Financial Market Frictions and Learning from the Stock Price

Lets “Pay attention to the plumbing” (Levitt 2003)

Dominik Rösch

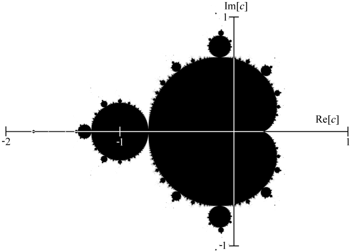
University at Buffalo

drosch@buffalo.edu

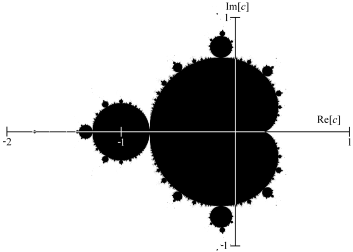
February 13, 2020

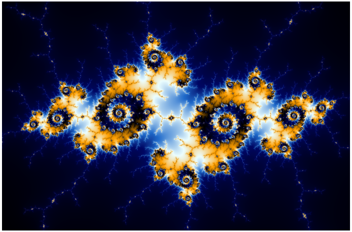
This work is supported in part by NSF ACI-1541215.

Mandelbrot Set

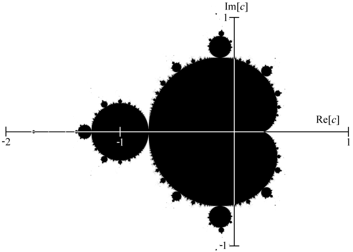


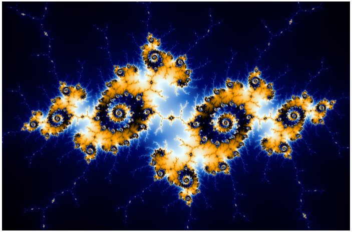
Mandelbrot Set





Mandelbrot Set

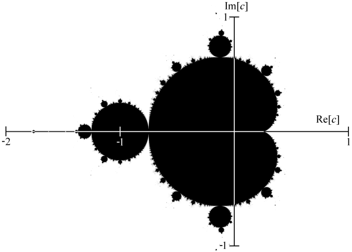




coloured by how fast $c \in \mathbb{C} = (\mathbb{R}, \mathbb{R})$ converges to infinity in:

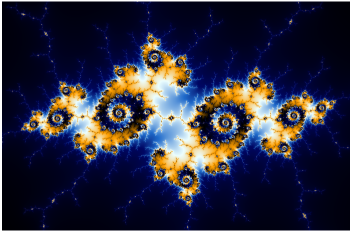
$$z_{n+1} = z_n^2 + c, \; \textit{with } z_0 = 0$$

Mandelbrot Set



coloured by how fast $c \in \mathbb{C} = (\mathbb{R}, \mathbb{R})$ converges to infinity in:

$$z_{n+1} = z_n^2 + c, \ \textit{with } z_0 = 0$$



“Can there be a more striking demonstration of the enormous complexity hidden in the simplest laws?” (Peitgen, Richter 1986)

Data: Stock Price

- What does it mean to learn from the stock price?

Data: Stock Price

- What does it mean to learn from the stock price?
- Berkshire trades at both over USD 300,000 for its A-share and just a bit over USD 200 for its B-share

Data: Stock Price

- What does it mean to learn from the stock price?
- Berkshire trades at both over USD 300,000 for its A-share and just a bit over USD 200 for its B-share
    - Companies often have different share classes, such companies are often dropped from empirical analysis.
- Prices are (almost) meaningless as they are under the control of the company

Data: Stock Price

- What does it mean to learn from the stock price?
- Berkshire trades at both over USD 300,000 for its A-share and just a bit over USD 200 for its B-share
    - Companies often have different share classes, such companies are often dropped from empirical analysis.
- Prices are (almost) meaningless as they are under the control of the company
- Prices are often more understood as ratios, e.g., price to earnings, price to book, or just returns.

Data: Returns

- Imagine the following investment, each day you flip a coin on head you get 11% and on tail you lose 10%
- Is this a good investment?

Data: Returns

- Imagine the following investment, each day you flip a coin on head you get 11% and on tail you lose 10%
- Is this a good investment?
- The expected return is 0.5% ... annualized more than 100%
- Starting with USD 1, after less than 30 years you are a billionaire!

Data: Returns

- Imagine the following investment, each day you flip a coin on head you get 11% and on tail you lose 10%
- Is this a good investment?
- The expected return is 0.5% ... annualized more than 100%
- Starting with USD 1, after less than 30 years you are a billionaire!
- Or are you? Actually, in the long-term you lose, the expected continously compounded return is negative!
- Empirically, should you use simple or log-returns? What about dividends? (convention: assumed re-invested, except in special papers)

In our case Stock Price: Tobins Q (Corporate Finance)

- Tobins Q is the “market value of equity (price times shares outstanding from CRSP) plus book value of assets minus the book value of equity, scaled by book assets.”

In our case Stock Price: Tobins Q (Corporate Finance)

- Tobins Q is the "market value of equity (price times shares outstanding from CRSP) plus book value of assets minus the book value of equity, scaled by book assets."
- "$Q = (mktcap + ATQ - bvse)/ATQ$, where ATQ is quarterly total assets from Compustat, and bvse is stockholders' equity from Compustat"

In our case Stock Price: Tobins Q (Corporate Finance)

- Tobins Q is the "market value of equity (price times shares outstanding from CRSP) plus book value of assets minus the book value of equity, scaled by book assets."
- "$Q = (mktcap + ATQ - bvse)/ATQ$, where ATQ is quarterly total assets from Compustat, and bvse is stockholders' equity from Compustat"
- Actually, $Q = (mktcap/1000 + atq - bvse)/atq$ with $mktcap = |Price| \times shrout$

In our case Stock Price: Tobins Q (Corporate Finance)

- Tobins Q is the "market value of equity (price times shares outstanding from CRSP) plus book value of assets minus the book value of equity, scaled by book assets."
- "$Q = (mktcap + ATQ - bvse)/ATQ$, where ATQ is quarterly total assets from Compustat, and bvse is stockholders' equity from Compustat"
- Actually, $Q = (mktcap/1000 + atq - bvse)/atq$ with $mktcap = |Price| \times shrout$
- because of scaling and in CRSP *Price* can be negative

Financial Market Frictions (Market Microstructre)

- Illiquidity
  - Quoted Spreads
  - Effective Spreads
- Inefficiency
  - OIB Predictability
  - Variance ratio
  - Hasbrouck
  - Put-call parity
- Not necessarily, Inefficiency = Illiquidity.

“Financial Market Frictions and Learning from the Stock Price”

- “Learning”: Investment-Sensitivity to Q (ISQ), Q-Theory

“Financial Market Frictions and Learning from the Stock Price”

- “Learning”: Investment-Sensitivity to Q (ISQ), Q-Theory
- $Inv_{i,q} = \beta_{0,q} + \beta_{1,q} Fric_{i,q-1} \times Q_{i,q-1} + \beta_{2,q} Fric_{i,q-1} + \beta_{3,q} Q_{i,q-1} + \Sigma \beta_q Contr_{i,q-1} + \epsilon_{i,q}$

"Financial Market Frictions and Learning from the Stock Price"

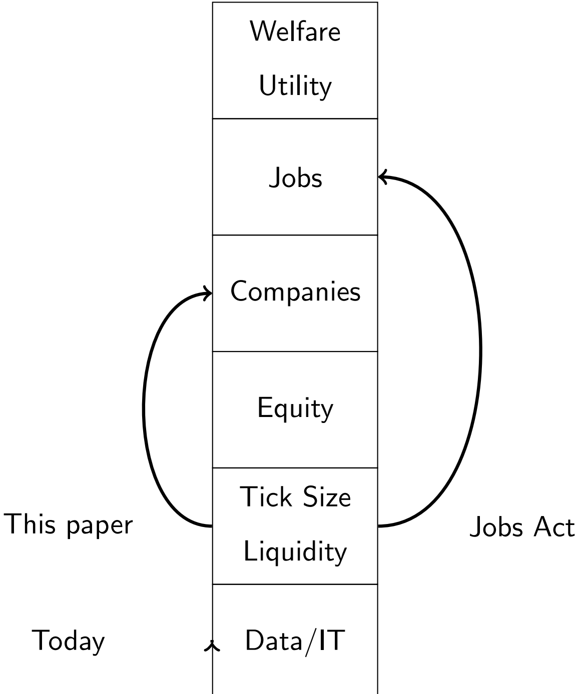
- "Learning": Investment-Sensitivity to Q (ISQ), Q-Theory
- $Inv_{i,q} = \beta_{0,q} + \beta_{1,q} Fric_{i,q-1} \times Q_{i,q-1} + \beta_{2,q} Fric_{i,q-1} + \beta_{3,q} Q_{i,q-1} + \Sigma \beta_q Contr_{i,q-1} + \epsilon_{i,q}$
- Investment is measured, e.g., as CAPX which is reported as a running sum within each fiscal year.

“Financial Market Frictions and Learning from the Stock Price”

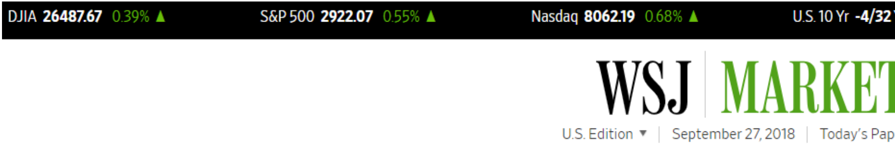
- “Learning”: Investment-Sensitivity to Q (ISQ), Q-Theory
- $Inv_{i,q} = \beta_{0,q} + \beta_{1,q} Fric_{i,q-1} \times Q_{i,q-1} + \beta_{2,q} Fric_{i,q-1} + \beta_{3,q} Q_{i,q-1} + \Sigma \beta_q Contr_{i,q-1} + \epsilon_{i,q}$
- Investment is measured, e.g., as CAPX which is reported as a running sum within each fiscal year.
- Do frictions increase or decrease ISQ ($\beta_{1,q}$)

“Financial Market Frictions and Learning from the Stock Price”

- “Learning”: Investment-Sensitivity to Q (ISQ), Q-Theory
- $Inv_{i,q} = \beta_{0,q} + \beta_{1,q} Fric_{i,q-1} \times Q_{i,q-1} + \beta_{2,q} Fric_{i,q-1} + \beta_{3,q} Q_{i,q-1} + \Sigma \beta_q Contr_{i,q-1} + \epsilon_{i,q}$
- Investment is measured, e.g., as CAPX which is reported as a running sum within each fiscal year.
- Do frictions increase or decrease ISQ ($\beta_{1,q}$)
- We find: $\beta_{1,q} > 0$, ISQ increases with frictions, ISQ is lower for more liquid stocks (because of noise trading)
- “[w]hat’s needed for a liquid market causes prices to be less efficient”(Black, 1986, p. 532).
- Normally, now argue that this relation is causal.


Welfare
Utility
Jobs
Companies
Equity
Tick Size
Liquidity
Data/IT
This paper
Jobs Act
Today

4 Bytes of Financial Data: 2922.07

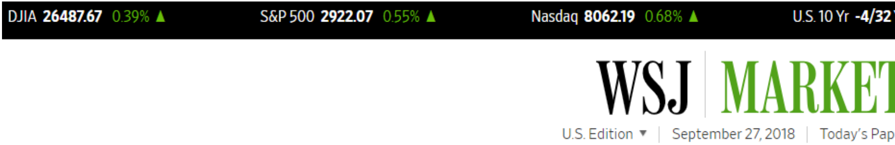


DJIA **26487.67** 0.39% ▲ S&P 500 **2922.07** 0.55% ▲ Nasdaq **8062.19** 0.68% ▲ U.S. 10 Yr **-4/32**

WSJ | MARKET

U.S. Edition ▼ | September 27, 2018 | Today's Pap

4 Bytes of Financial Data: 2922.07



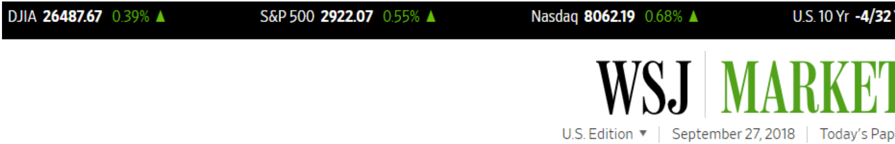
DJIA **26487.67** 0.39% ▲ S&P 500 **2922.07** 0.55% ▲ Nasdaq **8062.19** 0.68% ▲ U.S. 10 Yr **-4/32**

WSJ MARKET

U.S. Edition ▼ | September 27, 2018 | Today's Pap

- 2kb: 2922.07 is an average of across around 500 prices
- 20kb: Each stock is priced on around 10 venues.
- 400kb: At each time each stock has around 20 prices [depth].
- 4mb/second: Prices often change 10 times per second
- 93GB/day: A trading day has 6.5 hours.

4 Bytes of Financial Data: 2922.07



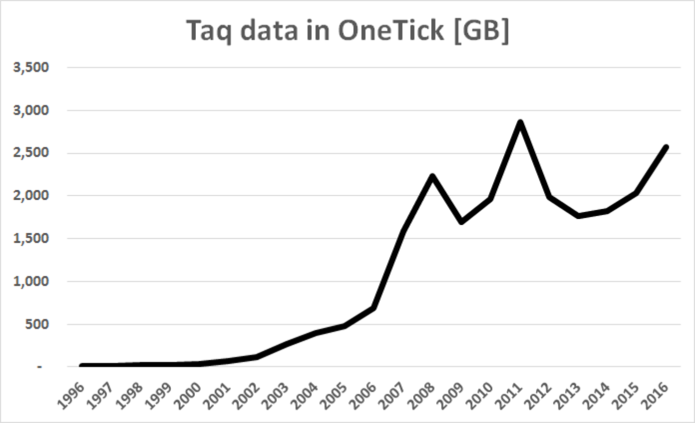
DJIA **26487.67** 0.39% ▲ S&P 500 **2922.07** 0.55% ▲ Nasdaq **8062.19** 0.68% ▲ U.S. 10 Yr **-4/32**

WSJ | MARKET

U.S. Edition ▼ | September 27, 2018 | Today's Pap

- 2kb: 2922.07 is an average of across around 500 prices
- 20kb: Each stock is priced on around 10 venues.
- 400kb: At each time each stock has around 20 prices [depth].
- 4mb/second: Prices often change 10 times per second
- 93GB/day: A trading day has 6.5 hours.
- if you compress the data you need around 10GB/day.

US tick-by-tick data (TAQ)



- Data contains more than just prices, such as sizes, and other identifiers. Data does not contain depth, just the best available buy and sell prices per venue.

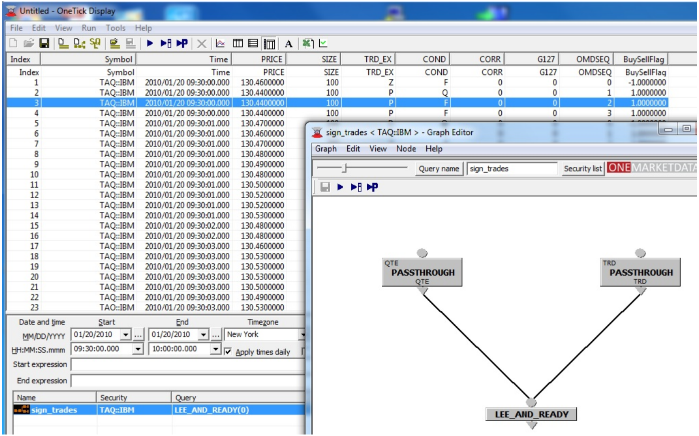
My first attempt: Mysql framework

- I started writing my stored procedure:
  - get_ohlc_for(ticker, between, bucket)
- when I coded:

```
SUBSTRING_INDEX(
        MAX(CONCAT(time, ’_’, price)), ’_’, -1
) AS ‘close‘
```

- I realized MySQL might not be the best DB for financial data.

My second attempt: OneTick framework



Index	Symbol	Time	PRICE	SIZE	TRD_EX	COND	CORR	G127	OMDSEQ	BuySellFlag
Index	Symbol	Time	PRICE	SIZE	TRD_EX	COND	CORR	G127	OMDSEQ	BuySellFlag
1	TAQ::IBM	2010/01/20 09:30:00.000	130.4600000	100	Z	F	0	0	0	-1.0000000
2	TAQ::IBM	2010/01/20 09:30:00.000	130.4400000	100	P	Q	0	0	1	1.0000000
3	TAQ::IBM	2010/01/20 09:30:00.000	130.4400000	100	P	F	0	0	2	1.0000000
4	TAQ::IBM	2010/01/20 09:30:00.000	130.4400000	100	P	F	0	0	3	1.0000000
5	TAQ::IBM	2010/01/20 09:30:01.000	130.4700000							
6	TAQ::IBM	2010/01/20 09:30:01.000	130.4600000							
7	TAQ::IBM	2010/01/20 09:30:01.000	130.4700000							
8	TAQ::IBM	2010/01/20 09:30:01.000	130.4800000							
9	TAQ::IBM	2010/01/20 09:30:01.000	130.4900000							
10	TAQ::IBM	2010/01/20 09:30:01.000	130.4800000							
11	TAQ::IBM	2010/01/20 09:30:01.000	130.5000000							
12	TAQ::IBM	2010/01/20 09:30:01.000	130.5200000							
13	TAQ::IBM	2010/01/20 09:30:01.000	130.5200000							
14	TAQ::IBM	2010/01/20 09:30:01.000	130.5300000							
15	TAQ::IBM	2010/01/20 09:30:02.000	130.4800000							
16	TAQ::IBM	2010/01/20 09:30:02.000	130.4800000							
17	TAQ::IBM	2010/01/20 09:30:03.000	130.4600000							
18	TAQ::IBM	2010/01/20 09:30:03.000	130.5300000							
19	TAQ::IBM	2010/01/20 09:30:03.000	130.5300000							
20	TAQ::IBM	2010/01/20 09:30:03.000	130.5300000							
21	TAQ::IBM	2010/01/20 09:30:03.000	130.5000000							
22	TAQ::IBM	2010/01/20 09:30:03.000	130.4900000							
23	TAQ::IBM	2010/01/20 09:30:03.000	130.5300000							

Date and time	Start	End	Timezone
MM/DD/YYYY	01/20/2010	01/20/2010	New York
HH:MM:SS.mmm	09:30:00.000	10:00:00.000	☑ Apply times daily
Start expression			
End expression			

Name	Security	Query
sign_trades	TAQ::IBM	LEE_AND_READY(0)

Query name: sign_trades

Step 0: Setting up reference data

- Exchange trading times (e.g., NYSE: 09:00-16:30)
- Primary Exchange (e.g., IBM is NYSE)
- Symbology mappings (PERMNO to TICKER, e.g. Citigroup)
    - 70519——19861029000000—19890119000000—CCC—
    - 70519——19890120000000—19931231000000—PA—
    - 70519——19940103000000—19981007000000—TRV—
    - 70519——19981008000000—19981203000000—CCI—
    - 70519——19981204000000—20161230000000—C—
- Corporate actions (stock splits, dividends)

Step 0: Setting up reference data

- Exchange trading times (e.g., NYSE: 09:00-16:30)
- Primary Exchange (e.g., IBM is NYSE)
- Symbology mappings (PERMNO to TICKER, e.g. Citigroup)
    - 70519——19861029000000—19890119000000—CCC—
    - 70519——19890120000000—19931231000000—PA—
    - 70519——19940103000000—19981007000000—TRV—
    - 70519——19981008000000—19981203000000—CCI—
    - 70519——19981204000000—20161230000000—C—
- Corporate actions (stock splits, dividends)
- After REF_DATA and importing TAQ: Lets compute effective spreads...

Effective spreads: Data Processing

- $ESPR_{i,t} = 2 \times sign_{i,t} \times (TRD_{i,t} - MID_{i,t-})$
- Need TRD prices and midpoint prices prevailing TRD, $MID_{i,t-}$

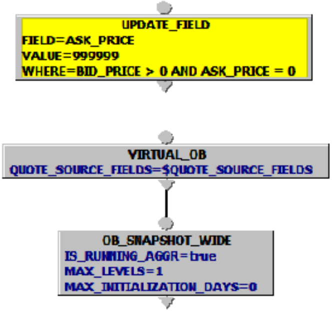
Effective spreads: Data Processing

- $ESPR_{i,t} = 2 \times sign_{i,t} \times (TRD_{i,t} - MID_{i,t-})$
- Need TRD prices and midpoint prices prevailing TRD, $MID_{i,t-}$
- "For data before 1996, we delay quotes by 5 seconds following Lee and Ready (1991)."
- "We use monthly TAQ data from 1993 to 2014, with trades and quotes timestamped by the second, and daily TAQ data from 2014 onwards."
- "We apply the Holden and Jacobsen (2014) interpolated time technique for all data from 1983 to 2014"
- $sign_{i,t}$ use Lee and Ready (1991) using NBBO (?), or take absolute values (equivalent?)

Step 1.0: QTE need National Best Bid Offer (NBBO)

- Each exchange reports prices.
- Easy for TRD: “sum” them up.
- Difficult for QTE: at each point in time look at all ASK (BID) across exchanges and take best one, i.e., lowest (highest)
- This is called National Best Bid and Offer (NBBO)
- TAQ has a file with NBBO prices, but with errors.
- Before NBBO, do not ignore 0’s, this means orders are cancelled and current exchange doesnt have valid price.

Step 1.0: Compute NBBO



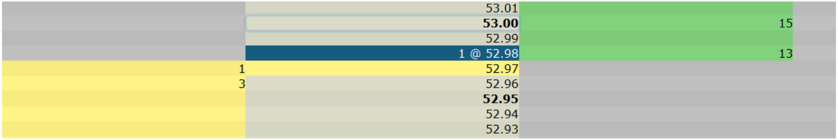
UPDATE_FIELD
FIELD=ASK_PRICE
VALUE=999999
WHERE=BID_PRICE > 0 AND ASK_PRICE = 0

VIRTUAL_OB
QUOTE_SOURCE_FIELDS=$QUOTE_SOURCE_FIELDS

OB_SNAPSHOT_WIDE
IS_RUNNING_AGGR=true
MAX_LEVELS=1
MAX_INITIALIZATION_DAYS=0

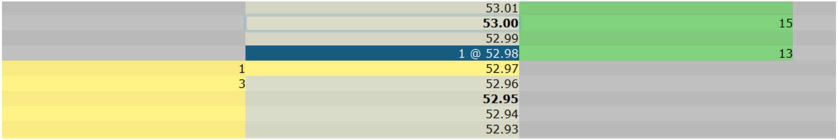
And others

Step 1.0: Compute NBBO, “Limit order book”



	53.01	
	53.00	15
	52.99	
	1 @ 52.98	13
1	52.97	
3	52.96	
	52.95	
	52.94	
	52.93	

Step 1.0: Compute NBBO, “Limit order book”



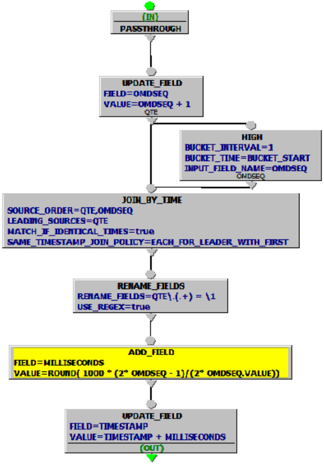
	53.01	
	53.00	15
	52.99	
	1 @ 52.98	13
1	52.97	
3	52.96	
	52.95	
	52.94	
	52.93	

- construct a virtual order book based on Best Bid Ask quotes from different exchanges.

Step 1.1: MTAQ vs DTAQ: Holden and Jacobsen (2014) interpolated time

- MTAQ is timestamped at second accuracy, DTAQ at least milliseconds.
- Imagine one trade t with many quotes in second s.
- You want quote prevailing t, but which one?

Step 1.1: Compute interpolated time



```
PASSTHROUGH

UPDATE_FIELD
FIELD=OMDSEQ
VALUE=OMDSEQ + 1

HIGH
BUCKET_INTERVAL=1
BUCKET_TIME=BUCKET_START
INPUT_FIELD_NAME=OMDSEQ

JOIN_BY_TIME
SOURCE_ORDER=QTE,OMDSEQ
LEADING_SOURCES=QTE
MATCH_IF_IDENTICAL_TIMES=true
SAME_TIMESTAMP_JOIN_POLICY=EACH_FOR_LEADER_WITH_FIRST

RENAME_FIELDS
RENAME_FIELDS=QTE\.(.+) = \1
USE_REGEX=true

ADD_FIELD
FIELD=MILLISECONDS
VALUE=ROUND( 1000 * (2* OMDSEQ - 1)/(2* OMDSEQ.VALUE))

UPDATE_FIELD
FIELD=TIMESTAMP
VALUE=TIMESTAMP + MILLISECONDS
```

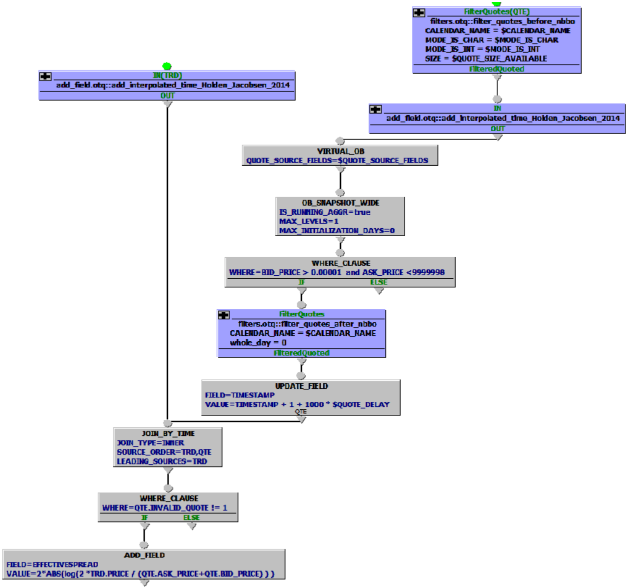
Step 1.2: Delay quotes

- Why? TRD and QTE are reported separately. TRD not as important, therefore reported with delay.

Step 1.2: Delay quotes

- Why? TRD and QTE are reported separately. TRD not as important, therefore reported with delay.
- Timestamps of both are when published by SIP (Securities Information Processor, see Bartlett and McCrary JFM, 2019)
- DTAQ provides several timestamps like “Participant Timestamp”, “Trade Reporting Facility (TRF) Timestamp” (see NYSE)

Step 2: Compute Effective spreads per trade



FilterQuotes(QTE)
filters.otq::filter_quotes_before_nbbo
CALENDAR_NAME = $CALENDAR_NAME
MODE_IS_CHAR = $MODE_IS_CHAR
MODE_IS_INT = $MODE_IS_INT
SIZE = $QUOTE_SIZE_AVAILABLE
FilteredQuoted

IN(TRD)
add_field.otq::add_interpolated_time_Holden_Jacobsen_2014
OUT

IN
add_field.otq::add_interpolated_time_Holden_Jacobsen_2014
OUT

VIRTUAL_OB
QUOTE_SOURCE_FIELDS=$QUOTE_SOURCE_FIELDS

OB_SNAPSHOT_WIDE
IS_RUNNING_AGGR=true
MAX_LEVELS=1
MAX_INITIALIZATION_DAYS=0

WHERE_CLAUSE
WHERE=BID_PRICE > 0.00001 and ASK_PRICE <9999998
IF ELSE

FilterQuotes
filters.otq::filter_quotes_after_nbbo
CALENDAR_NAME = $CALENDAR_NAME
whole_day = 0
FilteredQuoted

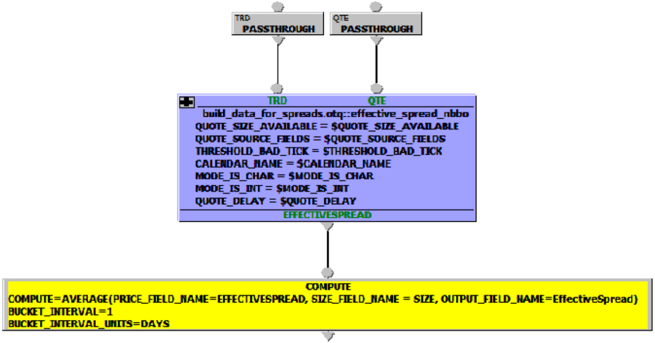
UPDATE_FIELD
FIELD=TIMESTAMP
VALUE=TIMESTAMP + 1 + 1000 * $QUOTE_DELAY
QTE

JOIN_BY_TIME
JOIN_TYPE=INNER
SOURCE_ORDER=TRD,QTE
LEADING_SOURCES=TRD

WHERE_CLAUSE
WHERE=QTE.INVALID_QUOTE != 1
IF ELSE

ADD_FIELD
FIELD=EFFECTIVESPREAD
VALUE=2*ABS(log(2 *TRD.PRICE / (QTE.ASK_PRICE+QTE.BID_PRICE)))

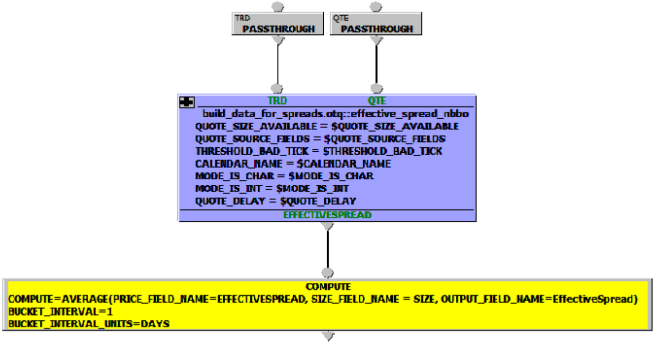
Step 3: Aggregate Effective spread per day



```
build_data_for_spreads.otq::effective_spread_nbbo
QUOTE_SIZE_AVAILABLE = $QUOTE_SIZE_AVAILABLE
QUOTE_SOURCE_FIELDS = $QUOTE_SOURCE_FIELDS
THRESHOLD_BAD_TICK = $THRESHOLD_BAD_TICK
CALENDAR_NAME = $CALENDAR_NAME
MODE_IS_CHAR = $MODE_IS_CHAR
MODE_IS_INT = $MODE_IS_INT
QUOTE_DELAY = $QUOTE_DELAY
```

```
COMPUTE=AVERAGE(PRICE_FIELD_NAME=EFFECTIVESPREAD, SIZE_FIELD_NAME = SIZE, OUTPUT_FIELD_NAME=EffectiveSpread)
BUCKET_INTERVAL=1
BUCKET_INTERVAL_UNITS=DAYS
```

Step 3: Aggregate Effective spread per day



```
build_data_for_spreads.otq::effective_spread_nbbo
QUOTE_SIZE_AVAILABLE = $QUOTE_SIZE_AVAILABLE
QUOTE_SOURCE_FIELDS = $QUOTE_SOURCE_FIELDS
THRESHOLD_BAD_TICK = $THRESHOLD_BAD_TICK
CALENDAR_NAME = $CALENDAR_NAME
MODE_IS_CHAR = $MODE_IS_CHAR
MODE_IS_INT = $MODE_IS_INT
QUOTE_DELAY = $QUOTE_DELAY
```

```
COMPUTE=AVERAGE(PRICE_FIELD_NAME=EFFECTIVESPREAD, SIZE_FIELD_NAME = SIZE, OUTPUT_FIELD_NAME=EffectiveSpread)
BUCKET_INTERVAL=1
BUCKET_INTERVAL_UNITS=DAYS
```

- How to measure effective spreads without trades?
- How to measure quoted spreads without quotes?
- Decision to trade/quote are endogeneous.

Step 4 Run on server (using GNU Parallel)

```
database = ${1}
query    = ${2}

for date in dates.txt ; do
    sem -k --id exp -P ot.cpus
        export.pl ${database} ${date} ${query}
done

sem --wait --id exp
```

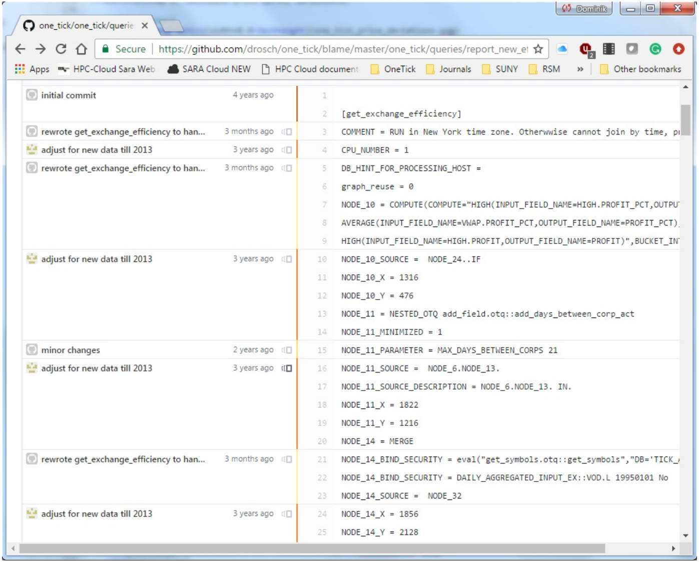
Step 5: Test! TDD, write a test suite.

```
lives_ok {
    $one_tick->import_csv();
}
’can import data into OneTick’;

my $results_summary = {
    ’total_ticks’    => 196232,
};

Test::More::is_deeply($one_tick->results_summary,
    $results_summary,
    ’Got summary results as expected’
);
```

Step 6: Write a paper

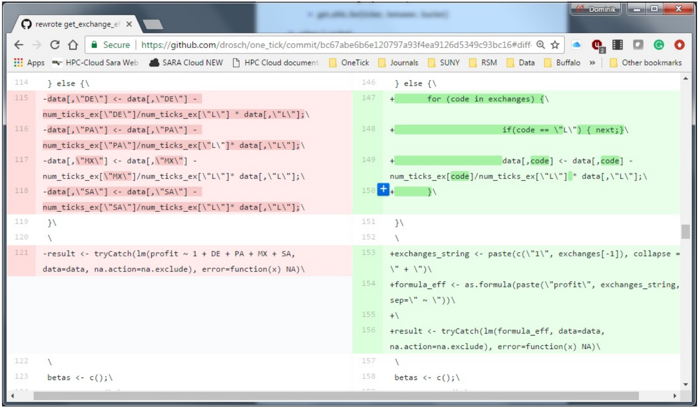
- RStudio (easiest at Vidia CCR or RStudio Cloud) within Rstudio:
- Version Control
  - "File" ⟶ "New Project" ⟶ "Version Control Git"
  - Use "https" address, better set up "ssh"
- Reproducible (R Sweave)
  - "New File" ⟶ "R Sweave"
- Access to Data (ODBC)
  - "Connections" ⟶ "New Connection" (right)

Step 6.1: How to make sure code is consistent? Version Control!



```
initial commit                                  4 years ago
                                                               [get_exchange_efficiency]
rewrote get_exchange_efficiency to han...       3 months ago   COMMENT = RUN in New York time zone. Otherwwise cannot join by time, pı
adjust for new data till 2013                   3 years ago    CPU_NUMBER = 1
rewrote get_exchange_efficiency to han...       3 months ago   DB_HINT_FOR_PROCESSING_HOST =
                                                               graph_reuse = 0
                                                               NODE_10 = COMPUTE(COMPUTE="HIGH(INPUT_FIELD_NAME=HIGH.PROFIT_PCT,OUTPU
                                                               AVERAGE(INPUT_FIELD_NAME=VWAP.PROFIT_PCT,OUTPUT_FIELD_NAME=PROFIT_PCT)
                                                               HIGH(INPUT_FIELD_NAME=HIGH.PROFIT,OUTPUT_FIELD_NAME=PROFIT)",BUCKET_IN
adjust for new data till 2013                   3 years ago    NODE_10_SOURCE =  NODE_24..IF
                                                               NODE_10_X = 1316
                                                               NODE_10_Y = 476
                                                               NODE_11 = NESTED_OTQ add_field.otq::add_days_between_corp_act
                                                               NODE_11_MINIMIZED = 1
minor changes                                   2 years ago    NODE_11_PARAMETER = MAX_DAYS_BETWEEN_CORPS 21
adjust for new data till 2013                   3 years ago    NODE_11_SOURCE =  NODE_6.NODE_13.
                                                               NODE_11_SOURCE_DESCRIPTION = NODE_6.NODE_13. IN.
                                                               NODE_11_X = 1822
                                                               NODE_11_Y = 1216
                                                               NODE_14 = MERGE
rewrote get_exchange_efficiency to han...       3 months ago   NODE_14_BIND_SECURITY = eval("get_symbols.otq::get_symbols","DB='TICK_
                                                               NODE_14_BIND_SECURITY = DAILY_AGGREGATED_INPUT_EX::VOD.L 19950101 No
                                                               NODE_14_SOURCE =  NODE_32
adjust for new data till 2013                   3 years ago    NODE_14_X = 1856
                                                               NODE_14_Y = 2128
```

Step 6.1: How to make sure code is consistent? git diff



```
 } else {\
-data[,\"DE\"] <- data[,\"DE\"] -
num_ticks_ex[\"DE\"]/num_ticks_ex[\"L\"] * data[,\"L\"];\
-data[,\"PA\"] <- data[,\"PA\"] -
num_ticks_ex[\"PA\"]/num_ticks_ex[\"L\"]* data[,\"L\"];\
-data[,\"MX\"] <- data[,\"MX\"] -
num_ticks_ex[\"MX\"]/num_ticks_ex[\"L\"]* data[,\"L\"];\
-data[,\"SA\"] <- data[,\"SA\"] -
num_ticks_ex[\"SA\"]/num_ticks_ex[\"L\"]* data[,\"L\"];\
 }\
 \
-result <- tryCatch(lm(profit ~ 1 + DE + PA + MX + SA,
data=data, na.action=na.exclude), error=function(x) NA)\
 \
 betas <- c();\
```

```
 } else {\
+        for (code in exchanges) {\
+                        if(code == \"L\") { next;}\
+                        data[,code] <- data[,code] -
num_ticks_ex[code]/num_ticks_ex[\"L\"] * data[,\"L\"];\
+        }\
 }\
 \
+exchanges_string <- paste(c(\"1\", exchanges[-1]), collapse =
\" + \")\
+formula_eff <- as.formula(paste(\"profit\", exchanges_string,
sep=\" ~ \"))\
+\
+result <- tryCatch(lm(formula_eff, data=data,
na.action=na.exclude), error=function(x) NA)\
 \
 betas <- c();\
```

Step 6.2: Pull results into favourite statistical software: ODBC

- Regardless which DB, use stored procedures.

```
library("RODBC")
channel <- odbcConnect("ONETICK_DEFAULT_DSN")

test <- sqlQuery(channel, "SELECT *
    FROM OTQ_FILES.examples::get_pespr QTE
    WHERE (QTE.TIMESTAMP>=’2004-01-01 00:00:00 GM
        AND (QTE.TIMESTAMP<’2004-01-03 00:00:00 G
        AND (param_assign(’query_paras’,’’)=1)"
    )
```

Step 6.3: How to make sure paper is reproducible? RSweave!

- Embed ODBC / R-code into Latex using RSweave.

Step 6.3: How to make sure paper is reproducible? RSweave!

- Embed ODBC / R-code into Latex using RSweave.

```
\begin{table}[H]
    \caption{\textbf{Caption ...} This table reports

    \#Stocks &      Mean    \\

    \Sexpr{pretty_numbers(format_integer, data_obs[["
    \Sexpr{pretty_numbers(format_integer, data_means[
```

What about sizes?

- Nasdaq Dealer Market: Double counting.

What about sizes?

- Nasdaq Dealer Market: Double counting.
- How are trades reported , an order of 100,000 shares will be broken into several trades.
- But will a trade against several limit orders reported for each limit order it hits?

Resources

- TAQ:

  nyse.com/publicdocs/nyse/data/Daily_TAQ_Client_Spec_v3.0d.pdf
- ODBC: en.wikipedia.org/wiki/Open_Database_Connectivity
- github.com
- rstudio.com
- Good video:

  rstudio.com/resources/webinars/working-with-big-data-in-r/