Minutely Stats - mstats

What is mstats?

From market data(trades and quotes), we calculate stats by **sym,minute (1 minute)**
These stats can be used for any derived analytics, instead of having to refer the entire marketdata DB every time a stat needs to be calculated

The result is a table that is stored in derived DB and available in

```
getderiveddata[`prod;`cme]"select from mstats where date=2021.08.26"
```

> mstats contains data for all instruments available in qbtradedinsts, for all ctypes(even the ctypes that are not part of qbtradedinsts)
> That is, restriction is only on inst level and not ctype level.
> E.g: For CL, we have data for OT SP BF. But we trade only OT,SP.
> You can find mstats for all ctypes of CL (OT,SP,BF)

How to query mstats[click here]?

Data availability in Database

- mstats table is available from 2021.08.02(Aug 2021) in sim & prod
- rtvol column was added to mstats and is available from 2022.05.23 in sim & 2022.06.02 in prod. Older days will have null rtvol(Some days older than above dates can have rtvol value in case mstats was rerun to fix a different issue)

Keynote: Mstats & Liquidity Metrics

Liquidity Metric… ibi.pdf
27 Aug 2021, 10:13 AM

Table Schema

Table column summary

Table Column	Name	Description	Method
sym	symbol		
time	time		
exchangetime	exchangetime		
open	open	first trade price in a bin	
high	high	highest trade price in a bin	
low	low	lowest trade price in a bin	
close	close	last trade price in a bin	
firstBid	bid at first trade	bid as of first trade in a bin	
firstAsk	ask at first trade	ask as of first trade in a bin	
lastBid	bid at last trade	bid as of last trade in a bin	
lastAsk	ask at last trade	ask as of last trade in a bin	
volume	volume	total volume in a bin	
numTrades	no. of trades	total no. of trades in a bin	
avgSize	average size	average trade size	
vwap	volume weighted average price	vwap of bin	
aggBuyVol	aggressive buy volume	total volume where aggress=`B	
aggSellVol	aggressive sell volume	total volume where aggress=`S	
depth	depth	average depth in a bin	depth of a single trade= quotesize as of trade / trade siz
numRndLots	no. of round lots	no. of round lot trades in a bin	
numContPx	no. of price continuations	no. of price continuations in a bin	If consecutive price changes are in same direction => continuation E.g: 98 99 99.5
numAltPx	no. of price alterations	no. of price alterations in a bin	If consecutive price changes are in opposite direction => alternating E.g: 98 99 98
avgMid	average mid price	average mid price of a bin	
avgQuotesize	average quotesize	average quotesize in a bin	
numQuotes	no. of quotes	total no. of quotes in a bin	
avgSpread	average spread	average bid-ask spread in a bin	

openBid	open bid	first bid in a bin	
openAsk	open ask	first ask in a bin	
highBid	high bid	highest bid in a bin	
highAsk	high ask	highest ask in a bin	
lowBid	low bid	lowest bid in a bin	
lowAsk	low ask	lowest ask in a bin	
closeBid	close bid	last bid in a bin	
closeAsk	close ask	last ask in a bin	
prevBid	last known bid	last known bid that existed before this bin	
prevAsk	last known ask	last known ask that existed before this bin	
variance	variance	price variance in a bin	
priceImpact1	price impact 1	average price impact in a bin	price impact of single trade = (midprice after PIsecs of trade) - (mid price as of trade)
priceImpact2	price impact 2	price impact weighed by size	
DpprevDP	temporary value	sum (Dp * prevDp) in a bin where Dp = Change in trade price prevDp = prev of the series Dp	
rtvol	real-time volatility	real time volatility generated using a vectorized logic of the feature rtvol in real_time_variance.q	rtvol value in each mstats bin depicts the last value of rtvol for that minute’s bin

How is mstats generated?

The minutely stats are generated by kdb/coreprocs/stats/qcode/mstats.q
The helper funcs are defined in kdb/coreprocs/stats/qcode/mstats_funcs.q
The part of code that helps define the cfg_tbl and builds the query_tbl is defined in qutil.q

Each row in cfg_tbl corresponds to an update operation to calculate the stat

For e.g:

Line 68 translates to:

```
update highBid:max bid by sym,0D00:01 xbar rcvtime from bq
```

Line 53 translates to:

```
update Deltap:(prc-prev prc) by sym from bq
```

Iteration:

update operations are performed in batches or iterations.
All stats in iteration 1 will be done first in one query.
For e.g: open high low close stats here are calculated in one go, hence belong to same iteration

```
update open:first prc,high:max prc,low:min prc,close:last prc by sym,0D:00:01 xbar rcvtime from tr
```

Stats which depend on other stats should be placed in iterations higher than the ones they depend on

Logic for some not-straight forwards stats:

Average quote size:

In general, average quotesize is avg 0.5*(bsiz+asiz)
During bins which don’t have any quotesize value, we need to extrapolate the last known quotesize into the bin and then calculate the average quotesize for those empty bins

Hence we use this function to use the last known quotesize and closing quotesize in each bin to fill the bins which don’t have any quotesize changes

Idea is that, in a bin with no quotesizes, the average quotesize is same as last known quotesize

Let us assume that we have a symbol for which there are no quotes for sometime after the open and before the close. Then we need to fill those bins.

The empty bins near close - get the last known closing quotesize from the last non-empty bin

The empty bins near open - get the last known quotesize from the last quote that existed before the open

The same filling logic is used to calculate Average spread as well