrete-time estimator. The end goal of Zhang, Mykland, and Ait-Sahalia (2005b) is to estimate the realized volatility of returns using as much data as possible. This approach follows Zhang et al. (2005a)'s paper on the optimal sampling frequency. In their paper, they develop five estimators. They start with the most inadequate, the "fifth-best" estimator, and count down to the most appropriate and consistent, the "first-best" estimator.

The "fifth-best" estimator is based on theory, since the sum of squared log returns for high-frequency data, in the absence of microstructure noise, should consistently estimate the integrated volatility of the return process. The problem with transaction's data is that data are noisy, thus, making this estimator biased. This shortcoming is widely known among researchers and Zhang, Mykland, and Ait-Sahalia (2005b) provide a review of the literature.

[6] Aït-Sahalia, Yacine; Mykland, Per A. (2009). T.G. Anderson et al., Handbook of Financial Time Series, DOI: 10.1007/978-3-540-71297-8_25, © Springer-Verlag Berlin Heidelberg 2009

The "fourth-best" estimator is the approach most popular among researchers, which is sampling sparsely, say every 5 minutes. Zhang, Mykland, and Ait-Sahalia (2005b) argue that sparse sampling takes care of the microstructure noise in the return process, but throws away most of the data. For instance, if one were to sample every 5 minutes, one would only be using 78 out of 23,400 observations[7] available in a trading day. A general rule of statistical analysis is: never throw away data.

While the "third-best" estimator uses arbitrary sampling, the "third-best" estimator aims to find an optimal sampling interval while using more data. This estimator has good properties, but needs an adjustment discussed later in this section.

The "second-best estimator," seeks to use more data in the volatility estimation. A compelling way to use all the data is by combining the properties of the "fifth and third-best" estimators. The result is a blend of optimal sampling and averaging across subsamples. The construction of the subsamples is straight forward. Let $G^k, k = 1, \dots, K$, be the subsamples of the original data set where $\frac{n}{K} \to \infty$ as $n \to \infty$. Start at the first observation G^1 and record an observation every 5 minutes; for G^2, start at the second observation and record an observation every 5 minutes, and so on. The average the estimators obtained from the subsamples. Unfortunately, this estimator remains biased.

Finally, the "first-best" estimator combines the fifth and second-best estimator: using all

[7] One day of trading is 6.5 hours, as is the case of the NYSE and NASDAQ exchanges. With data generated every second, this would amount to 300 seconds/5 minutes; 23,400 observations in a day.

the data in the fifth estimator and the averaged subsamples in the second estimator. The result is a consistent estimator with a bias-adjustment.

The mechanics of theses estimators are explained as follows:

Fifth-Best Estimator

It is widely known in the market microstructure literature that the sum of log squared returns, in the absence of market microstructure noise, should consistently estimate the volatility of returns (McDonald, 2006). In the presence of market microstructure noise, however, the sum of log squared returns fails to estimate the true variance of the returns,$\langle X, X\rangle_T^{(all)}$, computed as

$[X, X]_T \triangleq \sum_{t_i}(X_{t_{i+1}} - X_{t_i})^2$, where the X_{t_t}'s are all trade observations in [0, T], and instead estimates the variance of the microstructure noise, $E\epsilon^2$, scaled by $(2n)^{-1}$. More formally, it estimates,

$$\sum_{t_i, t_{i+1} \in [0,T]}(Y_{t_{i+1}} - Y_{t_i})^2 = 2nE\epsilon^2 + O_p\left(n^{\frac{1}{2}}\right), \quad (2)$$

where *n* is the number of sampling interval over [0,T]. The first term is the variance of the noise the noise and the second term is, as Zhang, Mykland, and Ait-Sahalia (2005b) argue, the asymptotically Gaussian term, which dwarfs the integrated volatility, O_p. This estimator has undesirable properties. Nonetheless, section 4 discusses important ramifications from this estimator. From this point forward, this estimator will be referred to as $[Y, Y]_T^{(all)}$.

Fourth-Best Estimator

This is the approach used by most researchers. By taking a subsample of the observations ensures one does not ignore the market microstructure noise. Nonetheless, for 6.5 hours trading

day with data generated every second, a total of 23,400 observations are produced. If one were to only sample every 5 minutes, one would the throwing away 299 out of 300 observations. Following Zhang's et al. (2005b) notation, $[Y,Y]_T^{(sparse)}$ or the "fourth-best" estimator only uses 78 observations. This is the shortcoming this paper seeks to address.

Third Best Estimator

This estimator, $[Y,Y]_T^{(spase,opt)}$, follows the same intuition as the fourth-best estimator. The difference is that it seeks to find an optimal sampling method, quantitatively, instead of arbitrarily selecting a number. The optimal sampling method Zhang et al. (2005a,b) computed is given by

$$n_{sparse}^{*} = \left(\frac{T}{4(E\epsilon^2)^2}\int_0^T \sigma_t^4 dt\right)^{\frac{1}{3}} \tag{3}$$

Zhang, Mykland, and Ait-Sahalia (2005b) recognize that due to the sample size in (3) a higher-order adjustment is needed.

Second-Best Estimator

Even after developing an optimal sampling method, many observations are thrown out. The second-best estimator, $[Y,Y]_T^{(avg)}$, addresses this issue by averaging estimators, $[Y,Y]_T^{(k)}$ across *K* grids of size $\bar{n}$. Although this estimator remains biased, it is a better estimator than $[Y,Y]_T^{(all)}$ in terms of bias.

$$[Y,Y]_T^{(avg)} \underset{\approx}{\mathcal{L}} \langle X,X\rangle_T + 2\bar{n}E\epsilon^2 + \left[4\frac{\bar{n}}{K}E\epsilon^4 + \frac{4T}{3\bar{n}}\int_0^T \sigma_t^4\, dt\right]^{\frac{1}{2}} Z_{total}, \tag{4}$$

where the first term is the matrix of squared log returns, the second is the biased due to the noise. Z_{total} is standard normal, indicating noise. The first term in the squared brackets represents noise

and second term is the result of discretization; these two terms combined represent the total variance.[8] See Zhang (2005b) for details, derivation and bias reduction techniques.

First-Best Estimator

The first-best estimator, $\widehat{\langle \mathrm{X}, \mathrm{X} \rangle_{\mathrm{T}}}$, controls for the market microstructure noise and uses all of the data by combining the "fifth-best" and the "second-best" estimators; thus the "Two Scale" name. Further, Zhang, Mykland, and Ait-Sahalia (2005b) apply a bias-correction method. The "first-best" estimator becomes,

$$\widehat{\langle \mathrm{X}, \mathrm{X} \rangle_{\mathrm{T}}} = [\mathrm{Y}, \mathrm{Y}]_{\mathrm{T}}^{\mathrm{avg}} - \frac{\bar{\mathrm{n}}}{\mathrm{n}} [\mathrm{Y}, \mathrm{Y}]_{\mathrm{T}}^{\mathrm{all}} \tag{5}$$

$$\bar{n} = \frac{1}{K} \sum_{k=1}^{K} n_k = \frac{n - K + 1}{K} \tag{6}$$

where K is the number of subsamples, the first term comes from the "second-best" estimator, slow time scale, and the second term is the "fifth-best" estimator, fast time scale. After a sample adjustment,

$$\langle \widehat{X, X} \rangle_T^{adj} = \left(1 - \frac{\bar{n}}{n}\right)^{-1} \langle \widehat{X, X} \rangle_T, \tag{6}$$

The estimator (6) is unbiased of higher order than the estimator (5)[9].

III. BASIS FOR THE MODEL

Theoretically, one could use the sum of squared log returns to get the variance. Zhang et

[8] Zhang, Lan; Mykland, Per A.; Aït-Sahalia, Yacine (2005a). "A Tale of Two Time Scales: Determining Integrated Volatility with Noise High-Frequency Data." American Statistical Association.
[9] Ibid

al. (2005a,b) and Aït-Sahalia (2009) argue that doing so ignores the market microstructure noise contained in the latent return process. Had this not been the case, the sum of squared log returns,

$$[X, X]_T \triangleq \sum_{t_i} (X_{t_{i+1}} - X_{t_i})^2, \tag{7}$$

where the X_{t_t}'s are all trade observations in [0, T],[10] estimate the integrated realized volatility. Further, they argue that as the sampling frequency increases, in the limit,

$$plim \sum_{t_i} (X_{t_{i+1}} - X_{t_i})^2 = \int_0^T \sigma_t^2 dt \tag{8}$$

is the best estimator, theoretically, for the integrated volatility. Nonetheless, this method fails to account for the microstructure noise contained in high-frequency trade data. Despite this shortcoming, important implications can be derived from this estimator, which are discussed in sections 3.

A common practice among finance researchers is to sample sparsely, say for instance every 5 minutes[11]. This sampling method tries to ignore the microstructure noise by only taking into account some, but not all, of the data. While this method addresses the market microstructure noise component, it discards too much of the available data; this otherwise would be optimal. As mentioned previously, in a normal trading day, 6.5 hours, with data being generated every second sampling every, say 5 minutes, one would reduce the sampled observations from 23,400 (all data) to 78 (5 minute intervals). Sampling every 5 minutes does not seem adequate when much more data are available.

[10] Ibid

[11] Aït-Sahalia, Yacine; Mykland, Per A.; Zhang, Lan. (2009). "High-frequency Market Microstructure Noise Estimates and Liquidity Measures."

IV. IMPLICATIONS AND PRACTICAL APPLICATIONS

While the "fifth-best" estimator fails to accurately estimate the variation of returns for high-frequency observations due to the market microstructure noise, this estimator has important and useful applications. For instance, Zhang et al., (2005b) argue that the "fifth-best" estimator, $[Y,Y]_T^{all}$, when used with high-frequency data, consistently estimators the variance of the microstructure noise. That is,

$$\widehat{E\epsilon^2} = \frac{1}{2n}[Y,Y]_T^{all}, \tag{9}$$

for where a consistent asymptotic variance estimator is also available, see Zhang et al., (2005b).

If Zhang et al., (2005b) are right, the "fifth-best" estimator has important applications in finance and economics. For instance, an immediate application of this estimator, beyond equities, is currencies. Economists and traders interested in studying the drivers and efficiency of currency trading may use equation (9) to separate the fundamental price drivers from the noise. In doing so, the researcher might find this estimator to be a straightforward method for estimating the variance of the noise term, and therefore, the efficiency of currency prices in the exchange rate market.

The implications for equity prices is more obvious. Researchers can take advantage of high-frequency trading to better understand the markets and its efficiency by observing investor and trading behavior. Nonetheless, this does not come without challenges. As the number of trade data increase, so does the noise. Bid-ask prices, the size of the trade, and direction of the trade are just but a few examples of the constituents of the market microstructure noise in the finance

literature. For instance, Roll (1984) claims the bid-ask spread is the sole responsible for the market microstructure noise.

Understanding how the variance of the noise affects prices when using high-frequency data has a direct impact in financial decisions. This is especially true for asset managers when they invest in equities and hedge positions against exchange rate fluctuations and for companies and governments engaging in foreign exchange transactions.

Another interesting application is the use of the first-best estimator for evaluating the efficiency of prices of stocks traded by institutional investors. See Hasbrouck, Boehmer and Kelley (2009) for research on this topic.

V. MONTE CARLO SIMULATION

To test the results obtained by Aït-Sahalia et al, (2005) a Monte Carlo Analysis is conducted. Using the the parameters below[12] 25,000 simulations are performed. To simulate stock prices, the Heston (1993) Model was used,

$$dX_t = (\mu - v_t)dt + \sigma_t dB_t \tag{10}$$

$$dv_t = \kappa(\alpha - v_t)dt + \gamma v_t^{\frac{1}{2}} dW_t \tag{11}$$

[12] See Zhang, Mykland, and Ait-Sahalia (2005b) page 1404 of the American Statistical Association Journal. Also these parameters and code are available in the Appendix A.

The same approach and parameter values used by Zhang, Mykland, and Ait-Sahalia (2005b) were employed. The Greek parameters below are assumed to be constant between Brownian motions, B and W: $\boldsymbol{\mu = 0.05, \kappa = 5}$, $\boldsymbol{\alpha = 0.04}$, $\boldsymbol{\gamma = 0.5}$, $\boldsymbol{\rho = -0.5}$, $\boldsymbol{E(\epsilon^2)^{\frac{1}{2}} = 0.0005, \Delta t = 1}$ second. For the market microstructure noise, ϵ, it is assumed Gaussian and small, the standard deviation is 0.05% of the value of the asset price. T=1/252, one trading day, 252 trading days a year. The Feller's condition is also assumed, $2\kappa\alpha \geq \gamma^2$, to prevent the volatility process from trespassing the zero boundary.

Table 1 shows the results of the Mote Carlo Simulation. These results are consistent with the findings of Zhang, Mykland, and Ait-Sahalia (2005b). The fifth best estimator fails to accurately estimate the variance of the true process. One can see how, generally, the results improve as one approaches the "first-best" estimator, especially after the small sample adjustment in the "first-best" estimator.

The TSRV estimator provides a novel approach in estimating the realized volatility using high-frequency financial data. The fourth best estimator, which uses sparse sampling is the most common in practice, but the results obtained in this simulation, show that while the effect of the market microstructure noise is reduced, one throws away too much data. Instead, the "first-best" estimator produces statistically better results, that while slightly biased, are asymptotically equivalent to the true integrated volatility. After a small sample adjustment and bias correction, the "first-best" estimator results in a major improvement over the current methods for estimating the realized volatility using high-frequency data.

Additionally, it is evident that the sample bias and sample variance become drastically smaller with each estimator, specially with the sample adjustment in the "first-best" estimator.

The same is true for the root-mean-square error (RMSE) and the relative statistics. Another point worth noticing is that there is a substantial difference between the sparse sample estimator, "third-best" estimator which Zhang, Mykland, and Ait-Sahalia (2005b) want to compare to, and the adjusted "first-best" estimator. Consistent with Zhang, Mykland, and Ait-Sahalia (2005b), the "first-best" estimator has a smaller sample bias and sample variance.

While "fifth-best" estimator is widely known not to be reliable in estimating the volatility of returns using high-frequency data, Zhang, Mykland, and Ait-Sahalia (2005b) found that it consistently estimates the market microstructure noise. This finding has a major implication for market efficiency. For instance, further research on this topic could use the equation (9) to asses if stocks of institutional investors are priced more efficiently and the extent that their trades contribute to the efficiency of the financial markets across the board.

VI. MARKET DATA RESULTS

Aït-Sahalia and Yu (2009) proved that liquid stocks tend to have a lower market microstructure noise. Noisy trading occurs as a result of diverse factors. These factors may include asymmetric information, macroeconomic developments, and industry specific factors. In addition to these factors, investors trade securities at difference prices depending on the side of the transaction they are in, buying or selling. Since there are many buyers and sellers with distinct sources of information, prices might deviate from the latent price. This paper uses the bid-ask price to as a measure of price efficiency. Stocks with lower bid-ask spreads are generally more liquid and thus should be priced more efficiently than stocks with higher bid-ask spreads.

Table 1. Monte Carlo Simulation for the Five Estimators plus the Small Sample Adjusted Estimator (FirstBestAdj)[13]

	Fifth Best	**Fourth Best**	**Third Best**	**Second Best**	**First Best**	**First Best Adj**
Sample Bias	0.0468	$1.5619\ x\ 10^{-4}$	$3.5169\ x\ 10^{-5}$	$3.0417\ x\ 10^{-5}$	$-3.2491\ x\ 10^{-6}$	$4.2509\ x\ 10^{-8}$
Sample Variance	$2.8115\ x\ 10^{-7}$	$3.2367\ x\ 10^{-9}$	$4.8898\ x\ 10^{-9}$	$2.476\ x\ 10^{-9}$	$1.9749\ x\ 10^{-10}$	$1.8918\ x\ 10^{-10}$
Sample RMSE	7.3992	0.0247	0.0056	0.0048	$5.1373\ x\ 10^{-4}$	$6.7213\ x\ 10^{-6}$
Sample Relative Bias	295.1631	0.9851	0.2218	−0.1919	−0.0205	$-2.6812\ x\ 10^{-4}$
Sample Relative	18.1706	0.1972	0.6090	0.4000	−0.0530	0.0158
Variance	$4.6668\ x\ 10^{4}$	154.7654	34.0732	29.3349	2.2403	−0.9576
Sample Replative RMSE						

[13] MATLAB code for this results is available in the Appendix.

Security dealers may quote different prices for securities based on how liquid is the stock, the size of the transaction or outstanding inventory. This paper argues that these factors, especially the bid-ask spread, contribute to the market microstructure noise. See Aït-Sahalia and Yu (2009), Sarr and Lybek (2002), and Roll (1984) attribute the microstructure noise to liquidity, trade size, noise-to-signal ratio and bid-ask bounce. This paper took a simple approach, just like Roll (1984) and focused on the bid-ask spread as a measure of liquidity and transaction costs.

To test this hypothesis, this paper looks at the divergence of the market microstructure noise in the components of the Dow Jones Industrial Average using the equation (9). Theoretically, stocks with lower bid-ask spreads should be priced more efficiently, and consequently, should exhibit less noise incorporated into the true return process.

Equation (9) is used as an estimator for market microstructure for the 30 stocks[14] of the Dow Jones Industrial Average (DJIA). These stocks are classified into two categories: stocks with an average bid-ask spread below the index's mean as group 1, and stocks with a bid-ask spread above the index's mean as group 2. The data was collected from CRSP[15] using the end of day bid-ask spread reported and TAQ for trade data from the Wharton Research Data Services database. The data only includes information for the one month, October 2013, due to computing and software limitations.

[14] These are the components of the Dow Jones Industrial Average as of January 22, 2016.
[15] Data for the month of October 2013. Center for Research in Security Prices (CRSP) and Trade and Quote (TAQ), Wharton Research Data Services, University of Pennsylvania.

The bid-ask spread is defined as the difference between the ask price, price at which a security can be bought in the market, and the bid price, price at which a security can be sold in the market. Thus, the bid-ask spread can be computed as

$$ASK - BID = SPREAD, \tag{12}$$

From Table 2, it can be seen that for the month of October 2013 the average bid-ask spread of the components of the DJIA, based on January 2016 components, is 1.6 cents. Group 1 is composed of 25 stocks and group 2 is composed of 5 stocks.

Table 2. Bid-Ask Spread: Summary Statistics for Dow Jones Industrial Average Components[16]

Variable	**Obs**	**Mean**	**Std. Dev.**	**Min**	**Max**
Spread	690	0.0158116	0.0256241	0	0.0600052
Spread = 1	575	0.0110261	0.0040239	0	0.0416708
Spread = 2	115	0.0397393	0.0565134	0	0.1100006

Table 3. Group categories and stocks in the Dow Jones Industrial Average (DJIA)

GROUPS	**Stock Symbol**
GROUP 1:	**AXP, BA, CAT, CSCO, CVX** **DD, DIS, GE, HD, INTC** **JNJ, JPM, KO, MCD, MMM** **MRK, MSFT, NKE, PFE, PG** **TRV, UNH, UTX, VZ, WMT**
GROUP 2:	**AAPL, IBM, GS, V, XOM**

[16] STATA code for this calculations available in the Appendix B.

Table 4. Components of the Dow Jones Industrial Average (DJIA)

Stock Symbol	***Company Name***
AXP	American Express Co.
AAPL	Apple Inc.
BA	Boeign Co.
CAT	Caterpillar Inc.
CSCO	Cisco System Inc.
Stock Symbol	***Company Name***
CVX	Chevron Corp.
DD	El du Pont de Nemours and Co.
DIS	Walt Disney Co.
XOM	Exxon Mobil Cop.
GE	General Electric Co.
GS	Goldman Sachs Group Inc.
HD	Home Depot Inc.
IBM	International Business Machines Corp.
INTC	Intel Corp.
JNJ	Johnson & Johnson
KO	Cola-Cola Co.
JPM	JPMorgan Chase and Co.
MCD	McDonald's Cop.
MMM	3M Co.
MRK	Merck & Co Inc.
MSFT	Microsoft Corp.
NKE	Nike Inc.
PFE	Pfizer Inc.
PG	Procter & Gamble Co.
TRV	Travelers Companies Inc.
UNH	UnitedHealth Group Inc.
UTX	United Technologies Corp.
VZ	Verizon Communications Inc.
V	Visa Inc.
WMT	Wal-Mart Stores Inc.

Table 3 shows the summary statistics for group 1 and group 2. Group 1 contains the stocks with a bid-ask spread below the mean, 0.016 cents, while Group 2 contains the stocks with a bid-ask spread above the mean. Due to how liquid these stocks are, the size of the groups should not create unbalance.

The DJIA is one of the most important stock indexes in the U.S. Due to its importance and the amount of research available on its constituents, the individual prices of the DJIA constituents should be close to the efficient price.

The results from table 4 are do not support the hypothesis. For Group 1, the estimated market microstructure noise from equation (9) is larger than stocks in Group 2. These results are likely affected by the high liquidity of the components of the Dow Jones Industrial Average and the size of the sample. This claim is supported by the estimated microstructure noise in both groups. The noise is virtually zero for both groups, indicating that the price in the market for these stocks is close to the efficient price.

Table 5. Microstructure Noise Estimates for Dow Jones Industrial Average Components[17]

Group	**Obs**	**Microstructure Noise**	**TSRV σ^2**
Group 1	5,411,634.8	$5.1399\ x\ 10^{-7}$	$1.0280\ x\ 10^{-6}$
Group 2	4,579,738	$4.9501\ x\ 10^{-7}$	$9.9003\ x\ 10^{-7}$

There is not conclusive evidence that the stocks with narrower bid-ask spread than stocks exhibit lower market microstructure noise than stocks with wider bid-ask spreads. The results of this study are likely affected by the limited sample size and the high liquidity of the stocks among the groups. Since constituents in the DJIA are closely monitored in the market, the bid-ask spread may not be the only conclusive factor affecting the market microstructure noise. Further, the length of time for the data collected, one month[18], might have not been sufficient to account for more

[17] Data from October 1st to October 31st, 2013. Python code for this estimation is available in the Appendix C.
[18] One more of data was used due to computing and software limitations.

variation in the bid-ask spread. Other factors such as volume, price changes, and research coverage could prove useful in estimating the market microstructure noise.

It is important to highlight that the above results might be influenced by the fact that certain stocks were not in the DJIA in October 2013. For instance, Apple was not included in the DJIA until spring of 2015. Further research should test whether inclusion in the DJIA decreases the microstructure noise contained in the stock price, and thus, returns. Further, increasing the amount of time in the study, sample size, and number of stocks may prove beneficial to corroborate or update the results.

VII. CONCLUSION

In this study, I looked into the characteristics of the Two Scales Realized Volatility estimator and how it can be used to estimate realized volatility with high-frequency financial and economic data. The "first-best" estimator with its small sample and bias adjustment uses all available data by using two time scales: a slow time scale, averaging across subsamples, and a fast time scale, using all available data using the sum of squared log returns. The Monte Carlo simulation results are consistent with those of Zhang, Mykland, and Ait-Sahalia (2005b) and Aït-Sahalia and Mykland (2009). The TSRV consistently estimates the realized volatility in high-frequency financial data while minimizing the market microstructure noise.

An important ramification of the TSRV estimator comes from the "fifth-best" estimator.

While this estimator does a poor job in estimating the volatility of returns at high-frequencies due to the noise contained in the return process, this estimator provides special insights into the market microstructure noise. Zhang, Mykland, and Ait-Sahalia (2005b) prove that using the using equation (9), which is based on the "fifth-best" estimator, one can consistently estimate the market microstructure noise in high-frequency data. Using equation (9), I tested the hypothesis that stocks that trade with narrower bid-ask spreads, more liquid with lower transaction costs, have less market microstructure noise than those with wider bid-ask spreads. Particularly, this paper looked at the constituents of the Dow Jones Industrial Average (DJIA). The study found that bid-ask spread, by itself, was not a conclusive measure to determine a marked difference in the market microstructure noise in the DJIA components. The study was likely affected by the high liquidity of the stocks, the number of stocks in the sample and the size of the data.

Further research should increase the length of time of the study to account for more variation in the bid-ask spread; include a large selection of stocks, and more rankings. Moreover, this study can be expanded by controlling for other variables such as economy policy intervention, research coverage, volume, price changes and other macroeconomic and market wide factors. Finally, further research may also include a comparison between the DJIA and other domestic, and international indexes and/or equities.

REFERENCES

Ait-Sahalia, Y.; Mykland, P. A., & Zhang, L. (2005a). How Often to Sample a Continuous Process in the Presence of Microstructure Noise. *"The Review of Financial Studies, Vol. 18, No. 2, 2005."* The Society of Financial Studies.

Aït-Sahalia, Yacine; Mykland, Per A. (2009). T.G. Anderson et al., Handbook of Financial Time Series, DOI: 10.1007/978-3-540-71297-8_25, © Springer-Verlag Berlin Heidelberg 2009

Aït-Sahalia, Y.; Yu, J. (2009). High-frequency Market Microstructure Noise Estimates and Liquidity Measures. *"The Annals of Applied Statistics, Vol. 3, No. 1, 422-457."* Institute of Mathemaatical Statistics.

Camponovo, L., Matsushita, Y., & Otsu, T. (2015). Nonparametric likelihood for volatility under high-frequency data Suntory and Toyota International Centres for Economics and Related Disciplines, LSE, STICERD - Econometrics Paper Series. Retrieved from http://search.proquest.com.dist.lib.usu.edu/docview/1660006981?accountid=14761

Hasbrouck, Joel (1993). Assessing the quality of a security market: A new approach to transactioncost measurement. Review of Financial Studies 6 191–212.

Li, Yingying, Per A. Mykland, Eric Renault, Lan Zhang, and Xinghua Zheng. 2014. Realized Volatility When Sampling Times Are Possibly Endogenous. *"Econometric Theory 30, no. 3: 580-605."* Business Source Premier, EBSCOhost (accessed January 22, 2016).

McDonald, Robert L. 2006. Derivative Markets, Second Edition. Pearson Education. ISBN 0-321-28030-

X, page 607.

Misaki, Hiroumi, and Naoto Kunitomo. 2015. "On robust properties of the SIML estimation of volatility under micro-market noise and random sampling." International Review Of Economics & Finance 40, 265-281. Business Source Premier, EBSCOhost (accessed January 22, 2016).

Richard, Roll (1984). A simple model of the implicit bid–ask spread in an efficient market. Journal of Finance 39 1127–1139.

Ysusi, C. (2008). Estimating integrated volatility using absolute high-frequency returns. International Journal of Monetary Economics and Finance, 1(2), 177-200. Retrieved from http://search.proquest.com.dist.lib.usu.edu/docview/56868740?accountid=14761

Yu, Chao, Yue Fang, Zeng Li, Bo Zhang, and Xujie Zhao. 2014. "Non-Parametric Estimation of High-Frequency Spot Volatility for Brownian Semimartingale with Jumps." *Journal Of Time Series Analysis* 35, no. 6: 572-591. *Academic Search Premier*, EBSCO*host*(accessed January 22, 2016).

Zhang, L., Mykland, P. A., & Ait-Sahalia, Y. (2005b). A tale of two time scales: Determining integrated volatility with noisy high-frequency data. *"American Statistical Association, Vol. 100, Issue 472, 2005."*

APPENDICES

APPENDIX A

% MATLAB CODE FOR TSRV ESTIMATION

```
% Five estimators

M = 25000; %number of MCs
T = 1/252; %length of time in years -- this is fixed, the sample size n is
T/delta

tic; %start time counter
randn('state',sum(100*clock));

nbseconds = 1;
delta = nbseconds/(60*60*(6.5)*252); % data available every 5 seconds, this
is dt
deltasparse = 5/(60*(6.5)*252); % arbitrary sparse sampling every 5 minutes
n = round(T/delta);

if nbseconds == 5;
    Kgrid =
[1,2,3,4,5,6,8,9,10,12,13,15,18,20,24,26,30,36,39,40,45,52,60,65,72,78,90,104
,117,120,130,156,180,195,234,260,312,360,390,468,520,585,780,936,1170,1560,23
40,4680]';
    % these are the divisors of n=4680, corresponding to nbseconds = 5
    % they are such that each value of Kgrid is integer and so is each value
of nsparse = n/Kgrid
    Ksparsek =
[1,12,60,65,72,78,90,104,117,120,130,156,180,195,234,260,312,360,390,468,520,
585,780,936]';
    % Ksparsek = grid used for RMSE plot, this is a subset of Kgrid
elseif nbseconds == 1;
    Kgrid =
[1,2,3,4,5,6,8,9,10,12,13,15,18,20,24,25,26,30,36,39,40,45,50,52,60,65,72,75,
78,90,100,104,117,120,130,150,156,180,195,200,225,234,260,300,312,325,360,390
,450,468,520,585,600,650,780,900,936,975,1170,1300,1560,1800,1950,2340,2600,2
925,3900,4680,5850,7800,11700,23400]';
    Ksparsek =
[1,60,120,200,300,360,390,450,468,520,585,600,650,780,900,936,975,1170,1300,1
560,1800,1950,2340,2600,2925,3900]';
else
    disp(['Error -- adjust nbseconds']); return;
end
nKgrid = size(Kgrid,1); nKsparsek =  size(Ksparsek,1);
% will always pick K as the closest value on Kgrid
nsparsek = n./Ksparsek; % nsparsek = sample size for the sparsek estimator

% the objective
XX=zeros(M,1);

% the five estimators
```

```
YYall=zeros(M,1); YYsparse=zeros(M,1); YYsparsek=zeros(M,nKsparsek);
YYsparseopt=zeros(M,1); YYavg=zeros(M,1); YYavgBC=zeros(M,1);
XXhat=zeros(M,1); XXhatadj=zeros(M,1);
% other quantities to export
Ee2hat=zeros(M,1); Ee4hat=zeros(M,1); a2hat=zeros(M,1);
quarticity=zeros(M,1); nsparseopt=zeros(M,1); Ksparseopt=zeros(M,1);
nbarstarNBC=zeros(M,1); KstarNBC=zeros(M,1);
nbarstar=zeros(M,1); Kstar=zeros(M,1); cstar=zeros(M,1); XXall=zeros(M,1);

if nbseconds == 5;        a2true = (0.15/100)^2; % a2 = variance of
microstructure noise term
elseif nbseconds == 1;    af2true = (0.10/100)^2; % a2 needs to decrease with
the sampling size for the optimal RMSE calculations to be meaningful
else
    disp(['Error -- adjust nbseconds']); return;
end
%a2true = 0; % no noise case



% dX(t) = (mu - V(t)/2) dt + sqrt(V(t)) dW(t),  dV(t) = kappa (alpha - V(t))
+ gamma*sqrt(V(t)) dB(t),  E[dW(t) dB(t)] = rho dt, E[e2] = a2
mutrue = 0.05;
kappatrue = 5;
alphatrue = 0.04;
gammatrue = 0.5;
% gammatrue = 0; % gammatrue = 0 makes volatility non-stochastic
% Feller's condition for 0 boundary of v is q >= 0
% where omegatrue = (2*kappatrue)/gammatrue^2; nutrue = gammatrue*alphatrue;
qtrue = nutrue-1;
rhotrue = - 0.5;

Ee2 = a2true;
Ee4 = 3 * a2true^2;
Equarticity = T * (alphatrue^2 + alphatrue*gammatrue^2/(2*kappatrue));



for iMC=1:M, %loop on Monte Carlos

    % n is the number of dY's -- to get there, need to simulate n+1 Y's
    % in fact, use 2*n to be able to take the central part (n/2):(3*n/2-1) so
that we can lead and lag properly for the avg estimator
    dB = sqrt(delta)*randn(2*n+1,1);
    dW = sqrt(delta)*randn(2*n+1,1);    % aka random('Normal',0,1,n,1);
    dB = sqrt(1-rhotrue^2)*dB + rhotrue*dW; % generates correlated dB and dW
such that E[dB*dW] = rho dt


    x = zeros(2*n+1,1); % x = log-price
    v = zeros(2*n+1,1); % v = local stochastic variance
    sigma = zeros(2*n+1,1); % sigma = sqrt(v) = local stochastic volatility

    x(1) = log(100); % initial log-price
```

```
    if gammatrue == 0; % case where volatility is non-stochastic
        v(1) = alphatrue;
    else
        omegatrue = (2*kappatrue)/gammatrue^2; nutrue = omegatrue*alphatrue;
        A = nutrue; B = 1/omegatrue;
        v(1) = gamrnd(A,B); % draws initial value from stationary
distribution which is Gamma since v is a CIR process
        % from Mathematica: omegatrue = (2*kappatrue)/gammatrue^2; nutrue =
omegatrue*alphatrue; qtrue = nutrue-1;
        % pidens[x_] = (omegatrue^nutrue/Gamma[nutrue])*x^qtrue*exp(-
omegatrue*x);
        % R = gamrnd(A,B) generates gamma random numbers with parameters A
and B.
        % gampdf(x,A,B) computes the gamma pdf at each of the values in x
using the corresponding parameters in A and B.
        % gampdf(x,A,B) = (1/(B^A * Gamma(A)) * x^(A-1) * exp(-x/B)
        % therefore A = nutrue; B = 1/omegatrue;
        % vplot = (1:n+1)*0.40^2/n; plot(vplot,gampdf(vplot,A,B)); return;
        % pX[del_, x_, x0_, K_] = Dg[x]*pY[del, g[x], g[x0], K];
        % Delta method: X = sigma, Y = v = g[X} = X^2, Dg[X] = 2*X
        % sigmaplot = (1:n+1)*0.40/n;
plot(sigmaplot,2*sigmaplot.*gampdf(sigmaplot.^2,A,B)); return;
    end;


    sigma(1) = sqrt(v(1));
    % simulate more that one day's (T) worth of data and we will use the
center observations from n/2 to 3*n/2 to allow for leads and lags
    for i=2:(2*n+1),
        v(i) = v(i-1) + kappatrue*(alphatrue - v(i-1))*delta +
gammatrue*sqrt(v(i-1))*dW(i);
        sigma(i) = sqrt(v(i));
        x(i) = x(i-1) + (mutrue - sigma(i-1)^2/2)*delta + sigma(i-1)*dB(i);
    end;


    %x = zeros(2*n+1,1); %to get no x

    epsilon = sqrt(a2true)*randn(2*n+1,1); % epsilon = market microstructure
noise
    y = x + epsilon; % y = noisy data

    %plot((n/2):(3*n/2+1),exp(x)); pause(1); plot((n/2):(3*n/2+1),sigma*100);
return;

    % use the middle part of the sample so we can go backward and forward by
up to n/2 obs when computing subgrids
    XX(iMC) = sum(sigma((n/2):(3*n/2-1)).^2)*delta; % this is the integral
from 0 to T of sigma^2 = v
    quarticity(iMC) = sum(sigma((n/2):(3*n/2-1)).^4)*delta; % this is the
integral from 0 to T of sigma^4 = v^2

    Ee2hat(iMC) = mean(epsilon((n/2):(3*n/2)).^2);
    Ee4hat(iMC) = mean(epsilon((n/2):(3*n/2)).^4);
```

```
    % this is the end simulated data


    % Fifth best estimator: YYall
    dY = y((n/2)+1:(3*n/2)) - y((n/2):(3*n/2-1)); % dY = increments of Y,
there are n increments
    YYall(iMC) = sum(dY.^2);
    a2hat(iMC) = YYall(iMC)/(2*n); % YYall is an estimator of the variance of
the noise, used later to center YYavg

    dX = x((n/2)+1:(3*n/2)) - x((n/2):(3*n/2-1));
    XXall(iMC) = sum(dX.^2);

    % Fourth best estimator: YYsparse
    Ksparse = deltasparse/delta; % goes from nbseconds seconds to 5 minutes
    nsparse = n/Ksparse; % nsparse = sample size for the sparse estimator,
numbers must be such that this is a round number
    % by construction 1 + (nsparse*Ksparse) = n + 1
    ysparse=zeros(nsparse+1,1);

    ysparse(1) = y(n/2);
    for i=2:(nsparse+1);
        ysparse(i) = y(n/2+(i-1)*Ksparse);
    end;

    dYsparse = ysparse(2:(nsparse+1)) - ysparse(1:nsparse);
    YYsparse(iMC) = sum(dYsparse.^2);


    % loop on Ksparse to get a RMSE curve for different values of Ksparsek
    % by construction (nsparsek*Ksparsek) = n

        for j=1:nKsparsek;
        ysparsek = zeros(nsparsek(j),1);

        ysparsek(1) = y(n/2);
        for i=2:(nsparsek(j)+1);
            ysparsek(i) = y(n/2+(i-1)*Ksparsek(j));
        end;

        dYsparsek = ysparsek(2:(nsparsek(j)+1)) - ysparsek(1:nsparsek(j));
        YYsparsek(iMC,j) = sum(dYsparsek.^2);
        end;



    % Third best estimator: YYsparseopt

    if a2true > 0;
    nsparseopt(iMC) =
max(1,round((T*quarticity(iMC)/(4*Ee2hat(iMC)^2))^(1/3))); % this is nstar =
sample size for the sparseopt estimator
else
```

```
    nsparseopt(iMC) = n/40;
    % normally, set nsparseopt(iMC) = n in this case, but use n/40 for
testing purposes
end;
    %nsparseopt(iMC) = max(1,round((T*Equarticity/(4*Ee2^2))^(1/3))); % this
is nstar = sample size for the sparseopt estimator
    Ksparseopt(iMC) = n/nsparseopt(iMC);

    [tmp,idx] = min(abs(Kgrid-Ksparseopt(iMC))); % idx is the index of the
vector Kgrid that gives the closest value to n/nsparseopt(iMC)
    Ksparseopt(iMC) = Kgrid(idx);
    nsparseopt(iMC) = n/Ksparseopt(iMC);


    ysparseopt=zeros(nsparseopt(iMC)+1,1);

    ysparseopt(1) = y(n/2);
    for i=2:(nsparseopt(iMC)+1);
        ysparseopt(i) = y(n/2+(i-1)*Ksparseopt(iMC));
    end;

    dYsparseopt = ysparseopt(2:(nsparseopt(iMC)+1)) -
ysparseopt(1:nsparseopt(iMC));
    YYsparseopt(iMC) = sum(dYsparseopt.^2);
    % the factor n/(nsparseopt(iMC)*Ksparseopt(iMC)) is to adjust for the
rounding
    % making sure that we integrate on 0 to T, not 0 to a smaller number due
to rounding


    % Second best estimator: YYavg

    % first compute YYgrid(k) over subgrid k, k=1,...,K
    % there are Kstar equally spaced grids each with size nbarstar
    % NBC means Not Bias Corrected

    if a2true > 0;
    nbarstarNBC(iMC) =
min(n,max(1,round((T*quarticity(iMC)/(6*Ee2hat(iMC)^2))^(1/3)))); % nbarstar
= size of each subgrid
else
    nbarstarNBC(iMC) = n/40;
    % normally, set nsparseopt(iMC) = n in this case, but use n/40 for
testing purposes
end;
    KstarNBC(iMC) = n/nbarstarNBC(iMC); % to ensure that the largest y point
used,  Kstar + nbarstar*Kstar = (nbarstar+1)*Kstar <= n + 1

    [tmp,idx] = min(abs(Kgrid-KstarNBC(iMC))); % idx is the index of the
vector Kgrid that gives the closest value to n/nsparseopt(iMC)
    KstarNBC(iMC) = Kgrid(idx);  % forces Kstar onto the grid
    nbarstarNBC(iMC) = n/KstarNBC(iMC); % n = nbar*K is exact by construction
of Kgrid

    % KstarNBC(iMC) = 300; % to test formula
```

```
    % without noise, large values of Kstar tend to generate skewness and
kurtosis of the standardized second best distribution
    % nbarstarNBC(iMC) = n/KstarNBC(iMC);

    ygridk = zeros(nbarstarNBC(iMC)+1,1); dYgridk=zeros(nbarstarNBC(iMC),1);
YYgrid = zeros(KstarNBC(iMC),1);
    for k=1:KstarNBC(iMC);
        ygridk(1) = y(n/2 - round(KstarNBC(iMC)/2) + k);
        for i=2:(nbarstarNBC(iMC)+1);
            ygridk(i) = y(n/2 - round(KstarNBC(iMC)/2) + k + (i-
1)*KstarNBC(iMC));
        end;
        dYgridk = ygridk(2:(nbarstarNBC(iMC)+1)) -
ygridk(1:nbarstarNBC(iMC));
        YYgrid(k) = sum(dYgridk.^2);
    end;
    % now YYavg is the average of YYgrid(k) over subgrid k, k=1,...,K
    YYavg(iMC) = mean(YYgrid);



    % First best estimator: XXhat
    % need to compute YYavg differently from the second best, between Kstar
is now K = cstar * n^(2/3)

    cstar(iMC) = ((12*Ee2hat(iMC)^2)/(T*quarticity(iMC)))^(1/3);
    % cstar(iMC) = ((12*Ee2^2)/(T*Equarticity))^(1/3);
    Kstar(iMC) = cstar(iMC) * n^(2/3); % nbarstar = n^(1/3) / cstar,   Kstar
= cstar * n^(2/3), nbarstar * Kstar = n

    [tmp,idx] = min(abs(Kgrid-Kstar(iMC))); % idx is the index of the vector
Kgrid that gives the closest value to n/nsparseopt(iMC)
    Kstar(iMC) = Kgrid(idx); % forces Kstar onto the grid
    nbarstar(iMC) = n/Kstar(iMC); % n = nbar*K is exact by construction of
Kgrid
    cstar(iMC) = Kstar(iMC)/n^(2/3);

    % Kstar(iMC) = 200; % to test formula
    % increasing Kstar gives better approx to bias, but increases non-
normality
    % nbarstar(iMC) = n/Kstar(iMC);
    % cstar(iMC) = Kstar(iMC)/n^(2/3);

    ygridk = zeros(nbarstar(iMC)+1,1); dYgridk=zeros(nbarstar(iMC),1);
YYgrid = zeros(Kstar(iMC),1);
    for k=1:Kstar(iMC);
        ygridk(1) = y(n/2 - round(Kstar(iMC)/2) + k);
        for i=2:(nbarstar(iMC)+1);
            ygridk(i) = y(n/2 - round(Kstar(iMC)/2) + k + (i-1)*Kstar(iMC));
        end;
        dYgridk = ygridk(2:(nbarstar(iMC)+1)) - ygridk(1:nbarstar(iMC));
        YYgrid(k) = sum(dYgridk.^2);
    end;
    % now YYavg is the average of YYgrid(k) over subgrid k, k=1,...,K
    % BC means this is for the Bias Corrected estimator
    YYavgBC(iMC) = mean(YYgrid);
```

```
    % finally:
    XXhat(iMC) = YYavgBC(iMC) - 2*nbarstar(iMC)*a2hat(iMC);
    % now adjusted estimator for small sample
    if nbarstar(iMC) < n;
        XXhatadj(iMC) = XXhat(iMC) / (1 - nbarstar(iMC)/n);
    else
        XXhatadj(iMC) = XXhat(iMC) / (1 - (n-1)/n);
    end;



    %disp([' ']); % add a blank line
    %disp(['true value = ',num2str(XX(iMC))]);
    %disp(['5th best   = ',num2str(YYall(iMC))]);
    %disp(['4th best   = ',num2str(YYsparse(iMC))]);
    %disp(['3rd best   = ',num2str(YYsparseopt(iMC))]);
    %disp(['2nd best   = ',num2str(YYavg(iMC))]);
    %disp(['1st best   = ',num2str(XXhat(iMC))]);
    %disp(['1st best (small sample bias-corrected) =
',num2str(XXhatadj(iMC))]);
    %disp([' ']); % add a blank line


end; %on iMC loop

%Mean Value for Estimators
trueValue = mean(XX(iMC))
fifthBest = YYall(iMC)
fourthBest = YYsparse(iMC)
thirdBest = YYsparseopt(iMC)
secondBest = YYavg(iMC)
firstBest = XXhat(iMC)
firstBestcorr = mean(XXhatadj(iMC))

%Sample Bias
sb5 = mean(YYall - XX)
sb4 = mean(YYsparse - XX)
sb3 = mean(YYsparseopt - XX)
sb2 = mean(YYavg - XX)
sb1 = mean(XXhat - XX)
sb1corr = mean(XXhatadj - XX)

%Sample Variance
d5 = (YYall- XX);
sv5 = mean(var(d5))
d4 = (YYsparse- XX);
sv4 = mean(var(d4))
d3 = (YYsparseopt- XX);
sv3 = mean(var(d3))
d2 = (YYavg- XX);
sv2 = mean(var(d2))
d1 = (XXhat- XX);
sv1 = mean(var(d1))
d1c = (XXhatadj- XX);
```

```
sv1 = mean(var(d1c))

%RMSE
r5 = mean(sqrt(sum(XX - YYall).^2/numel(XX)))
r4 = mean(sqrt(sum(XX - YYsparse).^2/numel(XX)))
r3 = mean(sqrt(sum(XX - YYsparseopt).^2/numel(XX)))
r2 = mean(sqrt(sum(XX - YYavg).^2/numel(XX)))
r1 = mean(sqrt(sum(XX - XXhat).^2/numel(XX)))
r1corr = mean(sqrt(sum(XX - XXhatadj).^2/numel(XX)))

%Sample Relative Bias
rb5 = mean((YYall - XX))/mean(XX)
rb4 = mean((YYsparse - XX))/mean(XX)
rb3 = mean((YYsparseopt - XX))/mean(XX)
rb2 = mean((YYavg - XX))/mean(XX)
rb1 = mean((XXhat - XX))/mean(XX)
rb1corr = mean((XXhatadj - XX))/mean(XX)

%Sample Relative Variance
rv5 = (var(YYall) - var(XX))/var(XX)
rv4 = (var(YYsparse) - var(XX))/var(XX)
rv3 = (var(YYsparseopt) - var(XX))/var(XX)
rv2 = (var(YYavg) - var(XX))/var(XX)
rv1 = (var(XXhat) - var(XX))/var(XX)
rvcorr = (var(XXhatadj) - var(XX))/var(XX)

%Relative RSME
rr5 = (r5 - mean(XX))/mean(XX)
rr4 = (r4 - mean(XX))/mean(XX)
rr3 = (r3 - mean(XX))/mean(XX)
rr2 = (r2 - mean(XX))/mean(XX)
rr1 = (r1 - mean(XX))/mean(XX)
rr1corr = (r1corr - mean(XX))/mean(XX)


%     disp(['mean true value = ',num2str(mean(XX))]);
%     disp(['mean 5th best   = ',num2str(mean(YYall))]);
%     disp(['mean 4th best   = ',num2str(mean(YYsparse))]);
%     disp(['mean 3rd best   = ',num2str(mean(YYsparseopt))]);
%     disp(['mean 2nd best   = ',num2str(mean(YYavg))]);
%     disp(['mean 1st best   = ',num2str(mean(XXhat))]);
%     disp(['mean 1st best (small sample bias-corrected) =
',num2str(mean(XXhatadj))]);
%

%%%%%%%%%%%%%%%%%% END OF MAIN ROUTINE %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

APPENDIX B

```
/*
STATA CODE FOR DETERMINING BID-ASK SPREAD AND GROUPING
        */

*Transform Data

*Load Data
use "/Users/aristides/Downloads/810c1a36d0d61346.dta", clear

*Browse data
browse
*Create ID compatible with STATA
encode ticker, gen (tic)
*Create Spread variable
gen spread = ask - bid
*Sort data
sort tic spread
*Clean data - additional stock, not in DJIA
drop in 70/92
*Take mean of each stock
egen aapl = mean(spread) if tic==1
egen axp = mean(spread) if tic==2
egen ba = mean(spread) if tic==3
egen cat = mean(spread) if tic==4
egen csco = mean(spread) if tic==5
egen cvx = mean(spread) if tic==6
egen dd = mean(spread) if tic==7
egen xom = mean(spread) if tic==8
egen ge = mean(spread) if tic==9
egen gs = mean(spread) if tic==10
egen hd = mean(spread) if tic==11
egen ibm = mean(spread) if tic==12
egen intc = mean(spread) if tic==13
egen jnj = mean(spread) if tic==14
egen ko = mean(spread) if tic==15
egen jpm = mean(spread) if tic==16
egen mcd = mean(spread) if tic==17
egen mmm = mean(spread) if tic==18
```

```
egen mrk = mean(spread) if tic==19
egen msft = mean(spread) if tic==20
egen nke = mean(spread) if tic==21
egen pfe = mean(spread) if tic==22
egen pg = mean(spread) if tic==23
egen trv = mean(spread) if tic==24
egen unh = mean(spread) if tic==25
egen utx = mean(spread) if tic==26
egen vz = mean(spread) if tic==27
egen v = mean(spread) if tic==28
egen wmt = mean(spread) if tic==29
egen dis = mean(spread) if tic==30

*Check the 50% percentile of the spread
sum spread, detail
*Make groups based on percentile
gen group = 2
#delimit;
replace group = 1 if aapl<= .0158116 | axp<=.0158116 | ba<=.0158116 | cat<=.0158116 |
csco<=.0158116 |
cvx<=.0158116 | dd<=.0158116 | xom<=.0158116 | ge<=.0158116 |gs<=.0158116 | hd<=.0158116|
ibm<=.0158116 | intc<=.0158116 | jnj<=.0158116 | ko<=.0158116 |
jpm <=.0158116 | mcd<=.0158116 | mmm<=.0158116 | mrk<=.0158116 | msft<=.0158116 |
nke<=.0158116 | pfe<=.0158116 | pg<=.0158116 |
trv<=.0158116 | unh<=.0158116 | utx<=.0158116 | vz<=.0158116 | v<=.0158116 |
wmt<=.0158116 | dis<=.0158116;
#delimit cr
*Sort based on percentile
sort tic group
browse group tic
*See group's info
tab group
sum group, detail
browse group tic if group==1

/*
Variables with lowest bid-ask spread
Group 1a: AXP, BA, CAT, CSCO, CVX,
Group 1b: DD, DIS, GE, HD, INTC,
Group 1c: JNJ, JPM, KO, MCD, MMM,
Group 1d: MRK, MSFT, NKE, PFE, PG,
Group 1e: TRV, UNH, UTX, VZ, WMT
        */

/*
Variables with above average bid-ask spread
Group 2: AAPL, IBM, GS, V, XOM
        */

*Summary Statistics of groups 1 and 2
sum spread, detail
```

```
sum spread if group==1, detail
sum spread if group==2, detail
```

APPENDIX C

#PYTHON CODE FOR MICROSTRUCTURE NOISE AND VOLATILITY ESTIMATION
#WITH 30 COMPONENTS OF THE DOW JONES INDUSTRIAL AVERAGE

```
Created on Wed Dec 23 21:19:45 2015

@author: aristides
"""

import csv
import numpy as np

#Read

def Returns():
    stock = np.genfromtxt ('group1a.csv', delimiter=",", skip_header=1, usecols=3)
    price = stock


    logprc = np.log(price[:-1])
    logprc_1 = np.log(price[1:])
    nret = logprc[:-1] - logprc_1[1:]

    return (nret)


## Fifth Best Estimator
def FifthBestEstimator(nret):
    rv = np.dot(nret, nret)

    return rv


def FourthBestEstimator(nret, incr):
    nobs = nret.size
    zt = nret[incr-1:nobs:incr]
    rv = FifthBestEstimator(zt)
    return rv
```

```
def SecondBestEstimator(nret, kvalue):

    nobs = int(nret.size)
    yyslow = np.zeros((kvalue,))
    n = np.zeros((kvalue,))

    for i in range(0, kvalue):
        ind = np.arange(start=i, stop=nobs, step=kvalue)
        yyslow[i] = FifthBestEstimator(nret[ind])
        n[i] = len(ind)

    yyavg = yyslow.mean()
    #nbar = n.mean() #Not sure what this does

    return yyavg

## Main
nret = Returns()
nobs = int(nret.size)

kvalue = 300 # in R code K
incr = 1 #In R code J

nbarK = (nobs - kvalue + 1)/(kvalue)
nbarJ = (nobs - incr + 1)/(incr)


yall = FifthBestEstimator(nret)
yyavg = SecondBestEstimator(nret, kvalue)
xxt = yyavg - (nbarK/nbarJ) * yall
xxtadj = (1.0 / (1.0 - nbarK/nbarJ)) * xxt;

firstBest = xxtadj
#RV
yall
#TSRV
firstBest
#Noise
Noise = yall / (2*nobs)
Noise
```