algoseek_minute_data

September 29, 2021

1 Processing Algoseek's Trade & Quote Minute Bar data

In this notebook, we load the high-quality NASDAQ100 minute-bar trade-and-quote data generously provided by Algoseek (available here) that we will use in Chapter 12 to develop an intraday trading strategy.

1.1 Imports & Settings

[1]:

```
import warnings

warnings.filterwarnings('ignore')
```

[2]:

```
% matplotlib inline

from pathlib import Path
from tqdm import tqdm

import numpy as np
import pandas as pd

import matplotlib.pyplot as plt
import seaborn as sns
```

[3]:

```
sns.set_style('whitegrid')
idx = pd.IndexSlice
```

1.2 Algoseek Trade & Quote Minute Bar Data

1.2.1 Data Dictionary

The Quote fields are based on changes to the NBBO (National Best Bid Offer) from the top-of-book price and size from each of the exchanges.

The enhanced Trade & Quote bar fields include the following fields: - **Field**: Name of Field. - **Q / T**: Field based on Quotes or Trades - **Type**: Field format - **No Value**: Value of field when there is no value or data. - Note: "Never" means field should always have a value EXCEPT for the first bar of the day. - **Description**: Description of the field.

id	Field	Q/T	Type	No Value	Description
1	`Date`		YYYYMMDD	Never	Trade Date
2	`Ticker`		String	Never	Ticker Symbol
3	`TimeBarStart`		HHMM HHMMSS HHMMSSMMM	Never	For minute bars: HHMM. For second bars: HHMMSS. Examples- One second bar 130302 is from time greater than 130301 to 130302.- One minute bar 1104 is from time greater than 1103 to 1104.
4	`OpenBarTime`	Q	HHMMSSMMM	Never	Open Time of the Bar, for example one minute:11:03:00.000
5	`OpenBidPrice`	Q	Number	Never	NBBO Bid Price as of bar Open
6	`OpenBidSize`	Q	Number	Never	Total Size from all Exchanges withOpenBidPrice
7	`OpenAskPrice`	Q	Number	Never	NBBO Ask Price as of bar Open
8	`OpenAskSize`	Q	Number	Never	Total Size from all Exchange withOpenAskPrice
9	`FirstTradeTime`	T	HHMMSSMMM	Blank	Time of first Trade
10	`FirstTradePrice`	T	Number	Blank	Price of first Trade
11	`FirstTradeSize`	T	Number	Blank	Number of shares of first trade
12	`HighBidTime`	Q	HHMMSSMMM	Never	Time of highest NBBO Bid Price
13	`HighBidPrice`	Q	Number	Never	Highest NBBO Bid Price
14	`HighBidSize`	Q	Number	Never	Total Size from all Exchanges with HighBidPrice
15	`AskPriceAtHighBidPrice`	Q	Number	Never	Ask Price at time of Highest Bid Price
16	`AskSizeAtHighBidPrice`	Q	Number	Never	Total Size from all Exchanges with `AskPriceAtHighBidPrice`
17	`HighTradeTime`	T	HHMMSSMMM	Blank	Time of Highest Trade
18	`HighTradePrice`	T	Number	Blank	Price of highest Trade
19	`HighTradeSize`	T	Number	Blank	Number of shares of highest trade
20	`LowBidTime`	Q	HHMMSSMMM	Never	Time of lowest Bid
21	`LowBidPrice`	Q	Number	Never	Lowest NBBO Bid price of bar.
22	`LowBidSize`	Q	Number	Never	Total Size from all Exchanges with `LowBidPrice`
23	`AskPriceAtLowBidPrice`	Q	Number	Never	Ask Price at lowest Bid price
24	`AskSizeAtLowBidPrice`	Q	Number	Never	Total Size from all Exchanges with `AskPriceAtLowBidPrice`
25	`LowTradeTime`	T	HHMMSSMMM	Blank	Time of lowest Trade
26	`LowTradePrice`	T	Number	Blank	Price of lowest Trade
27	`LowTradeSize`	T	Number	Blank	Number of shares of lowest trade
28	`CloseBarTime`	Q	HHMMSSMMM	Never	Close Time of the Bar, for example one minute: 11:03:59.999
29	`CloseBidPrice`	Q	Number	Never	NBBO Bid Price at bar Close
30	`CloseBidSize`	Q	Number	Never	Total Size from all Exchange with `CloseBidPrice`
31	`CloseAskPrice`	Q	Number	Never	NBBO Ask Price at bar Close
32	`CloseAskSize`	Q	Number	Never	Total Size from all Exchange with `CloseAskPrice`
33	`LastTradeTime`	T	HHMMSSMMM	Blank	Time of last Trade
34	`LastTradePrice`	T	Number	Blank	Price of last Trade
35	`LastTradeSize`	T	Number	Blank	Number of shares of last trade
36	`MinSpread`	Q	Number	Never	Minimum Bid-Ask spread size. This may be 0 if the market was crossed during the bar.If negative spread due to back quote, make it 0.
37	`MaxSpread`	Q	Number	Never	Maximum Bid-Ask spread in bar

id	Field	Q/T	Type	No Value	Description
38	CancelSize	T	Number	0	Total shares canceled. Default=blank
39	VolumeWeightPrice	T	Number	Blank	Trade Volume weighted average price Sum((Trade1Shares*Price)+(Trade2Shares*Price)+…)/TotalShares. Note: Blank if no trades.
40	NBBOQuoteCount	Q	Number	0	Number of Bid and Ask NNBO quotes during bar period.
41	TradeAtBid	Q,T	Number	0	Sum of trade volume that occurred at or below the bid (a trade reported/printed late can be below current bid).
42	TradeAtBidMid	Q,T	Number	0	Sum of trade volume that occurred between the bid and the mid-point:(Trade Price > NBBO Bid) & (Trade Price < NBBO Mid)
43	TradeAtMid	Q,T	Number	0	Sum of trade volume that occurred at mid.TradePrice = NBBO MidPoint
44	TradeAtMidAsk	Q,T	Number	0	Sum of ask volume that occurred between the mid and ask:(Trade Price > NBBO Mid) & (Trade Price < NBBO Ask)
45	TradeAtAsk	Q,T	Number	0	Sum of trade volume that occurred at or above the Ask.
46	TradeAtCrossOrLocked	Q,T	Number	0	Sum of trade volume for bar when national best bid/offer is locked or crossed. Locked is Bid = Ask Crossed is Bid > Ask
47	Volume	T	Number	0	Total number of shares traded
48	TotalTrades	T	Number	0	Total number of trades
49	FinraVolume	T	Number	0	Number of shares traded that are reported by FINRA. Trades reported by FINRA are from broker-dealer internalization, dark pools, Over-The-Counter, etc. FINRA trades represent volume that is hidden or not public available to trade.
50	UptickVolume	T	Integer	0	Total number of shares traded with upticks during bar.An uptick = (trade price > last trade price)
51	DowntickVolume	T	Integer	0	Total number of shares traded with downticks during bar.A downtick = (trade price < last trade price)
52	RepeatUptickVolume	T	Integer	0	Total number of shares where trade price is the same (repeated) and last price change was up during bar. Repeat uptick = (trade price == last trade price) & (last tick direction == up)
53	RepeatDowntickVolume	T	Integer	0	Total number of shares where trade price is the same (repeated) and last price change was down during bar. Repeat downtick = (trade price == last trade price) & (last tick direction == down)
54	UnknownVolume	T	Integer	0	When the first trade of the day takes place, the tick direction is “unknown” as there is no previous Trade to compare it to.This field is the volume of the first trade after 4am and acts as an initiation value for the tick volume directions.In future this bar will be renamed to UnkownTickDirectionVolume .

1.2.2 Notes

Empty Fields

An empty field has no value and is “Blank” , for example FirstTradeTime and there are no trades during the bar period. The field `Volume` measuring total number of shares traded in bar will be `0` if there are no Trades (see `No Value` column above for each field).

No Bid/Ask/Trade OHLC

During a bar timeframe there may not be a change in the NBBO or an actual Trade. For example, there can be a bar with OHLC Bid/Ask but no Trade OHLC.

Single Event

For bars with only one trade, one NBBO bid or one NBBO ask then Open/High/Low/Close price,size andtime will be the same.

`AskPriceAtHighBidPrice`, `AskSizeAtHighBidPrice`, `AskPriceAtLowBidPrice`, `AskSizeAtLowBidPrice` Fields

To provide consistent Bid/Ask prices at a point in time while showing the low/high Bid/Ask for the bar, AlgoSeek uses the low/high `Bid` and the corresponding `Ask` at that price.

1.2.3 FAQ

Why are Trade Prices often inside the Bid Price to Ask Price range?

The Low/High Bid/Ask is the low and high NBBO price for the bar range. Very often a Trade may not occur at these prices as the price may only last a few seconds or executions are being crossed at mid-point due to hidden order types that execute at mid-point or as price improvement over current `Bid/Ask`.

How to get exchange tradable shares?

To get the exchange tradable volume in a bar subtract `Volume` from `FinraVolume`. - `Volume` is the total number of shares traded. - `FinraVolume` is the total number of shares traded that are reported as executions by FINRA.

When a trade is done that is off the listed exchanges, it must be reported to FINRA by the brokerage firm or dark pool. Examples include: - internal crosses by broker dealer - over-the-counter block trades, and - dark pool executions.

1.3 Data prep

We use the ‘Trade and Quote’ dataset - see documentation for details on the definition of the numerous fields.

[5]:
```
tcols = ['openbartime',
         'firsttradetime',
         'highbidtime',
         'highasktime',
         'hightradetime',
         'lowbidtime',
         'lowasktime',
         'lowtradetime',
         'closebartime',
         'lasttradetime']
```

[6]:
```
drop_cols = ['unknowntickvolume',
             'cancelsize',
             'tradeatcrossorlocked']
```

[7]:
```
keep = ['firsttradeprice',
        'hightradeprice',
        'lowtradeprice',
        'lasttradeprice',
        'minspread',
        'maxspread',
        'volumeweightprice',
        'nbboquotecount',
        'tradeatbid',
        'tradeatbidmid',
        'tradeatmid',
        'tradeatmidask',
        'tradeatask',
        'volume',
        'totaltrades',
        'finravolume',
        'finravolumeweightprice',
        'uptickvolume',
        'downtickvolume',
        'repeatuptickvolume',
        'repeatdowntickvolume',
        'tradetomidvolweight',
        'tradetomidvolweightrelative']
```

We will shorten most of the field names to reduce typing:

[8]:
```
columns = {'volumeweightprice': 'price',
           'finravolume': 'fvolume',
           'finravolumeweightprice': 'fprice',
           'uptickvolume': 'up',
           'downtickvolume': 'down',
           'repeatuptickvolume': 'rup',
           'repeatdowntickvolume': 'rdown',
           'firsttradeprice': 'first',
           'hightradeprice': 'high',
           'lowtradeprice': 'low',
           'lasttradeprice': 'last',
           'nbboquotecount': 'nbbo',
           'totaltrades': 'ntrades',
           'openbidprice': 'obprice',
           'openbidsize': 'obsize',
           'openaskprice': 'oaprice',
           'openasksize': 'oasize',
           'highbidprice': 'hbprice',
           'highbidsize': 'hbsize',
           'highaskprice': 'haprice',
           'highasksize': 'hasize',
           'lowbidprice': 'lbprice',
           'lowbidsize': 'lbsize',
           'lowaskprice': 'laprice',
           'lowasksize': 'lasize',
           'closebidprice': 'cbprice',
           'closebidsize': 'cbsize',
           'closeaskprice': 'caprice',
           'closeasksize': 'casize',
           'firsttradesize': 'firstsize',
           'hightradesize': 'highsize',
           'lowtradesize': 'lowsize',
           'lasttradesize': 'lastsize',
           'tradetomidvolweight': 'volweight',
           'tradetomidvolweightrelative': 'volweightrel'}
```

The Algoseek minute-bar data comes in compressed csv files that contain the data for one symbol and day, organized in three directories for each year (2015-17). The function **extract_and_combine_data** reads the ~80K source files and combines them into a single **hdf5** file for faster access.

> The data is fairly large (>8GB), and if you run into memory constraints, please modify the code to process the data in smaller chunks. One options is to iterate over the three directories containing data for a single year only, and storing each year separately.

[4]:

```
nasdaq_path = Path('../../data/nasdaq100')
```

[14]:

```
def extract_and_combine_data():
    path = nasdaq_path / '1min_taq'
    if not path.exists():
        path.mkdir(parents=True)

    data = []
    # ~80K files to process
    for f in tqdm(list(path.glob('*/**/*.csv.gz'))):
        data.append(pd.read_csv(f, parse_dates=[['Date', 'TimeBarStart']])
                    .rename(columns=str.lower)
                    .drop(tcols + drop_cols, axis=1)
                    .rename(columns=columns)
                    .set_index('date_timebarstart')
                    .sort_index()
                    .between_time('9:30', '16:00')
                    .set_index('ticker', append=True)
                    .swaplevel()
                    .rename(columns=lambda x: x.replace('tradeat', 'at')))
    data = pd.concat(data).apply(pd.to_numeric, downcast='integer')
    data.index.rename(['ticker', 'date_time'], inplace=True)
    print(data.info(show_counts=True))
    data.to_hdf(nasdaq_path / 'algoseek.h5', 'min_taq')
```

[15]:

```
extract_and_combine_data()
```

```
80194it [20:55, 63.87it/s]
```

```
<class 'pandas.core.frame.DataFrame'>
MultiIndex: 31355463 entries, ('MSFT', Timestamp('2015-02-09 09:30:00')) to
('DISH', Timestamp('2016-10-11 16:00:00'))
Data columns (total 45 columns):
 #   Column          Non-Null Count     Dtype
---  ------          --------------     -----
 0   obprice         31355451 non-null  float64
 1   obsize          31355451 non-null  float64
 2   oaprice         31355457 non-null  float64
 3   oasize          31355457 non-null  float64
 4   first           30955838 non-null  float64
 5   firstsize       30955838 non-null  float64
 6   hbprice         31355463 non-null  float64
 7   hbsize          31355463 non-null  int32
 8   haprice         31355463 non-null  float64
 9   hasize          31355463 non-null  int32
 10  high            30955838 non-null  float64
 11  highsize        30955838 non-null  float64
 12  lbprice         31355463 non-null  float64
 13  lbsize          31355463 non-null  int32
 14  laprice         31355463 non-null  float64
 15  lasize          31355463 non-null  int32
 16  low             30955838 non-null  float64
 17  lowsize         30955838 non-null  float64
 18  cbprice         31355463 non-null  float64
 19  cbsize          31355463 non-null  int32
 20  caprice         31355463 non-null  float64
 21  casize          31355463 non-null  int32
 22  last            30955838 non-null  float64
 23  lastsize        30955838 non-null  float64
 24  minspread       31354810 non-null  float64
 25  maxspread       31355327 non-null  float64
 26  price           30386944 non-null  float64
 27  nbbo            31355463 non-null  int32
 28  atbid          31355463 non-null  int32
 29  atbidmid       31355463 non-null  int32
 30  atmid          31355463 non-null  int32
 31  atmidask       31355463 non-null  int32
 32  atask          31355463 non-null  int32
 33  volume         31355463 non-null  int32
 34  ntrades        31355463 non-null  int16
 35  fvolume        31355463 non-null  int32
 36  fprice         29561289 non-null  float64
 37  up             31355463 non-null  int32
 38  down           31355463 non-null  int32
 39  rup            31355463 non-null  int32
 40  rdown          31355463 non-null  int32
 41  volweight      30386944 non-null  float64
 42  volweightrel   30386944 non-null  float64
 43  timeweightbid  31355463 non-null  float64
 44  timeweightask  31355463 non-null  float64
dtypes: float64(26), int16(1), int32(18)
memory usage: 8.4+ GB
None
```

[9]:

```
df = pd.read_hdf(nasdaq_path / 'algoseek.h5', 'min_taq')
```

[10]:

```
df.info(null_counts=True)
```

```
<class 'pandas.core.frame.DataFrame'>
MultiIndex: 31355463 entries, ('MSFT', Timestamp('2015-02-09 09:30:00')) to
('DISH', Timestamp('2016-10-11 16:00:00'))
Data columns (total 45 columns):
 #   Column         Non-Null Count     Dtype
---  ------         --------------     -----
 0   obprice        31355451 non-null  float64
 1   obsize         31355451 non-null  float64
 2   oaprice        31355457 non-null  float64
 3   oasize         31355457 non-null  float64
 4   first          30955838 non-null  float64
 5   firstsize      30955838 non-null  float64
 6   hbprice        31355463 non-null  float64
 7   hbsize         31355463 non-null  int32
 8   haprice        31355463 non-null  float64
 9   hasize         31355463 non-null  int32
 10  high           30955838 non-null  float64
 11  highsize       30955838 non-null  float64
 12  lbprice        31355463 non-null  float64
 13  lbsize         31355463 non-null  int32
 14  laprice        31355463 non-null  float64
 15  lasize         31355463 non-null  int32
 16  low            30955838 non-null  float64
 17  lowsize         30955838 non-null  float64
 18  cbprice         31355463 non-null  float64
 19  cbsize          31355463 non-null  int32
 20  caprice         31355463 non-null  float64
 21  casize          31355463 non-null  int32
 22  last            30955838 non-null  float64
 23  lastsize        30955838 non-null  float64
 24  minspread       31354810 non-null  float64
 25  maxspread       31355327 non-null  float64
 26  price           30386944 non-null  float64
 27  nbbo            31355463 non-null  int32
 28  atbid           31355463 non-null  int32
 29  atbidmid        31355463 non-null  int32
 30  atmid           31355463 non-null  int32
 31  atmidask        31355463 non-null  int32
 32  atask           31355463 non-null  int32
 33  volume          31355463 non-null  int32
 34  ntrades         31355463 non-null  int16
 35  fvolume         31355463 non-null  int32
 36  fprice          29561289 non-null  float64
 37  up              31355463 non-null  int32
 38  down            31355463 non-null  int32
 39  rup             31355463 non-null  int32
 40  rdown           31355463 non-null  int32
 41  volweight       30386944 non-null  float64
 42  volweightrel    30386944 non-null  float64
 43  timeweightbid   31355463 non-null  float64
 44  timeweightask   31355463 non-null  float64
dtypes: float64(26), int16(1), int32(18)
memory usage: 8.4+ GB
```

1.4 NASDAQ 100 Constituents

The dataset contains 142 stocks because there were multiple changes to index membership over the 2015-17 period:

[11]:

```
len(df.index.unique('ticker'))
```

[11]:

```
142
```

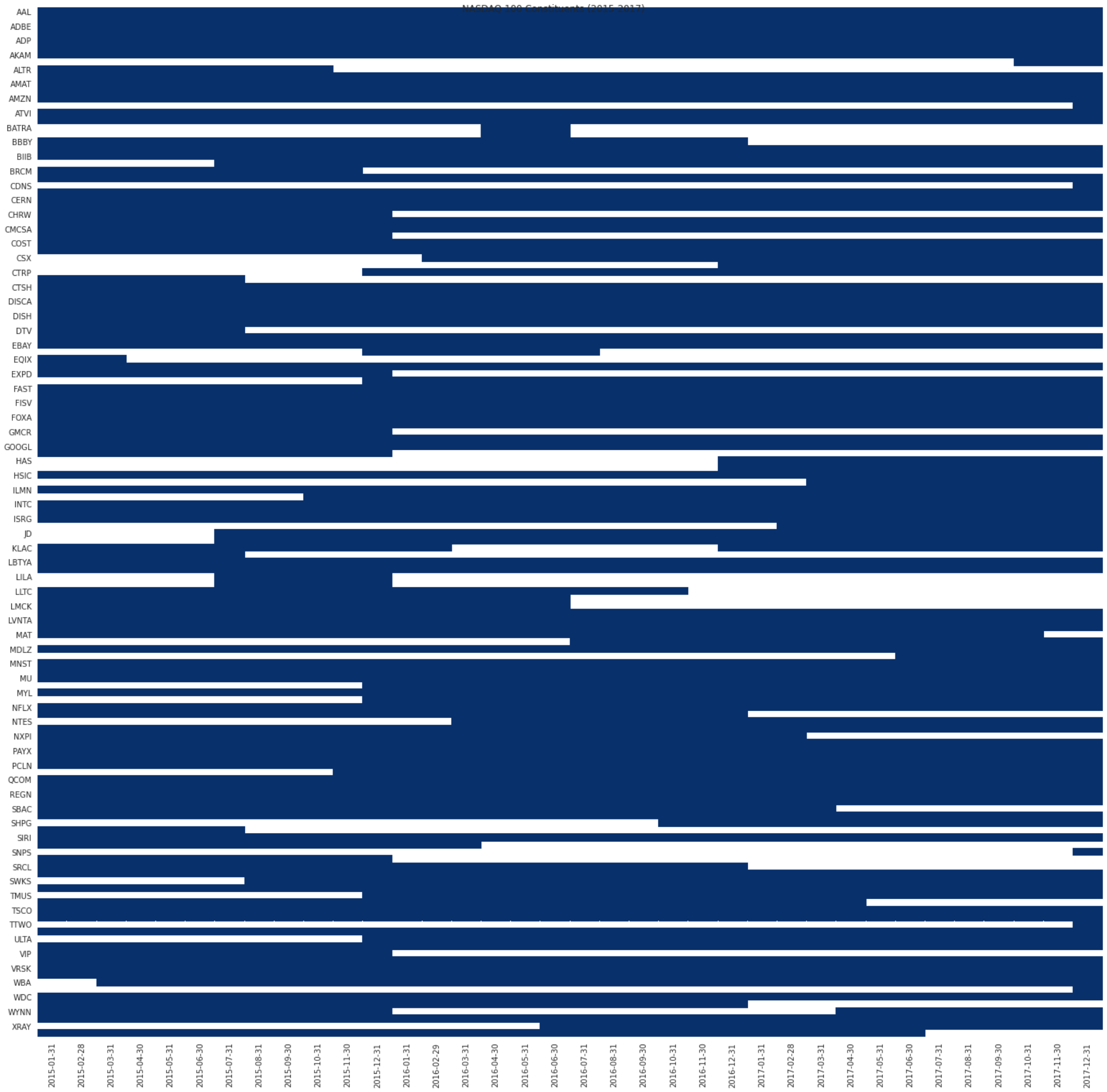
The below heatmap highlights the frequent entry/exit points of various securities, which emphasizes the need for a survivorship-free dataset.

[53]:

```
constituents = (df.groupby([df.index.get_level_values('date_time').date,␣
 ↪'ticker'])
                .size()
                .unstack('ticker')
                .notnull()
                .astype(int)
                .replace(0, np.nan))

constituents.index = pd.to_datetime(constituents.index)
constituents = constituents.resample('M').max()
constituents.index = constituents.index.date
```

[63]:
```
fig, ax = plt.subplots(figsize=(20, 20))
mask = constituents.T.isnull()
ax = sns.heatmap(constituents.T, mask=mask, cbar=False, ax=ax, cmap='Blues_r')
ax.set_ylabel('')
fig.suptitle('NASDAQ100 Constituents (2015-2017)')
fig.tight_layout();
```



[ ]:

```
```