Usage Docs

Goal

The goal of this project is to consume a trade blotter for a portfolio and calculate various financial metrics, chiefly the profit made (or loss taken).

Input Data

The format of the blotter is a CSV file that contains the following fields:

```
LocalTime,Symbol,EventType,Side,FillSize,FillPrice,FillExchange
9:30:00.000,AES,TRADE,t,100,11.14,NYSE
9:30:01.000,AES,TRADE,b,100,11.16,NASDAQ
```

When data corruption happens, e.g. missing, or wrong values, my algorithm will skip that row entirely.

Trading Stats Computes

My algorithm will compute the following stats in a data stream fashion, namely on a trade-by-trade basis with the results append on the same row.

a. SymbolBought
   - Number of shares of the stock bought
b. SymbolSold
   - Number of shares of the stock sold
c. SymbolPosition
   - Number of shares of the stock that the portfolio currently holds or owes
   - Dependent on the number of shares we bought vs. sold, the position can be:
     i. Long (positive number) if we bought more shares than we sold
     ii. Short (negative number) if we sold more than we bought
     iii. Flat (zero) if we don't have a position in the given stock
d. SymbolNotional
   - Value of the shares bought or sold, == FillSize * FilledPrice
e. ExchangeBought
   - Number of shares bought on the current exchange, across all symbols
f. ExchangeSold
   - Number of shares sold on the current exchange, across all symbols
g. TotalBought
   - Total number of shares bought across all symbols
h. TotalSold
   - Total number of shares sold across all symbols
i. TotalBoughtNotional

- Total value (SymbolNotional) of all shares bought across all symbols

j. TotalSoldNotional
   - Total value (SymbolNotional) of all shares sold across all symbols

In addition, the summary trade stats will also be computed simultaneously

a. Shares Bought: Total number of shares bought
b. Shares Sold: Total number shares sold
c. Notional Bought: Total value of all shares bought
d. Notional Sold: Total value of all shares sold
e. Per Exchange Volumes:
   a. For each exchange, the total number of shares bought and sold
   b. Sorted by the exchange name
f. Average fill size
g. Median fill size
h. Top 10 most active stocks: List of stocks with most volume (in total shares traded), in descending order and including the actual volume shares traded in parenthesis

Specialities

Computing running median used to be a hard problem in practice and is a hard problem in the Leetcode coding training platform. Here I used the `bisect` insert algorithm adapted to this probken and achieved top 80% performance in Leetcode.

Future work A robust data cleaning/checking function when processing trades

How to run

Assume python already installed in computer

- python calcStats.py inputFile outputFile
- from calcStats import calcTradeStats calcTradeStats(inputFile, outputFile)