IB9BL0 ADVANCED TOPICS IN FINANCE:

Market Microstructure

Deadline: 01 April 2016

Warwick ID: 1257635

WBS PhD Finance

"I declare that this work is entirely my own in accordance with the University's Regulation 11 and the WBS guidelines on plagiarism and collusion. All external references and sources are clearly acknowledged and identified within the contents. "

"No substantial part(s) of the work submitted here has also been submitted by me in other assessments for accredited courses of study, and I acknowledge that if this has been done it may result in me being reported for self-plagiarism and an appropriate reduction in marks may be made when marking this piece of work."

Contents

1 Introduction

Textbook asset pricing models adopt an idealised picture of a frictionless and self-equilibrating market to analyse returns and prices of financial assets. In that idealised version of the trading process, all potential participants are present on the market; these participants use market orders that reflect their demand or supply of securities, and they are not affected by behaviours of other market players; and an auctioneer balances the quantities demanded and supplied at a single equilibrium price which reflects a consensus view of the security's "fundamental value." Real-world markets do not work like this for two main reasons: 1) market participants are not present on the market at the same time, and 2) the participants who are present at any instant in a real-world financial market have heterogeneous information about the security's fundamentals.

Accordint to Foucault et al. (2013), the study of market microstructure sheds light on two key aspects of real-world markets that are neglected by textbook asset pricing models: liquidity and price discovery. *Liquidity* is the degree to which an order can be executed within a short time period at a price close to the security's consensus value. *Price discovery* is the speed and accuracy with which transaction prices incorporate information available to market participants, which is closely related to *market efficiency.*

The two aspects, liquidity and market efficiency, are the focus of this work, which offers numerical, graphical and comparative analyses of different measures of liquidity and market efficiency for two stocks with different trading volumes:

- A frequently traded stock, Procter and Gamble Co. (company ticker PG),
- An infrequently trade stock, Material Science Co. (company ticker MSC).

For each stock, the sample contains TAQ data for one trading week in each year over 2006-2008:

- 06 March to 10 March in 2006,
- 05 March to 09 March in 2007,
- 03 March to 07 March in 2008.

2 Averages of Spread Measures: Empirical and Graphical Analysis

This section focuses on three spread measures to estimate liquidity:

- Quoted Spread: $s = \dfrac{a-b}{m}$, where a and b are ask and bid prices, m is midprice,

  which reflects the liquidity available at a given point in time for a hypothetical transaction.

- Effective Spread: $S_e = 2\left(\dfrac{d(p-m)}{m}\right)$, $d = 1 \ or \ -1$ for a buy or sell order,

  which is defined as the difference between the price at which a market order executes and the midquote on the market the instant before and which measures trading costs using the prices actually obtained by investors.

- Realised Spread: $S_r = 2\left(\dfrac{d_t(p_t - m_{t+\Delta})}{m_t}\right)$,

  which is defined as the difference between the transaction price and the midprice at some time, Δ, after the transaction (say five later), where the interval Δ should be long enough to ensure that market quotes have adjusted to reflect the price impact of the transaction. It can be seen as a measure of the profit earned by the *liquidity supplier* on the transaction at time t if he unwinds his position at the midprice at $t + \Delta$.

2.1 Daily and Hourly Averages of Spread Measures: Quoted Spread (QS), Effective Spread (ES) and Realised Spread (RS)

Table 2.1: Daily Averages of Spread Measures

	Illiquid: MSC			Liquid: PG		
	2006	2007	2008	2006	2007	2008
			Panel A: Quoted Spread (QS)			
QS (1)	0.693%	0.473%	0.656%	0.0263%	0.0197%	0.0183%
QS (2)	0.713%	0.420%	0.736%	0.0248%	0.0175%	0.0175%
QS (3)	0.834%	0.435%	0.747%	0.0248%	0.0191%	0.0191%
QS (4)	0.438%	0.393%	0.580%	0.0226%	0.0184%	0.0181%
QS (5)	0.375%	0.516%	0.651%	0.0232%	0.0191%	0.0180%
			Panel B: Effective Spread (ES)			
ES (1)	0.526%	0.387%	0.385%	0.0216%	0.0213%	0.0184%
ES (2)	0.589%	0.318%	0.546%	0.0220%	0.0187%	0.0181%
ES (3)	0.564%	0.332%	0.573%	0.0216%	0.0192%	0.0198%
ES (4)	0.326%	0.317%	0.464%	0.0200%	0.0181%	0.0184%
ES (5)	0.296%	0.334%	0.466%	0.0203%	0.0182%	0.0186%
			Panel C: Realised Spread (RS)			
RS (1)	0.127%	0.236%	0.218%	0.0121%	0.0031%	0.0102%
RS (2)	0.321%	0.092%	-0.102%	0.0078%	0.0111%	0.0076%
RS (3)	0.396%	0.255%	0.007%	-0.0013%	0.0072%	0.0138%
RS (4)	0.245%	0.117%	0.314%	0.0048%	0.0092%	0.0138%
RS (5)	0.039%	0.192%	0.440%	-0.0048%	0.0041%	0.0112%

(1) to (5) = day 1 to 5.

From Table 2.1 Panel A,

- For the inactively traded stock MSC, the daily averaged quoted spread (QS) decreases from 2006 to 2007 then increases from 2007 to 2008, for the same period in each (except day 5).

- For the actively traded stock PG, however, the quoted spread decreases monotonically from 2006 to 2008 for any given day.

From Panel B,

- For the inactively traded stock MSC, the daily averaged effective spread (ES) shares the same patterns with quoted spread for the 2nd, 3rd and 4th day of a trading week. Nonetheless, there is a monotonic increase in ES from 2006 to 2008 for the 1st day and a monotonic decrease for the last (5th) trading day.

- For the actively traded stock PG, the pattern is similar to that for QS.

From Panel C,

- For the inactively traded stock MSC, there is no clear pattern across three years. However, it is worth noting that RS can be very small or even negative.
- For the actively traded stock PG, however, the daily averaged realised spread (RS) tends to increase from 2006 to 2008, opposite to the patterns for QS and ES.

In general, the three types of spreads are good measures of liquidity. Because MSC has much larger spread measures, and thus higher trading costs, than PG, we can refer to MSC and PG as illiquid and liquid stocks.

Since the macroeconomic conditions in U.S. started to deteriorate from the end of 2007, it seems that the inactively traded stock MSC begun to suffer from the liquidity loss much earlier than the actively traded stock PG as investors reduced risky investment in response to increase systematic risk. The liquidity loss is again captured by the spread measures.

The intrady dynamics of hourly averages of spread measures for MSC and PG are shown in Figure 2.1 and 2.2. Some points are worth highlighting:

- Hourly spread measures for illiquid stock MSC are much larger than those for liquidity stock PG.
- For both MSC and PG, and for almost any trading day, QS and ES tend to be much higher for the beginning 1st hour of the day.
- QS and ES follow each other closely most of the time, whereas there is in general a big difference between RS and ES (as well as QS).

The last point can be explained by the following decomposition of realised spread:

$$S_r = \frac{d_t(p_t - m_{t+\Delta})}{m_t} = \frac{d_t(p_t - m_t)}{m_t} - \frac{d_t(m_{t+\Delta} - m_t)}{m_t}$$

$$\Rightarrow \quad \text{Realised Spread} = \text{Effective Spread - Price Impact}$$

$$\textit{or} \;\; \text{Price Impact} = \text{Effective Spread - Realised Spread}$$

The difference between ES and RS tends increase as the price impact increases, and vice versa.

Figure 2.1: MATERIAL SCIENCES CORP (MSC), Intraday Dynamics of Hourly Averages of QS, ES and RS

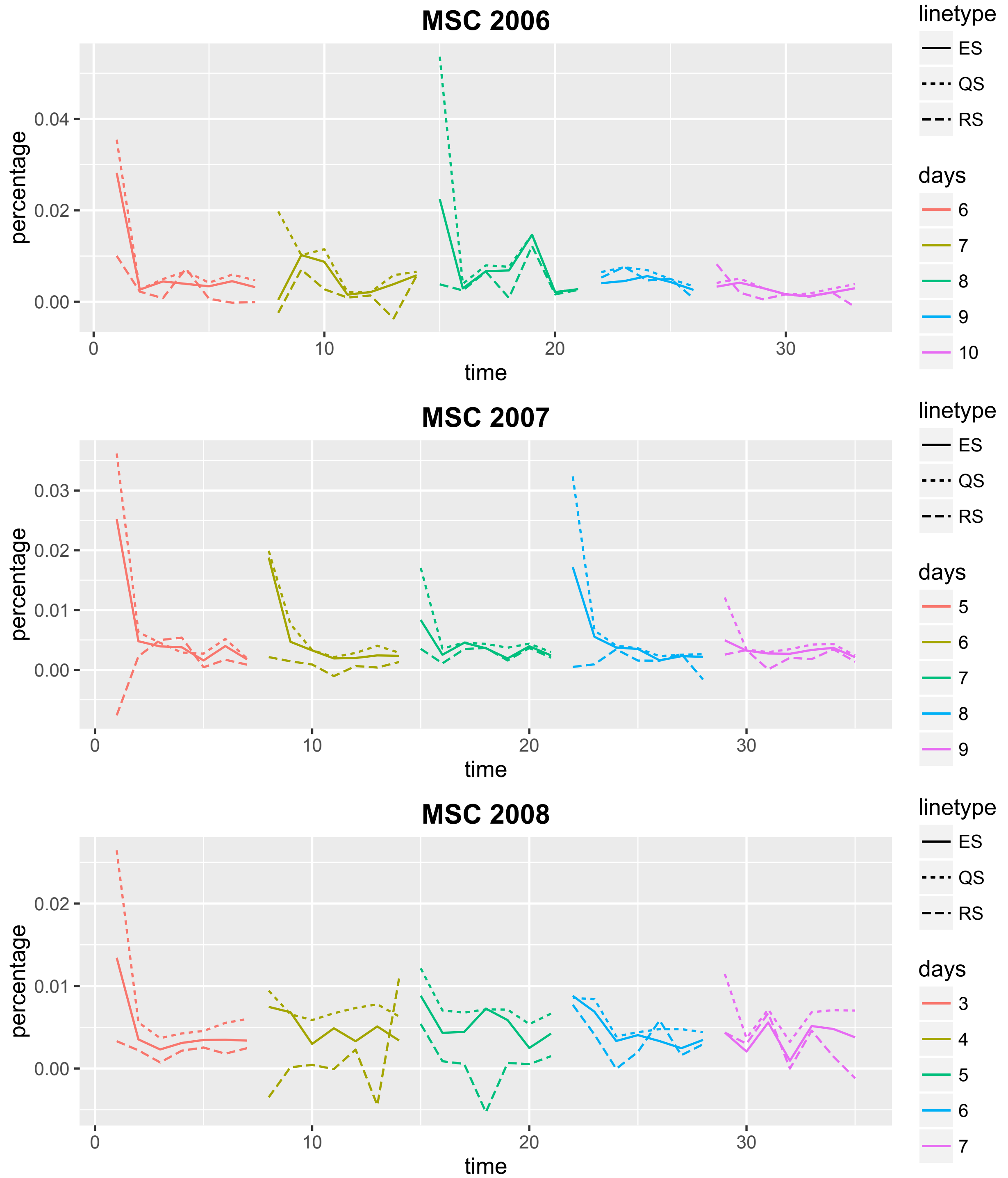
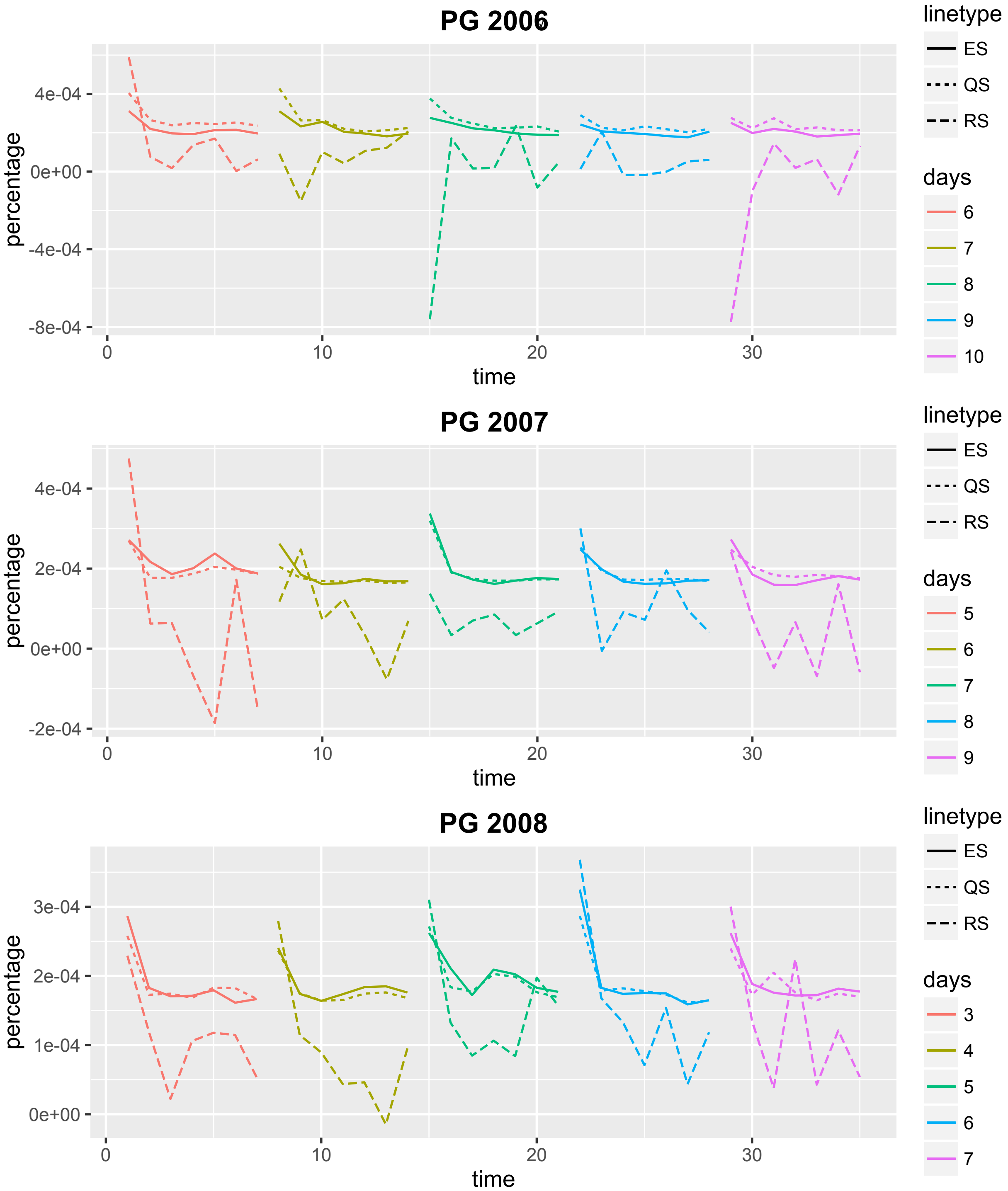


Figure 2.2: PROCTER and GAMBLE CO (PG), Intraday Dynamics of Hourly Averages of QS, ES and RS



2.2 Correlations of Spread Measures with Amihud Illiquidity Ratio

Table 2.2: Correlation with Amihud (hourly)

	Illiquid: MSC			Liquid: PG		
	2006	2007	2008	2006	2007	2008
Quoted Spread (QS)	0.124	0.676	0.852	0.508	0.503	0.468
Effective Spread (ES)	0.122	0.708	0.633	0.502	0.424	0.501
Realised Spread (RS)	0.265	-0.706	0.074	0.414	0.448	0.288

The Amihud illiquidity ratio (denoted as Amihud) is a daily price impact measure and is computed as

$$Amihud_d = \frac{|Return_d|}{\$Volume_d} = \frac{|Return_d|}{Volume_d \cdot Midprice_d}$$

Hence, the Amihud can be interpreted as a measure of the price change associated with one additional unit of dollar volume. The higher the Amihud, the more illiquid.

From Table 2.2,

- For illiquid stock MSC, Amihud is highly positively correlated with QS and ES only in 2007 and 2008. The correlations of Amihud with QS and ES are weak in 2006. Moreover, the correlation between Amihud and RS is weak in 2006 and 2008 and is highly negative in 2007.

- For liquid stock PG, there is a moderately high correlation of Amihud with all three spread measures across all the three years.

3 Effective Spread (ES) for Transactions of Different Sizes

This section focuses on the intraday dynamics of hourly averages of ES for three trade size categories: between 0 and 500 (0-500), 500 and 1000 (500-1000), and 1000 and above (1000+).

The intraday graphs for MSC and PG are shown in Figure 3.1 and 3.2. The open circle and close circle are introduced for lines representing hourly averages of ES with size 500-1000 and with size 1000+, respectively. This leads to clearer graphical presentation since the number of trades for sizes over 500 is substantially less than that for sizes between 0 and 500. Larger orders with sizes over 1000 are very rare and may not exist in many cases, especially for the inactively traded stock MSC.

The most important implication from the two graphs is that the larger the trade size, the higher the effective spread. This pattern is common for both inactively and actively traded stocks and across years, although some exceptions exist. This is consistent with the fact that large trade on average has high price impact and contains high asymmetric information.

It is also clear that the less the stock is actively traded, the less frequent a large order is placed by investors.

Figure 3.1: MATERIAL SCIENCES CORP (MSC),
Hourly Averages of ES with Different Sizes: 0-500, 500-1000 and 1000+

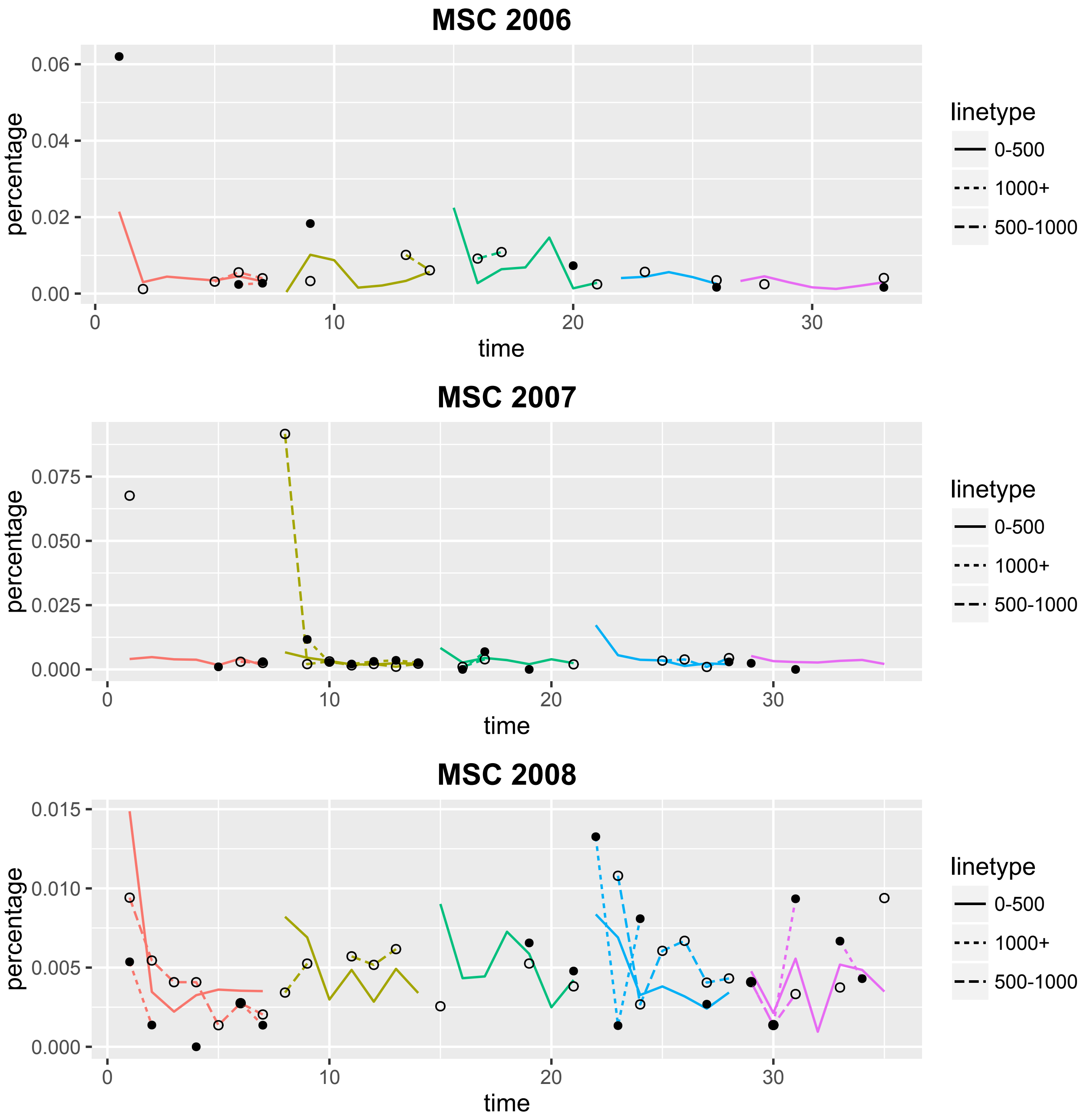
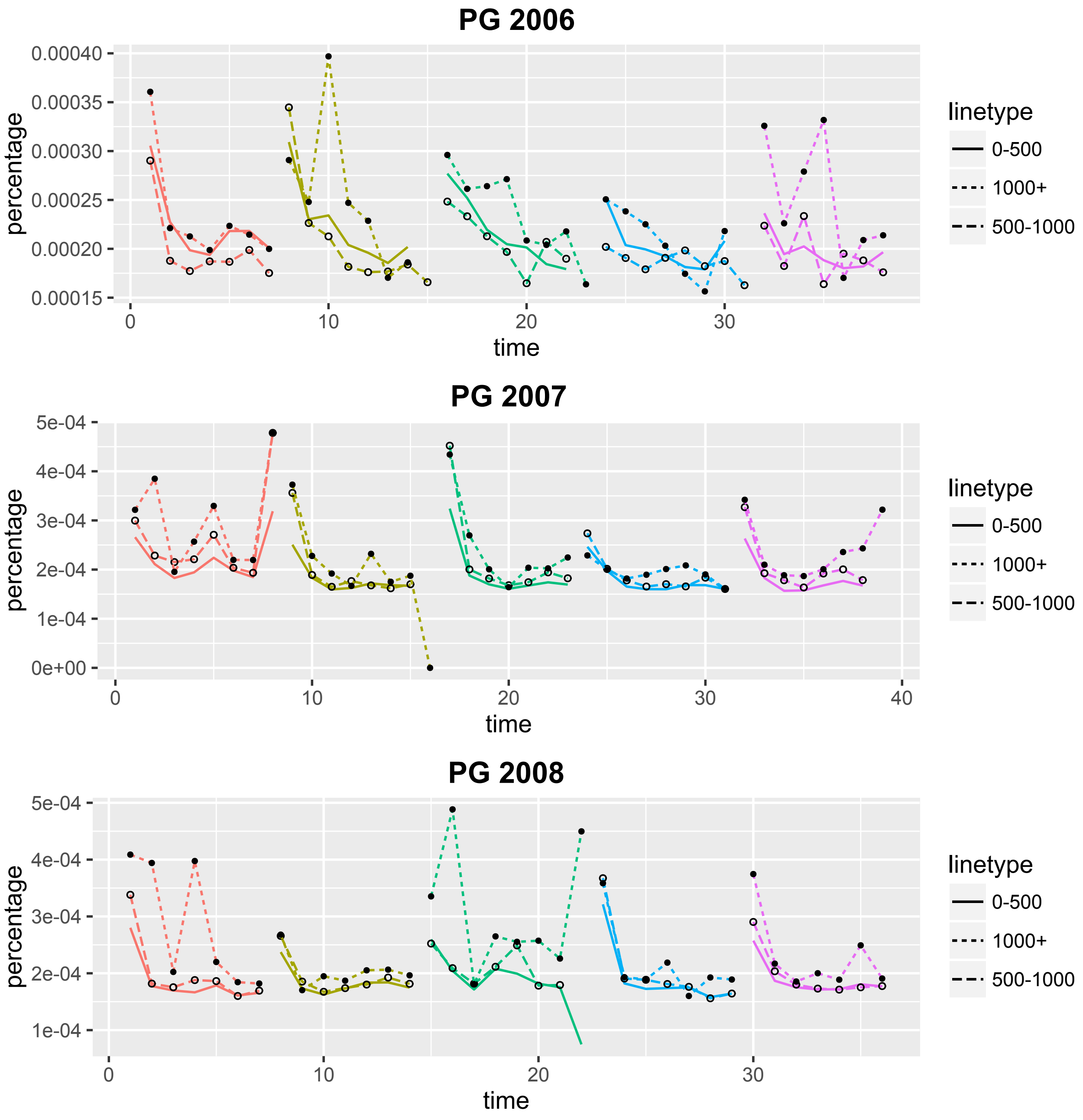


Figure 3.2: PROCTER and GAMBLE CO (PG),
Hourly Averages of ES with Different Sizes: 0-500, 500-1000 and 1000+


PG 2006

PG 2007

PG 2008

4 Volume-Weighted Average Price (VWAP)

Previous spread measures of liquidity and trading costs require data on bid and ask quotes, but these are sometimes simply unavailable. However, we can still measure implicit trading costs use time series of recent transaction prices and trading volume. Volume-weighted average price (VWAP) is one of the measures of implicit trading costs.

The normal VWAP as well as VWAP based on buyer-initiated trades ($VWAP_{buy}$) and on seller-initiated trader ($VWAP_{sell}$) are computed and compared. A general pattern can be seen from Figure 4.1 and 4.2 that $VWAP_{buy} > VWAP > VWAP_{sell}$, although the reverse is true on some occasions. The pattern is expected since a buy (sell) order tends to more the price up (down).

Figure 4.1: MATERIAL SCIENCES CORP (MSC), Daily $VWAP, VWAP_{buy}, VWAP_{sell}$

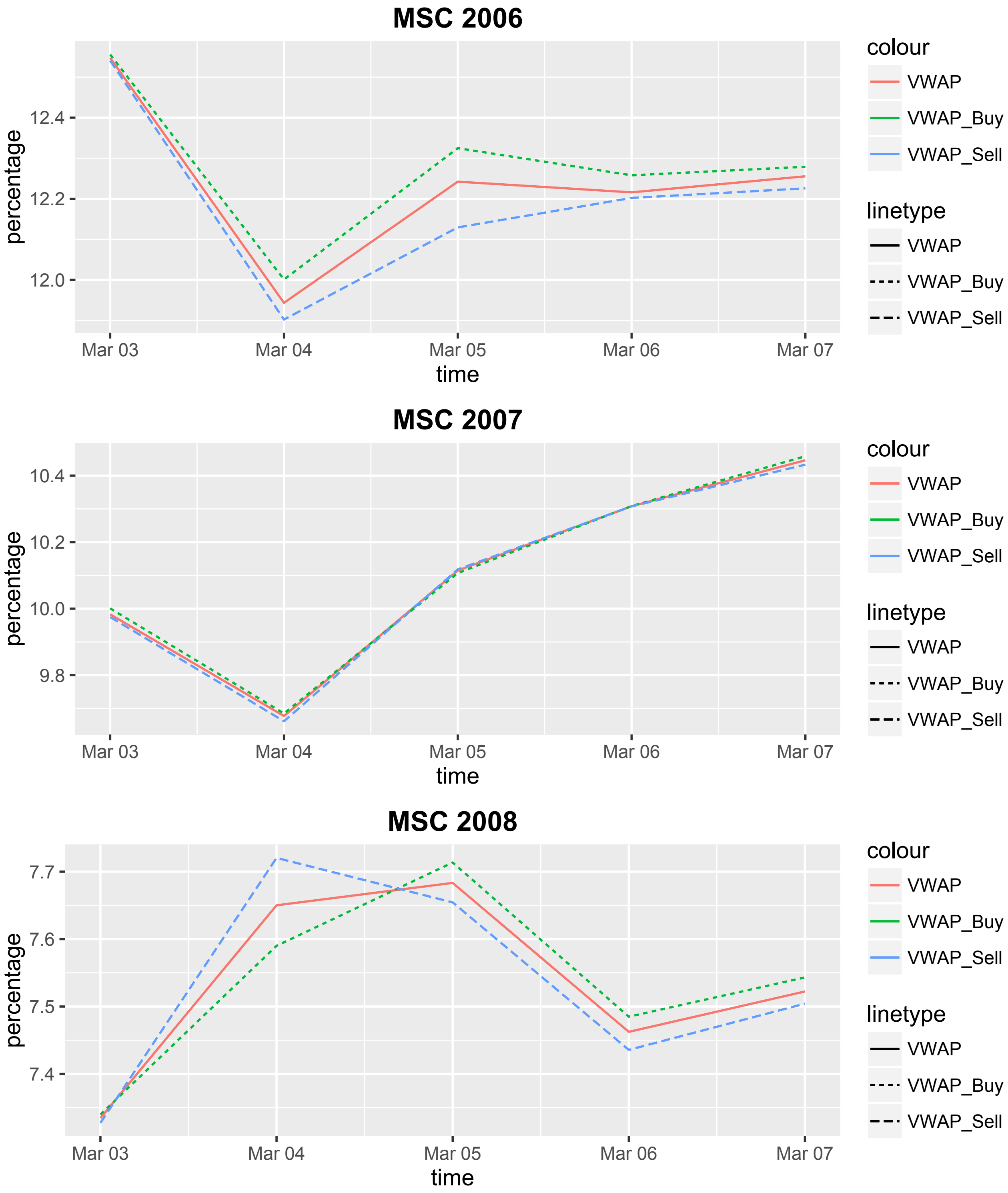
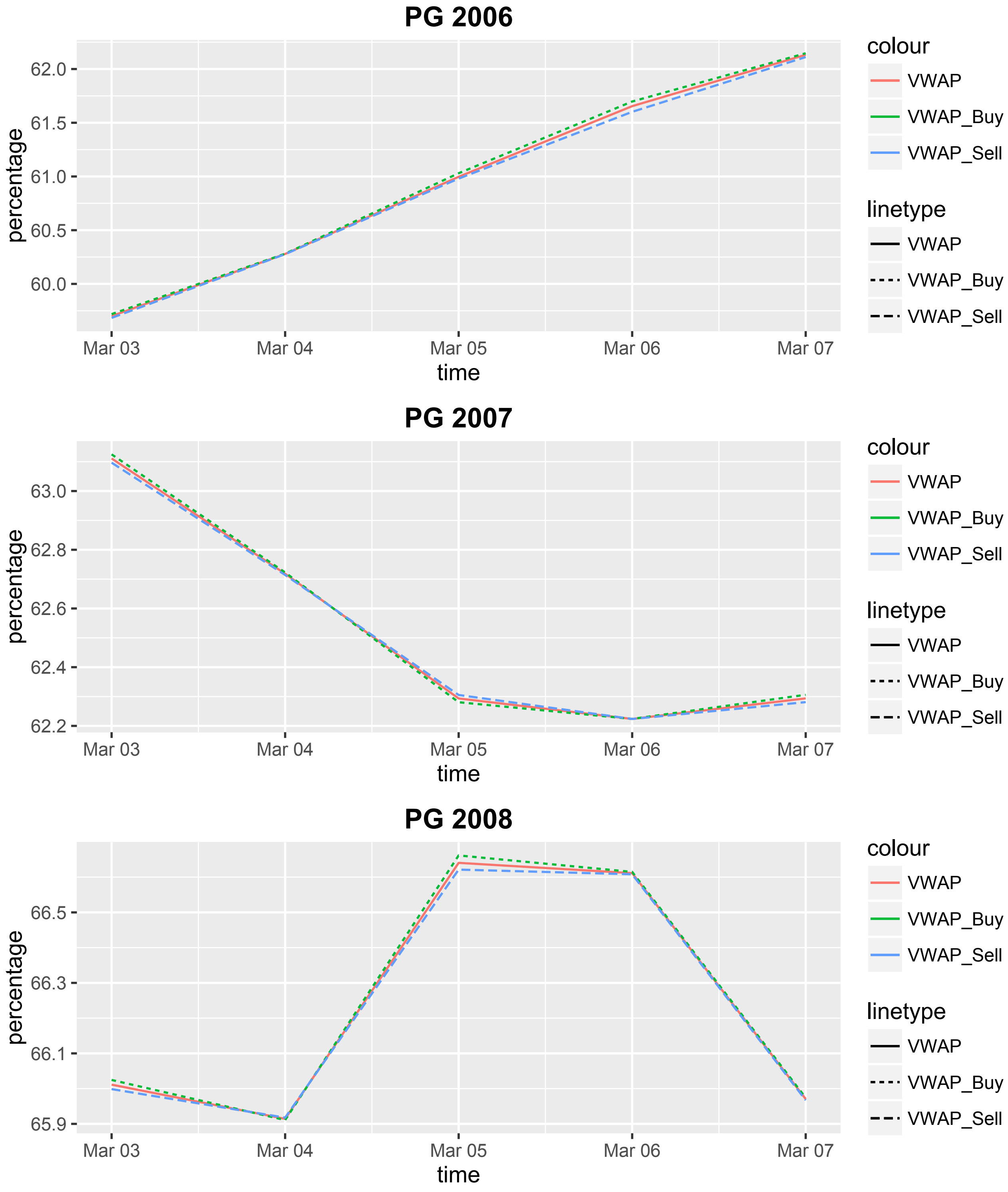


Figure 4.2: PROCTER and GAMBLE CO (PG), Daily $VWAP, VWAP_{buy}, VWAP_{sell}$

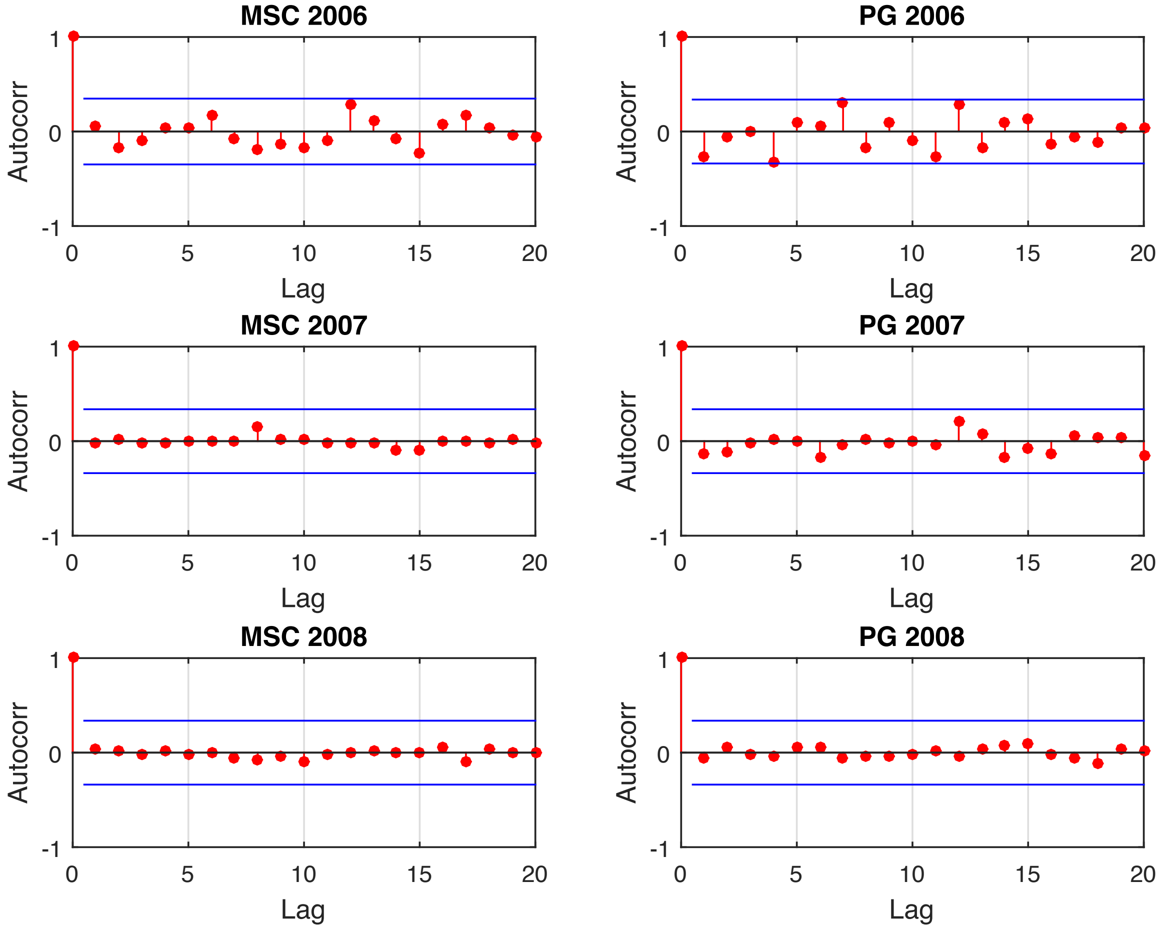


5 Market Efficiency Tests

5.1 Return Autocorrelations

The return autocorrelations are estimated using hourly returns for both MSC and PG. Surprisingly, there is no autocorrelation up to lag 20 for illiquid stock MSC, implying that the returns of illiquid stock follow a random walk and thus the prices are informationally efficient. This is unexpected because illiquid or inactively traded stocks are in general considered informationally efficient compared to liquid stocks. This may be caused by the estimation inaccuracy due to the small sample used in this work.

Figure 5.1: Return Autocorrelation Function (ACF)



5.2 Variance Ratios

The variance ratio is calculated based on the formula in Rösch et al., 2013 and the hourly return variance used in Bessembinder (2003):

$$Variance\ Ratio = \left| 1 - 6.5 \cdot \frac{VAR(1-hour)}{VAR(OC)} \right|,$$

where $VAR(1-hour)$ is the return variance estimated from 1-hour midprice returns within a day and $VAR(OC)$ is the return variance estimated from daily open-to-close midprice returns.

A value of zero indicates random walk and price efficiency. The higher the value, the less the price efficiency.

From Figure 5.2,

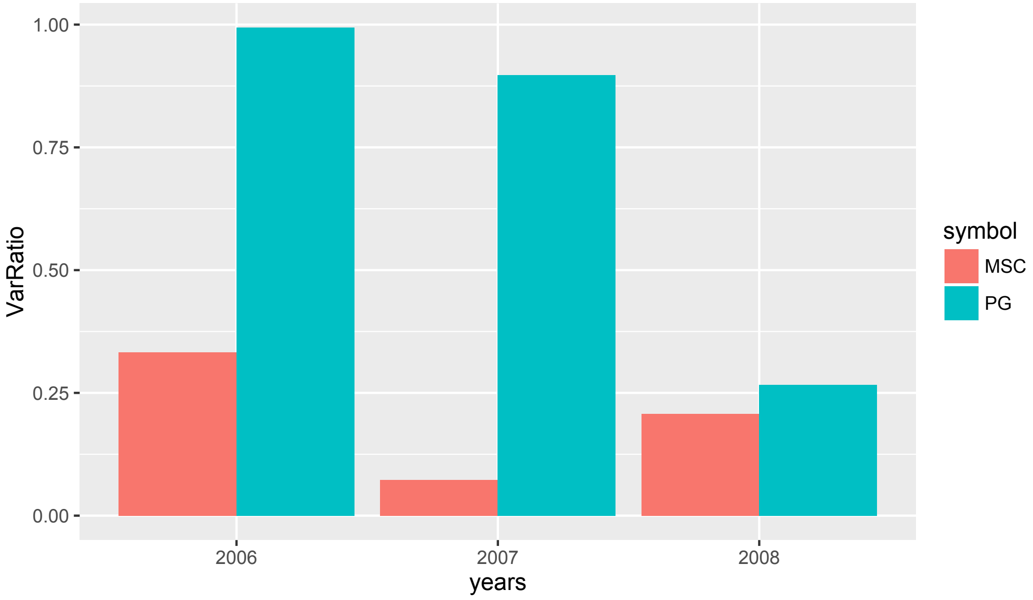
- For illiquid stock MSC, the variance ratio decreased in the same period from 2006 to 2007 and then increased from 2007 to 2008.

- For liquid stock PG, the variance ratio decreased monotonically from 2006 to 2008.

- Unexpectedly, the ratios for illiquid stock MSC are much lower than those for liquid stock PG.

Again, this unanticipated result may be due to small sample measurement biases, especially for the illiquid stock MSC that has much less intraday data.

Figure 5.2: Variance Ratio: MSC vs PG



5.3 Rösch et al. (2013) Regression: A Goodness-of-Fit Test

The market efficiency can also be measured by the R^2 obtained from the return predictability regression in Rösch et al. (2013) that regresses 1-minute midprice return[1] on lagged order imbalance (computed as the difference between the total dollar amount of trades initiated by buyers and sellers):

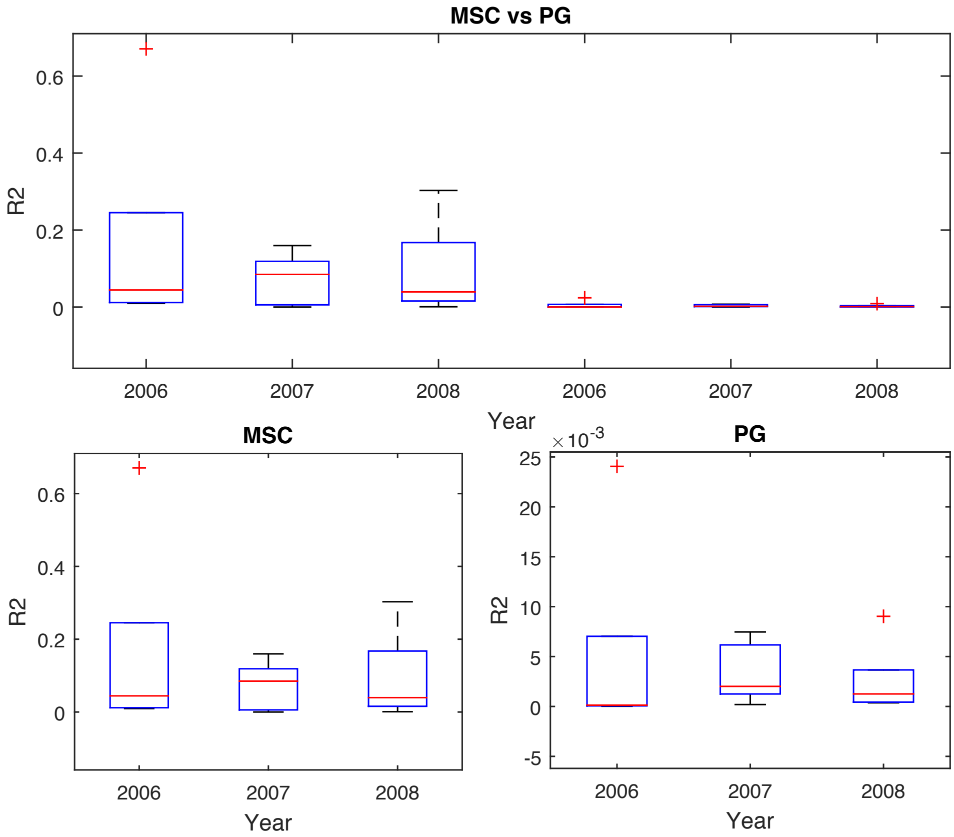
$$R_{i,d,t} = \alpha_{i,d} + \beta_{i,d} OIB_{i,d,t-1} + \varepsilon_{i,d,t}.$$

The less the R^2, the higher the price efficiency.

As is expected, the R^2 for liquid stock PG is much smaller than that for illiquid stock MSC (from Figure 5.3). This result is consistent with the fact that liquid stocks are generally more informationally efficient. However, There is no clear pattern across the years.

[1] The 1-minute return is applied to the liquid stock PG. For the illiquid stock MSC, the 1-hour returns are computed due to the infrequency of trades.

Figure 5.3: R^2 from Rösch et al. (2013) Regression: MSC vs PG



6 Conclusion

Main findings from each section are summarised as follows:

- **Section 2:**
  - The liquidity, measured by (lowered) quoted spread, effective spread and realised spread, tends to increase over the years,
  - Cross sectionally, spread measures are lower (i.e. liquidity is higher) for more actively traded stocks,
  - Regarding the intraday dynamics, there is a general pattern that quoted and effective spread are much higher for the beginning 1st hour than the rest time of the day,
  - A difference exists generally between effective and realised spread, which represents the price impact,
  - Overall, there are high, positive correlations between Amihud illiquidity ratio and quoted/effective spread; the correlation between Amihud and realised spread can sometimes be weak and even highly negative.
- **Section 3:**

  - Larger the trade size, the higher the effective spread, for both inactively and actively traded stocks; this is consistent with the fact that large trade on average has high price impact and contains high asymmetric information,
  - The less the stock is actively traded, the less frequency a large order is placed by investors; this is consistent with the fact that large investors tend to focus on large cap, actively traded stocks.

- **Section 4:**
  - There is a general pattern that $VWAP_{buy} > VWAP > VWAP_{sell}$, which is expected since a buy (sell) order tends to more the price up (down).

- **Section 5:**
  - There is no return autocorrelation for both illiquid stock MSC and liquid stock PG; the estimation of zero autocorrelation for the illiquid stock may be caused by the inaccuracy due to the small sample used in the current work,
  - In terms of variance ratio, the estimates for illiquid stock MSC are much lower than those for liquid stock PG for all the three years, which is unexpected as low ratio (close to zero) indicate high price efficiency; this, again, may be due to small sample measurement biases, especially for the illiquid stock MSC that has much less intraday data.
  - Regarding the Rösch et al. (2013) regression, the R^2 for liquid stock PG is much smaller than that for illiquid stock MSC; this result is consistent with the notion that liquid stocks are generally more informationally efficient.
  - Overall, this is an inconsistency in results with different measures of market efficiency, which is believed to result from the small sample biase, in particular for the illiquid stock MSC.

References

Bessembinder, H. (2003). Trade execution costs and market quality after decimalization, *Journal of Financial and Quantitative Analysis* **38**(04): 747–777.

Foucault, T., Pagano, M. and Röell, A. (2013). *Market liquidity: theory, evidence, and policy*, Oxford University Press.

Rösch, D. M., Subrahmanyam, A. and van Dijk, M. A. (2013). An empirical analysis of comovements in high-and low-frequency measures of market efficiency, *Unpublished working paper, Rotterdam School of Management, and University of California, Los Angeles* .

7 Appendix

7.1 Market Efficiency Test Results

	Illiquid: MSC			Liquid: PG		
	2006	2007	2008	2006	2007	2008
	Panel A: Volume-Weighted Average Price (VWAP)					
VWAP (1)	12.55	9.98	7.33	59.70	63.11	66.01
VWAP (2)	11.94	9.68	7.65	60.28	62.72	65.91
VWAP (3)	12.24	10.11	7.68	61.00	62.29	66.64
VWAP (4)	12.22	10.31	7.46	61.65	62.22	66.61
VWAP (5)	12.26	10.45	7.52	62.13	62.29	65.97
VWAPbuy (1)	12.56	10.00	7.34	59.72	63.13	66.02
VWAPbuy (2)	12.00	9.69	7.59	60.28	62.72	65.91
VWAPbuy (3)	12.32	10.11	7.71	61.03	62.28	66.66
VWAPbuy (4)	12.26	10.31	7.48	61.70	62.22	66.61
VWAPbuy (5)	12.28	10.46	7.54	62.15	62.31	65.97
VWAPsell (1)	12.54	9.97	7.33	59.68	63.10	66.00
VWAPsell (2)	11.90	9.66	7.72	60.28	62.71	65.92
VWAPsell (3)	12.13	10.12	7.65	60.98	62.31	66.62
VWAPsell (4)	12.20	10.31	7.44	61.60	62.22	66.61
VWAPsell (5)	12.23	10.43	7.50	62.11	62.28	65.97
	Panel B: Market Efficiency Tests					
Variance Ratio	0.33	0.07	0.21	0.99	0.90	0.27
R2 (1)	0.0125	0.0849	0.0207	0.0014	0.0002	0.0013
R2 (2)	0.0094	0.0000	0.3029	0.0000	0.0016	0.0019
R2 (3)	0.6707	0.1051	0.0008	0.0001	0.0075	0.0004
R2 (4)	0.0444	0.0077	0.1224	0.0001	0.0020	0.0090
R2 (5)	0.1034	0.1598	0.0395	0.0241	0.0057	0.0005

R2 = R squared from Rosch et al (2013) regressions,
(1) to (5) = day 1 to 5.

7.2 SAS Codes for Data Retrieval, Processing and Numerical Computation

```
/* HOLDEN AND JACOBSEN SAS CODE FOR NATIONAL BEST BID AND OFFER (NBBO)
   AND LIQUIDITY MEASURE COMPUTATION

   Based on: Holden, C. and S. Jacobsen, 2013, Liquidity Measurement
   Problems in Fast, Competitive Markets: Expensive and Cheap Solutions,
   Forthcoming in the Journal of Finance

   We welcome any comments or suggestions. We can be reached at:
   Holden: cholden@indiana.edu
   Jacobsen: staceyj@cox.smu.edu
   Of course, you use this code at your own risk.            */


/* Assignment for Advanced Topics in Finance - Market Microstructure

   Module Code: IB9BL0;

   Author: Dalong Sun (Warwick ID 1257635) */


/* Global settings */
options errors=50;
libname project 'C:\Users\phd15ds\Downloads\SAS_Database\';

/* Question 3: RETRIEVE MONTHLY TRADE AND QUOTE (MTAQ) FILES FROM WRDS
               AND DOWNLOAD TO PC */

/* Connect to WRDS */
%let wrds = wrds.wharton.upenn.edu 4016;
options comamid=TCP remote=WRDS;
signon username=_prompt_;

/* Submit SAS statements to WRDS */
rsubmit;
    libname taq '/wrds/taq/sasdata';
    option msglevel=i mprint source;

    /* Retrieve quote data */

    data monthlyquote;

        /* Enter MTAQ dates in YYYYMMDD format for Consolidated Quote files ("cq")*/
        set taq.cq_20060306 - taq.cq_20060310
            taq.cq_20070305 - taq.cq_20070309
            taq.cq_20080303 - taq.cq_20080307;

        /* Enter company tickers */

        /* 'MSC' = MATERIAL SCIENCES CORP
           'PG'  = PROCTER & GAMBLE CO */
        where symbol in ('MSC','PG') and

        /* Quotes are retrieved prior to market open time to ensure NBBO
        Quotes are available for beginning of the day trades*/
        (("9:00:00"t) <= time <= ("16:00:00"t)); format date date9.;
    run;

/* Retrieve trade data */
    data monthlytrade (drop=G127 COND);
```

```
        /* Enter the same MTAQ dates as above in YYYYMMDD format for
        Consolidated Trade files ("ct")*/
        set taq.ct_20060306 - taq.ct_20060310
            taq.ct_20070305 - taq.ct_20070309
            taq.ct_20080303 - taq.ct_20080307;

        /* Enter same company tickers as above */
        where symbol in ('MSC','PG') and

        /* Retrieve trades during normal market hours */
        (("9:30:00"t) <= time <= ("16:00:00"t));
        type='T';
        format date date9.;
    run;

    /* Download to PC */
    proc download data=monthlyquote out=project.monthlyquote;
    run;
    proc download data=monthlytrade out=project.monthlytrade;
    run;
endrsubmit;

/* Question 4: MERGE TRADES AND QUOTES AND IDENTIFY TRADE DIRECTIONS

/* STEP ONE: CLEAN QUOTE DATA */
data project.monthlyquote;
    set project.monthlyquote;

    /* Quote Filter 1: Abnormal Modes. Quotes with abnormal modes
       (i.e., abnormal quote conditions) are set to extreme values
       so that they will not enter the NBBO */
    if mode in (4,7,9,11,13,14,15,19,20,27,28) then do; OFR=9999999; BID=0; end;

    /* Quote Filter 2: Crossed Quotes on the Same Exchange.
        Quotes from a given exchange with positive values in which
        the Bid is greater than the Ask (i.e., crossed quotes) are set to
        extreme values so that they will not enter the NBBO */
    If BID>OFR and BID>0 and OFR>0 then do; OFR=9999999; BID=0; end;

    /* Quote Filter 3: One-Sided Bid Quotes.
        One-sided bid quotes (i.e., quotes in which the Bid is a positive value
        and the Ask is set to '0') are allowed to enter the NBBO; The Ask is set to
        an extreme value so that it will not enter the NBBO*/
    If BID>0 and OFR=0 then OFR=9999999;

    /* Quote Filter 4: Abnormally Large Spreads.
        Quotes with positive values and large spreads (i.e., spreads greater than $5.00)
        are set to extreme values so that they will not enter the NBBO */
    spr=OFR-BID;
    If spr>5 and BID>0 and OFR>0 and OFR ne 9999999 then do; BID=0; OFR=9999999; end;

    /* Quote Filter 5: Withdrawn Quotes.
        This is when an exchange temporarily has no quote, as indicated by quotes with price
        or depth fields containing values less than or equal to 0 or equal to '.'.
        See discussion in Holden and Jacobsen (2013), page 10. They are set to extreme values
        so that they will not enter the NBBO.
        They are NOT deleted, because that would incorrectly allow the prior quote from
        that exchange to enter the NBBO.
        NOTE: Quote Filter 5 must come last */
    if OFR le 0 then OFR=9999999;
    if OFR =. then OFR=9999999;
    if OFRSIZ le 0 then OFR=9999999;
    if OFRSIZ =. then OFR=9999999;
    if BID le 0 then BID=0;
    if BID =. then BID=0;
    if BIDSIZ le 0 then BID=0;
    if BIDSIZ =. then BID=0;
run;

/* STEP TWO: CLEAN TRADE DATA */
data project.monthlytrade;
    set project.monthlytrade;

    /* Trade Filter: Keep only trades in which the Correction field contains '00'
       and the Price field contains a value greater than zero */
    where corr eq 0 and price gt 0;
run;

/* STEP THREE: CREATE INTERPOLATED TIME VARIABLES
   Based on: Holden and Jacobsen (2013), pages 22-24 */
/* Create Interpolated Quote Time for quote dataset */
proc sort data=project.monthlyquote;
    by symbol date time;
run;

/* 'J' indexes the order of quotes within a given second */
data project.monthlyquote (drop=MODE spr);
    set project.monthlyquote;
    retain J;
    by symbol date time;
    if first.symbol or first.date or first.time then J=1; else J=J+1;
run;
proc sort data=project.monthlyquote;
    by symbol date time descending J;
run;

/* 'N' is the total number of quotes within a given second */
data project.monthlyquote;
    set project.monthlyquote;
    retain N; by symbol date time;
    if first.symbol or first.date or first.time then N=J; else N=N+0;
run;
proc sort data=project.monthlyquote;
    by symbol date time J;
run;
data project.monthlyquote (drop=J N);
    set project.monthlyquote;
    InterpolatedTime=time+((2*J-1)/(2*N));
    format InterpolatedTime best15.;
run;

/* Create Interpolated Trade Time for trade dataset */
proc sort data=project.monthlytrade;
    by symbol date time;
run;

/* 'I' indexes the order of trades within a given second */
data project.monthlytrade;
    set project.monthlytrade;
    retain I;by symbol date time;
    if first.symbol or first.date or first.time then I=1; else I=I+1;
run;
proc sort data=project.monthlytrade;
    by symbol date time descending I;
run;
```

```
/* 'N' is the total number of trades within a given second */
data project.monthlytrade;
    set project.monthlytrade;
    retain N; by symbol date time;
    if first.symbol or first.date or first.time then N=I;else N=N+0;
run;
proc sort data=project.monthlytrade;
    by symbol date time I;
run;
data project.monthlytrade (drop=I N);
    set project.monthlytrade;
    InterpolatedTime=time+((2*I-1)/(2*N));
    tradetime=time;
    format InterpolatedTime best15.;
    format tradetime time.;
run;

/* STEP FOUR: NATIONAL BEST BID AND OFFER (NBBO) CALCULATION */
/* Assign ID to Each Unique Exchange or Market Maker and Find Maximum Number of Exchanges*/
proc sort data=project.monthlyquote;
    by ex mmid;
run;

data project.monthlyquote;
    set project.monthlyquote;
    retain ExchangeID;
    if _N_=1 then ExchangeID=0;
    if first.ex or first.mmid then ExchangeID=ExchangeID+1;
    by ex mmid;
run;

data _null_;
    set project.monthlyquote end=eof;
    retain MaxExchangeID;
    if ExchangeID gt MaxExchangeID then MaxExchangeID=ExchangeID;
    if eof then call symput('MaxExchangeID',MaxExchangeID);
run;
%put &MaxExchangeID;
proc sort data=project.monthlyquote;
    by symbol date time InterpolatedTime;
run;

%macro BBO; /* Create Dataset that has a Column for Each Exchange ID's Bid and Offer Quote
               for All Interpolated Times*/
data project.monthlyquote;set project.monthlyquote;
    by symbol date;
    array exbid(&MaxExchangeID);exbid(ExchangeID)=bid;
    array exofr(&MaxExchangeID);exofr(ExchangeID)=ofr;
    array exbidsz(&MaxExchangeID);exbidsz(ExchangeID)=bidsiz;
    array exofrsz(&MaxExchangeID);exofrsz(ExchangeID)=ofrsiz;
/* For Interpolated Times with No Quote Update, Retain Previous Quote Outstanding*/
%do i=1 %to &MaxExchangeID;
        retain exbid2&i exofr2&i exbidsz2&i exofrsz2&i;
        if first.symbol or first.date then exbid2&i=exbid&i;
            if exbid&i ge 0 then exbid2&i=exbid&i; else exbid2&i=exbid2&i+0;
        if first.symbol or first.date then exofr2&i=exofr&i;
            if exofr&i ge 0 then exofr2&i=exofr&i; else exofr2&i=exofr2&i+0;
        if first.symbol or first.date then exbidsz2&i=exbidsz&i;
            if exbidsz&i ge 0 then exbidsz2&i=exbidsz&i; else exbidsz2&i=exbidsz2&i+0;
        if first.symbol or first.date then exofrsz2&i=exofrsz&i;
            if exofrsz&i ge 0 then exofrsz2&i=exofrsz&i; else exofrsz2&i=exofrsz2&i+0;
%end;
```

```
/* Find Best Bid and Offer Across All Exchanges and Market Makers*/
%do i=&MaxExchangeID %to &MaxExchangeID;
    BestBid = max(of exbid21-exbid2&i);
    BestOfr = min(of exofr21-exofr2&i);
%end;
/* Find Best and Total Depth Across All Exchanges and Market Makers that are at the NBBO*/
%do i=1 %to &MaxExchangeID;
    if exbid2&i=BestBid then MaxBidDepth=max(MaxBidDepth,exbidsz2&i);
    if exofr2&i=BestOfr then MaxOfrDepth=max(MaxOfrDepth,exofrsz2&i);
    if exbid2&i=BestBid then TotalBidDepth=sum(TotalBidDepth,exbidsz2&i);
    if exofr2&i=BestOfr then TotalOfrDepth=sum(TotalOfrDepth,exofrsz2&i);
%end;
run;
%mend BBO;
%BBO;

/* Only Output Changes in NBBO Records (e.g., changes in quotes or depth)*/
data FinalNBBO (keep=symbol date time InterpolatedTime
    BestBid BestOfr MaxBidDepth MaxOfrDepth TotalBidDepth TotalOfrDepth);
    set project.monthlyquote;
    if symbol eq lag(symbol)
        and date eq lag(date)
        and BestOfr eq lag(BestOfr)
        and BestBid eq lag(BestBid)
        and MaxOfrDepth eq lag(MaxOfrDepth)
        and MaxBidDepth eq lag(MaxBidDepth)
        and TotalOfrDepth eq lag(TotalOfrDepth)
        and TotalBidDepth eq lag(TotalBidDepth)
    then delete;
run;

/* If Abnormal Quotes Enter the NBBO Then Set To ".". There Will Be NO NBBO */
data FinalNBBO;
    set FinalNBBO;
    if BestBid < .00001 then do;
    BestBid=.;
    BestOfr=.;
    MaxOfrDepth=.;
    MaxBidDepth=.;
    TotalOfrDepth=.;
    TotalBidDepth=.;
    end;
else if BestOfr > 9999998 then do;
    BestBid=.;
    BestOfr=.;
    MaxOfrDepth=.;
    MaxBidDepth=.;
    TotalOfrDepth=.;
    TotalBidDepth=.;
    end;
run;

/* STEP FIVE: INTERWEAVE TRADES WITH QUOTES: TRADES AT INTERPOLATED TIME TMMM
   ARE MATCHED WITH QUOTES IN FORCE AT INTERPOLATED TIME TMM(M-1)
   To Do This, Increase Interpolated Quote Time in Quotes Dataset by One Millisecond = .001*/
data FinalNBBO;
    set FinalNBBO;
    type='Q';
    InterpolatedTime+.001;
run;

/* Sort and Stack Quotes and Trades Datasets */
proc sort data=FinalNBBO;
```

```
    by symbol date InterpolatedTime;
run;
proc sort data=project.monthlytrade;
    by symbol date InterpolatedTime;
run;
data Tradesquotes;
    set project.monthlytrade FinalNBBO;
    by symbol date InterpolatedTime;
run;

/* Sort Stacked Dataset by "Type" So that Quotes (Type=Q) Proceed Trades (Type=T) */
proc sort data=Tradesquotes;
    by symbol date InterpolatedTime type;
run;

/* For Each Trade, Identify the Outstanding NBBO, Best Depth and Total Depth */
data Tradesquotesv2 (drop=time BestOfr BestBid MaxOfrDepth MaxBidDepth
                          TotalOfrDepth TotalBidDepth corr);
    set Tradesquotes;
    by symbol date;
    retain quotetime BestOfr2 BestBid2 MaxOfrDepth2 MaxBidDepth2 TotalOfrDepth2 TotalBidDepth2;
    if type='Q' then quotetime=time;else quotetime=quotetime;
    if type='Q' then BestOfr2=BestOfr;else BestOfr2=BestOfr2;
    if type='Q' then BestBid2=BestBid;else BestBid2=BestBid2;
    if type='Q' then MaxOfrDepth2=MaxOfrDepth;else MaxOfrDepth2=MaxOfrDepth2;
    if type='Q' then MaxBidDepth2=MaxBidDepth;else MaxBidDepth2=MaxBidDepth2;
    if type='Q' then TotalOfrDepth2=TotalOfrDepth; else TotalOfrDepth2=TotalOfrDepth2;
    if type='Q' then TotalBidDepth2=TotalBidDepth; else TotalBidDepth2=TotalBidDepth2;
    format quotetime time.;
run;
data tradesquotes;
    set _null_;
run;

/* STEP SIX: Classify Trades as "Buys" or "Sells" Using Lee & Ready (LR);
   Determine NBBO Midpoint and Locked and Crossed NBBOs */

data TSpread1;
    set Tradesquotesv2;
    where type='T'; midpoint=(BestOfr2+BestBid2)/2;
    if BestOfr2=BestBid2 then lock=1;else lock=0;
    if BestOfr2<BestBid2 then cross=1;else cross=0;
run;

/* Determine Whether Trade Price is Higher or Lower than Previous Trade Price,
   or "Trade Direction" */
data TSpread1;
    set TSpread1;
    by symbol date; direction=dif(price);
    if first.symbol then direction=.;
    if first.date then direction=.;
run;
data TSpread1;
    set TSpread1;
    retain direction2;
    if direction ne 0 then direction2=direction;
    else direction2=direction2;
run;

/* First Classification Step: Classify Trades Using Tick Test */
data TSpread1 (drop=direction);
    set TSpread1;
    if direction2>0 then tradeLR='buy';
    if direction2<0 then tradeLR='sell';
    if direction2=. then tradeLR=.;
run;

/* Second Classification Step: Update Trade Classification
   When Conditions are Met as Specified by LR */
data TSpread1;
    set TSpread1;
    if lock=0 and cross=0 and price gt midpoint then tradeLR='buy';
    if lock=0 and cross=0 and price lt midpoint then tradeLR='sell';
run;

/* Question 5: COMPUTE FOUR LIQUIDITY MEASURES FOR EACH TRANSACTION */;
data QSpread1;
    set FinalNBBO;
    if time lt ("9:30:00"t) then delete;
run;

/* Determine Time Each Quote is In Force Based on Interpolated Time */
proc sort data=QSpread1;
    by symbol date descending InterpolatedTime;
run;
data QSpread1;
    set QSpread1;
    by symbol date;
    inforce=abs(dif(InterpolatedTime));
run;

/* If Last Quote of Day, then Quote is Inforce Until 4:00 pm */
data QSpread1;
    set QSpread1;
    by symbol date;
    end=57600; /* 4:00 pm = 57,600 seconds after midnight */
    if first.symbol or first.date then inforce=max((end-InterpolatedTime),0);
run;
proc sort data=QSpread1 (drop=end);
    by symbol date InterpolatedTime;
run;

/* Find Midpoint */
data QSpread1;
    set QSpread1;
    midpoint=(BestOfr+BestBid)/2;
run;

/* Question 5.1 and 5.2: RELATIVE QUOTED SPREAD AND RELATIVE EFFECTIVE SPREAD */
data TSpread1;
    set TSpread1;
    wQuotedSpread_Percent = (BestOfr2-BestBid2)/midpoint;
    wEffectiveSpread_Dollar=(abs(price-midpoint))*2;
    wEffectiveSpread_Percent=wEffectiveSpread_Dollar/midpoint;
run;

/* Delete Trades Associated with Locked or Crossed Best Bids or Best Offers */
data TSpread2;
    set TSpread1;
    if lock=1 or cross=1 then delete;
run;

/* Question 5.3 and 5.4: RELATIVE REALISED SPREAD AND FIVE-MINUTE PRICE IMPACT */
data MidQ_compare(keep=symbol date type midpointnew tradetime BestOfrnew BestBidnew);
    set QSpread1;
    midpointnew=midpoint;
    tradetime=time-300;
    format tradetime time.;
    BestOfrnew=BestOfr;
    BestBidnew=BestBid;
run;

data MidT (keep=symbol date tradetime type midpoint price tradeLR tradeEOH tradeCLNV
                wEffectiveSpread_Dollar size dollar);
    set TSpread1;
run;
proc sort data=MidQ;
    by symbol date tradetime type;
run;
proc sort data=MidT;
    by symbol date tradetime type;
run;

/* Stack Trades at Time T with NBBO Quotes at Time T+5 */
data Mid1;
    set MidT MidQ;
    by symbol date tradetime type;
run;

/* For Each Trade at Time T, Identify the Outstanding NBBO at Time T+5 */
data Mid1;
    set Mid1;
    by symbol date;
    retain midpoint5 BestOfr5 BestBid5;
    if type='Q' then midpoint5=midpointnew; else midpoint5=midpoint5;
    if type='Q' then BestOfr5=BestOfrnew;else BestOfr5=BestOfr5;
    if type='Q' then BestBid5=BestBidnew;else BestBid5=BestBid5;
run;

/* Delete Trades at T Associated with Locked or Crossed Best Bids or Best Offers at T+5 */
data Spread_Each (drop=midpointnew BestOfrnew BestBidnew);
    set Mid1;
    if BestOfr5=BestBid5 or BestOfr5<BestBid5 then delete;
run;

/* Create Indicator Variable "D" equal to "1" if Trade is a Buy
   and "-1" if Trade is a Sell for LR, EOH, and CLNV */
data Spread_Each;
    set Spread_Each;
    where type='T';
    if tradeLR='buy' then Dlr=1;
    if tradeLR='sel' then Dlr=-1;

    /* Compute Dollar and Percent Realized Spread for LR, EOH, and CLNV */
    wDollarRealizedSpread_LR=Dlr*(price-midpoint5)*2;
    wPercentRealizedSpread_LR=wDollarRealizedSpread_LR/midpoint5;

    /* Compute Dollar and Percent Price Impact for LR, EOH, and CLNV */
    wDollarPriceImpact_LR=wEffectiveSpread_Dollar-wDollarRealizedSpread_LR;
    wPercentPriceImpact_LR=wDollarPriceImpact_LR/midpoint5;
run;

/* Question 6: COMPUTE DAILY AND HOURLY AVERAGES OF SPREAD MEASURES
               & CORRELATIONS OF SPREAD MEASURES WITH AMIHUDE  */
Data Spread_Each;
    set Spread_Each;
    hours = hour(tradetime);
    if tradetime gt ("15:59:59"t) then delete;
run;
```

```
Data Tspread2;
    set Tspread2;
    hours = hour(tradetime);
    if tradetime gt ("15:59:59"t) then delete;
run;

/* Question 6.1: daily averages  */

*daily ave of relative quoted and effective spread;
proc sql;
    create table QS_ES_Daily
    as select symbol,date,
    sum(size) as size_daily,
    mean(wQuotedSpread_Percent) as PercentQuotedSpread_Daily,
    mean(wEffectiveSpread_Percent) as PercentEffectiveSpread_Daily
    from TSpread2
    group by symbol,date
    order by symbol,date;
quit;

*daily ave of relative realised spread and five-minute price impact;
proc sql;
    create table RS_PI_Daily
    as select symbol,date,
    sum(size) as size_daily,
    mean(wPercentRealizedSpread_LR) as PercentRealizedSpread_Daily,
    mean(wPercentPriceImpact_LR) as PercentPriceImpact_Daily
    from Spread_Each
    group by symbol,date
    order by symbol,date;
quit;

/* Question 6.2: hourly averages  */

*hourly ave of relative quoted and effective spread;
proc sql;
    create table QS_ES_Hourly
    as select symbol,date,hours,
    sum(size) as size_hourly,
    mean(wQuotedSpread_Percent) as PercentQuotedSpread_Hourly,
    mean(wEffectiveSpread_Percent) as PercentEffectiveSpread_Hourly
    from TSpread2
    group by symbol,date,hours
    order by symbol,date,hours;
quit;

*hourly ave of relative realised spread and five-minute price impact;
proc sql;
    create table RS_PI_Hourly
    as select symbol,date,hours,
    sum(size) as size_hourly,
    mean(wPercentRealizedSpread_LR) as PercentRealizedSpread_Hourly,
    mean(wPercentPriceImpact_LR) as PercentPriceImpact_Hourly
    from Spread_Each
    group by symbol,date,hours
    order by symbol,date,hours;
quit;

data Spread_Each;
    set Spread_Each;
    lagmidp = lag(midpoint);
    ret     = log(midpoint) - log(lagmidp);
```

```
run;

/* Question 6.3: Amihud Illiquidity Ratio  */

/* Preliminary: Data Manipulation*/

/* Data MSC */
data data_msc;
    set Spread_each;
    keep symbol date price midpoint size tradetime tradeLR hours mins;
    where symbol = 'MSC';
run;
data data_msc;
    set data_msc;
    days   = day(date);
    lagmid = lag(midpoint);
    ret    = log(midpoint) - log(lagmid);
run;
proc sql;
    create table ret_h_msc as
    select symbol,date,days,hours,
    sum(ret) as ret_h
    from data_msc
    group by symbol,date,days,hours
    order by symbol,date,days,hours;
quit;
proc sql;
    create table ret_d_msc as
    select symbol,date,
    sum(ret) as ret_d
    from data_msc
    group by symbol,date
    order by symbol,date;
quit;

/* Data PG*/
data data_pg;
    set Spread_each;
    keep symbol date price midpoint size tradetime tradeLR hours mins;
    where symbol = 'PG';
run;
data data_pg;
    set data_pg;
    days   = day(date);
    lagmid = lag(midpoint);
    ret    = log(midpoint) - log(lagmid);
run;
proc sql;
    create table ret_h_pg as
    select symbol,date,days,hours,
    sum(ret) as ret_h
    from data_pg
    group by symbol,date,days,hours
    order by symbol,date,days,hours;
quit;
proc sql;
    create table ret_d_pg as
    select symbol,date,
    sum(ret) as ret_d
    from data_pg
    group by symbol,date
    order by symbol,date;
quit;
```

```
/* combine data of two companies*/
data data_combine;
    set data_msc data_pg;
run;
data ret_d;
    set ret_d_msc ret_d_pg;
run;
data ret_h;
    set ret_h_msc ret_h_pg;
run;

/* daily Amihud */

*Compute hourly return, volume and midpoint price ;
proc sql;
    create table Amihud_Daily
    as select symbol,date,
    sum(ret) as ret_daily,
    sum(size) as size_daily,
    mean(midpoint) as midp_daily
    from data_combine
    group by symbol,date
    order by symbol,date;
quit;

data Amihud_Daily;
    set Amihud_Daily;
    dollarvol = size_daily*midp_daily;
    Amihud_daily = abs(ret_daily)/dollarvol;
    if ret_daily = 0 then delete;
run;

/* hourly Amihud */

*Compute hourly return, volume and midpoint price ;
proc sql;
    create table Amihud_Hourly
    as select symbol,date,hours,
    sum(ret) as ret_hourly,
    sum(size) as size_hourly,
    mean(midpoint) as midp_hourly
    from data_combine
    group by symbol,date,hours
    order by symbol,date,hours;
quit;

data Amihud_Hourly;
    set Amihud_Hourly;
    dollarvol = size_hourly*midp_hourly;
    Amihud_hourly = abs(ret_hourly)/dollarvol;
    if ret_hourly = 0 then Amihud_hourly =.;
run;

/* Question 6.4: Correlation of 4 spread measures with Amihud  */

/* daily correlations */
data Corr_Daily;
    set Amihud_Daily;
    set QS_ES_Daily;
    set RS_PI_Daily;
    year = year(date);
run;
```

```
proc corer data = Corr_Daily outp=CorrOutpD;
    var  Amihud_daily PercentQuotedSpread_Daily
         PercentEffectiveSpread_Daily PercentRealizedSpread_Daily;
    by symbol year;
run;
* print the output;
title 'Output Data Set from PROC CORR Daily';
proc print data=CorrOutpD noobs;
run;

/* hourly correlations */
data Corr_Hourly;
    set Amihud_Hourly;
    set QS_ES_Hourly;
    set RS_PI_Hourly;
    year = year(date);
run;
proc corr data = Corr_Hourly outp=CorrOutpH;
    var  Amihud_Hourly PercentQuotedSpread_Hourly
         PercentEffectiveSpread_Hourly PercentRealizedSpread_Hourly;
    by symbol year;
run;
* print the output;
title 'Output Data Set from PROC CORR Hourly';
proc print data=CorrOutpH noobs;
run;

/*data Amihud_hourly;*/
/*  set Amihud_hourly;*/
/*  lagret_hourly = lag(ret_hourly);*/
/*  lagret_hourly2 = lag2(ret_hourly);*/
/*run;*/

/* Question 7: COMPUTE EFFECTIVE SPREADS FOR TRANSACTIONS OF DIFFERENT SIZES:
   0   - 500,
   500 - 1000,
   1000 and above. */

/* single out transactions of different sizes */
data ES_0to500;
    set TSpread1;
    where size >= 0 and size <500;
    drop wQuotedSpread_Percent wEffectiveSpread_Percent wEffectiveSpread_Dollar
         lagprice ret
         MaxOfrDepth2 MaxBidDepth2 TotalOfrDepth2 TotalBidDepth2;
run;
data ES_500to1000;
    set TSpread1;
    where size >= 500 and size <1000;
    drop wQuotedSpread_Percent wEffectiveSpread_Percent wEffectiveSpread_Dollar
         lagprice ret
         MaxOfrDepth2 MaxBidDepth2 TotalOfrDepth2 TotalBidDepth2;
run;
data ES_1000above;
    set TSpread1;
    where size >= 1000;
    drop wQuotedSpread_Percent wEffectiveSpread_Percent wEffectiveSpread_Dollar
         lagprice ret
         MaxOfrDepth2 MaxBidDepth2 TotalOfrDepth2 TotalBidDepth2;
run;

/* compute ES*/
data ES_0to500;
```

```
    set ES_0to500;
    wEffectiveSpread_Dollar  = (abs(price-midpoint))*2;
    wEffectiveSpread_Percent = wEffectiveSpread_Dollar/midpoint;
    hours = hour(tradetime);
run;
data ES_500to1000;
    set ES_500to1000;
    wEffectiveSpread_Dollar  = (abs(price-midpoint))*2;
    wEffectiveSpread_Percent = wEffectiveSpread_Dollar/midpoint;
    hours = hour(tradetime);
run;
data ES_1000above;
    set ES_1000above;
    wEffectiveSpread_Dollar  = (abs(price-midpoint))*2;
    wEffectiveSpread_Percent = wEffectiveSpread_Dollar/midpoint;
    hours = hour(tradetime);
run;

/* Delete Trades Associated with Locked or Crossed Best Bids or Best Offers */
data ES_0to500;
    set ES_0to500;
    if lock=1 or cross=1 then delete;
run;
data ES_500to1000;
    set ES_500to1000;
    if lock=1 or cross=1 then delete;
run;
data ES_1000above;
    set ES_1000above;
    if lock=1 or cross=1 then delete;
run;

* daily ave;
proc sql;
    create table ES_0to500_Daily
    as select symbol,date,
    sum(size) as size_daily,
    mean(wEffectiveSpread_Percent) as Percent_ES_0to500_Daily
    from ES_0to500
    group by symbol,date
    order by symbol,date;
quit;
proc sql;
    create table ES_500to1000_Daily
    as select symbol,date,
    sum(size) as size_daily,
    mean(wEffectiveSpread_Percent) as Percent_ES_500to1000_Daily
    from ES_500to1000
    group by symbol,date
    order by symbol,date;
quit;
proc sql;
    create table ES_1000above_Daily
    as select symbol,date,
    sum(size) as size_daily,
    mean(wEffectiveSpread_Percent) as Percent_ES_1000above_Daily
    from ES_1000above
    group by symbol,date
    order by symbol,date;
quit;

/* combine */
data ES_Diff_Sizes_Daily;
```

```
    set ES_0to500_Daily;
    set ES_500to1000_Daily;
    set ES_1000above_Daily;
    drop size_daily;
run;

* hourly ave;
proc sql;
    create table ES_0to500_Hourly
    as select symbol,date,hours,
    sum(size) as size_Hourly,
    mean(wEffectiveSpread_Percent) as Percent_ES_0to500_Hourly
    from ES_0to500
    group by symbol,date,hours
    order by symbol,date,hours;
quit;
proc sql;
    create table ES_500to1000_Hourly
    as select symbol,date,hours,
    sum(size) as size_Hourly,
    mean(wEffectiveSpread_Percent) as Percent_ES_500to1000_Hourly
    from ES_500to1000
    group by symbol,date,hours
    order by symbol,date,hours;
quit;
proc sql;
    create table ES_1000above_Hourly
    as select symbol,date,hours,
    sum(size) as size_Hourly,
    mean(wEffectiveSpread_Percent) as Percent_ES_1000above_Hourly
    from ES_1000above
    group by symbol,date,hours
    order by symbol,date,hours;
quit;

/* combine */
data ES_Diff_Sizes_Hourly;
    set ES_0to500_Hourly;
    set ES_500to1000_Hourly;
    set ES_1000above_Hourly;
    drop size_Hourly;
run;

/* Question 8: COMPUTE AND COMPARE VWAP,VWAP_BUY,VWAP_SELL FOR THE WHOLE DAY */
data VWAP_prelim;
    set TSpread2;
    keep symbol date price size tradeLR;
run;

/* VWAP Overall */
proc sql;
    create table VWAP
    as select *,
    (select sum(size) from VWAP_prelim as sub where sub.symbol = main.symbol
     and sub.date = main.date)
    as vol_daily from VWAP_prelim as main;
quit;

data VWAP;
    set VWAP;
    weight         = size/vol_daily;
    weighted_price = weight*price;
run;
```

```
proc sql;
    create table VWAP_daily as
    select symbol,date,
    sum(weighted_price) as VWAP_daily,
    mean(price) as EWAP_daily
    from VWAP
    group by symbol,date
    order by symbol,date;
quit;

/* VWAP Buy */
data VWAP_buy_prelim;
    set VWAP_prelim;
    where tradeLR = 'buy';
run;

proc sql;
    create table VWAP_buy
    as select *,
    (select sum(size) from VWAP_buy_prelim as sub where sub.symbol = main.symbol
     and sub.date = main.date)
    as vol_buy_daily from VWAP_buy_prelim as main;
quit;

data VWAP_buy;
    set VWAP_buy;
    weight         = size/vol_buy_daily;
    weighted_price = weight*price;
run;

proc sql;
    create table VWAP_buy_daily as
    select symbol,date,
    sum(weighted_price) as VWAP_buy_daily
    from VWAP_buy
    group by symbol,date
    order by symbol,date;
quit;

/* VWAP Sell */
data VWAP_sell_prelim;
    set VWAP_prelim;
    where tradeLR = 'sel';
run;

proc sql;
    create table VWAP_sell
    as select *,
    (select sum(size) from VWAP_sell_prelim as sub where sub.symbol = main.symbol
     and sub.date = main.date)
    as vol_sell_daily from VWAP_sell_prelim as main;
quit;

data VWAP_sell;
    set VWAP_sell;
    weight         = size/vol_sell_daily;
    weighted_price = weight*price;
run;

proc sql;
    create table VWAP_sell_daily as
    select symbol,date,
    sum(weighted_price) as VWAP_sell_daily
    from VWAP_sell
    group by symbol,date
    order by symbol,date;
quit;

/* Question 9: TESTING PRICE EFFICIENCY  */

/* 9.1 acf and pacf plot */

/* Export to Excel file*/
proc export data = Ret_h (keep = symbol date ret_h)
    outfile = 'C:\Users\phd15ds\Downloads\SAS_Database\MMS.xlsx'
    dbms    = xlsx
    replace;
    sheet   = 'Ret_ACF';
run;

/* 9.2 Variance Ratio */

/* Export to Excel file: daily returns*/
proc export data = Ret_d
    outfile = 'C:\Users\phd15ds\Downloads\SAS_Database\MMS.xlsx'
    dbms    = xlsx
    replace;
    sheet   = 'DailyRet';
run;

/* Export to Excel file: hourly returns*/
proc export data = Ret_h
    outfile = 'C:\Users\phd15ds\Downloads\SAS_Database\MMS.xlsx'
    dbms    = xlsx
    replace;
    sheet   = 'HourlyRet';
run;

/* 9.3 Regression, Rosch et al 2013 */

/* MSC: One-hour Return (computed from previous steps) */
data Ret_h_msc;
    set Ret_h_msc;
    years = year(date);
run;
/* Export to Excel */
proc export data = Ret_h_msc
    outfile = 'C:\Users\phd15ds\Downloads\SAS_Database\MMS.xlsx'
    dbms    = xlsx
    replace;
    sheet   = 'RetHourlyMSC';
run;

/* PG: One-minute Return */
proc sql;
    create table Ret_m_PG as
    select symbol,date,days,hours,mins,
    sum(ret) as ret_1min
    from Data_combine
    where symbol = 'PG'
    group by symbol,date,days,hours,mins
    order by date,days,hours,mins;
quit;
data Ret_m_PG;
    set Ret_m_PG;
    years = year(date);
run;
/* Export to Excel */
proc export data = Ret_m_PG
    outfile = 'C:\Users\phd15ds\Downloads\SAS_Database\MMS.xlsx'
    dbms    = xlsx
    replace;
    sheet   = '1MinRetPG';
run;

/* Order Imbalance */
data OIB_prelim;
    set Data_combine;
    mins = minute(tradetime);
    drop MaxOfrDepth2 MaxBidDepth2 TotalOfrDepth2 TotalBidDepth2
         lock cross
         wQuotedSpread_Percent wEffectiveSpread_Dollar wEffectiveSpread_Percent;
run;
data OIB_prelim;
    set OIB_prelim;
    days       = day(date);
    if tradeLR = 'buy' then dollar_oib  = size*price;
    if tradeLR = 'sel' then dollar_oib = -1*size*price;
run;

/* One-hour OIB for MSC */
proc sql;
    create table OIB_hourly_MSC as
    select symbol,date,days,hours,
    sum(dollar_oib) as oib
    from OIB_prelim
    where symbol = 'MSC'
    group by symbol,date,days,hours
    order by symbol,date,days,hours;
quit;
data OIB_hourly_MSC;
    set OIB_hourly_MSC;
    years = year(date);
run;
/* Export to Excel */
proc export data = OIB_hourly_MSC
    outfile = 'C:\Users\phd15ds\Downloads\SAS_Database\MMS.xlsx'
    dbms    = xlsx
    replace;
    sheet   = 'OIBhourlyMSC';
run;

/* One-minute OIB for PG */
proc sql;
    create table OIB_1min_PG as
    select symbol,date,days,hours,mins,
    sum(dollar_oib) as oib
    from OIB_prelim
    where symbol = 'PG'
    group by symbol,date,days,hours,mins
    order by symbol,date,days,hours,mins;
quit;
data OIB_1min_PG;
    set OIB_1min_PG;
    years = year(date);
run;
/* Export to Excel */
proc export data = OIB_1min_PG
```

```
    outfile = 'C:\Users\phd15ds\Downloads\SAS_Database\MMS.xlsx'
    dbms    = xlsx
    replace;
    sheet   = 'OIB1minPG';
run;


/* ADDITIONAL: Codes for Plotting Purpose */

/* Figure 1: Intraday Dynamics of Hourly Averages of QS, ES and RS */

data Qs_es_rs_hourly_msc;
    set Qs_es_hourly(where=(symbol='MSC') drop = size_hourly days);
    set Rs_pi_hourly(where=(symbol='MSC') drop = PercentPriceImpact_Hourly
                                                 size_hourly days);
    years = year(date);
    days  = day(date);
    hours = hours * 3600;
    format hours hhmm.;
    DateTime=dhms(date,0,0,hours);
    format DateTime datetime13.;
    PercentQS_Hourly = PercentQuotedSpread_Hourly;
    PercentES_Hourly = PercentEffectiveSpread_Hourly;
    PercentRS_Hourly = PercentRealizedSpread_Hourly;
    drop PercentQuotedSpread_Hourly PercentEffectiveSpread_Hourly
         PercentRealizedSpread_Hourly;
run;
data Qs_es_rs_hourly_pg;
    set Qs_es_hourly(where=(symbol='PG') drop = size_hourly days);
    set Rs_pi_hourly(where=(symbol='PG') drop = PercentPriceImpact_Hourly size_hourly days);
    years = year(date);
    days  = day(date);
    PercentQS_Hourly = PercentQuotedSpread_Hourly;
    PercentES_Hourly = PercentEffectiveSpread_Hourly;
    PercentRS_Hourly = PercentRealizedSpread_Hourly;
    drop PercentQuotedSpread_Hourly PercentEffectiveSpread_Hourly
         PercentRealizedSpread_Hourly;
run;

/* Export to Excel*/
proc export data = Qs_es_rs_hourly_msc
    outfile = 'C:\Users\phd15ds\Downloads\SAS_Database\MMS_Plots.xlsx'
    dbms    = xlsx
    replace;
    sheet   = 'F1_QsEsRS_H_MSC';
run;
proc export data = Qs_es_rs_hourly_pg
    outfile = 'C:\Users\phd15ds\Downloads\SAS_Database\MMS_Plots.xlsx'
    dbms    = xlsx
    replace;
    sheet   = 'F1_QsEsRS_H_PG';
run;

/* Figure 2: Hourly Averages of ES with Different Sizes: 0-500, 500-1000 and 1000+ */

data Es_diffsizes_hourly_msc;
    merge Es_0to500_hourly(where=(symbol='MSC') drop = size_hourly)
        Es_500to1000_hourly(where=(symbol='MSC') drop = size_hourly)
        Es_1000above_hourly(where=(symbol='MSC') drop = size_hourly);
    by date hours;
    years = year(date);
    days  = day(date);
run;
```

```
data Es_diffsizes_hourly_pg;
    merge Es_0to500_hourly(where=(symbol='PG') drop = size_hourly)
        Es_500to1000_hourly(where=(symbol='PG') drop = size_hourly)
        Es_1000above_hourly(where=(symbol='PG') drop = size_hourly);
    by date hours;
    years = year(date);
    days  = day(date);
run;

/* Export to Excel*/
proc export data = Es_diffsizes_hourly_msc
    outfile = 'C:\Users\phd15ds\Downloads\SAS_Database\MMS_Plots.xlsx'
    dbms    = xlsx
    replace;
    sheet   = 'Es_DiffSizes_H_MSC';
run;
proc export data = Es_diffsizes_hourly_pg
    outfile = 'C:\Users\phd15ds\Downloads\SAS_Database\MMS_Plots.xlsx'
    dbms    = xlsx
    replace;
    sheet   = 'Es_DiffSizes_H_PG';
run;

/* Figure 3: Daily VWAP, VMAP_buy and VMAP_sell */
data VWAP_MSC;
    set VWAP_daily(where = (symbol = 'MSC') drop = EWAP_daily);
    set VWAP_buy_daily (where = (symbol = 'MSC'));
    set VWAP_sell_daily (where = (symbol = 'MSC'));
    years = year(date);
run;
data VWAP_PG;
    set VWAP_daily(where = (symbol = 'PG') drop = EWAP_daily);
    set VWAP_buy_daily (where = (symbol = 'PG'));
    set VWAP_sell_daily (where = (symbol = 'PG'));
    years = year(date);
run;

/* Export to Excel*/
proc export data = VWAP_MSC
    outfile = 'C:\Users\phd15ds\Downloads\SAS_Database\MMS_Plots.xlsx'
    dbms    = xlsx
    replace;
    sheet   = 'VWAP_MSC';
run;
proc export data = VWAP_PG
    outfile = 'C:\Users\phd15ds\Downloads\SAS_Database\MMS_Plots.xlsx'
    dbms    = xlsx
    replace;
    sheet   = 'VWAP_PG';
run;

/* Table2: Daily Averages of QS, ES and RS */
data Qs_Es_Rs_Daily;
    set Qs_es_daily;
    set Rs_pi_daily (drop = PercentPriceImpact_Daily);
    drop size_daily ;
run;
proc export data = Qs_ES_RS_Daily
    outfile = 'C:\Users\phd15ds\Downloads\SAS_Database\MMS.xlsx'
    dbms    = xlsx
    replace;
    sheet   = 'QsEsRs_Daily';
run;
```

7.3 MATLAB Codes for Variance Ratio and Rosch et al (2013) Regression

```
%% Assignment for Advanced Topics in Finance - Market Microstructure
%  Module Code: IB9BL0;
%  Author: Dalong Sun (Warwick ID 1257635)

%% Question 9

%% Load Data
[ret_hourly,titles] = xlsread('MMS.xlsx','Ret_ACF');

%% Data Management
ret_cell        = num2cell(ret_hourly);
titles(2:end,2) = ret_cell;

%% 9.1 Return Autocorrelation

%% ACF for MSC
ret_MSC06 = ret_hourly(1:33);
ret_MSC07 = ret_hourly(34:68);
ret_MSC08 = ret_hourly(69:103);

subplot(3,2,1),autocorr(ret_MSC06),title('MSC 2006'),ylabel('Autocorr')
subplot(3,2,3),autocorr(ret_MSC07),title('MSC 2007'),ylabel('Autocorr')
subplot(3,2,5),autocorr(ret_MSC08),title('MSC 2008'),ylabel('Autocorr')

%  ACF for PG
ret_PG06 = ret_hourly(104:138);
ret_PG07 = ret_hourly(139:173);
ret_PG08 = ret_hourly(174:208);

subplot(3,2,2),autocorr(ret_PG06),title('PG 2006'),ylabel('Autocorr')
subplot(3,2,4),autocorr(ret_PG07),title('PG 2007'),ylabel('Autocorr')
subplot(3,2,6),autocorr(ret_PG08),title('PG 2008'),ylabel('Autocorr')

%% Save Plot
print -depsc F4.eps;

%% 9.2 Variance Ratio

%% Load Data
[data_d,titles_d] = xlsread('MMS.xlsx','DailyRet');
[data_h,titles_h] = xlsread('MMS.xlsx','HourlyRet');

%% Data Management
ret_cell          = num2cell(data_d);
titles_d(2:end,3) = ret_cell;

ret_cell          = num2cell(data_h(:,3));
titles_h(2:end,5) = ret_cell;

%% MSC
years    = {1:33;34:68;69:103}; % 3 years for MSC
days     = {6:10;5:9;3:7};
VR_Days  = zeros(5,1);
VR_Yrs_H = zeros(3,1);
%% Numerator
for i = 1:3
    data        = data_h(years{i},[3,1]);

    VR_Days(1)  = var(data(data(:,2)==days{i}(1),1));
    VR_Days(2)  = var(data(data(:,2)==days{i}(2),1));
    VR_Days(3)  = var(data(data(:,2)==days{i}(3),1));
    VR_Days(4)  = var(data(data(:,2)==days{i}(4),1));
    VR_Days(5)  = var(data(data(:,2)==days{i}(5),1));

    VR_Yrs_H(i) = mean(VR_Days);
end

%% Denominator
VR_Yrs_D     = zeros(3,1);
VR_Yrs_D(1)  = var(data_d(1:5));
VR_Yrs_D(2)  = var(data_d(6:10));
VR_Yrs_D(3)  = var(data_d(11:15));

%% VarRatio_MSC
VarRatio_MSC  = abs(1-6.5*(VR_Yrs_H./VR_Yrs_D));
VarRatio_MSC2 = 6.5*(VR_Yrs_H./VR_Yrs_D);

%% PG
years         = {104:138;139:173;174:208}; % 3 years for PG
days          = {6:10;5:9;3:7};
VR_Days  = zeros(5,1);
VR_Yrs_H = zeros(3,1);

%% Numerator
for i = 1:3
    data       = data_h(years{i},[3,1]);

    VR_Days(1) = var(data(data(:,2)==days{i}(1),1));
    VR_Days(2) = var(data(data(:,2)==days{i}(2),1));
    VR_Days(3) = var(data(data(:,2)==days{i}(3),1));
    VR_Days(4) = var(data(data(:,2)==days{i}(4),1));
    VR_Days(5) = var(data(data(:,2)==days{i}(5),1));

    VR_Yrs_H(i) = mean(VR_Days);
end

%% Denominator
VR_Yrs_D    = zeros(3,1);
VR_Yrs_D(1) = var(data_d(16:20));
VR_Yrs_D(2) = var(data_d(21:25));
VR_Yrs_D(3) = var(data_d(26:30));

%% VarRatio_PG
VarRatio_PG  = abs(1-6.5*(VR_Yrs_H./VR_Yrs_D));
VarRatio_PG2 = 6.5*(VR_Yrs_H./VR_Yrs_D);

%% Overall
VarRatio_Overall  = [VarRatio_MSC;VarRatio_PG];
VarRatio_Overall2 = [VarRatio_MSC2,VarRatio_PG2];

%%
VarRatio_Overall(:,2) = [2006:2008,2006:2008];

%% Write into Excel
xlswrite('MMS_Plots.xlsx',VarRatio_Overall,'VarRatio')
%% 9.3 Regression: Rosch et al (2013)

%% MSC: Midprice One-hour Returns on Lagged Order Imbalances

%% Load Data
[OIB_h_MSC,titles_OIB_h_MSC] = xlsread('MMS.xlsx','OIBhourlyMSC');
[Ret_MSC,titles_Ret_MSC]     = xlsread('MMS.xlsx','RetHourlyMSC');
```

```
%% OLG Regression
yrs  = [2006;2007;2008];
days = [6:10;5:9;3:7]';

R2_MSC   = zeros(5,3);

for y = 1:3
    for d = 1:5
        OIB_y = OIB_h_MSC(OIB_h_MSC(:,4)==yrs(y),:);
        Ret_y = Ret_MSC(Ret_MSC(:,4)==yrs(y),:);

        OIB = OIB_y(OIB_y(:,1)==days(d,y),3);
        Ret = Ret_y(Ret_y(:,1)==days(d,y),3);
        [~,~,~,~,R2_MSC(d,y)] = OLSReg(OIB(1:end-1),Ret(2:end),'Y');
    end
end

%% PG: Midprice One-minute Returns on Lagged Order Imbalances

%% Load Data
[OIB_m_PG,titles_OIB_m_PG] = xlsread('MMS.xlsx','OIB1minPG');
[Ret_PG,titles_Ret_PG]     = xlsread('MMS.xlsx','1MinRetPG');

%% OLS Regression
R2_PG   = zeros(5,3);

for y = 1:3
    for d = 1:5
        OIB_y = OIB_m_PG(OIB_m_PG(:,5)==yrs(y),:);
        Ret_y = Ret_PG(Ret_PG(:,5)==yrs(y),:);

        OIB = OIB_y(OIB_y(:,1)==days(d,y),4);
        Ret = Ret_y(Ret_y(:,1)==days(d,y),4);
        [~,~,~,~,R2_PG(d,y)] = OLSReg(OIB(1:end-1),Ret(2:end),'Y');
    end
end

%% Boxplot
subplot(2,2,[1,2]),boxplot([R2_MSC,R2_PG]),title('MSC vs PG')
subplot(2,2,3),boxplot(R2_MSC),title('MSC')
subplot(2,2,4),boxplot(R2_PG),title('PG')

%% Save Plot
print -depsc F6.eps;

%% Combine
R2_Overall          = [R2_MSC(:);R2_PG(:)];
R2_Overall([1:5,16:20],2)   = 2006;
R2_Overall([6:10,21:25],2)  = 2007;
R2_Overall([11:15,26:30],2) = 2008;

%% Write into Excel
xlswrite('MMS_Plots.xlsx',R2_Overall,'R2')
```

```
% Author: Dalong Sun (Warwick ID:1257635)
%
% This is a function for the standard OLS regression;
%
% Input: 1.'X'           = a matrix of all the independent variables;
%        2.'Y'           = a vector of dependent variable;
%        3.'ConstYorN'   = Y, including the constant term;
%                          N, excluding the constant term;
%
% Output: 1.'beta'       = coefficient estimates;
%         2.'se'         = standard errors;
%         3.'t_stats'    = t-ratio statistics;
%         4.'pval'       = t-ratio p-value;
%         5.'R2'         = R-squared;
%         6.'adjR2'      = adjusted R-squared;
%         7.'resid'      = regression residuals;
%         8.'RSS'        = residual sum of squares;
%         9.'se_reg'     = standard error of regression;

function [beta,se,t_stats,pval,R2,adjR2,resid,RSS,se_reg] ...
    = OLSReg(X,Y,ConstYorN)

if ConstYorN == 'Y'
    X    = [ones(length(X),1), X];
end

beta     = lscov(X,Y); % 1. *OLS regression*

[T,k]    = size(X);

Y_hat    = X*beta;
resid    = Y - Y_hat;
RSS      = sum(resid.^2); % residual sum of squares
Cov      = (RSS/(T-k))*inv(X'*X); % assume homoskedasticity
se_reg   = sqrt(RSS/(T-k)); % standard error of regression

se       = diag(sqrt(Cov)); % 2. *standard error of coeff. estimates*

t_stats = beta./se; % 3. *t-ratio statistics*

pval     = 2*(1 - tcdf(t_stats,T-k));

Y_bar    = mean(Y);
R2       = 1 - RSS/(((Y-Y_bar)')*(Y-Y_bar)); % 4. *R-squared*

adjR2    = 1 - (1-R2)*( (T-1)/(T-k) ); % 5. *adjusted R-squared*

end
```

7.4 R Codes for Graphical Analysis

```
### Assignment for Advanced Topics in Finance - Market Microstructure ###
### Module Code: IB9BL0                                                ###
### Author: Dalong Sun (Warwick ID 1257635)                            ###

# Install Additionial Packages from CRAN
install.packages("xlsx")
install.packages("ggplot2")
install.packages("gridExtra")

# Load Packages Every Time
library(xlsx)
library(ggplot2)
library(gridExtra)
library(scales)

# Set the Working Directory to:
# setwd("C:/Users/phd15ds/Downloads/WBS Doctoral Programme
# /1st Year/WBS Phd Finance Training Courses (Term 2)
# /Advanced Topics in Finance/Market Microstructure/Works In Progress/Written Assignment")
setwd(paste("C:/Users/phd15ds/Downloads/WBS Doctoral Programme",
            "/1st Year/WBS Phd Finance Training Courses (Term 2)",
            "/Advanced Topics in Finance/Market Microstructure/Works In Progress",
            "/Written Assignment", sep=""))


###### Figure 1: Intraday Dynamics of Hourly Averages of QS, ES and RS ###### ------------

F1Spr_MSC <- read.xlsx("MMS_Plots.xlsx", sheetName = "F1_QsEsRS_H_MSC", header = 1)
F1Spr_PG  <- read.xlsx("MMS_Plots.xlsx", sheetName = "F1_QsEsRS_H_PG", header = 1)

# F1Spr_MSC$DateTime <- strftime(F1Spr_MSC$DateTime,"%Y-%m-%d %H:%M")

### For MSC, illiquid stock ### ----------------------------------------

# qplot(DateTime, PercentQS_Hourly, data = F1Spr_MSC06) +
# theme(axis.text.x = element_text(angle = 90))

# Save Plot
# postscript("F1.eps") # this generates plots in 'Landscape' format
cairo_ps("F11.eps", height = 8.0) # this generates plots in 'Portrait' format

F1Spr_MSC$days <- as.factor(F1Spr_MSC$days) # KEY for plot
F1Spr_MSC06    <- F1Spr_MSC[F1Spr_MSC$years == 2006,]
p_MSC06        <- ggplot(F1Spr_MSC06, aes(1:nrow(F1Spr_MSC06))) +
  geom_line(aes(y = PercentQS_Hourly, colour = days, linetype = "QS")) +
  geom_line(aes(y = PercentES_Hourly, colour = days, linetype = "ES")) +
  geom_line(aes(y = PercentRS_Hourly, colour = days, linetype = "RS")) +
  labs(x = "time", y = "percentage") +
  ggtitle("MSC 2006") +
  theme(plot.title = element_text(face="bold"))

F1Spr_MSC07 <- F1Spr_MSC[F1Spr_MSC$years == 2007,]
p_MSC07     <- ggplot(F1Spr_MSC07, aes(1:nrow(F1Spr_MSC07))) +
  geom_line(aes(y = PercentQS_Hourly, colour = days, linetype = "QS")) +
  geom_line(aes(y = PercentES_Hourly, colour = days, linetype = "ES")) +
  geom_line(aes(y = PercentRS_Hourly, colour = days, linetype = "RS")) +
  labs(x = "time", y = "percentage") +
  ggtitle("MSC 2007") +
  theme(plot.title = element_text(face="bold"))
```

```
F1Spr_MSC08 <- F1Spr_MSC[F1Spr_MSC$years == 2008,]
p_MSC08     <- ggplot(F1Spr_MSC08, aes(1:nrow(F1Spr_MSC08))) +
  geom_line(aes(y = PercentQS_Hourly, colour = days, linetype = "QS")) +
  geom_line(aes(y = PercentES_Hourly, colour = days, linetype = "ES")) +
  geom_line(aes(y = PercentRS_Hourly, colour = days, linetype = "RS")) +
  labs(x = "time", y = "percentage") +
  ggtitle("MSC 2008") +
  theme(plot.title = element_text(face="bold"))

# Create a 3 by 1 Multiplots
grid.arrange(p_MSC06, p_MSC07, p_MSC08, nrow = 3, ncol = 1)

dev.off()   # turn off the postscript device

### For PG, liquid stock ### ------------------------------------------

# Save Plot
cairo_ps("F12.eps", height = 8.0) # this generates plots in 'Portrait' format

F1Spr_PG$days <- as.factor(F1Spr_PG$days) # KEY for plot
F1Spr_PG06    <- F1Spr_PG[F1Spr_PG$years == 2006,]
p_PG06        <- ggplot(F1Spr_PG06, aes(1:nrow(F1Spr_PG06))) +
  geom_line(aes(y = PercentQS_Hourly, colour = days, linetype = "QS")) +
  geom_line(aes(y = PercentES_Hourly, colour = days, linetype = "ES")) +
  geom_line(aes(y = PercentRS_Hourly, colour = days, linetype = "RS")) +
  labs(x = "time", y = "percentage") +
  ggtitle("PG 2006") +
  theme(plot.title = element_text(face="bold"))

F1Spr_PG07 <- F1Spr_PG[F1Spr_PG$years == 2007,]
p_PG07     <- ggplot(F1Spr_PG07, aes(1:nrow(F1Spr_PG07))) +
  geom_line(aes(y = PercentQS_Hourly, colour = days, linetype = "QS")) +
  geom_line(aes(y = PercentES_Hourly, colour = days, linetype = "ES")) +
  geom_line(aes(y = PercentRS_Hourly, colour = days, linetype = "RS")) +
  labs(x = "time", y = "percentage") +
  ggtitle("PG 2007") +
  theme(plot.title = element_text(face="bold"))

F1Spr_PG08 <- F1Spr_PG[F1Spr_PG$years == 2008,]
p_PG08     <- ggplot(F1Spr_PG08, aes(1:nrow(F1Spr_PG08))) +
  geom_line(aes(y = PercentQS_Hourly, colour = days, linetype = "QS")) +
  geom_line(aes(y = PercentES_Hourly, colour = days, linetype = "ES")) +
  geom_line(aes(y = PercentRS_Hourly, colour = days, linetype = "RS")) +
  labs(x = "time", y = "percentage") +
  ggtitle("PG 2008") +
  theme(plot.title = element_text(face="bold"))

# Create a 3 by 1 Multiplots
grid.arrange(p_PG06, p_PG07, p_PG08, nrow = 3, ncol = 1)

dev.off()   # turn off the postscript device



###### Figure 2: Hourly Averages of ES with Different Sizes: 0-500, 500-1000 and 1000+ ######

F2ES_MSC <- read.xlsx("MMS_Plots.xlsx", sheetName = "Es_DiffSizes_H_MSC", header = 1)
F2ES_PG  <- read.xlsx("MMS_Plots.xlsx", sheetName = "Es_DiffSizes_H_PG", header = 1)

### For MSC, illiquid stock ### ---------------------------------------------------------

# Save Plot
```

```
cairo_ps("F21.eps") # this generates plots in 'Portrait' format

F2ES_MSC$days <- as.factor(F2ES_MSC$days) # KEY for plot
F2ES_MSC06    <- F2ES_MSC[F2ES_MSC$years == 2006,]
p_MSC06       <- ggplot(F2ES_MSC06, aes(1:nrow(F2ES_MSC06))) +
  geom_line(aes(y = Percent_ES_0to500_Hourly, colour = days, linetype = "0-500")) +
  geom_line(aes(y = Percent_ES_500to1000_Hourly, colour = days, linetype = "500-1000")) +
  geom_point(aes(y = Percent_ES_500to1000_Hourly), shape = 1, size = 1.5) +
  geom_line(aes(y = Percent_ES_1000above_Hourly, colour = days, linetype = "1000+")) +
  geom_point(aes(y = Percent_ES_1000above_Hourly), shape = 16, size = 1.5) +
  labs(x = "time", y = "percentage") +
  ggtitle("MSC 2006") +
  theme(plot.title = element_text(face="bold")) +
  guides(color = FALSE) # remove the legend for the aesthetics 'color'

F2ES_MSC07 <- F2ES_MSC[F2ES_MSC$years == 2007,]
p_MSC07    <- ggplot(F2ES_MSC07, aes(1:nrow(F2ES_MSC07))) +
  geom_line(aes(y = Percent_ES_0to500_Hourly, colour = days, linetype = "0-500")) +
  geom_line(aes(y = Percent_ES_500to1000_Hourly, colour = days, linetype = "500-1000")) +
  geom_point(aes(y = Percent_ES_500to1000_Hourly), shape = 1, size = 1.5) +
  geom_line(aes(y = Percent_ES_1000above_Hourly, colour = days, linetype = "1000+")) +
  geom_point(aes(y = Percent_ES_1000above_Hourly), shape = 16, size = 1.5) +
  labs(x = "time", y = "percentage") +
  ggtitle("MSC 2007") +
  theme(plot.title = element_text(face="bold")) +
  guides(color = FALSE) # remove the legend for the aesthetics 'color'

F2ES_MSC08 <- F2ES_MSC[F2ES_MSC$years == 2008,]
p_MSC08    <- ggplot(F2ES_MSC08, aes(1:nrow(F2ES_MSC08))) +
  geom_line(aes(y = Percent_ES_0to500_Hourly, colour = days, linetype = "0-500")) +
  geom_line(aes(y = Percent_ES_500to1000_Hourly, colour = days, linetype = "500-1000")) +
  geom_point(aes(y = Percent_ES_500to1000_Hourly), shape = 1, size = 1.5) +
  geom_line(aes(y = Percent_ES_1000above_Hourly, colour = days, linetype = "1000+")) +
  geom_point(aes(y = Percent_ES_1000above_Hourly), shape = 16, size = 1.5) +
  labs(x = "time", y = "percentage") +
  ggtitle("MSC 2008") +
  theme(plot.title = element_text(face="bold")) +
  guides(color = FALSE) # remove the legend for the aesthetics 'color'

# Create a 3 by 1 Multiplots
grid.arrange(p_MSC06, p_MSC07, p_MSC08, nrow = 3, ncol = 1)

dev.off()   # turn off the postscript device

### For PG, liquid stock ### ----------------------------------------------------------

# Save Plot
cairo_ps("F22.eps") # this generates plots in 'Portrait' format

F2ES_PG$days <- as.factor(F2ES_PG$days) # KEY for plot
F2ES_PG06    <- F2ES_PG[F2ES_PG$years == 2006,]
p_PG06       <- ggplot(F2ES_PG06, aes(1:nrow(F2ES_PG06))) +
  geom_line(aes(y = Percent_ES_0to500_Hourly, colour = days, linetype = "0-500")) +
  geom_line(aes(y = Percent_ES_500to1000_Hourly, colour = days, linetype = "500-1000")) +
  geom_point(aes(y = Percent_ES_500to1000_Hourly), shape = 1, size = 1) +
  geom_line(aes(y = Percent_ES_1000above_Hourly, colour = days, linetype = "1000+")) +
  geom_point(aes(y = Percent_ES_1000above_Hourly), shape = 16, size = 1) +
  labs(x = "time", y = "percentage") +
  ggtitle("PG 2006") +
  theme(plot.title = element_text(face="bold")) +
  guides(color = FALSE) # remove the legend for the aesthetics 'color'

F2ES_PG07 <- F2ES_PG[F2ES_PG$years == 2007,]
```

```
p_PG07    <- ggplot(F2ES_PG07, aes(1:nrow(F2ES_PG07))) +
  geom_line(aes(y = Percent_ES_0to500_Hourly, colour = days, linetype = "0-500")) +
  geom_line(aes(y = Percent_ES_500to1000_Hourly, colour = days, linetype = "500-1000")) +
  geom_point(aes(y = Percent_ES_500to1000_Hourly), shape = 1, size = 1) +
  geom_line(aes(y = Percent_ES_1000above_Hourly, colour = days, linetype = "1000+")) +
  geom_point(aes(y = Percent_ES_1000above_Hourly), shape = 16, size = 1) +
  labs(x = "time", y = "percentage") +
  ggtitle("PG 2007") +
  theme(plot.title = element_text(face="bold")) +
  guides(color = FALSE) # remove the legend for the aesthetics 'color'

F2ES_PG08 <- F2ES_PG[F2ES_PG$years == 2008,]
p_PG08    <- ggplot(F2ES_PG08, aes(1:nrow(F2ES_PG08))) +
  geom_line(aes(y = Percent_ES_0to500_Hourly, colour = days, linetype = "0-500")) +
  geom_line(aes(y = Percent_ES_500to1000_Hourly, colour = days, linetype = "500-1000")) +
  geom_point(aes(y = Percent_ES_500to1000_Hourly), shape = 1, size = 1) +
  geom_line(aes(y = Percent_ES_1000above_Hourly, colour = days, linetype = "1000+")) +
  geom_point(aes(y = Percent_ES_1000above_Hourly), shape = 16, size = 1) +
  labs(x = "time", y = "percentage") +
  ggtitle("PG 2008") +
  theme(plot.title = element_text(face="bold")) +
  guides(color = FALSE) # remove the legend for the aesthetics 'color'

# Create a 3 by 1 Multiplots
grid.arrange(p_PG06, p_PG07, p_PG08, nrow = 3, ncol = 1)

dev.off()



###### Figure 3: Daily VWAP, VMAP_buy and VMAP_sell ######-------------------------------

F3VWAP_MSC <- read.xlsx("MMS_Plots.xlsx", sheetName = "VWAP_MSC", header = 1);
F3VWAP_PG  <- read.xlsx("MMS_Plots.xlsx", sheetName = "VWAP_PG", header = 1);

### For MSC, illiquid stock ### ---------------------------------------------------------

# Save Plot
cairo_ps("F31.eps", height = 8.0) # this generates plots in 'Portrait' format

F3VWAP_MSC$years <- as.factor(F3VWAP_MSC$years) # KEY for plot
F3VWAP_MSC06     <- F3VWAP_MSC[F3VWAP_MSC$years == 2006,]
date             <- as.Date(F3VWAP_MSC06$DATE)
p_MSC06          <- ggplot(F3VWAP_MSC06, aes(date)) +
  geom_line(aes(y = VWAP_daily, colour = "VWAP", linetype = "VWAP")) +
  geom_line(aes(y = VWAP_buy_daily, colour = "VWAP_Buy", linetype = "VWAP_Buy")) +
  geom_line(aes(y = VWAP_sell_daily, colour = "VWAP_Sell", linetype = "VWAP_Sell")) +
  labs(x = "time", y = "percentage") +
  ggtitle("MSC 2006") +
  theme(plot.title = element_text(face="bold"))

F3VWAP_MSC07     <- F3VWAP_MSC[F3VWAP_MSC$years == 2007,]
date             <- as.Date(F3VWAP_MSC07$DATE)
p_MSC07          <- ggplot(F3VWAP_MSC07, aes(date)) +
  geom_line(aes(y = VWAP_daily, colour = "VWAP", linetype = "VWAP")) +
  geom_line(aes(y = VWAP_buy_daily, colour = "VWAP_Buy", linetype = "VWAP_Buy")) +
  geom_line(aes(y = VWAP_sell_daily, colour = "VWAP_Sell", linetype = "VWAP_Sell")) +
  labs(x = "time", y = "percentage") +
  ggtitle("MSC 2007") +
  theme(plot.title = element_text(face="bold"))

F3VWAP_MSC08     <- F3VWAP_MSC[F3VWAP_MSC$years == 2008,]
date             <- as.Date(F3VWAP_MSC08$DATE)
```

```
p_MSC08          <- ggplot(F3VWAP_MSC08, aes(date)) +
  geom_line(aes(y = VWAP_daily, colour = "VWAP", linetype = "VWAP")) +
  geom_line(aes(y = VWAP_buy_daily, colour = "VWAP_Buy", linetype = "VWAP_Buy")) +
  geom_line(aes(y = VWAP_sell_daily, colour = "VWAP_Sell", linetype = "VWAP_Sell")) +
  labs(x = "time", y = "percentage") +
  ggtitle("MSC 2008") +
  theme(plot.title = element_text(face="bold"))

# Create a 3 by 1 Multiplots
grid.arrange(p_MSC06, p_MSC07, p_MSC08, nrow = 3, ncol = 1)

dev.off()   # turn off the postscript device

### For PG, liquid stock ### ----------------------------------------------------------

# Save Plot
cairo_ps("F32.eps", height = 8.0) # this generates plots in 'Portrait' format

F3VWAP_PG$years <- as.factor(F3VWAP_PG$years) # KEY for plot
F3VWAP_PG06     <- F3VWAP_PG[F3VWAP_PG$years == 2006,]
date            <- as.Date(F3VWAP_PG06$DATE)
p_PG06          <- ggplot(F3VWAP_PG06, aes(date)) +
  geom_line(aes(y = VWAP_daily, colour = "VWAP", linetype = "VWAP")) +
  geom_line(aes(y = VWAP_buy_daily, colour = "VWAP_Buy", linetype = "VWAP_Buy")) +
  geom_line(aes(y = VWAP_sell_daily, colour = "VWAP_Sell", linetype = "VWAP_Sell")) +
  labs(x = "time", y = "percentage") +
  ggtitle("PG 2006") +
  theme(plot.title = element_text(face="bold"))

F3VWAP_PG07     <- F3VWAP_PG[F3VWAP_PG$years == 2007,]
date            <- as.Date(F3VWAP_PG07$DATE)
p_PG07          <- ggplot(F3VWAP_PG07, aes(date)) +
  geom_line(aes(y = VWAP_daily, colour = "VWAP", linetype = "VWAP")) +
  geom_line(aes(y = VWAP_buy_daily, colour = "VWAP_Buy", linetype = "VWAP_Buy")) +
  geom_line(aes(y = VWAP_sell_daily, colour = "VWAP_Sell", linetype = "VWAP_Sell")) +
  labs(x = "time", y = "percentage") +
  ggtitle("PG 2007") +
  theme(plot.title = element_text(face="bold"))

F3VWAP_PG08     <- F3VWAP_PG[F3VWAP_PG$years == 2008,]
date            <- as.Date(F3VWAP_PG08$DATE)
p_PG08          <- ggplot(F3VWAP_PG08, aes(date)) +
  geom_line(aes(y = VWAP_daily, colour = "VWAP", linetype = "VWAP")) +
  geom_line(aes(y = VWAP_buy_daily, colour = "VWAP_Buy", linetype = "VWAP_Buy")) +
  geom_line(aes(y = VWAP_sell_daily, colour = "VWAP_Sell", linetype = "VWAP_Sell")) +
  labs(x = "time", y = "percentage") +
  ggtitle("PG 2008") +
  theme(plot.title = element_text(face="bold"))

# Create a 3 by 1 Multiplots
grid.arrange(p_PG06, p_PG07, p_PG08, nrow = 3, ncol = 1)

dev.off()   # turn off the postscript device



###### Figure 5: Variance Ratio ######-------------------------------------------------

F5VR       <- read.xlsx("MMS_Plots.xlsx", sheetName = "VarRatio", header = 1);

F5VR$years <- as.factor(F5VR$years) # This time is optional and trivial

# Save Plot
```

```
cairo_ps("F5.eps", height = 4.0) # this generates plots in 'Portrait' format

ggplot(data = F5VR, aes(x = years, y = VarRatio, fill = symbol)) +
  geom_bar(stat = "identity", position = "dodge")

dev.off()   # turn off the postscript device



###### Figure 6: R Squared from Rosch et al (2013) Regression ######-------------------

F6R2       <- read.xlsx("MMS_Plots.xlsx", sheetName = "R2", header = 1);

F6R2$years <- as.factor(F6R2$years)

p1         <- ggplot(data = F6R2, aes(x = years, y = R2)) + geom_boxplot(aes(fill = symbol))

F6R2_PG    <- F6R2[F6R2$symbol == "PG",]
p2         <- ggplot(data = F6R2_PG, aes(x = years, y = R2)) + geom_boxplot(fill = "cyan")

# Create a 2 by 1 Multiplots
grid.arrange(p1, p2, nrow = 1, ncol = 2)
```