

CME Data for ORF474 High Frequency Trading

Robert Almgren*

Spring 2019

For our course, we have seven weeks of CME tick data from fall 2017. This data is recorded and processed by Quantitative Brokers, and is shared for educational purposes with special permission from CME Group. We host it in the Kdb+ (aka “Q”) database program, generously made available through an academic license from `kx.com`.

1 Server

To use the server, you need two pieces of software:

1. Client software for the Q database program, also known as Kdb+, from

`https://code.kx.com/q`

(the first link, “Download kdb+”).

2. The connector `rkdb` between the R statistical language and Q, from

`https://github.com/KxSystems/rkdb`

The installation process is fairly straightforward.

You can access our server at

	Host	<code>orf474.princeton.edu</code>
Interest Rate contracts on ports		<code>6000, ..., 6009</code>
Non-Rate contracts on ports		<code>7000, ..., 7009</code>

This is accessible only from within the Princeton network. From outside you need a VPN.

From a Q console, to access the server (that is a backtick in front of the dollar sign)

```
q)h: hopen `"$":orf474.princeton.edu:6000"
```

The dollar sign and the quotes are necessary because the server name has dashes in it. This creates a database handle `h` which you can use to run queries, for example

*ralmgren@princeton.edu

```
q)h"tables[]"
`instinfo`matchalgos`quote`trade`trdorders
```

Or from R, once you have the client software installed, you should be able to do

```
> db <- open_connection('orf474.princeton.edu', 6000)
> execute(db, 'tables[]')
[1] "instinfo" "matchalgos" "quote" "trade" "trdorders"
```

2 Data

The data we have comes from Quantitative Brokers' direct connection to CME. These are the exact databases that we use in our trading system and for our trading research. They include all top of book CME tick data for seven full weeks, from Monday Sep. 18 through Friday Nov. 3 2017, plus some reference data.

The tables on the server are

`instinfo` Reference information on futures products.

`matchalgos` Reference names for different match algorithms.

`trade` Trade data.

`quote` Top-of-book quote data.

`trdords` Breakdown of individual trades.

2.1 Reference tables

The `instinfo` table has reference information on the futures products in the trade and quote tables. Table 1 shows an extract from this table, produced by the query

```
`class`subclass`name xasc select from instinfo
```

This table does not include every single product in the tables, only the most significant ones. Here are descriptions of the fields:

`inst` The instrument name, in the CME Globex electronic trading system. Different symbols are sometimes used for floor trading or for clearing, but this is all that we need in our databases. "Instrument" denotes a broad product category. "Symbol" denotes a specific traded product, an instrument plus an expiration code. An expiration code is a one-letter month as in Table 2, plus the last digit of the expiration year. Thus ESZ7 would be the trading symbol for an SP500 futures contract (ES) whose expiration month is December (Z) 2017. The actual expiration date will generally be within the expiration month but not always: some of the energy contracts actually expire in the month before their expiration month. Details can be found on <http://www.cmegroup.com>. In Sept/Oct 2017, the Dec. contracts are the "front month" and will generally be the most actively traded.

Mapping from instrument to symbol is straightforward: append the expiration code. For the reverse, we provide the `sym2inst` dictionary: `sym2inst['ESZ7']` gives ``ES`.

inst	class	name	subclass	cur	notional	minpxincr	minpxincrvol	dispfactor	matchalgo
ZC	AG	Corn	Grain And Oilseed	USD	50	0.25	12.5	1	K
KE	AG	KC Hard Red Winter (HRW) Wheat	Grain And Oilseed	USD	50	0.25	12.5	1	O
ZS	AG	Soybean	Grain And Oilseed	USD	50	0.25	12.5	1	K
ZM	AG	Soybean Meal	Grain And Oilseed	USD	10	1	10	0.1	K
ZL	AG	Soybean Oil	Grain And Oilseed	USD	6	1	6	0.01	K
ZW	AG	Wheat	Grain And Oilseed	USD	50	0.25	12.5	1	K
GF	AG	Feeder Cattle	Livestock	USD	0.5	25	12.5	0.001	K
HE	AG	Lean Hog	Livestock	USD	0.4	25	10	0.001	K
LE	AG	Live Cattle	Livestock	USD	0.4	25	10	0.001	K
CL	EN	Crude Oil	Crude Oil	USD	10	1	10	0.01	F
CLT	EN	TAS Crude Oil	Crude Oil	USD	10	1	10	0.01	F
NG	EN	Natural Gas (Henry Hub) Physical	Natural Gas	USD	10	1	10	0.001	F
NGT	EN	TAS Natural Gas (Henry Hub) Physical	Natural Gas	USD	10	1	10	0.001	F
HO	EN	NY Harbor ULSD	Refined Products	USD	4.2	1	4.2	0.0001	F
RB	EN	RBOB Gasoline Physical	Refined Products	USD	4.2	1	4.2	0.0001	F
HOT	EN	TAS NY Harbor ULSD	Refined Products	USD	4.2	1	4.2	0.0001	F
RBT	EN	TAS RBOB Gasoline Physical	Refined Products	USD	4.2	1	4.2	0.0001	F
RTY	EQ	R2000 Ind Mini		USD	0.5	10	5	0.01	F
NKD	EQ	Nikkei/USD	International Index	USD	5	5	25	1	F
NIY	EQ	Nikkei/Yen	International Index	JPY	500	5	2500	1	F
YM	EQ	E-mini Dow (\$5)	US Index	USD	5	1	5	1	F
NQ	EQ	E-mini NASDAQ 100	US Index	USD	0.2	25	5	0.01	F
ES	EQ	E-mini S&P 500	US Index	USD	0.5	25	12.5	0.01	F
SMC	EQ	E-mini S&P 600 SmallCap	US Index	USD	1	10	10	0.01	F
EMD	EQ	E-mini S&P MidCap 400	US Index	USD	1	10	10	0.01	F
6M	FX	Mexican Peso	Emerging Market	USD	0.5	10	5	1e-06	F
6A	FX	Australian Dollar	Majors	USD	10	1	10	0.0001	F
6B	FX	British Pound	Majors	USD	6.25	1	6.25	0.0001	F
6C	FX	Canadian Dollar	Majors	USD	10	0.5	5	0.0001	F
6E	FX	Euro FX	Majors	USD	12.5	0.5	6.25	0.0001	F
6J	FX	Japanese Yen	Majors	USD	12.5	0.5	6.25	1e-06	F
6N	FX	New Zealand Dollar	Majors	USD	10	1	10	0.0001	F
6S	FX	Swiss Franc	Majors	USD	12.5	1	12.5	0.0001	F
...									

Table 1: The instinfo table.

F	G	H	J	K	M	N	Q	U	V	X	Z
Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec

Table 2: Futures expiration codes. These are fairly standard across futures exchanges. The highlighted quarterly maturities are the most active.

`class` The broad product category as in Table 3. Products of class IR are on the services accessed through ports 6000,...,6009; products of the other classes EQ, EN, FX, AG, and MT are on the services accessed through 7000,...,7009. The tables `instinfo` and `matchalgs` are identical across all services.

`name` The general name.

`subclass` The sub-category within `class`.

`cur` Currency in which the product is priced. This is almost always US dollar (USD).

`notional` The dollar (or `cur`) value associated with a unit change in the contract price, as reported in the market data (before multiplying by `minpxincrval`). For example, the notional of the Canadian dollar contract (6C) is \$10. If you buy one lot at 7940, and the closing price on that day is 7945, then the clearing house will send you $(7945 - 7940) \times \$10 = \50 . This is because each contract represents a position in 100,000 Canadian dollars¹, so a price change from US \$0.7940 to \$0.7945 per CAD has value $\$0.0005 \times 100,000 = \50 . The notional may be null for less active products for which we have not bothered to maintain the correct values.

`minpxincr` The minimum price increment or “tick” size; the discretization value of the price grid, in market data units. For example, the Canadian dollar contract has `minpxincr` = 0.5, so prices can be 7940, 7940.5, 7941, *etc.* A few specific maturities within a product class, or calendar spreads, may have smaller price increments. For example, near-term Eurodollar futures (GE) have tick size 0.25 rather than 0.5.

`minpxincrval` The dollar (or `cur`) value of `minpxincr`. It should always be that

$$\text{minpxincrval} = \text{notional} \times \text{minpxincr}$$

`dispfactor` The multiplier to be applied to price values in trade and quote data to get the actual contract price. This reduces the number of decimal points and leading zeros in market data. For example, the Japanese Yen contract (6J) has `dispfactor` = $1e-6$. A market data price of 9203 represents USD 0.009203 per yen, or 108.7 yen per USD.

`matchalgo` The match algorithm used for that product, at least for the outright contracts (calendar spreads can be different). The table `matchalgs` contains the description. To see how many products use each match algorithm, you can do

¹http://www.cmegroup.com/trading/fx/g10/canadian-dollar_contract_specifications.html

IR	EQ	EN	AG	FX	MT
Interest Rates	Equity Index	Energy	Agriculturals	Foreign Exchange	Metals

Table 3: Product classes

```

q)`n xdesc (select n:count i by matchalgo from instinfo) lj matchalgos
matchalgo| n   algoname
-----|-----
F         | 947 FIFO
K         | 145 Split FIFO and Pro-Rata
A         | 10  Allocation
O         | 2   Threshold Pro-Rata
Q         | 1   Threshold Pro-Rata with LMM
S         | 1   FIFO with Top Order and LMM
T         | 1   FIFO with LMM

```

Most products are F for pure time priority (FIFO, “first-in-first-out”), or K for mixed time priority/pro rata. Eurodollars (GE) have a pure pro rata match algorithm.

2.2 Trades and Quotes

These data are received by Quantitative Brokers from a direct market data feed, and hence are about the highest quality that it is possible to get. The tables are

quote Top-of-book quotes. These are actually produced from Level 2 data (quotes at each price level into the book) but this simplified form is enough for many purposes.

trade Trades. This table contains “aggregated” trades: when an aggressive market order matches several passive limit orders, it shows the aggressive order size.

trdords Shows the separate fills on the passive orders.

The following fields are common to trade and quote tables:

date Trading date. CME trading hours are from 5 PM Chicago time, through the night until 4 PM the next day. Trade date Wednesday August 30 contains data from 5 PM on Tuesday August 29 to 4 PM on Wednesday August 30.

sym The trading symbol, like ESZ7.

seq A sequence number, common between trades and quotes for the same date and symbol, but not between different symbols. This is the best way to sort trades and quotes into proper order. Sequence numbers are not consecutive in this top-of-book data; the missing numbers are updates to quotes deeper in the order book.

time The time stamp from the exchange match engine. CME documentation² says

²<https://www.cmegroup.com/confluence/display/EPICSANDBOX/MDP+3.0+-+Market+Data+Messaging>

q)h:hopen `"\$":orf474.princeton.edu:6005"

q)\c 72 120

q)h"6 # select from trade where date=2017.10.27,sym=`ZNZ7,time>=10:00:00"

date sym seq time prc siz aggr

2017.10.27	ZNZ7	11164931	0D10:00:00.000237107	124.65625	7	B
2017.10.27	ZNZ7	11165321	0D10:00:04.182057291	124.640625	10	S
2017.10.27	ZNZ7	11165494	0D10:00:06.501767397	124.640625	2	S
2017.10.27	ZNZ7	11165498	0D10:00:06.915766493	124.640625	5	S
2017.10.27	ZNZ7	11165520	0D10:00:07.649553269	124.640625	1	S
2017.10.27	ZNZ7	11165548	0D10:00:08.633265123	124.640625	5	S

q)h"6 # select from quote where date=2017.10.27,sym=`ZNZ7,time>=10:00:00"

date sym seq time bid ask bsiz asiz dbid dask dbsiz dasiz ibid iask ibsiz iasiz

2017.10.27	ZNZ7	11164933	0D10:00:00.000237107	124.640625	124.65625	1270	2431	124.640625	124.65625	1270	2422	124.625	124.65625	29	9
2017.10.27	ZNZ7	11164934	0D10:00:00.000535137	124.640625	124.65625	1270	2428	124.640625	124.65625	1270	2422	124.625	124.65625	29	6
2017.10.27	ZNZ7	11164936	0D10:00:00.042149149	124.640625	124.65625	1281	2428	124.640625	124.65625	1281	2422	124.625	124.65625	29	6
2017.10.27	ZNZ7	11164937	0D10:00:00.052252561	124.640625	124.65625	1281	2429	124.640625	124.65625	1281	2422	124.625	124.65625	29	7
2017.10.27	ZNZ7	11164941	0D10:00:00.062260843	124.640625	124.65625	1281	2428	124.640625	124.65625	1281	2422	124.625	124.65625	29	6
2017.10.27	ZNZ7	11164943	0D10:00:00.068048955	124.640625	124.65625	1281	2431	124.640625	124.65625	1281	2425	124.625	124.65625	29	6

q)h"tqmerge[2017.10.27; `ZNZ7; 10297170; 10297187]"

date sym seq time bsiz bid siz prc aggr ask asiz

2017.10.27	ZNZ7	10297171	0D07:54:40.365008139	2211	124.296875				124.3125	212					
2017.10.27	ZNZ7	10297173	0D07:54:40.365008525			50	124.3125	B							
2017.10.27	ZNZ7	10297175	0D07:54:40.365008525	2211	124.296875				124.3125	162					
2017.10.27	ZNZ7	10297176	0D07:54:40.365038857	2200	124.296875				124.3125	162					
2017.10.27	ZNZ7	10297178	0D07:54:40.365040309	2204	124.296875				124.3125	162					
2017.10.27	ZNZ7	10297180	0D07:54:40.365097753			99	124.3125	B							
2017.10.27	ZNZ7	10297182	0D07:54:40.365097753	2204	124.296875				124.3125	64					
2017.10.27	ZNZ7	10297183	0D07:54:40.365171693			20	124.3125	B							
2017.10.27	ZNZ7	10297186	0D07:54:40.365171693	2204	124.296875				124.3125	44					
2017.10.27	ZNZ7	10297187	0D07:54:40.365176223	2207	124.296875				124.3125	44					

Table 4: Example queries from the trade and quote tables.

Tag 60-TransactionTime All market data messages that are the result of a single incoming order action have consistent tag-60 TransactionTime values that represent the time CME Globex started processing the given event in nanosecond granularity.

In a production database system, timestamps would be UTC datetimes, to allow coordination of market data across time zones and daylight savings time changes. In this data set, for your convenience I have converted timestamps to Q timespan values, representing the time offset from midnight Chicago time. To see the range of data, do for example

```
q)h"select tmin:min time,tmax:max time from trade where date=2017.10.26"
tmin                tmax
-----
-0D06:59:59.999379153 0D15:59:59.866622693
```

The minimum time is just smaller than negative 7 hours, representing 24:00 – 7 hrs = 17:00 or 5 PM Chicago time on the previous day. The maximum time is just short of 16:00 or 4 PM Chicago time.

Fields specific to trade data are

prc Price of the trade. This should be multiplied by the corresponding **dispfactor** to get the actual contract price.

siz Size of the aggressive trade, if one aggressive order matches several limit orders.

aggr The aggressor side; that is, the side of the incoming order that generated the trade. Every trade has one or more buyers and one or more sellers. But if a buy market order matches against one or more resting sell limit orders, this field will show B.

Fields specific to quote data are

bid,ask Bid and ask prices, with the same multiplier as trade prices.

bsiz,asiz Bid and ask sizes.

There are some additional values in the quote table which we may talk about later. (Values with prefix **d** denote “direct” quotes, and values with prefix **i** denote “implied” quotes.)

Table 4 shows some sample queries from the **trade** and **quote** tables. These extract data for the December contract on the 10-year Treasury note ZNZ7 on Sep 27. At the top, we open a connection to one of the servers containing market data for interest rate products. The command **\c** sets the rows and columns of the console display. We specify “**time>=10:00:00**” to obtain data starting at 10 AM Chicago time, in the middle of the trading morning, to eliminate possible strange behavior around the market open at 5 PM the previous evening. The modifier “**6 #**” gives us the first 6 rows of the result set.

For your convenience in matching trades and quotes, I provide the function **tqmerge**:

```
{[d;s;l;sR] `seq xasc
  (select date,sym,seq,time,bsiz:0Ni,bid:0n,siz,prc,aggr,ask:0n,asiz:0Ni from trade
    where date=d,sym=s,seq within (sL;sR)),
  `date`sym`seq`time xcols 0!
  select last seq,time,last bsiz,last bid,siz:0Ni,prc:0n,aggr:`,last ask,last asiz
    by date,sym,time from quote
    where date=d,sym=s,seq within (sL;sR)};
```

For a given date, symbol, and minimum and maximum sequence numbers, this extracts trade data in that sequence number range, concatenates it with quote data from the same range (the comma , concatenates), and sorts the result in order by sequence number. Fields that do not exist in each table are filled with nulls, so that the concatenated tables have the same structure. You can see that time is identical for a trade, and the following quote update. You can also see that quote sizes often decrease by the size of a trade. But often the decrease is smaller than the trade size; this is due to hidden liquidity which we shall talk about at some point.

2.3 Order decomposition

The `trdords` table shows the separate passive fills for a given aggressive order. As an example, consider the 99-lot trade shown in Table 4. To see the breakdown of this order, select from `trdords` with the same date, symbol, and sequence number:

```
q)h"select from trade where date=2017.10.27,sym=`ZNZ7`,seq=10297180"
date      sym  seq      time      prc      siz aggr
```

```
-----
2017.10.27 ZNZ7 10297180 0D07:54:40.365097753 124.3125 99 B
```

```
q)h"select from trdords where date=2017.10.27,sym=`ZNZ7`,seq=10297180"
date      sym  seq      prc      siz ordid      is_aggr aggr
```

```
-----
2017.10.27 ZNZ7 10297180 124.3125 99 844620770303 1 B
2017.10.27 ZNZ7 10297180 124.3125 9 844620770077 0 B
2017.10.27 ZNZ7 10297180 124.3125 15 844620770080 0 B
2017.10.27 ZNZ7 10297180 124.3125 1 844620770082 0 B
2017.10.27 ZNZ7 10297180 124.3125 1 844620770090 0 B
2017.10.27 ZNZ7 10297180 124.3125 73 844620770094 0 B
```

The first row in each section of `trdords` always matches the trade shown in `trade`. The remaining rows show the breakdown into passive fills. In this case, the 99-lot aggressive order matched against 5 different resting orders (there are 6 rows, of which the first is the total fill). The passive fill sizes were 9, 15, 1, 1, 73. The prices and aggressiveness in `trdords` are always the same and the same as the price and aggressiveness in `trade`.

The `ordid` field in `trdords` can be used to track fills of a specific resting limit order. From the above example, looking at the fills for `ordid` 844620770077:

```
q)h"(select from trdords where date=2017.10.27,sym=`ZNZ7`,ordid=844620770077) lj
      select first time by date,sym,seq from trade where date=2017.10.27,sym=`ZNZ7"
date      sym  seq      prc      siz ordid      is_aggr aggr time
```

```
-----
2017.10.27 ZNZ7 10297173 124.3125 11 844620770077 0 B 0D07:54:40.365008525
2017.10.27 ZNZ7 10297180 124.3125 9 844620770077 0 B 0D07:54:40.365097753
```

This passive sell order participated in two different trades:

```
q)h"select from trade where date=2017.10.27,sym=`ZNZ7`,seq within 10297173 10297180"
date      sym  seq      time      prc      siz aggr
```

```
-----
2017.10.27 ZNZ7 10297173 0D07:54:40.365008525 124.3125 50 B
2017.10.27 ZNZ7 10297180 0D07:54:40.365097753 124.3125 99 B
```

First, an aggressive buy order for 50 lots filled 11 lots of this passive sell order. Next, another aggressive buy order of 99 lots filled the remaining 9 lots of this order.

What is missing in this data set is the corresponding information for quotes. We do not know when order 844620770077 was entered, what was its initial size, or whether it was modified or cancelled during its lifetime. We only know that it was filled for a total of $11 + 9 = 20$ lots. We may suppose that the last fill above completed this order, but it may have been incompletely filled and cancelled after the last fill we see. That additional information is available in CME market data, but we have not included it in the course databases because of its size and complexity.

Each query in the above includes the clause

```
where date=2017.10.27,sym=`ZNZ7
```

This is essential, since sequence numbers are not consistent across different dates or symbols (order ID's may be but let us not assume that).

3 Overall statistics

Figure 1 shows the overall number of trade and quote records per day. The numbers are quite substantial.

Figure 2 shows traded volume for different symbols of two specific instruments: 10-year Treasury and Crude Oil.

- For the Treasury futures, the December contract ZNZ7 is the front month through the entire period. (The September contract expires near the end of September, but its trading volume is negligible following the roll period at the end of August.)
- For Crude Oil, the November contract (CLX7) expires in the middle of October, and volume shifts to the December contract (CLZ7).

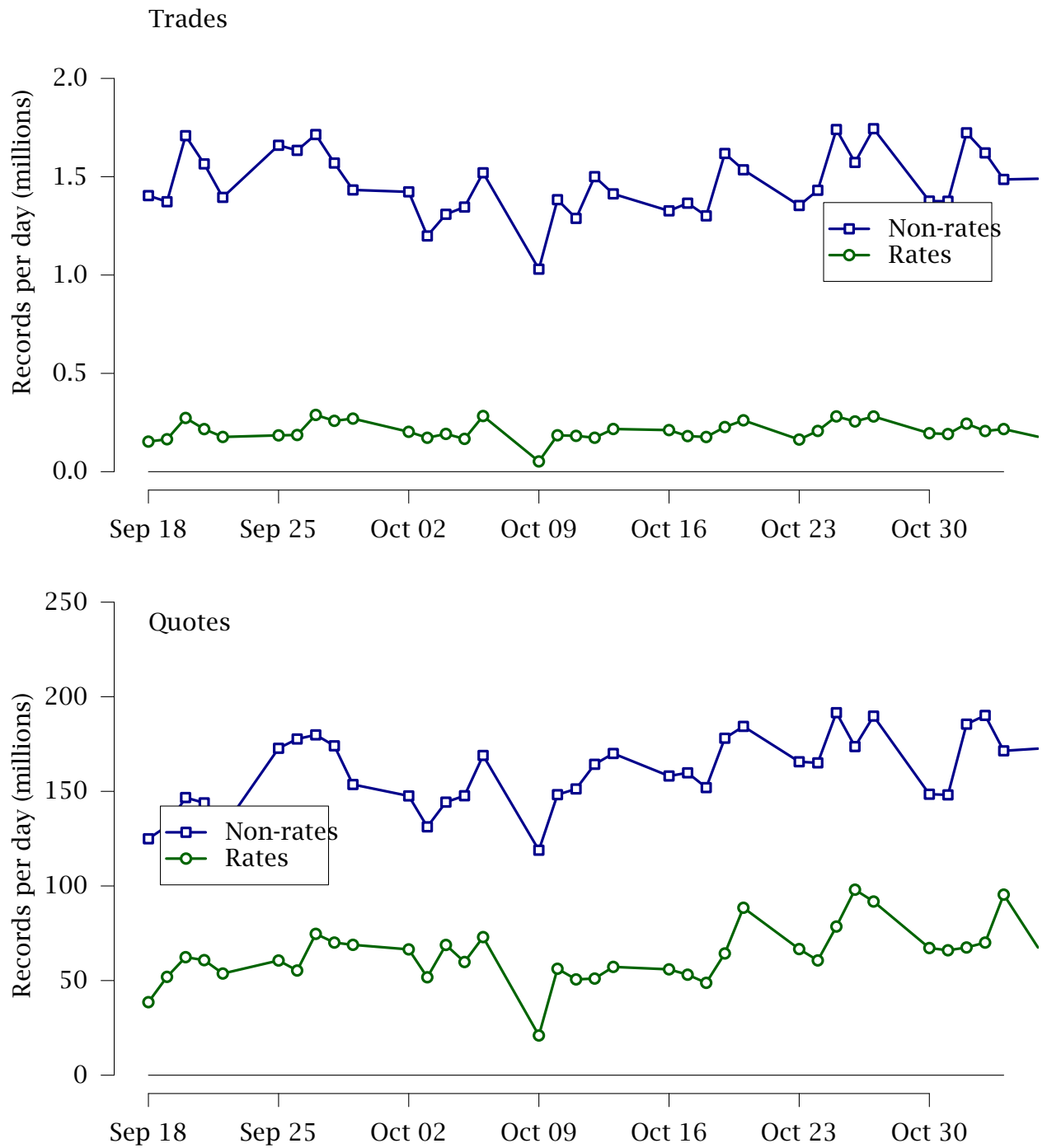


Figure 1: Number of trade and quote records per day, in Sep-Oct 2017. There are about 150 million quote records per day for non-rates, and about 50 million for rates, which requires a fairly high-performance database. The number of trades is much smaller.

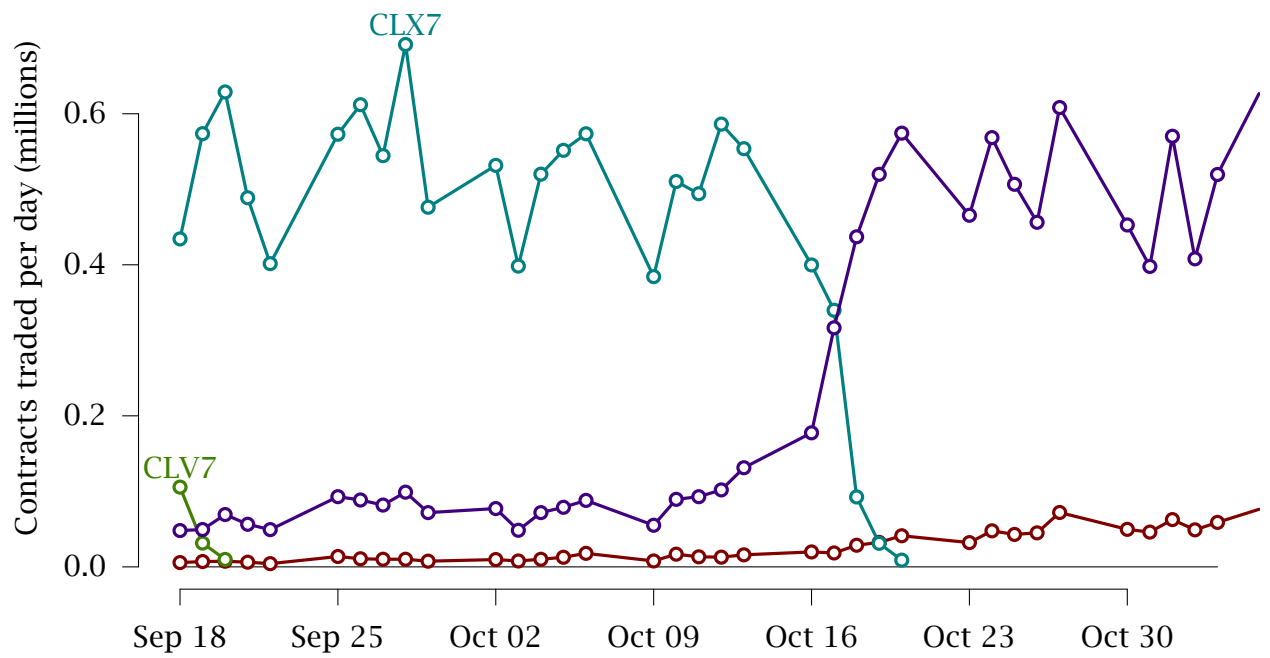
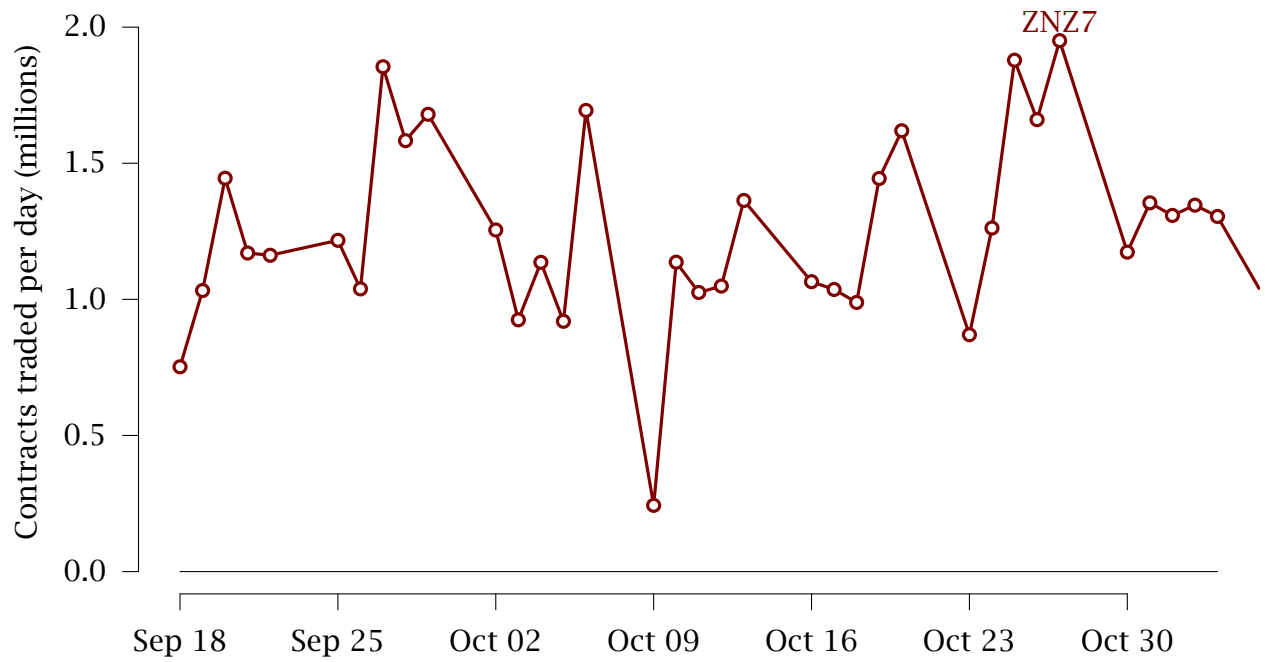


Figure 2: Trade volumes, in millions of lots per day, for two typical contracts: 10-year Treasury (ZN) and Crude Oil (CL).