**Abstract**

Market Makers provide values to the overall economic system by providing liquidities to the market. They facilitate trading activities and the transferring of risk exposures among counterparties. In return, market makers earn a bid-ask spread for the services they provide, and for the risks they take. Most noticeably, market makers are perennially exposed to the risk of being the counterparties of informed traders, for informed traders, by definition, have valuable information about future price changes. Playing a game of asymmetric information puts market makers at a significant disadvantage. Therefore, a thorough understanding of the market microstructure, and the ability to detect and predict price changes in the market is extremely valuable. In this paper, we introduce a model based on the imbalance of buy and sell information to help predicting price changes. Moreover, and speed has always been a decisive factor in determining the competitiveness and profitability of a market making entity. The recent rise of high frequency machines is a perfect example of the prior statement. Fast and efficient execution of orders are always of paramount importance. Without quality executions, price prediction is not of much value. In this paper, we also strive to improve influence execution quality by examining liquidity characteristics across different trading venues, and finding out where are securities trading at.

**Predicting price changes with Signed Order Imbalance for equity securities**

**Background information:**

Signed Order Imbalances (SOI) is an idea inspired by Easley, Lopez de Prado and O’Hara’s paper on “Flow Toxicity and Liquidity in a High Frequency World”. Intuitively, SOI measures the magnitude of imbalance between buy and sell orders within a specified time period. We believe a large portion of the short term equity price change is driven by such imbalances.

We restrict the range of this measure to be between 1 and -1. An SOI of 1 would indicate all orders are originated from buy requests, where as an SOI of -1 would indicate the opposite, and numbers in between quantify general imbalances with positive figures signaling preponderant buy orders, and vice versa.

**Definition of SOI:**

**(1)**

**Explanation of the definition:**

defines the total volume of a bucket. A bucket is nothing more than an aggregation of adjacent smaller chunks of trades, known as bin. SOI is measured at the bucket level, and individual weighted bins contribute to the SOI measure over the bucket. There are numerous different ways of aggregating these bins into a bucket, and depending on the total trading volume of the specific stock at interest, the optimal parameters used in aggregation differ. The below table listed ways of aggregating trades into bins and buckets based on either time interval or volume interval, note tick data for bin choice implies using the each trade transaction directly as a bin unit without doing any aggregation:

|  |  |
| --- | --- |
| **Bin Choice** | **Bucket Choice** |
| Time | Time |
| Trade Volume | Trade Volume |
| Tick Data |  |

A mixing of volume and time aggregating technique can be applied separately to bins and volume. In this paper, we primarily explore two different approaches: 1) tick data + time bucket, 2) tick data + volume bucket. While any combination of the above can be function, it is important to ensure a single bucket always contain at least one bin in order for the above methodology to be rational. And once the units are chosen, they should be fixed throughout the day. The advantage of using bins as oppose to tick data can be found in a paper written by Easley, López de Prado, and O’Hara [2012]. In their research, they claimed trade by trade classification is likely to result in misclassification (PG 22). However, ELO did not provide concrete evidence, but merely an intuitive suggestion. Without sufficient evidence that trade by trade classification is inferior, we choose to proceed with trade by trade (tick data) classification, and we will show trade by trade classification actually produces better regression fit than aggregating by time bins. Also by using tick data, we avoid unnecessary task of optimizing the fixed bin size within each bucket.

When predicting equity price changes, we first obtain SOI information over a bucket covering time period t to t+1, and regress against price change over a bucket covering time period t+1 to t+2. More details will be provided in the analysis section.

In equation (1), b(i) is a discrete function only takes value {+1,-1, 0}, which indicates the act (Bull/Sell/Neutral) of the corresponding bin/trade.

The classification method is explained in detail in the following section.

**Buy/Sell/Neutral Classification:**

Multiple buy/sell/neutral classification schemes are explored:

1. Modified “Lee-Ready” type classification based solely on trade data.

In this algorithm, transactional level prices for trades are used. A trade is classified as a buy trade if the transactional price at time t is higher than the price at t-1, and classified as a sell trade if the transactional price at time t is lower than the price at t-1. In case both transactional prices are identical, we look back one period (i.e. t-2 to t-1 period) and follow the same procedure. The rationale behind this is in case of the tie, we implicitly assume the momentum will continue. If the t-2 to t-1 period return is still 0, then we classify the bin of t-1 to t as neutral and return 0. (I.e. this particular trade has no weight in calculating signed order imbalance)

Note our classification algorithm is inherently different from Lee-Ready’s, because it skips the first step in the Lee-Ready algorithm, which is to first compare the transaction price with the nearest available quote midpoint, and only proceed to trades comparison if the price falls in the middle of the quote midpoint, whereas in our algorithm we only utilize trades.

Interested readers are encouraged to refer to Lee and Ready’s research paper (Lee and Ready [1991]). In the appendix, we also provide a short description of Lee-Ready algorithm for readers’ convenience.

2a) Nearest quotes with different range of quotes delays.

For each trade, this algorithm looks at the nearest available quote ahead of it (in terms of exchange time). If the transacted price is closer to the bid of the quote, we classify the trade as a sell by setting b to -1 (i.e. we are taking the bid). Alternatively if the transacted price is closer to the ask of the quote, the trade is classified as a buy (or b = 1). If the transacted price is equidistant, we assigned the trade as neutral.

The trades reported to the consolidated tape are usually later than the reported quotes. In order to remedy the problem, we introduce different range of delays to the quotes and follow the same algorithm for classification. Interested readers can refer to Lee-Ready’s paper (Lee and Ready [1991]), in which they discussed about how to calculate the optimal delay.

2b) Time weighted EMA quotes classification.

The third classification scheme first involves calculating time weighted EMAs of all the quotes, and starting with the nearest EMA quote, technique 2) is then applied, with the only change being the usage of EMA quotes instead of regular quotes.

The EMA calculation is defined as:

**(2)**

**(3)**

1. Classic Lee-Ready with and without delay.
2. Classic Lee-Ready with and without delay using EMA mid quotes.

**Tick Data with Time bucketing Analysis:**

Before proceeding to building an actual predictive model, it is important to first ascertain our intuition that SOI is well correlated with price returns. The simple linear regression model we used to verify is as follows:

, **(4)**

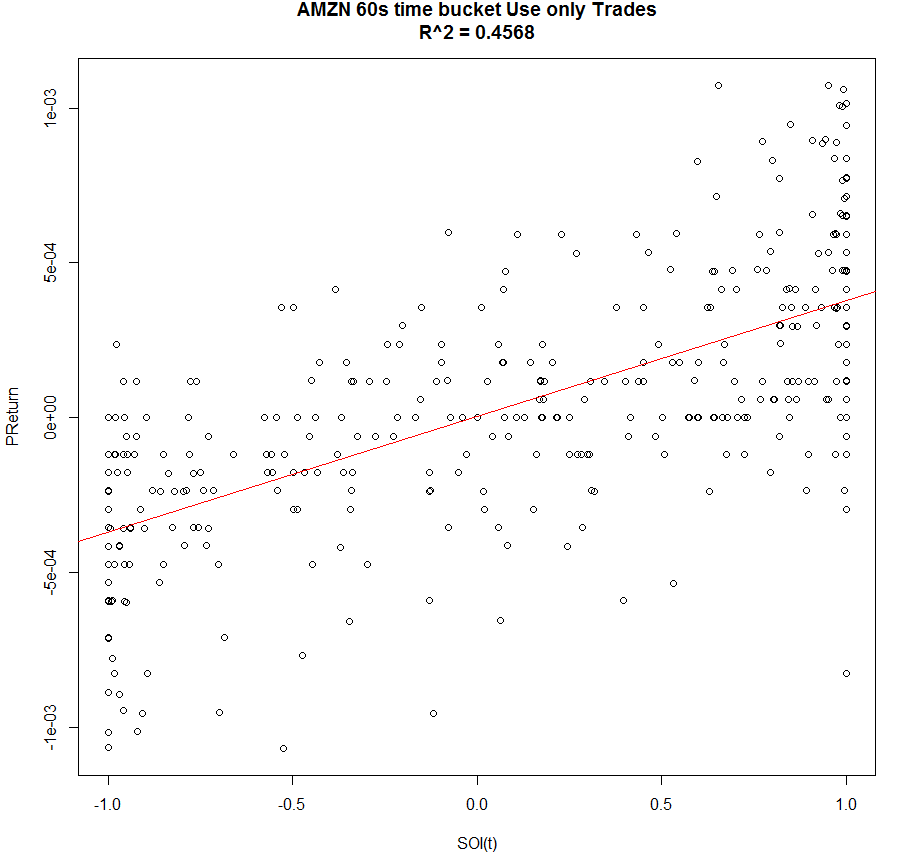
**(5)**

Here we define using tick level data (i.e. each bin contains only one trade), and time buckets. The size of the time bucket is dependent on the average trading volume of the stock of interest:

**(6)**

In the above equation, we fixate AMZN stock’s time bucket to be 60s. For other stocks with less trading volumes, their time buckets will be larger, for we would need to aggregate more data points. Since we are using linear scaling, we also want to cap the minimum bucket size to a constant to avoid bucket overflows.

To help establishing feasibility, we tested AMZN stock for 4/23/2013, ran the simple regression model, and apply classification rule 1. Below plot is the regression outcome (we take out all outliers that are 2 standard deviations away from the mean):



It is clear from the above plot SOI is strongly correlated with PReturn defined as mid-quote bucket return.

In order to precisely depict this relationship, we have first studied the optimized delay time for quotes. Since the exchange time field in the data is the time consolidated tape receive the message rather than the actual time that the quotes/trades happened.

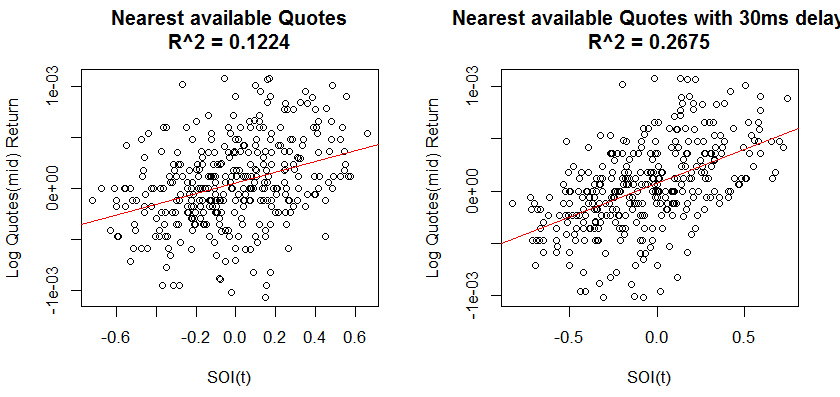
The process of transmitting data from Exchanges to the tape could introduce additional delays, presumably both trades and quotes. For simplicity, assume all transactions take place at trades, but the quotes are reported a constant time distance ahead. We run the same regression as above with all the quotes delayed by a constant time. (In addition, we also exclude the points Preturn/SOI is outside the two standard deviation range from its mean.) We run the regression on various symbols including TIF, AMZN, AB, and AGN, on different dates from April the 23rd to the 30th, and plot the R^2 against delay time.

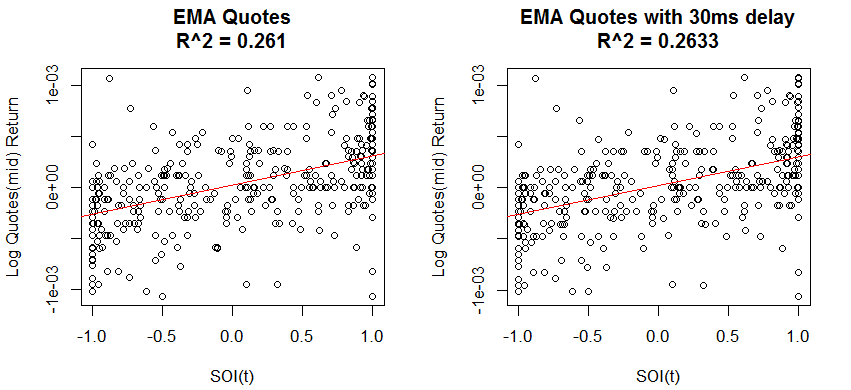
The plots of R^2 against delays are as the following: the left is second level delays, and the right is the zoom in part looking at milliseconds level. (The plots of all tested data exhibit the same pattern as the AMZN result on April 23rd we gave bellow)

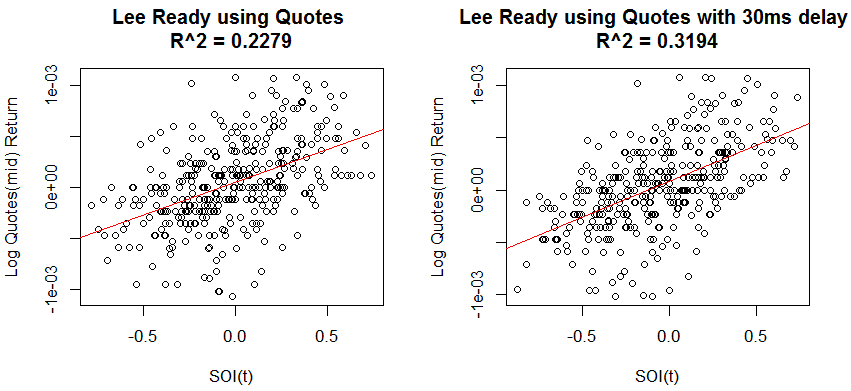
|  |  |
| --- | --- |
|  |  |

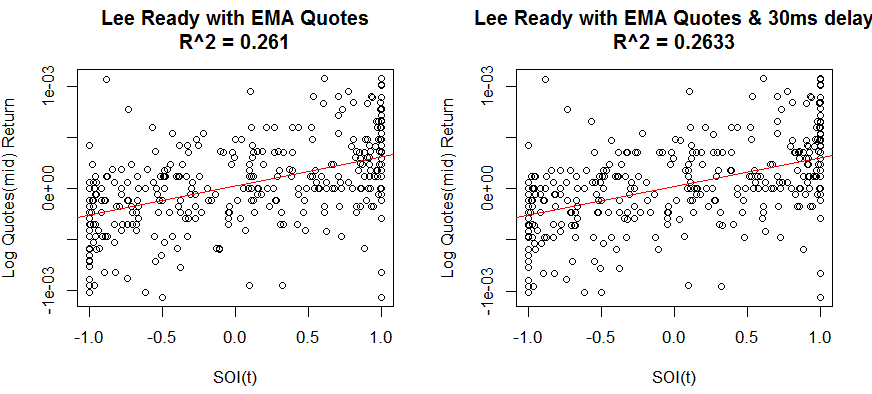
It is clear that the regression is locally optimized at e1 millisecond level when delay is around 30 milliseconds, and the universal optimized point is at e1 seconds level, around 30 seconds. However, the 30 milliseconds is chosen as the “optimized quotes delay time” because e1 seconds level is contradictory with the fact that the time to transmit data from Exchanges to the tape is at a millisecond level. (In later part of this research report related to venue liquidity, we will use a different methodology in calculating the optimal delay, but we will show that both end results are consistent with each other. As for the 30 seconds delay introduces such a high R^2, we defer the discussion to the next follow up report.

Below we show the regression results with only changes in the classification rules employed (taking out outliers that are 2 standard deviations from the mean), and according to R^2, classification rule #1 was the clear winner:



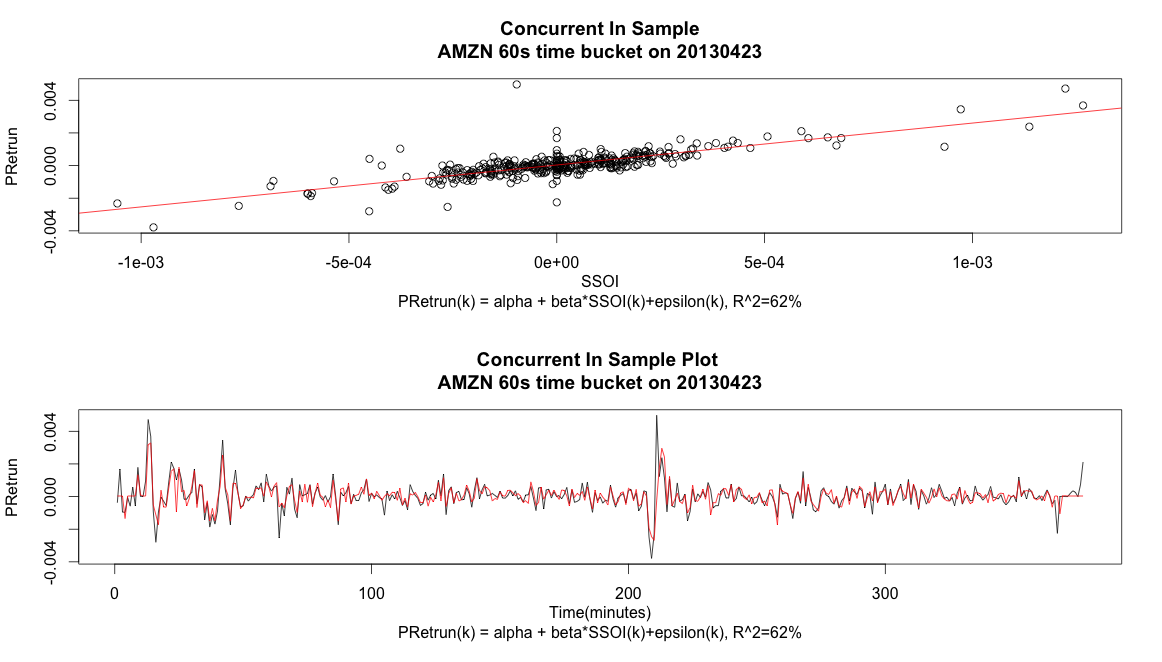






Classification rule #1 serves our purpose well. Note it is very likely all these classification methodologies that incorporates quotes delay of 5 seconds are likely to generate similar results; but since we were content with regression result, we moved on to improve this existing model. We were able to further adjust our model by scaling the SOI. The scaled SOI (SSOI) is obtained by multiplying SOI with the trade price volatility. And the price volatility of a bucket is defined as the standard deviation of all the trade prices within same bucket.

**(7)**



We then tested the model robustness across time by assessing the beta stability. And 4/29/2013’s AMZN data is used as an out of sample testing set.

Using the same SSOI sensitivity (beta), the concurrent regression model captures the price movement direction well, and correctly predicts 94.2% of the price movement direction on 29th. This ratio is calculated as . Therefore, we can conclude the SSOI sensitivity coefficient is relatively stable across days.

This model can also be successfully applied to any stock, and in Appendix I, we include the same concurrent model regression results for a list of different stocks across different sectors. The results corroborate to the conclusion that SOI scaled by volatility has a great explanatory power over mid quote returns.

Next we explore how to use this information to predict mid quote price returns. And the equation below summarizes our prediction model:

**(8)**

where measures partial SSOI, which only incorporates trades from the second half of a time bucket.

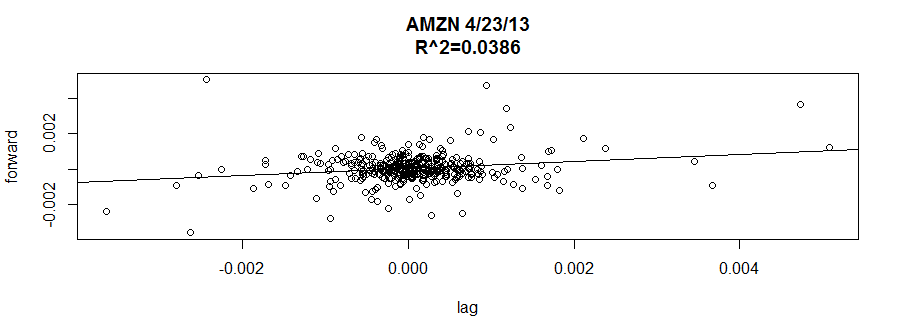
And we compare this model (equation (8)) with a simple autoregressive model:

**(9)**

The reason we even consider an autoregressive model is because our initial concurrent plots show good correlation between price movement and SSOI. And one important conclusion we can reach from the previous observation about predictive model based on SSOI is that such a model implicitly makes the assumption that there exists autocorrelation within the price changes! And the value SSOI can add is to make the price prediction more accurate, assuming there exists autocorrelation among the quote prices. To provide an example, we again use AMZN stock for 4/23/13 with 60 seconds time bucket.

Here are the results:

If we use only the lag return, the model R



The R^2 = 3.86%, with a p-value = 4.08e-5

After employing model **(8)**, the R^2 increases to 6.42%, with an adjusted R^2 of 5.6%, and a p-value = 4.084e-5. We expand the analysis to multiple stocks for six different days, and we record adjusted R^2 improvement resulted from using our model against R^2 obtained using simple auto-regression. (See Appendix I for details)

The results are satisfactory. For p-values < 0.1, the model achieves higher adjusted R^2 in 82.5% of the cases. And for p-values < 0.05, this model achieves higher adjusted R^2 in 90.91% of the cases. These results suggest when there is autocorrelation present in mid quote returns, adding SSOI as additional independent variables can improve the performance of our predictive model.

Discussion of SSOI measures as a predictor:

SSOI measure as a price change predictor is not strong. This is clearly evident by the weak R^2 results presented in the Appendix. Its weak predictive power can potentially be explained by SSOI’s strong correlation with concurrent price change.

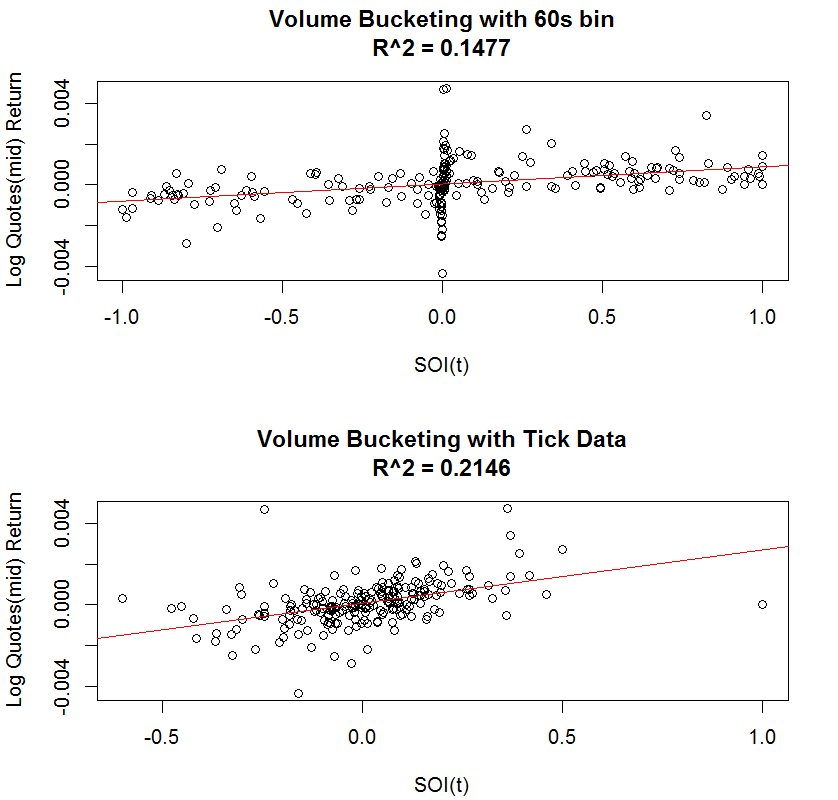
There is significant multicollinearity in this model. Moticollinearity is a statistical phenomenon when two or more predicting variables are highly correlated. In this model, high correlations exist among all three predictors. The correlation between lag quote returns and are already demonstrated, and the correlation between and is strong since both measures share the data used for computation. One problematic consequence of multicollinearity is to effectively estimate the coefficients of each independent variable. From our observation, these coefficients fluctuate widely on a daily basis:

Multicollinearity, however, does not undermine the model’s predictive power as a whole, and R^2 improvements after adjustment introduced by both SSOI measures suggest values in using SSOI as a predictive variable.

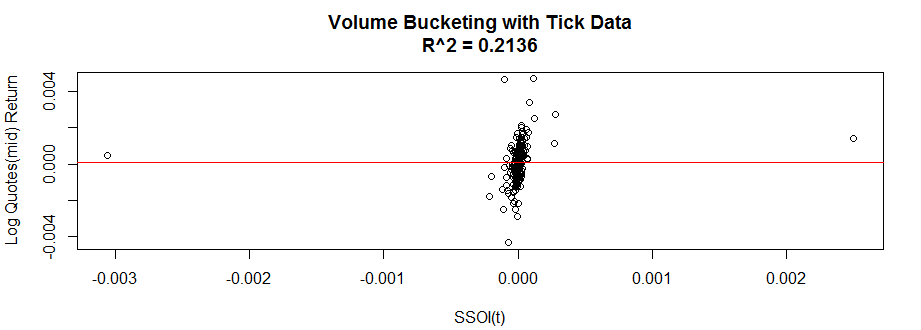
**Tick Data with Volume bucketing Analysis:**

In this section, we strive to examine the effect of using time bin instead of tick data; and we explore the adjustment we need to make to the previous predictive model and what are the essential differences between the two bucketing mechanisms.

First, we compare the difference in using time bins and tick data. We have again used 4/23/2013’s AMZN data as a testing set. Unlike ELO has suggested that using trade-by-trade classification can lead to great misclassification, trade-by trade classification clearly is a winner here:



Another phenomenon we have noticed is scaling the SOI measure by a volatility measure does not increase model fitting accuracies, the following plot is a clear demonstration:



And for the prediction model, we modify equation **(8)** to be

**(10)**

And we repeat the same procedure. This time, for p-value < 0.1, the predictive model beats the autoregressive model only 46.67% of the times, and for p-value < 0.05, this number drops to 40%. (for details, please check Appendix I). Yet we have also noticed the general drop in the % of p-values < 0.1. This is what is happening: by using volume bucketing, we are creating unequally spaced time forecasting intervals (unless trades occur uniformly throughout the day, and this is clearly false). Therefore, it is not realistic to employ the same autocorrelation model as a benchmark. Yet, if lag bucket mid quote returns no longer have any predictive power, then the same model based on SOI falls apart as well. This is the conclusion we have reached in the previous section.

Therefore, we conclude Tick Data/time bin with time bucketing should be the ideal way to construct SOI, which can be further utilized to improve an existing price change prediction model, given we can observe autocorrelation among short-term lags.