Lecture 5

Tolerance Functions

Factor Models

VWAP using Factor Models

Lee-Ready

COLUMBIA UNIVERSITY
IN THE CITY OF NEW YORK

Administrative

- First Homework due Feb 29th
- Group Project Proposal to be assigned this week, due ~week 8 or 9.
  - 1 page or less, describe what you want to do, any background research, what market data you need, etc.
  - No rush, but start thinking about it.

Lecture 5 Agenda

1. Finish Tolerance Functions and Factors
2. WRDS Tick Data
3. VWAP algo With Tick Data
4. Lee Ready algo and redo example

Tolerance Functions and Factor Models

Back to Lecture 4…

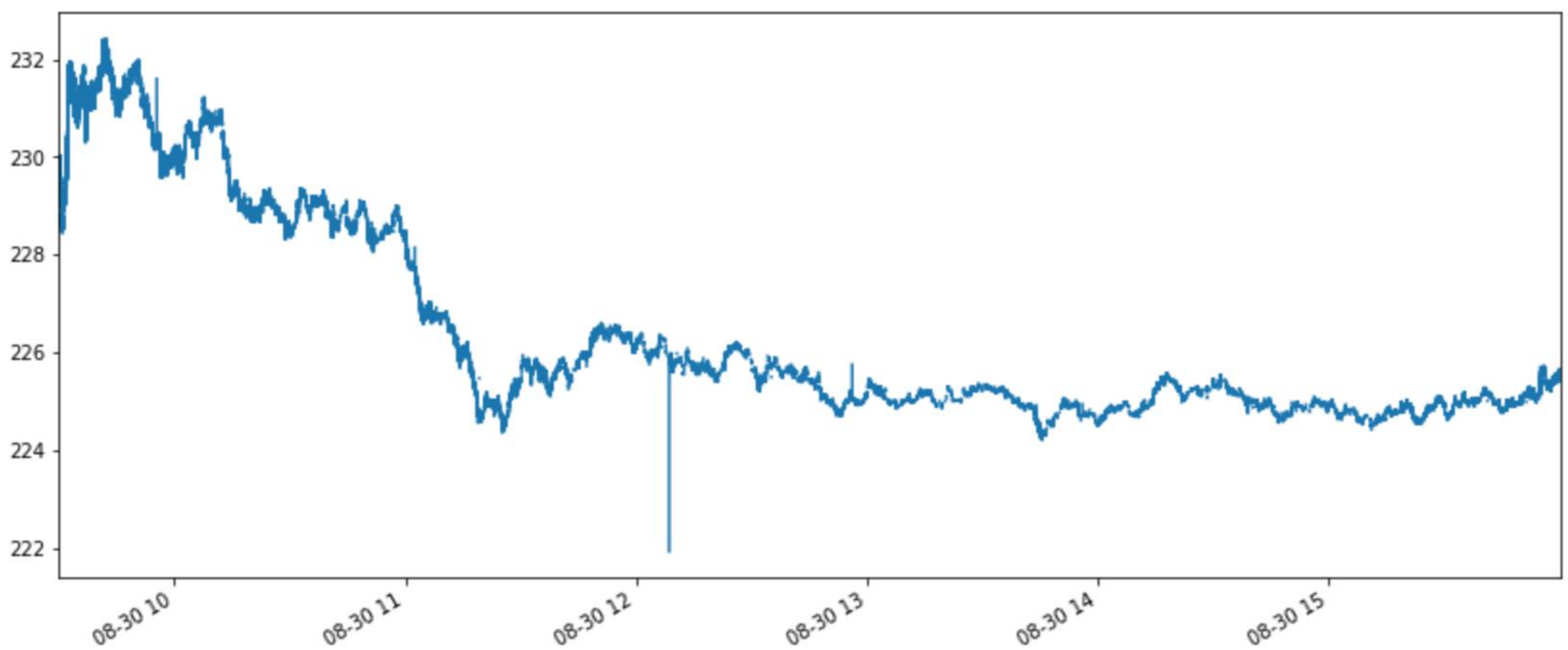
WRDS Tick Data

Principles of New Data

1. Always read the specification
2. This is ~~even~~ (especially!) true when you think you know what the data is.
3. Then, before you test your algo, sanity check your data.
   - Visually
   - Quantitatively ideally.
4. Previously I went to port some old algo code to use the WRDS data….

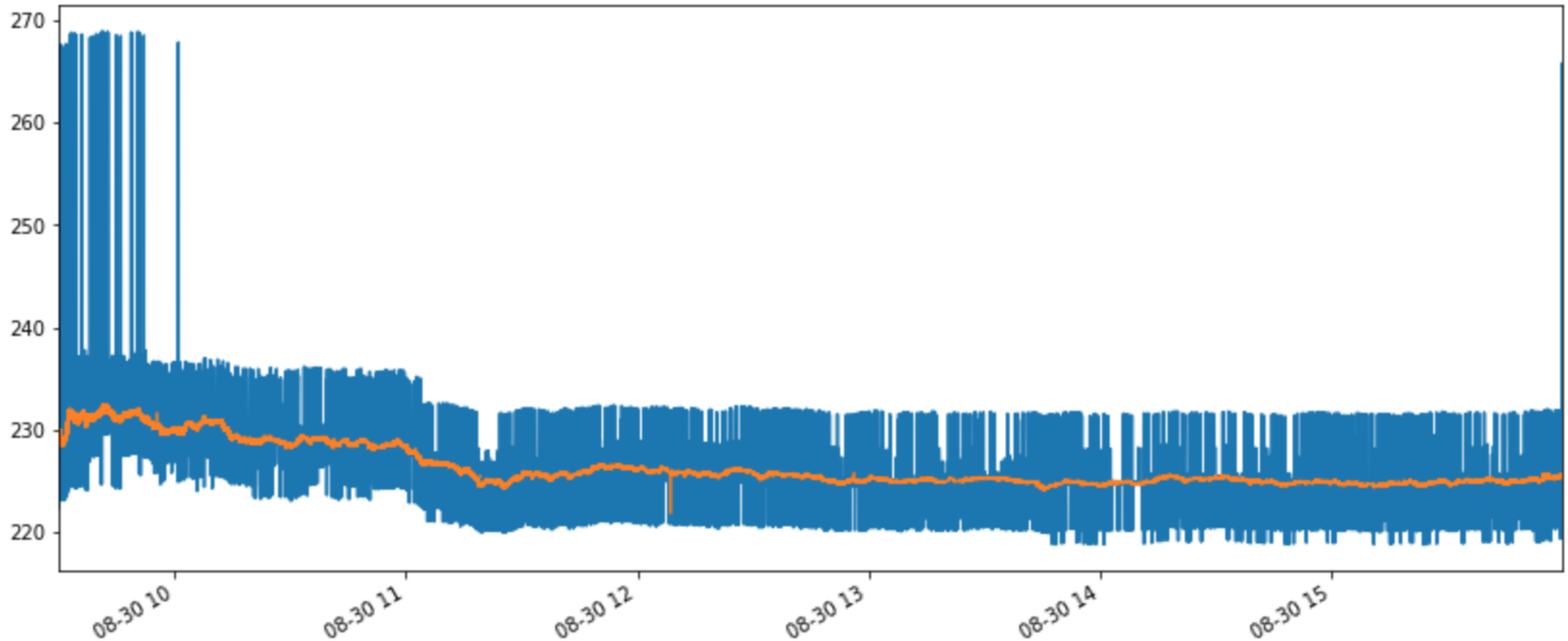
Principles of New Data, continued

I got some TSLA data… it looks ok mostly….



Principles of New Data, continued

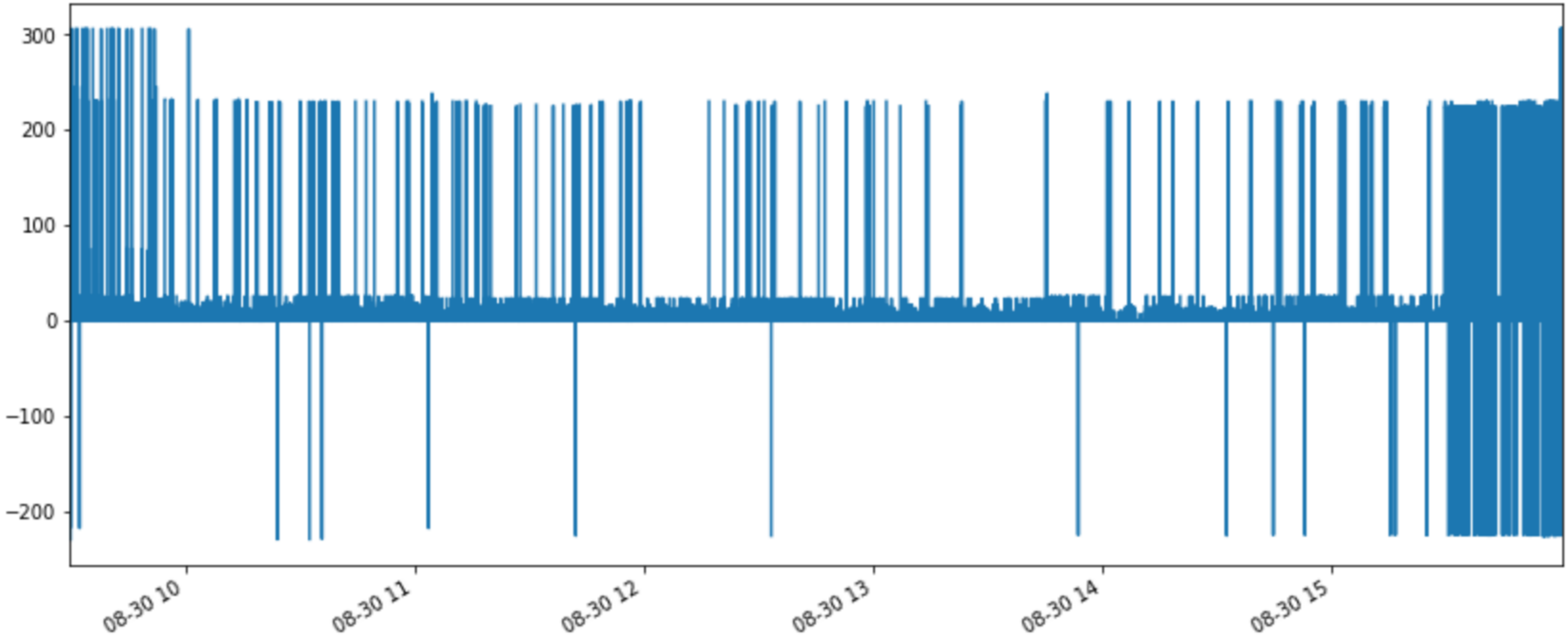
So then I loaded it into the sim and plotted my midpoints vs the stock price values…



Well THAT doesn't look right… what are the quotes doing?

Principles of New Data, continued

Yikes!



Principles of New Data, continued

It turns our we need to filter on some other conditions because **this isn't NBBO tick data!**

That's good news and bad news:

- Good news: we can theoretically build an order book with depth.
- Bad news: if we want to just read the NBBO we have some more work to do.

We can Read The Fine Manual and figure out what to do:
https://wrds-www.wharton.upenn.edu/documents/774/Daily_TAQ_Client_Spec_v2.2a.pdf

WRDS Tick Data Redux

1. Reload the tick data to get some extra fields (see the new version of *simtools.py* in Courseworks)

2. Change our logic slightly when we read the quote data

```
# skip if not NBBO
if not ( ( row.qu_source == 'N' ) and ( row.natbbo_ind == 4 ) ):
    continue
# set our local NBBO variables
if ( row.bid_px > 0 and row.bid_size > 0 ):
    bid_price = row.bid_px
    bid_size = row.bid_size * round_lot
if ( row.ask_px > 0 and row.ask_size > 0 ):
    ask_price = row.ask_px
    ask_size = row.ask_size * round_lot
```

WRDS Tick Data Redux

The Moral of the Story:

1. Read your data spec!

NOTE: my solution for the WRDS data is an approximation and is not perfect. It also ignores quote deletes and other things that we'll return to later….

Factor-Based VWAP

See VWAP v1 – Schedule Factor.ipynb

Lee-Ready: Trade Identification

Trade Identification

Inferring Trade Direction from Intraday Data

Charles M. C. Lee; Mark J. Ready

The Journal of Finance, Vol. 46, No. 2 (Jun., 1991), 733-746.

- All Trades are a match between one or more buy orders and one or more sell orders.
- Refer back to our concepts of passive (providing) and aggressive (taking)
- The liquidity taking order will cause the the trade to occur
- <u>The question</u>: can we identify "directionality" of trades, i.e. did a buy or a sell order trigger the trade?
- If we can identify buys and sells from trade data then we can potentially use that information in our analyses and trading signals

Trade Identification

We will consider three general cases

1. Identifying Trades without Quotes
2. Identifying Trades with Quotes
3. Trades inside the Spread

Trade Identification without Quotes

- We often may have only trades, not quotes
- In these cases, we can derive information from the trades themselves
- Let's classify trades relative to the trades <u>immediately preceding</u> them by two methods:
  - Tick Test
  - Reverse Tick Test

Tick Test

Look at trade price compared with the immediately preceding trade:

1. The trade is an **uptick** if the price is *higher* than the previous trade.
2. The trade is a **downtick** if the price is *lower* than the previous trade.
3. The trade is a **zero-uptick** if the previous trade was at the same price, but the previous price change was an *uptick*.
4. The trade is a **zero-downtick** if the previous trade was the same price, but the previous price change was a *downtick*.

Classification

A trade is a **buy** if it occurred on an uptick or a zero-uptick

A trade is a **sell** if it occurred on an downtick or zero-downtick

Reverse Tick Test

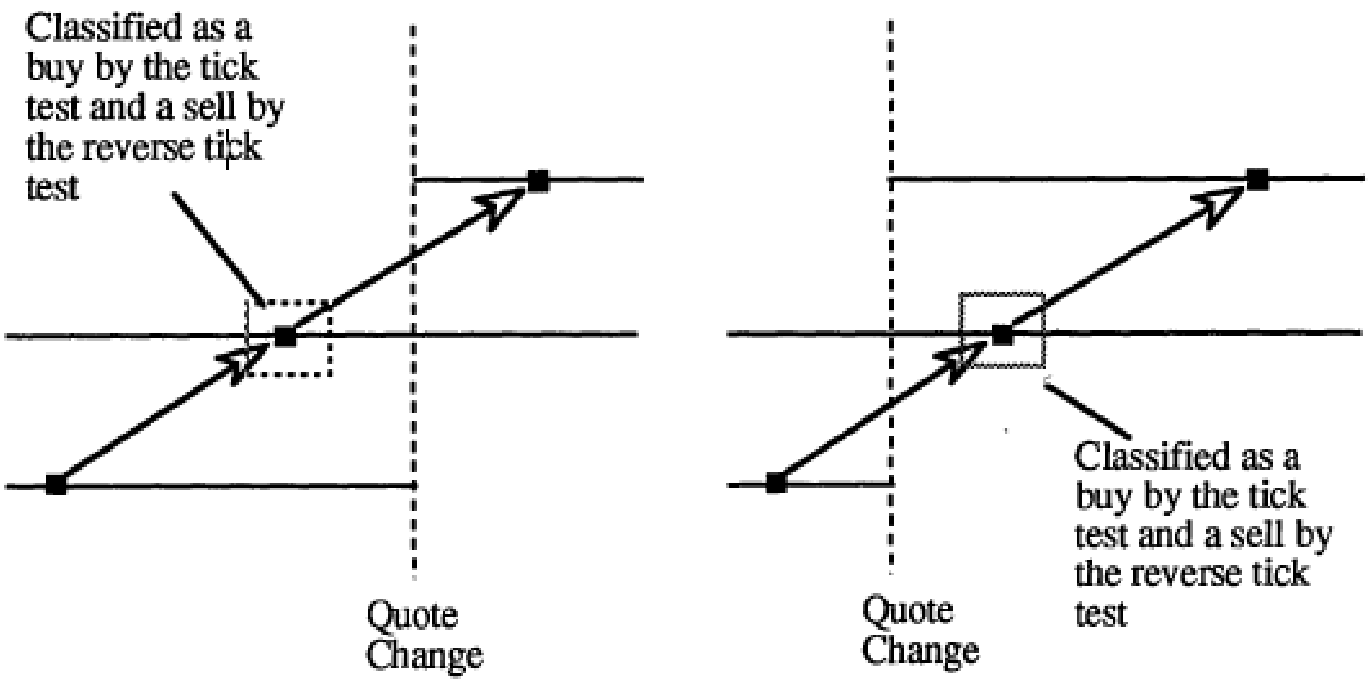
The reverse tick test compared to the immediately **following** trade.

1. If the following trade is at a *higher* price, the current trade is classified as a **sell**.

2. If the following trade is at a lower price, the current trade is classified as a **buy.**

Tick Test and Reverse Tick Test

Classification of Trades in Trending Markets with Quote Changes



Source: Lee, Ready (1991)

Tick Test and Reverse Tick Test

Table I

A Comparison of the Tick Test and the Reverse Tick Test When Identification of the Prevailing Quote is Unambiguous

For a sample of 150 NYSE firms during 1988, this table reports the percentage of trades classified as buys and sells by the tick test and the reverse tick test when the identity of the prevailing quote is unambiguous. A trade is classified as a buy (sell) by the tick test if the *prior* price change is positive (negative). A trade is classified as a buy by the reverse tick test if the *next* price change is negative (positive). A prevailing quote is considered unambiguously identified if the most recent quote revision occurred more than 5 seconds before the trade. This situation applied to 90.9 percent of the classified (nonopening) trades in the sample.

Classification Based on the Prevailing Quote	Tick Test Classification	Reverse Tick Test Classification
Buy—Above Ask	98.8% buy	91.3% buy
Buy—At Ask	92.1% buy	80.0% buy
Inside the Spread	52.4% sell	52.0% buy
Sell—At Bid	90.2% sell	79.4% sell
Sell—Below Bid	98.7% sell	94.5% sell

Source: Lee, Ready (1991)

COLUMBIA UNIVERSITY
IN THE CITY OF NEW YORK

Trade Identification with Quotes

We may have quotes, but which ones are the right ones to use?

Lee-Ready asks: are quotes at the time of a trade the <u>*cause*</u> or the <u>*result*</u> of a trade?

Using their *isolated trades*, they established that if a quote is *less than 5 seconds old*, it was probably caused by the trade being examined.

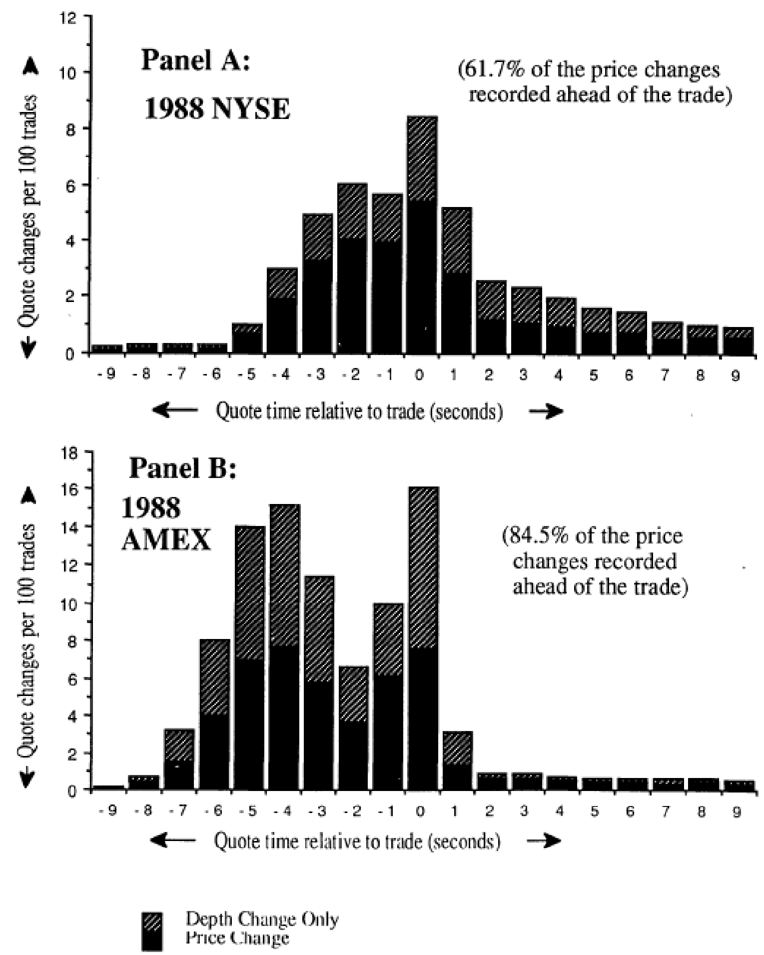


Figure 2. The frequency of quote revisions around isolated trades. The frequency of quote revisions in a 19-second window centered on isolated trades for 150 NYSE firms and all AMEX firms during 1988. An isolated trade is defined as the first trade after 11:00 a.m. but before 2:30 p.m., with no other trades within a 2-minute window centered on the trade. Both quote revisions which reflect price changes and quote revisions which affect depth changes only are shown.

Source: Lee, Ready (1991)

COLUMBIA UNIVERSITY
IN THE CITY OF NEW YORK

Trades Inside the Spread

Proposed options

- Location relative to quote ("bidness" and "askness")
  - A trade is a **buy** if the price is closer to the ask
  - A trade is a **sell** if the price is closer to the bid
  - Trades at midpoint are discarded

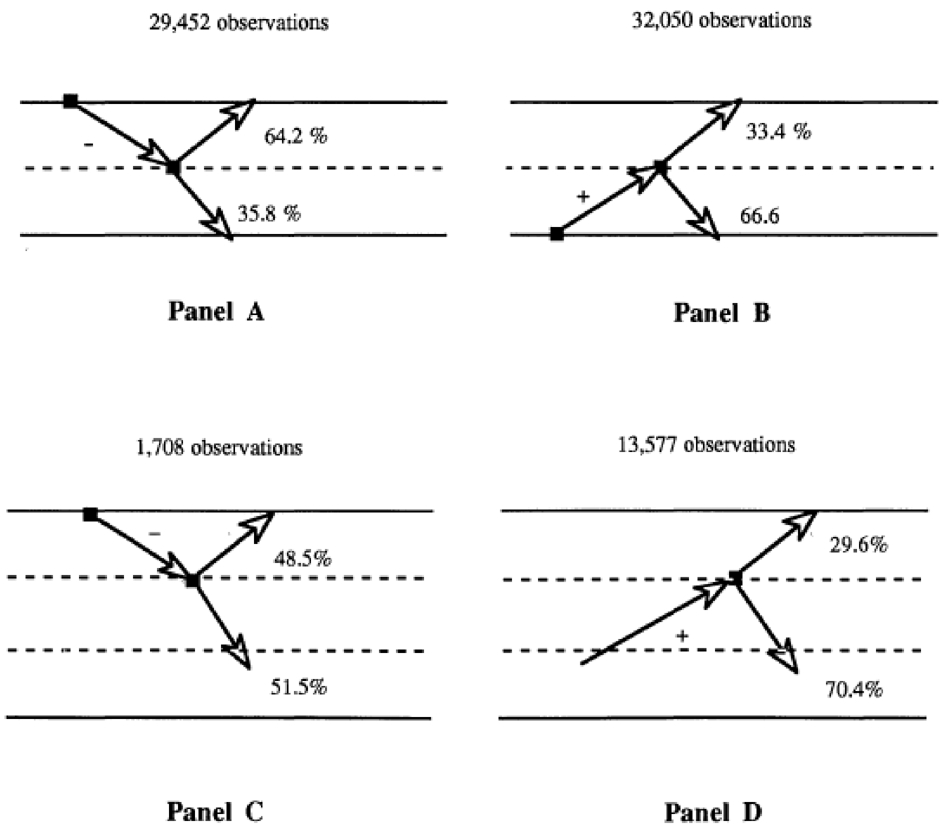


Figure 3. Frequency of price reversals for trades inside the spread. For a sample of 150 NYSE firms during 1988, these figures report the frequency of trades at the midpoint of a 1/4 spread (Panels A and B) and at the upper inside point of a 3/8 spread (Panels C and D). Panels A and C show the number of trades which occurred on a downtick and the proportion of uptick and downtick movements in the next price change. Panels B and D show the number of trades which occurred on an uptick and the proportion of uptick and downtick movements in the next price change.

Source: Lee, Ready (1991) *(NOTE: ¼ spread = \$0.25, 3/8 spread = \$0.375)*

COLUMBIA UNIVERSITY
IN THE CITY OF NEW YORK

Trades at Midpoint

According to the scenarios put forth in the paper, the tick test is effective at classifying trades inside the spread as well.

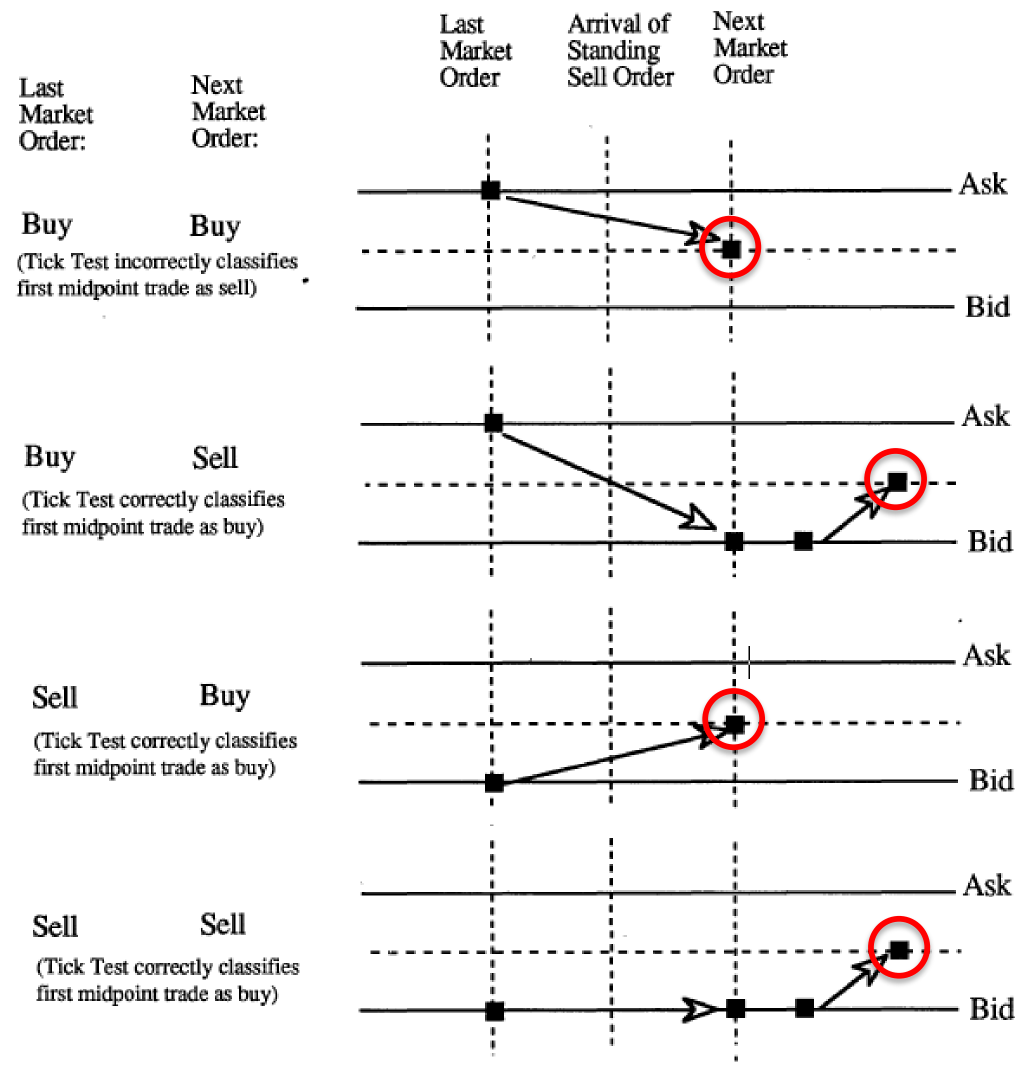


Figure 4. Classifying the first midpoint trade after the arrival of a standing sell order. Assume constant bid and ask prices and a ¼ spread. Further assume market buy and sell orders follow independent Poisson processes with the same arrival rate. Standing buy and sell orders also follow independent Poisson processes with a lower average arrival rate than market orders. The four equally likely patterns of market orders immediately before and after the arrival of a standing order are depicted above (buy/buy, buy/sell, sell/buy, and sell/sell). In three of the four cases, the tick test classification for the first midpoint trade is correct.

Source: Lee, Ready (1991)

Summary of the Lee-Ready Algorithm

1. If quotes are available, use them:
    a) If quotes are < 5 seconds (!) old, use previous quote
    b) Trades at the bid are **sells**
    c) Trades at the offer are **buys**
2. If trade is inside the spread but not at midpoint, classify by proximity to the bid or offer, apply same logic as #1
3. If trade is at midpoint or no quotes are available, use the tick test.

See the original paper for more detail

Limitations of Lee Ready

Changing Market Structure

- Narrower spreads
- Decimalization (2001)
- The original paper's 5-second window!
- Replication with isolated trades
- Market fragmentation
  - Non-displayed liquidity
  - Latency considerations

That said, we can still make use of this signal, in updated form.

Signal Factor Example: Tick Test

How do we use the theoretical value in trading decisions?

1. Consider a simplified conceptual version of a tick test signal T_j
2. Derive a signal from autocorrelation of trade directions:
   - State 1: Next tick predicted up
   - State 2: Next tick predicted down
   - State 3: Next tick indeterminate
3. Assume for the moment we want to trade immediately based on that signal
4. For a stock *j*, define price s_j to be the midpoint within the current quote.

   We can then express our three states as:

   - State 1: Next tick predicted up (buy): FV = the offer price = midpoint + spread/2
   - State 2: Next tick predicted down (sell): FV = the bid price = midpoint − spread/2
   - State 3: Next tick indeterminate: FV = midpoint

Measuring Autocorrelation

Let's look at the autocorrelation of the tick data:

1. Load Tick Data, (just trades)

2. Apply Lee-Ready (tick test) to do up/down
   - Use shift() operator to obtain previous row
   - Calculate difference
   - Use numpy sign() operator to retrieve the sign for each record
   - Apply from tick 2 to n

3. Calculate autocorrelation for n lags
   - Use autocorr(lag=n) for specific lags

4. Examine and Plot

See "Tick Test.ipynb"

Signal Factor Implementation: Tick Test

Once again, we can express this the signal as a continuous function:

$$FV = midpoint_j + b_{jF} F_{tick}$$

Where

b_{jF} = loading of the tick signal factor

F_{tick} = tick signal factor described as buy: 1, sell: -1

Calculate the exponential moving average of the tick value, something like this:

$$F_n = \frac{1}{\tilde{\tau}_w} f_n + w_n F_{n-1}$$

Where the weighting factor is defined as a function of a volume window $\tilde{\tau}_w$:

$$w_n = e^{-(t_n - t_{n-1})/\tilde{\tau}_w}$$

In practice you would calibrate the decay factor once and store the value.

f_n is by construction in {-1,1}, so multiply by $\frac{spread_{avg}}{2}$ to scale the factor.

Source: Derived from Michael Sotiropoulos BAML Lecture on trading models

Determining Volume Window

For the volume window, we want trade time, not clock time.

During simulation we can count trades tick by tick and construct a rolling window of *n* trades.

But if you want to convert to clock time:

$$\tilde{\tau}_w = max\left(\tau_{min}, \left(\tau_{max}, \frac{T_{day}}{N_{trd}} n\right)\right)$$

Where:

τ_{min}, τ_{max} are user defined min and max values

T_{day} = length of the trading day (390 minutes for the US stock exchanges)

N_{trd} = avg number of trades per day

n = target number of trades to include

Source: Derived from Michael Sotiropoulos presentation on trading models

Our Two Factor Model

So now we have:

$$FV_j = midpoint_j + b_j F_{tick} + b_j F_{schedule}$$

Where:

F_k = factors for each signal or adjustment

b_{jk}= stock *j's* sensitivity to factor F_k

Questions?