**Documentation for VPIN**

1. **Introduction**

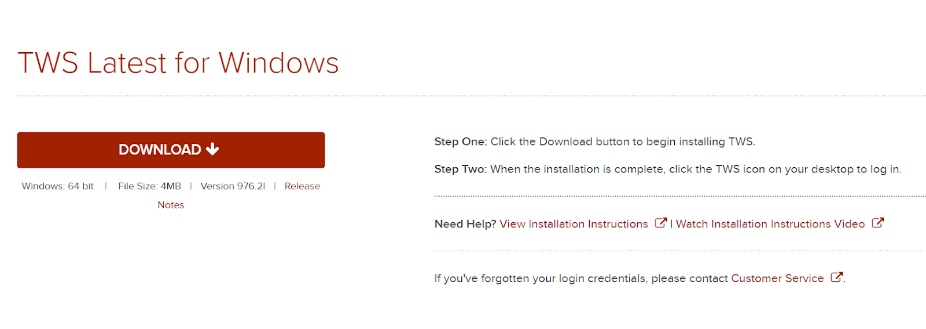
VPIN (Volume synchronized Probability of Informed trading) is a leading indicator of liquidity-induced volatility for financial products (such as stocks, futures, options and so on).

Interactive Brokers (IB) TWS is a trading platform which could provide us with historical and market data of these products, which would be further used for calculating VPIN as an indicator of special event-driven phenomenon to predict risks in the market.

Firstly we should download the IB TWS software to the computer, then use module ib\_insync to connect IB with Python, receive tick data from IB, find optimal parameters for VPIN to do the calculation, update the live data as time goes by and finally plot streaming charts for VPIN.

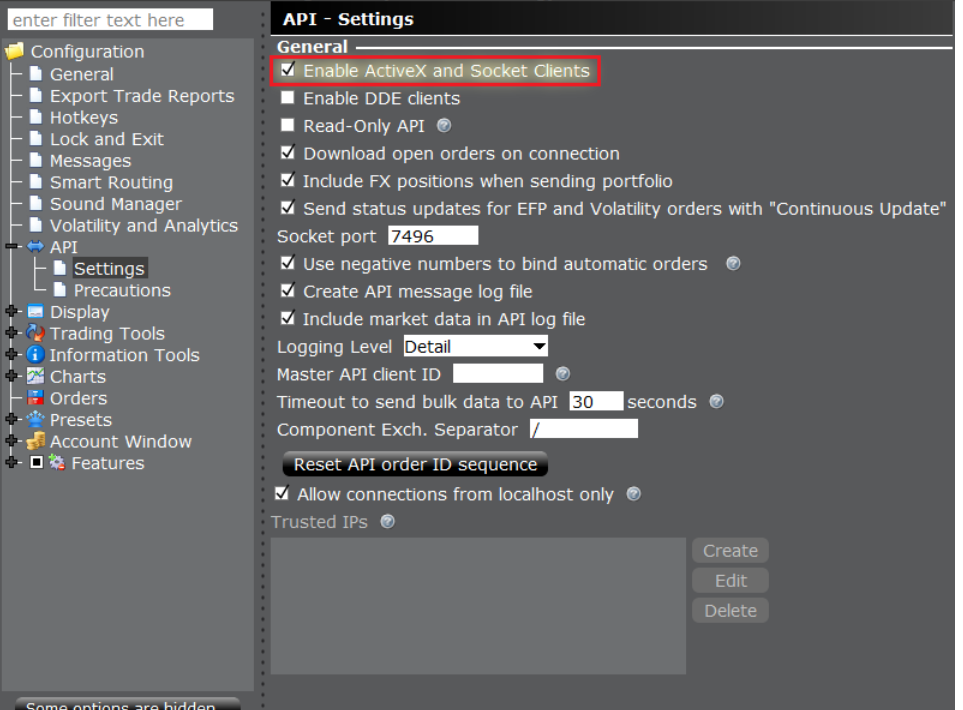
The whole process is based on two papers ***Parameter Analysis of the VPIN (Volume synchronized Probability of Informed Trading) Metric*** and ***A Big Data Approach to Analyzing Market Volatility.***

1. **Installation of IB**
2. **Download TWS and log in**

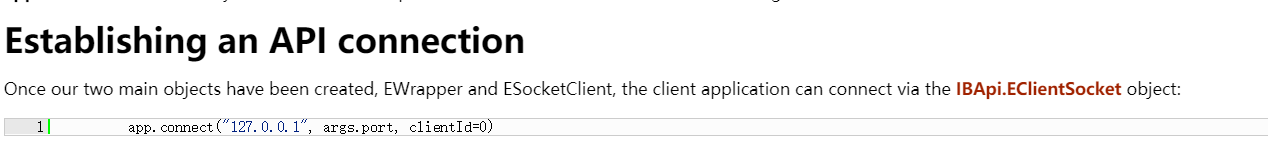


1. **Settings**

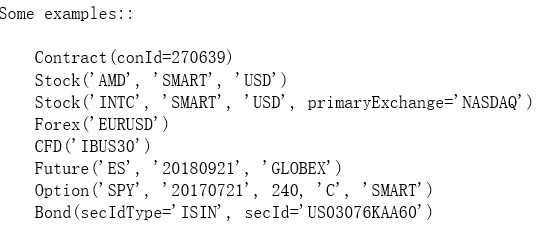
Socket port 7496 for paper trading, 7497 for real account



**c. Establish connection**



**d. Syntax of searching for securities**



1. **Program for history data**

**Key Parameters for VPIN (suggested in paper):**

•Nominal price of a bar 𝜋

• Parameter for the Bulk Volume Classification (BVC) 𝜈

• Buckets per day (BPD) 𝛽

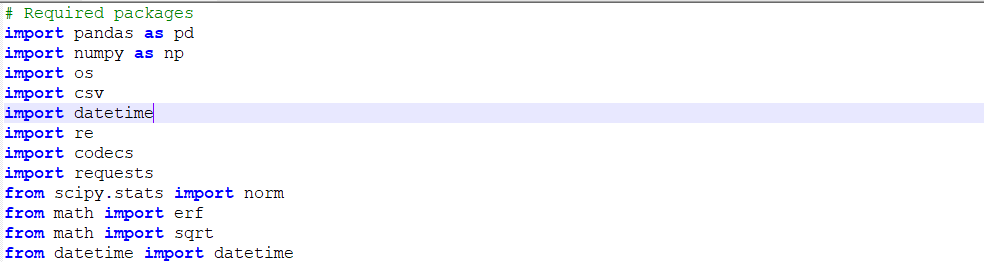
• Threshold for VPIN 𝜏

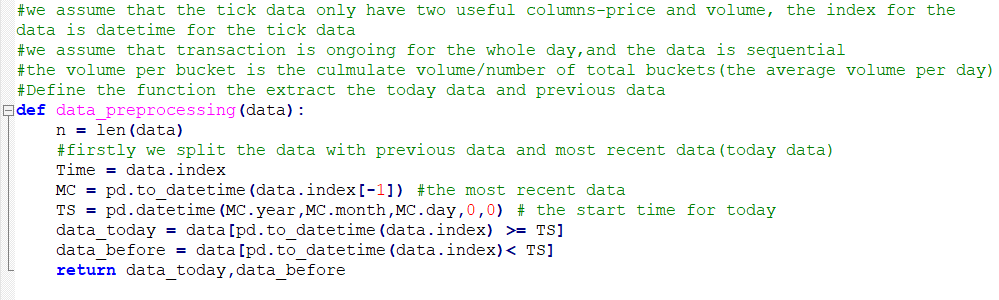
• Support window 𝜎

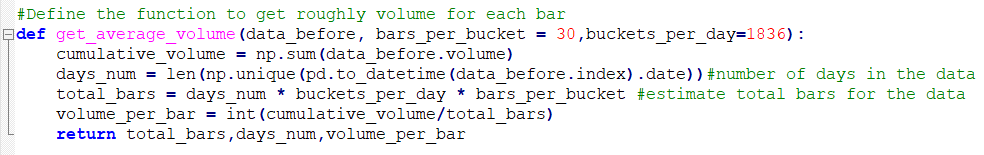
• Event horizon 𝜂

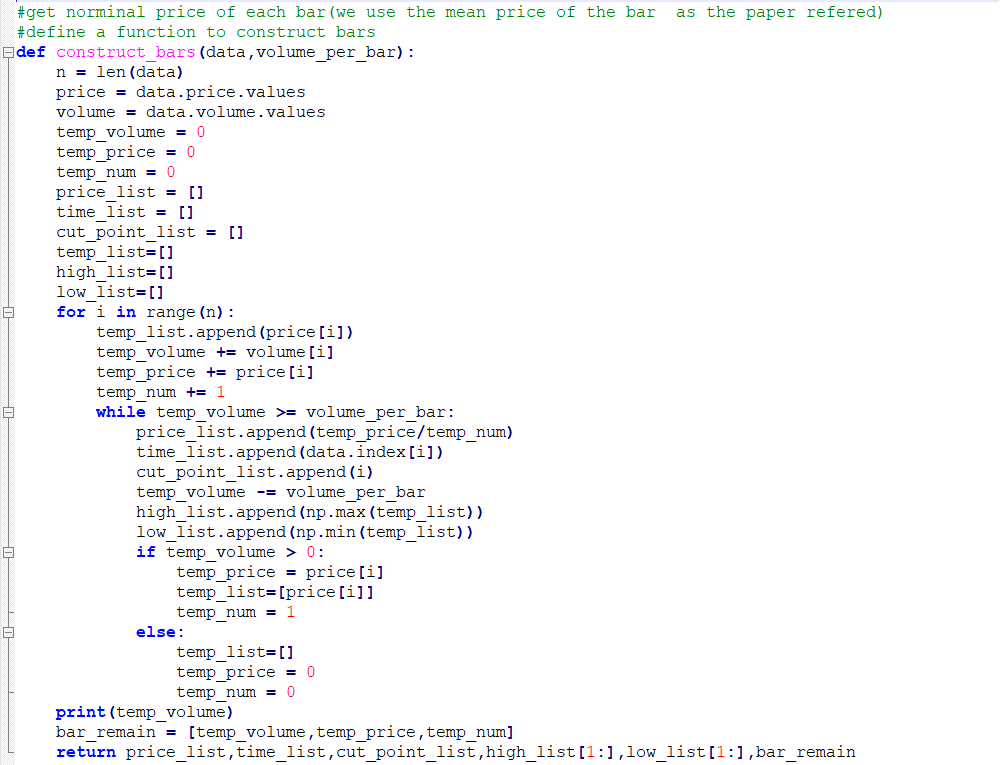
The calculation of these parameters is subject to changes in historical tick data and may be adjusted as more real-time data enter into the database. The following function is used to calculate historical VPIN values and save necessary values for future update.

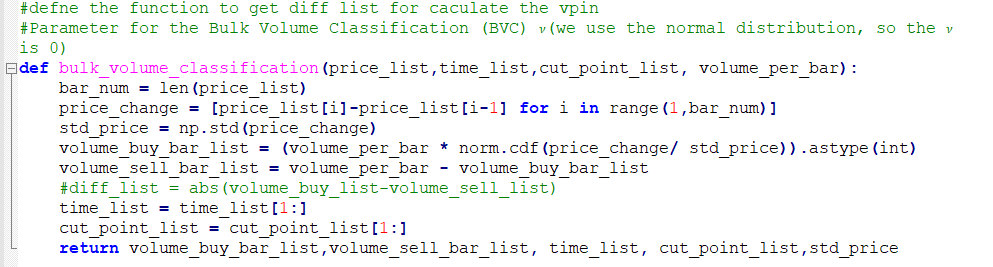
All codes in the Tick\_VPIN.py (before update)

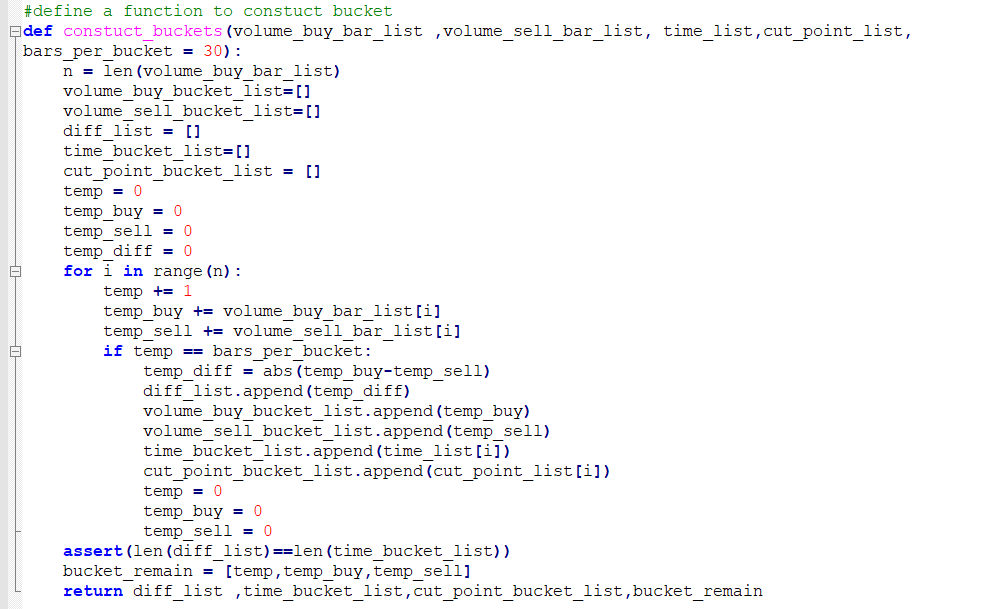


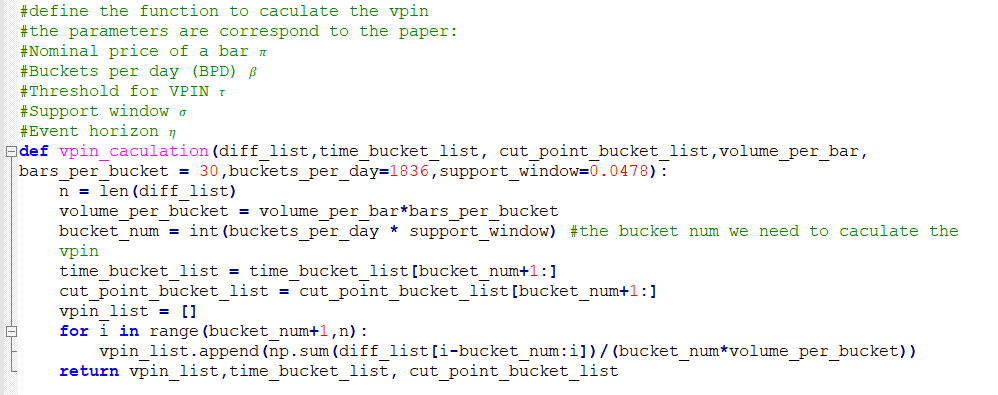


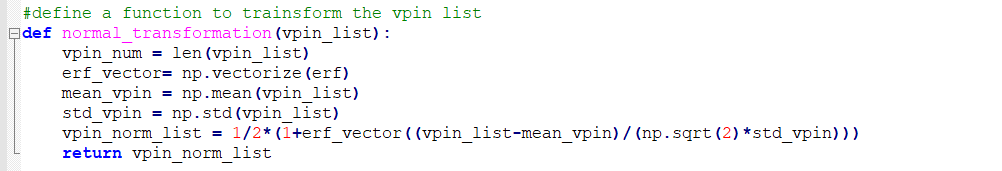


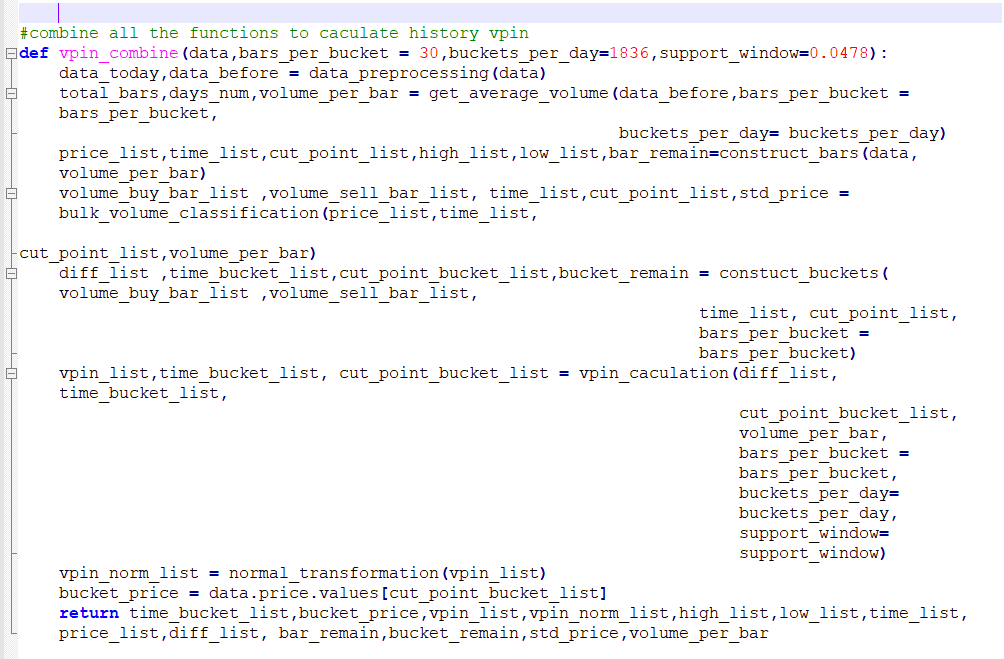




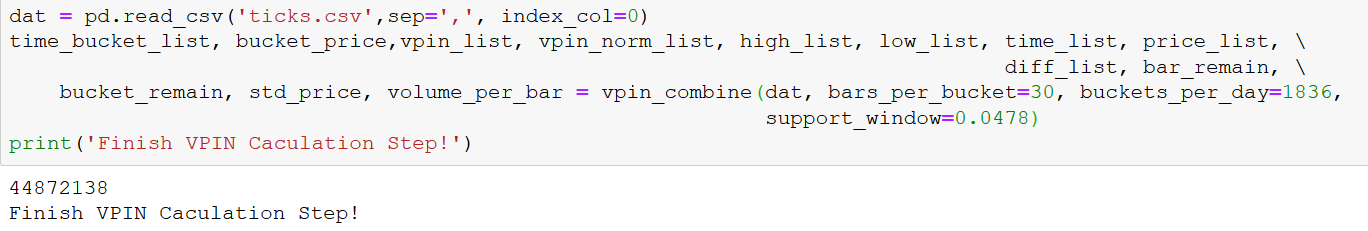








Example for historical VPIN calculation:

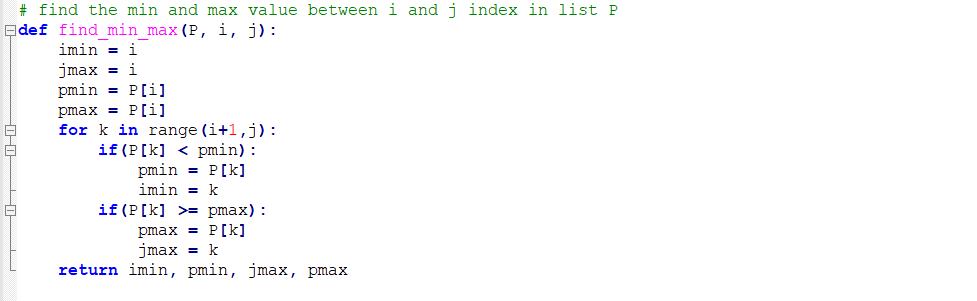


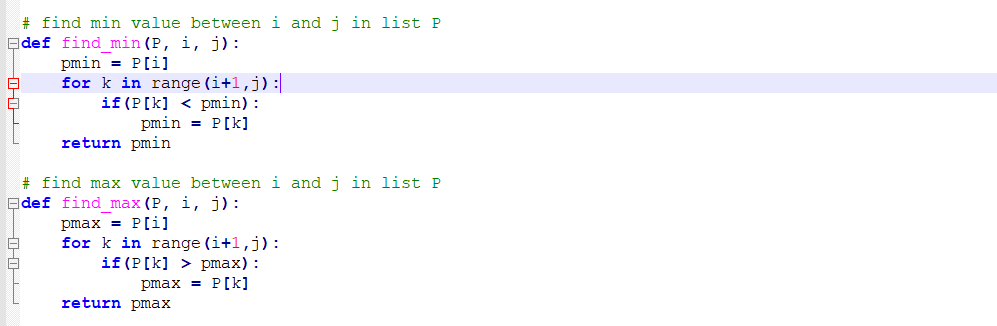
1. **Program for test VPIN program**

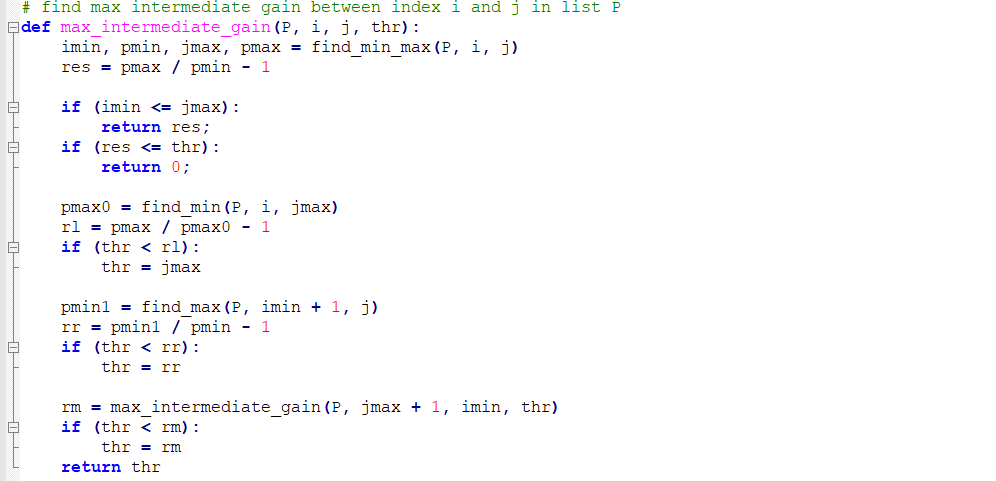
This program is used for testing the performance of VPIN. According to the paper, we should measure the performance through calculation of MIR and FPR. The formula could be referenced in the papers.

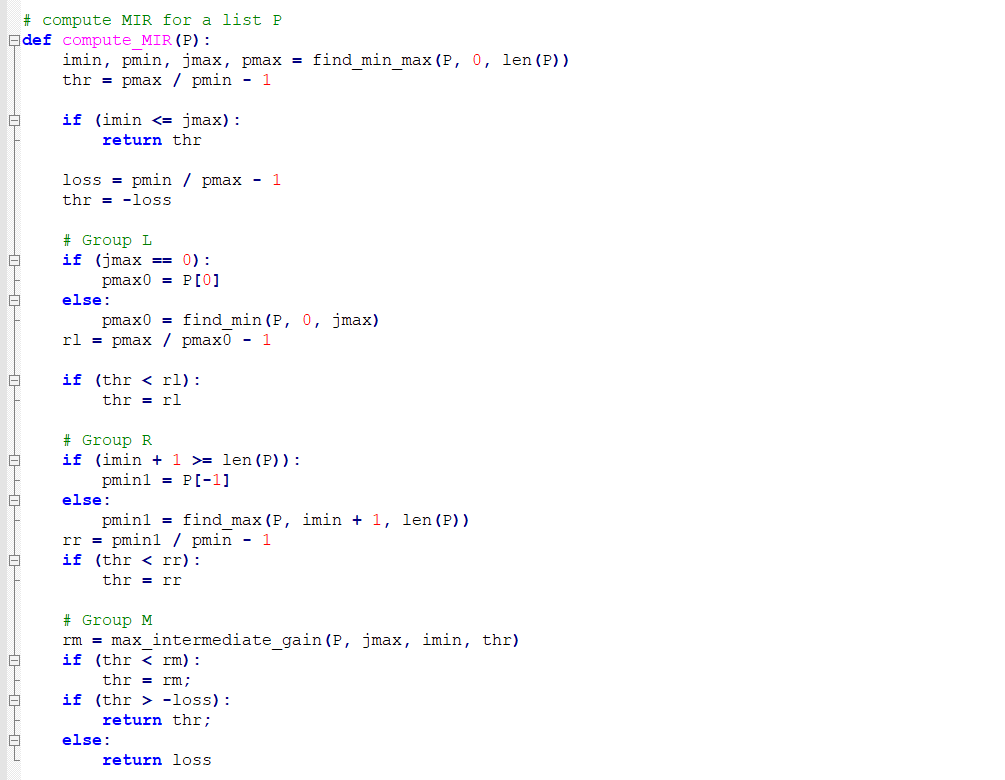
MIR computation is included in the file Parallezing\_MIR.py. In order to accelerate the computation, we use multi-processing (not applicable in Jupyter Notebook, but in Pycharm)

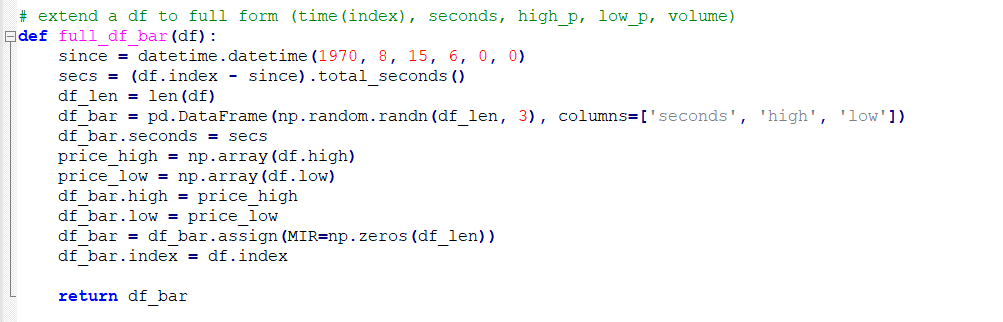


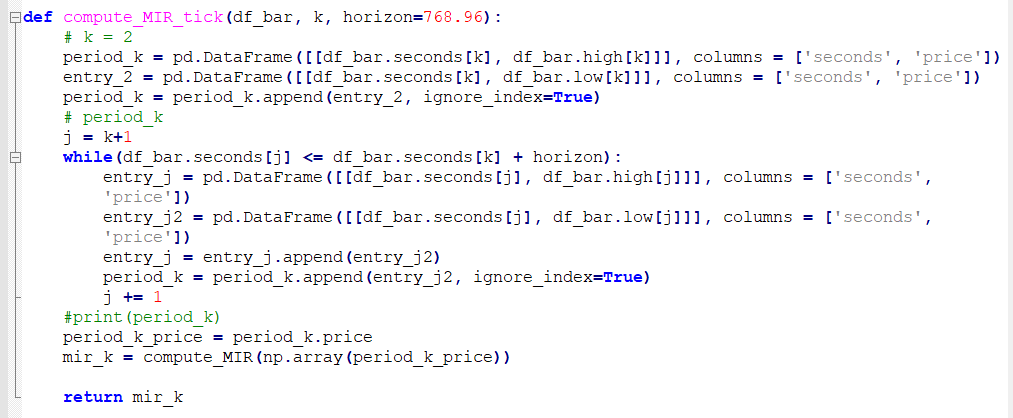


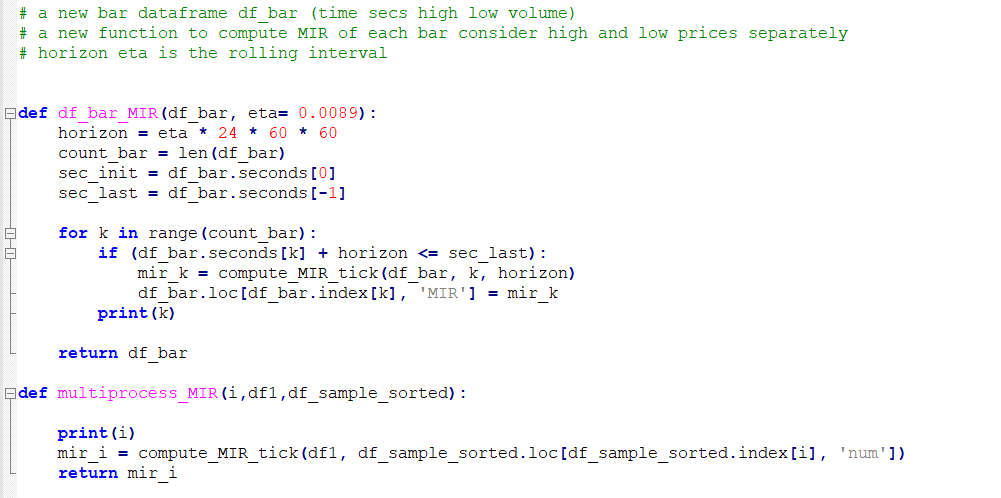


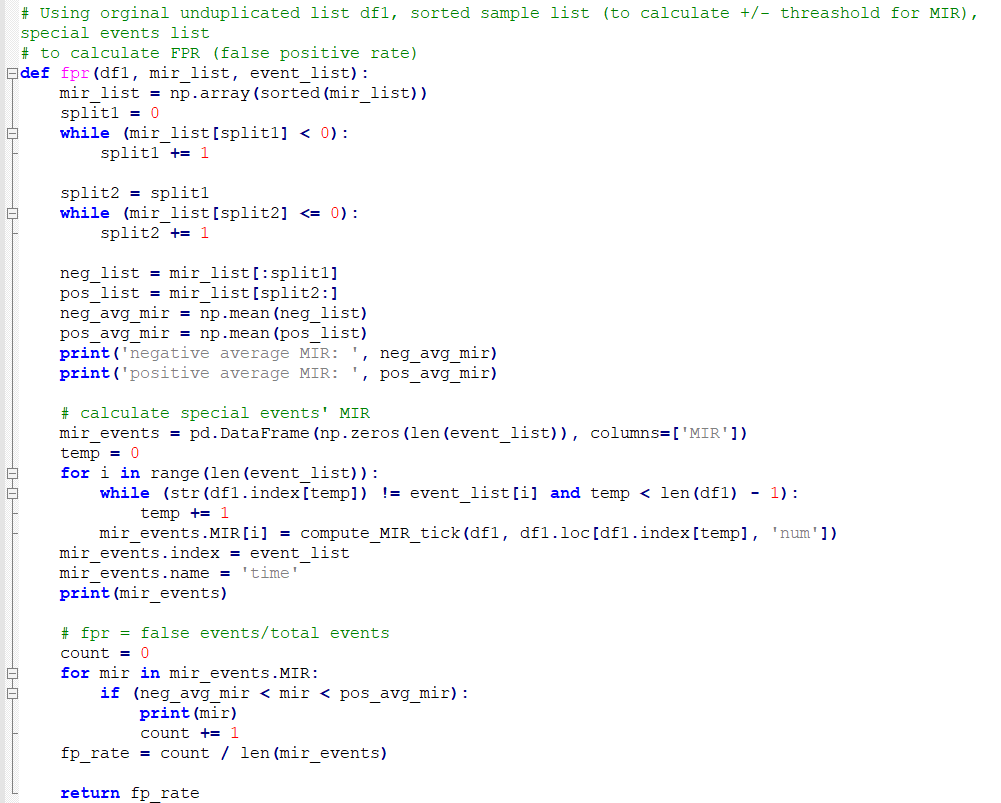




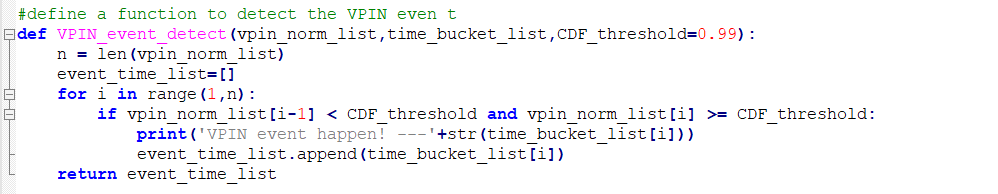




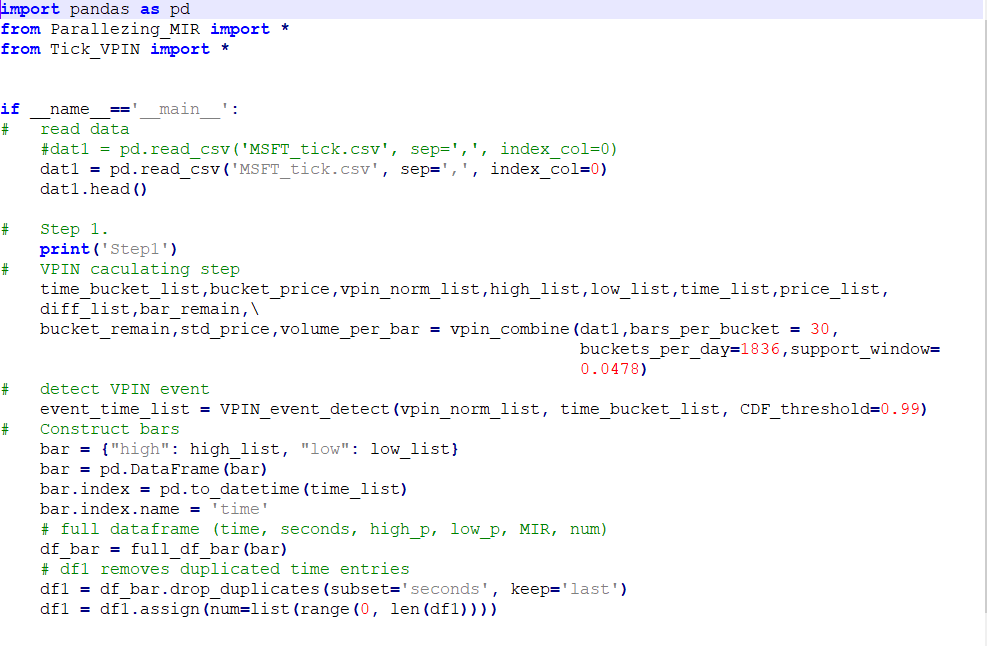


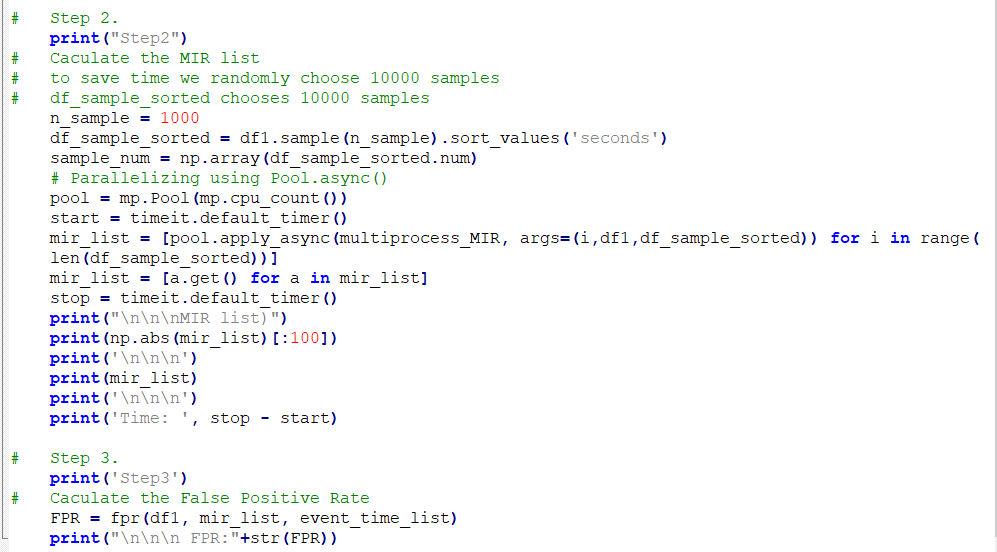


Tick\_VPIN.py would check the validation of VPIN events.



Examples of testing VPIN performance (codes in VPIN\_test.py):

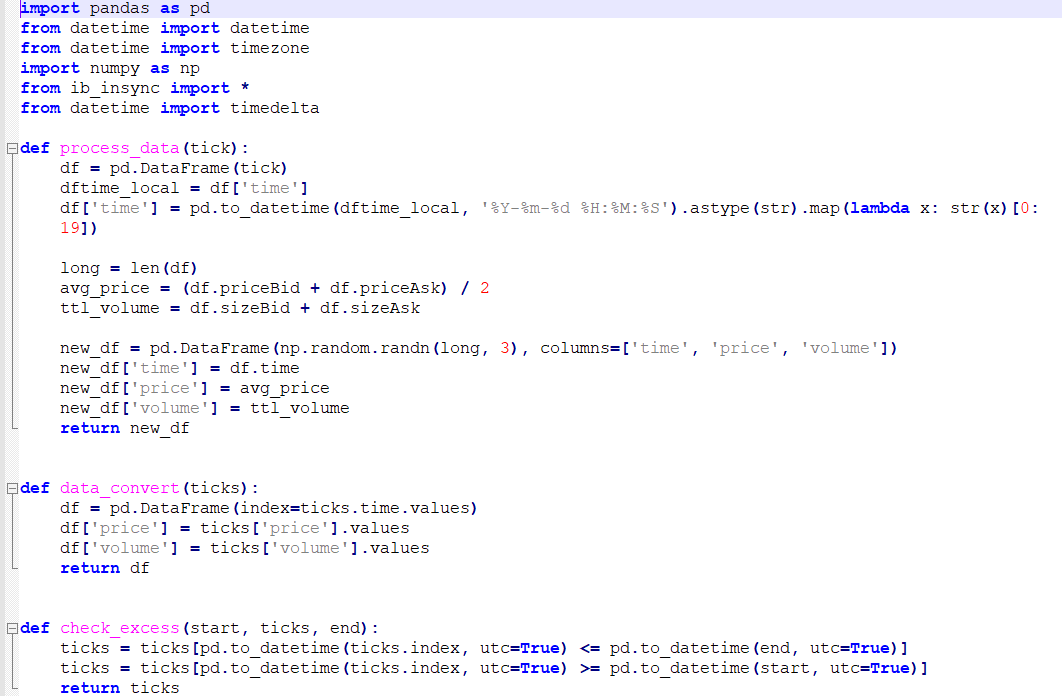


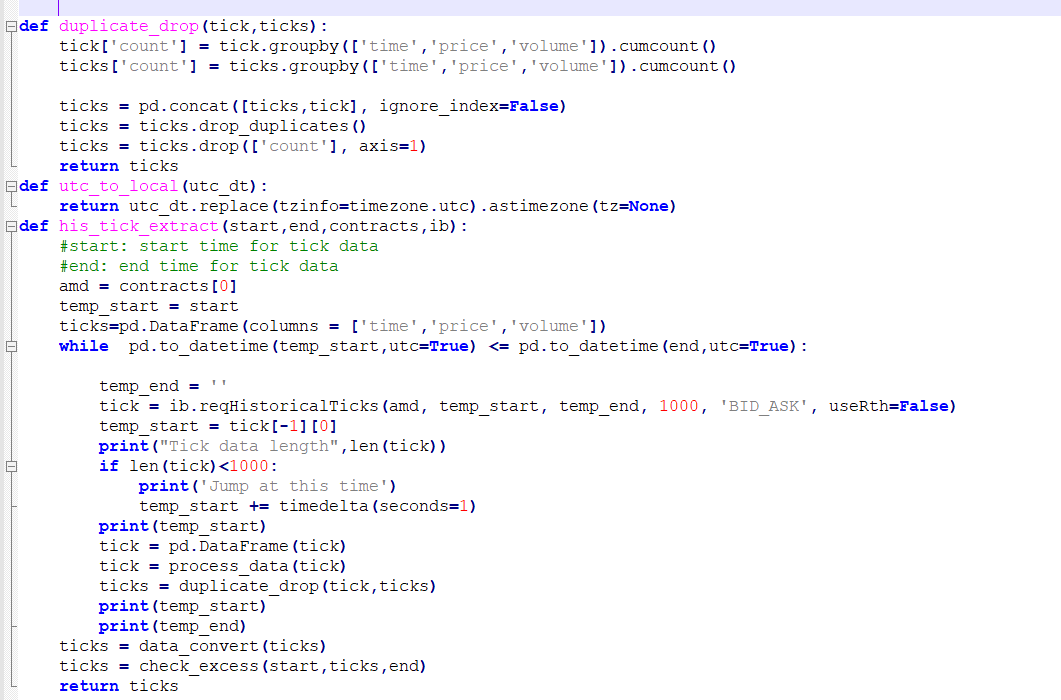


1. **Codes for extracting tick data from IB**

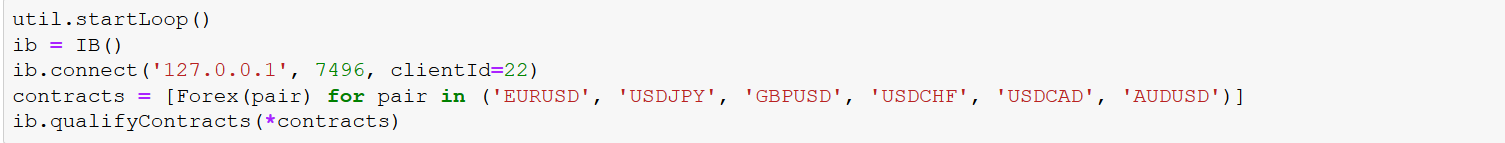
This file is used for requesting historical tick data from IB. If we assing the “end time” as now (datetime.datetime.now() ), we could get the most recent (about 1000) tick data till now. Notice: 1000 is the maximum limit for single time tick data request set by IB system.

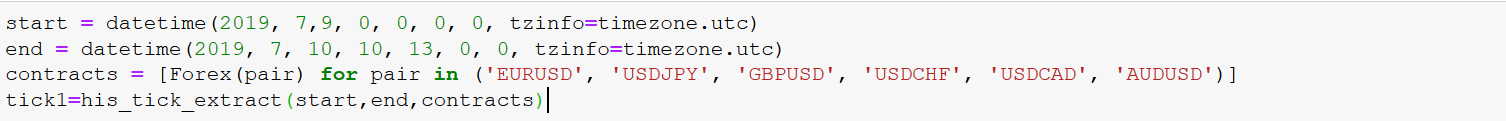
Codes are included in History\_tick.py





Examples for extracting historical tick data:

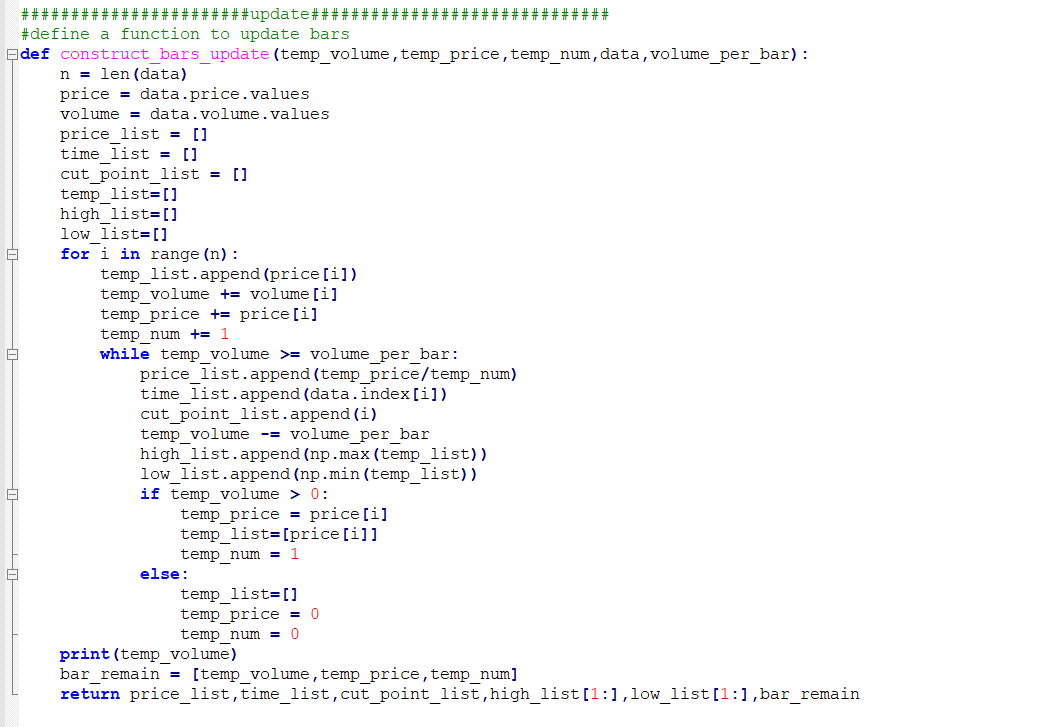


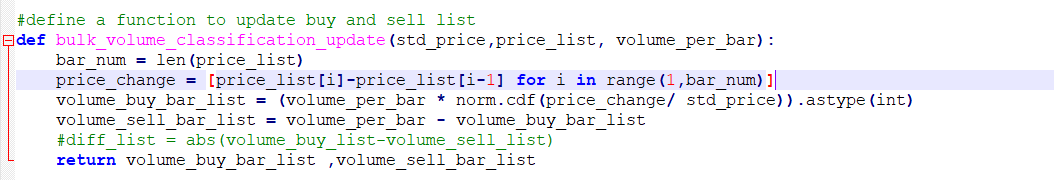


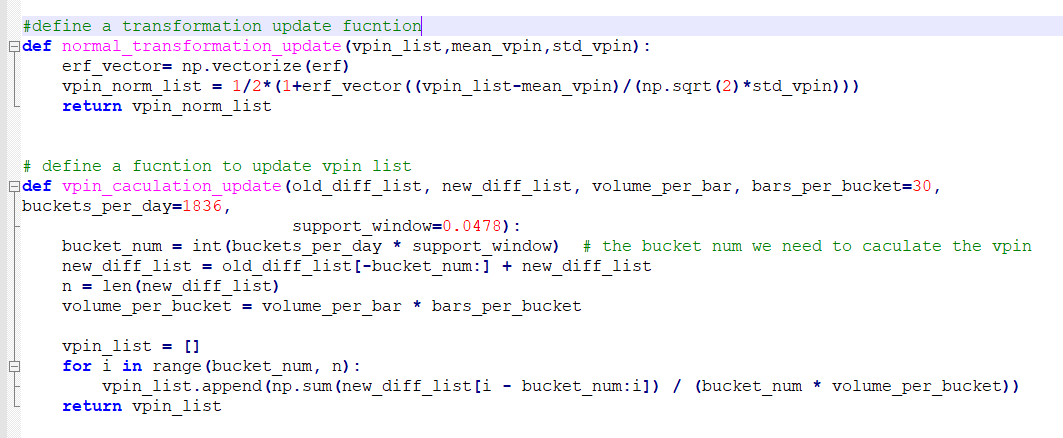
1. **Codes for updating VPIN**

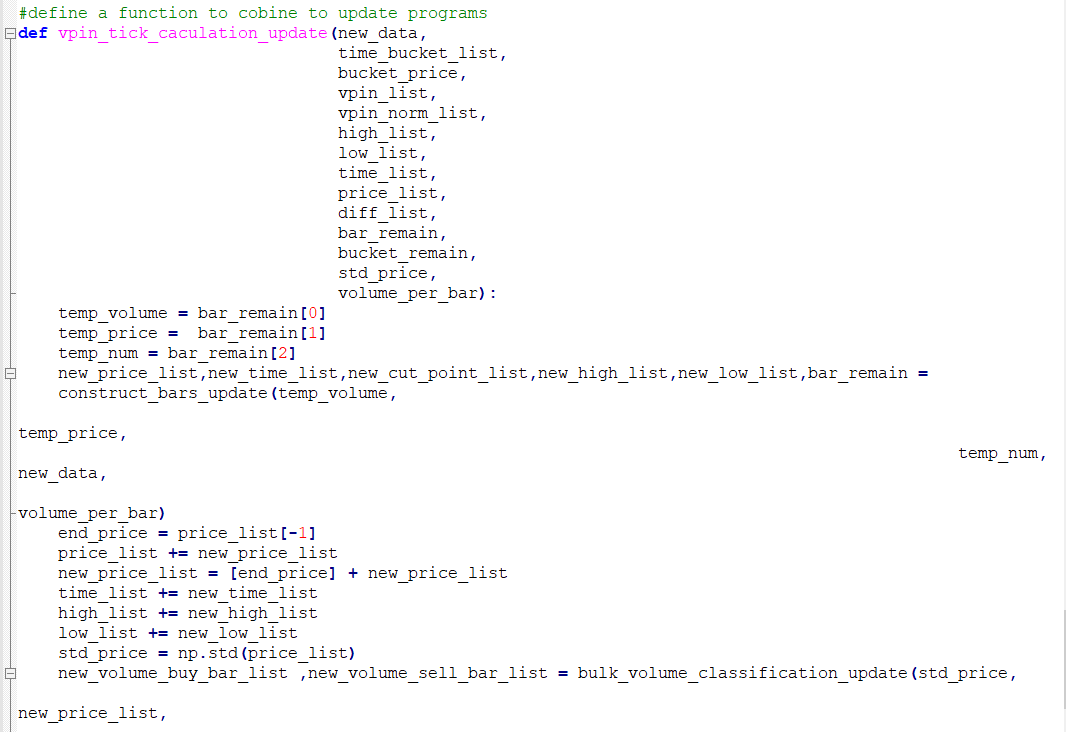
This function could calculate the updated VPIN if new data come in.

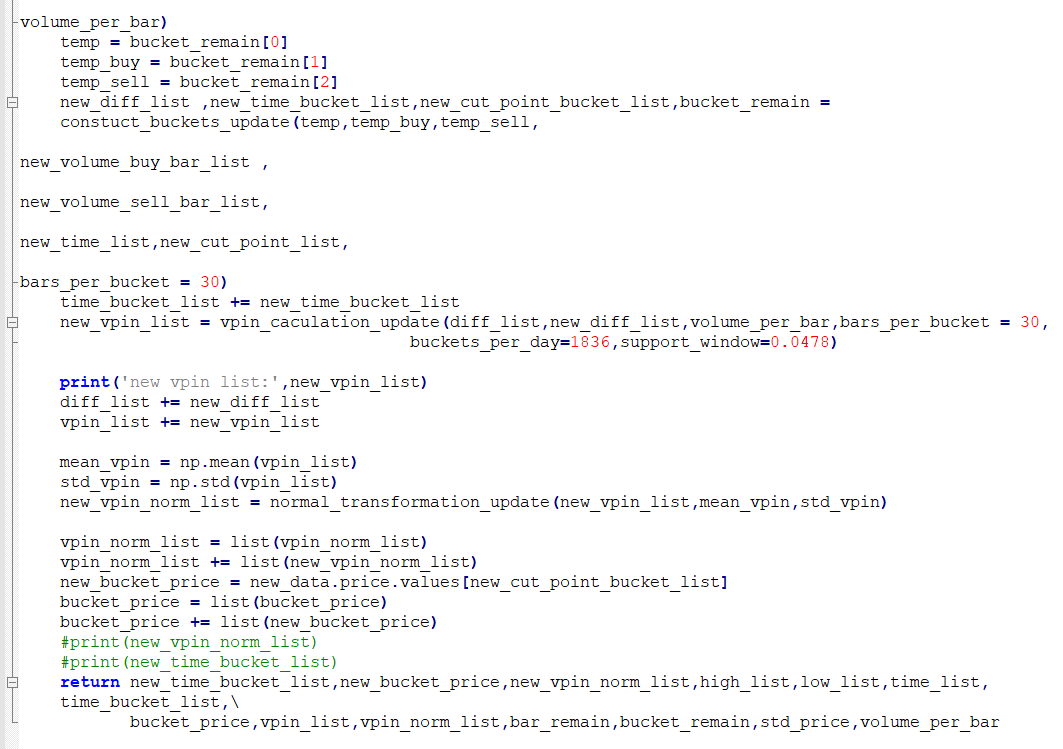
Codes are included inTick\_VPIN.py









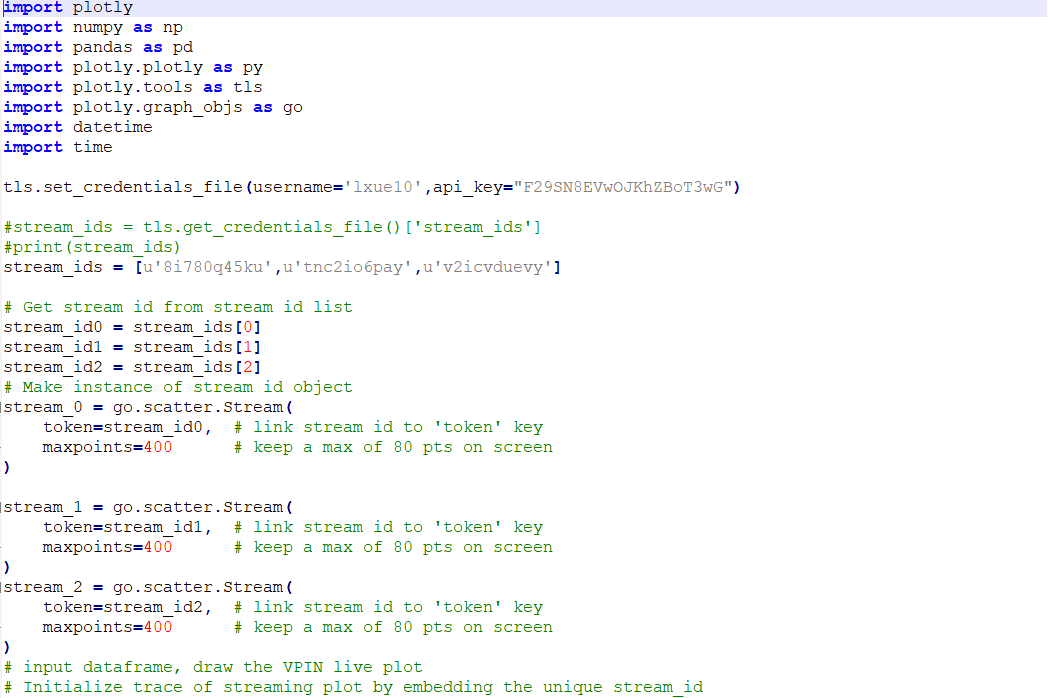


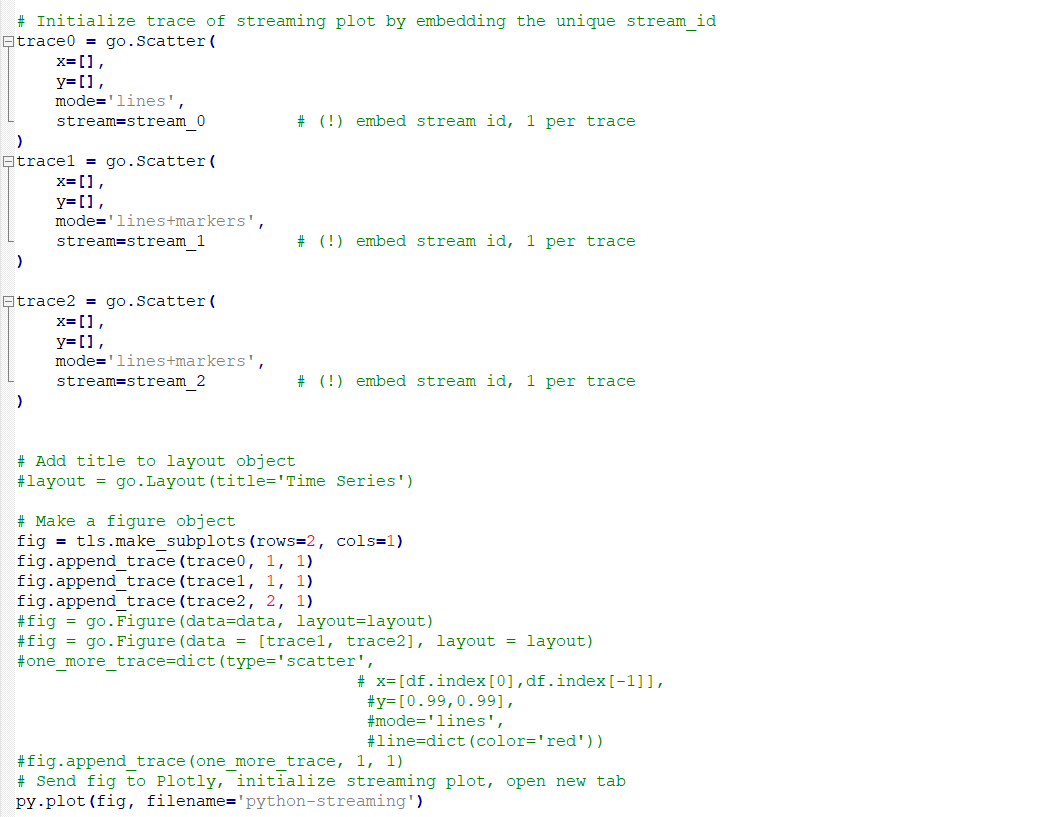
1. **Codes for plotting streaming data**

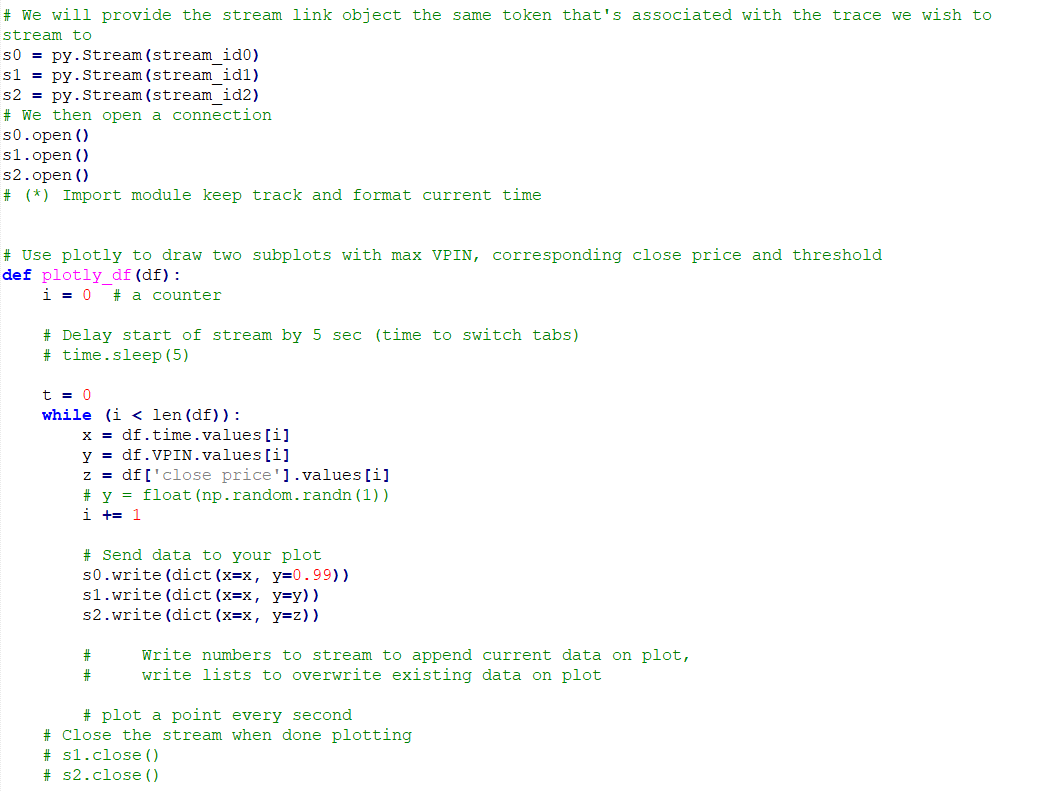
This file could draw live plots as time goes by and more tick data is included in our database. It not only reflects the most recent values of VPIN and close price, but also exhibits the trends in the market.

Notice, the plots could delay for several seconds as requesting data from IB and updating plots both need some running time.

Codes are included in plotly\_dynamic.py







1. **Codes for main function**

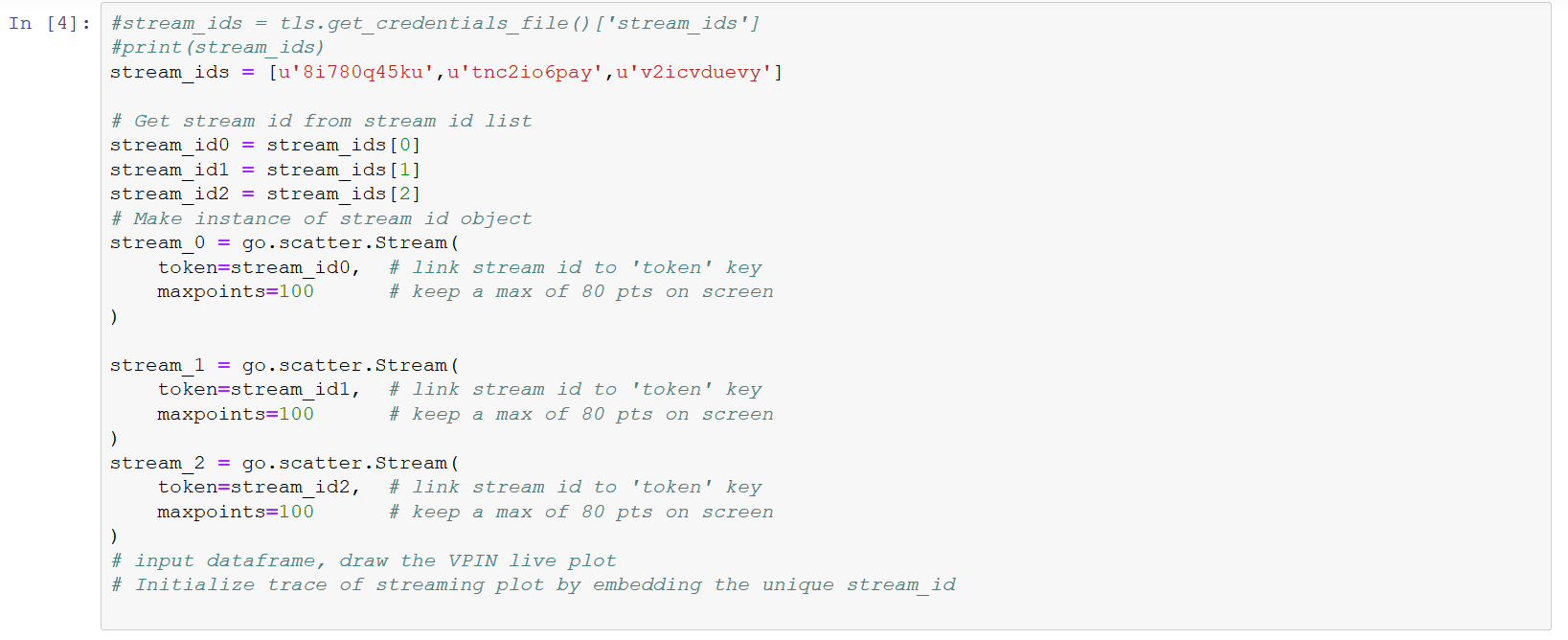
This file includes the main function we would run and it executes all functions we have written before. The output of this function are data frames consist of tick data and live plots reflecting the real-time lines of VPIN and close price.

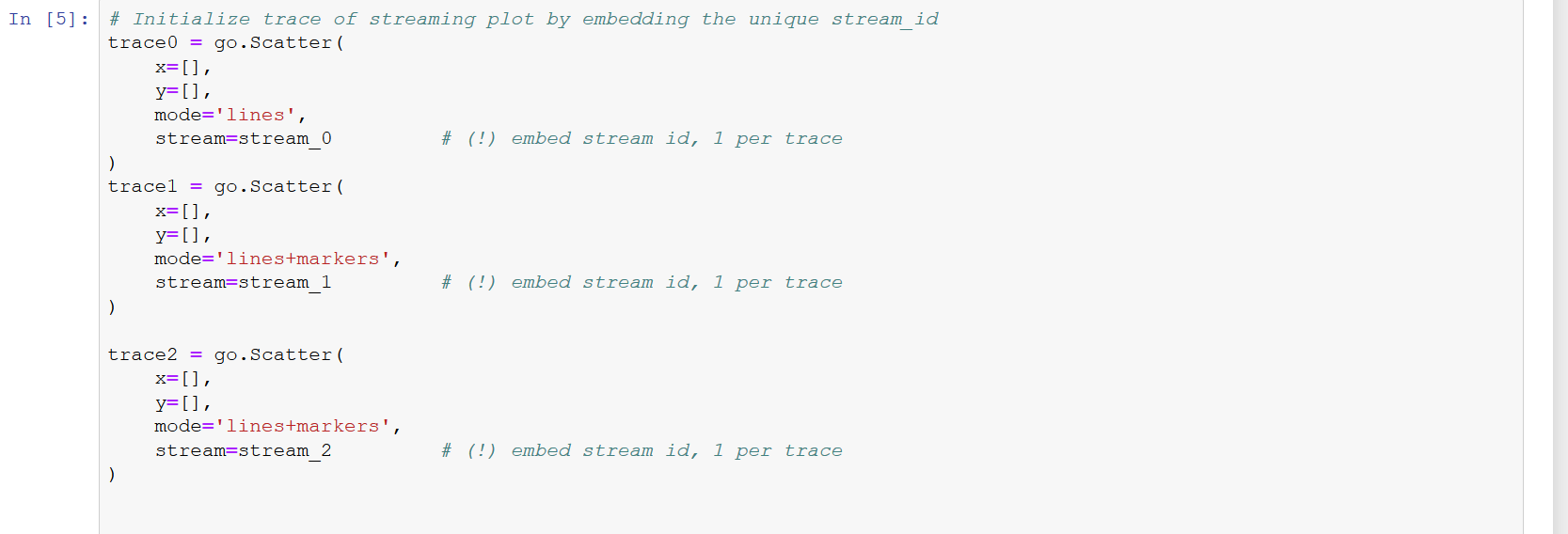
We set the threshold as 0.99 and if VPIN hits the line, it would be treated as extreme events in the market which have great influence on the volatility.

Codes are included in TEST.ipynb.

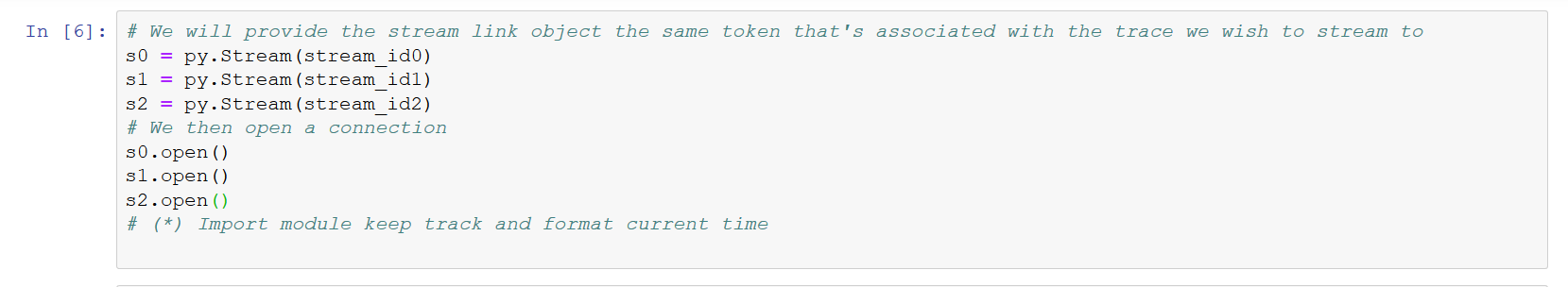


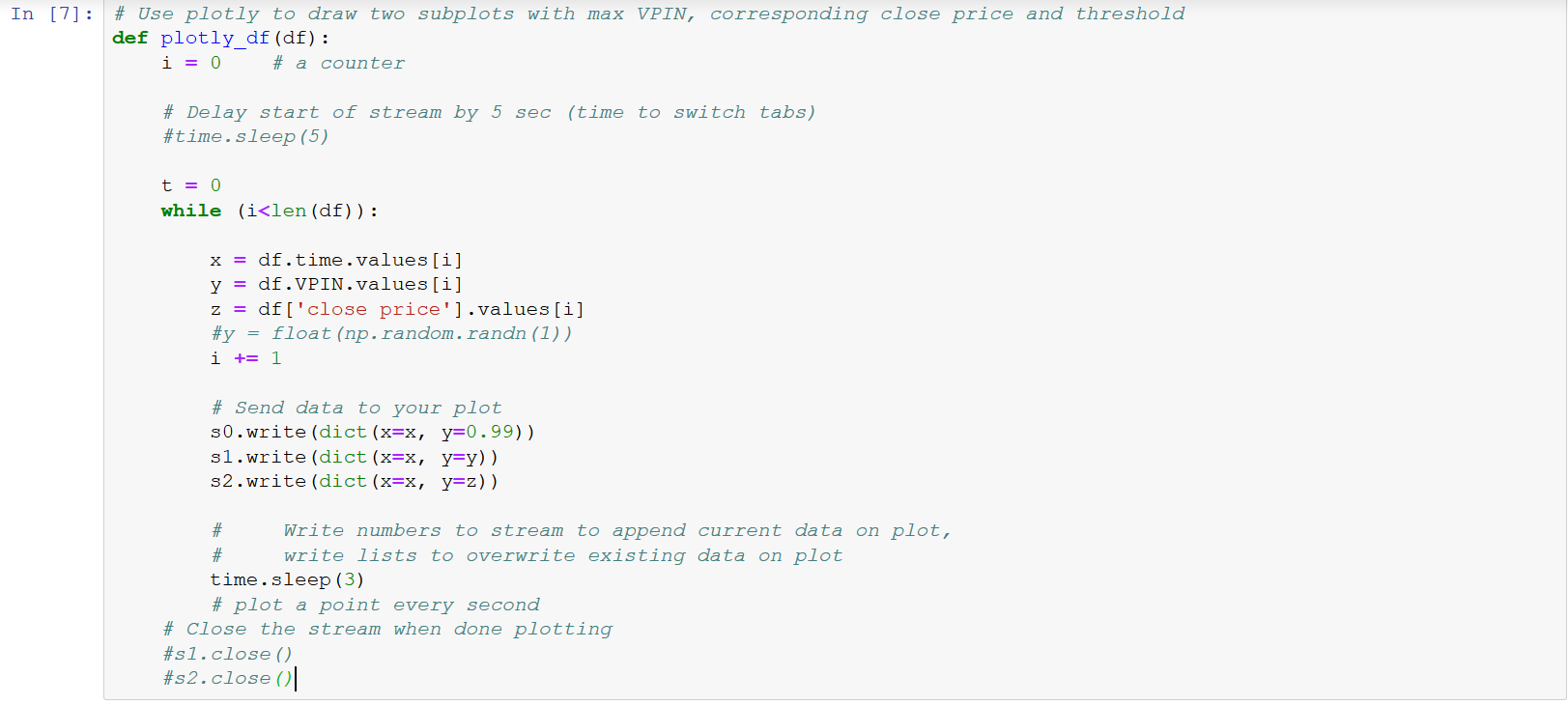




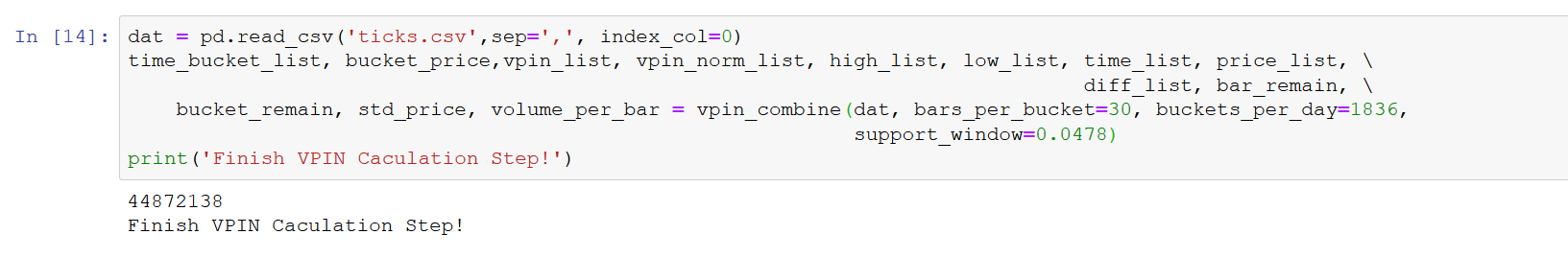


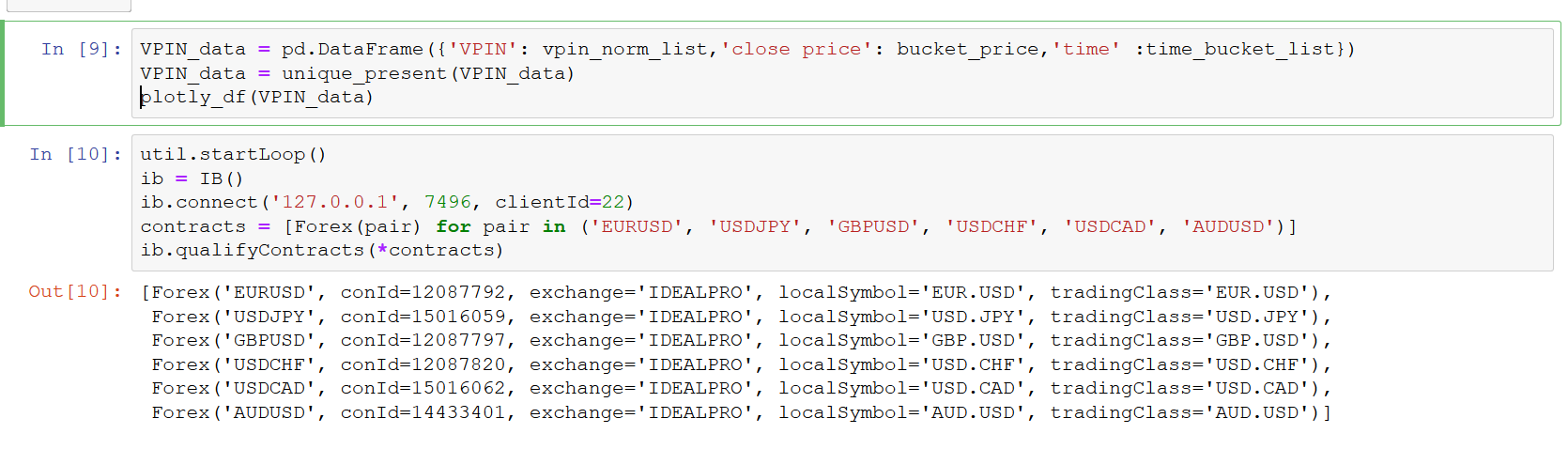


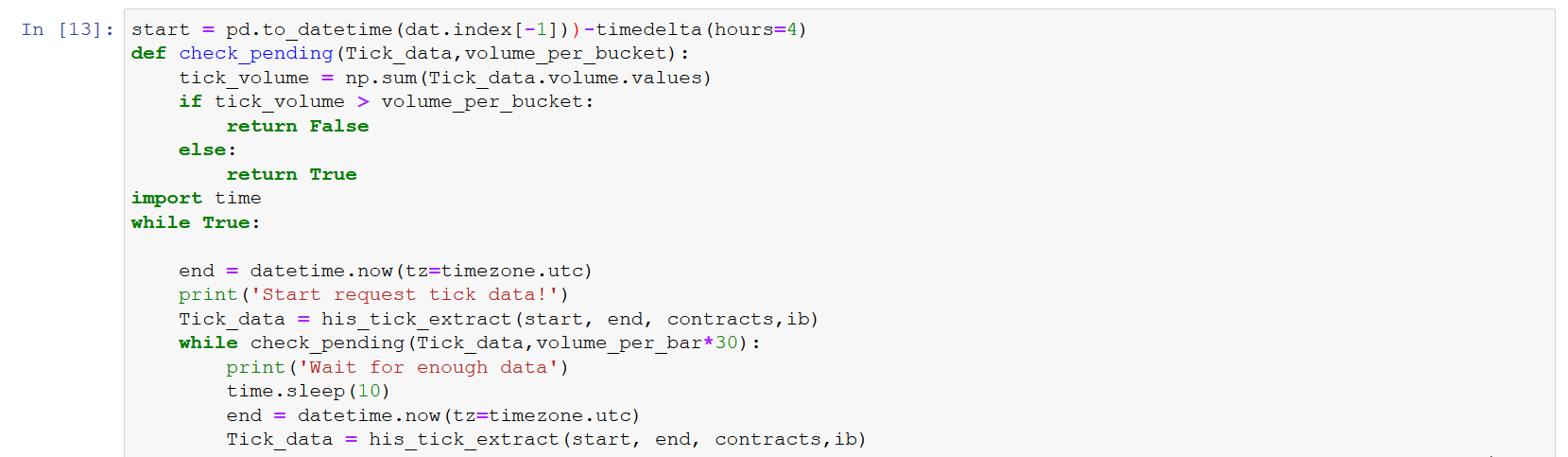


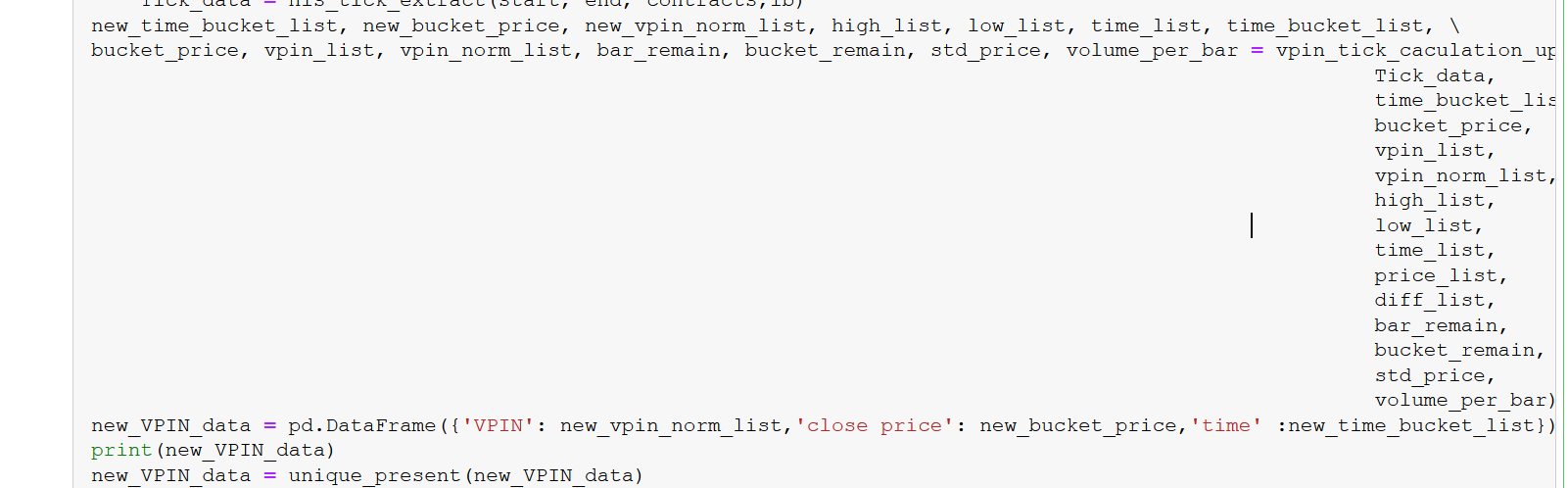


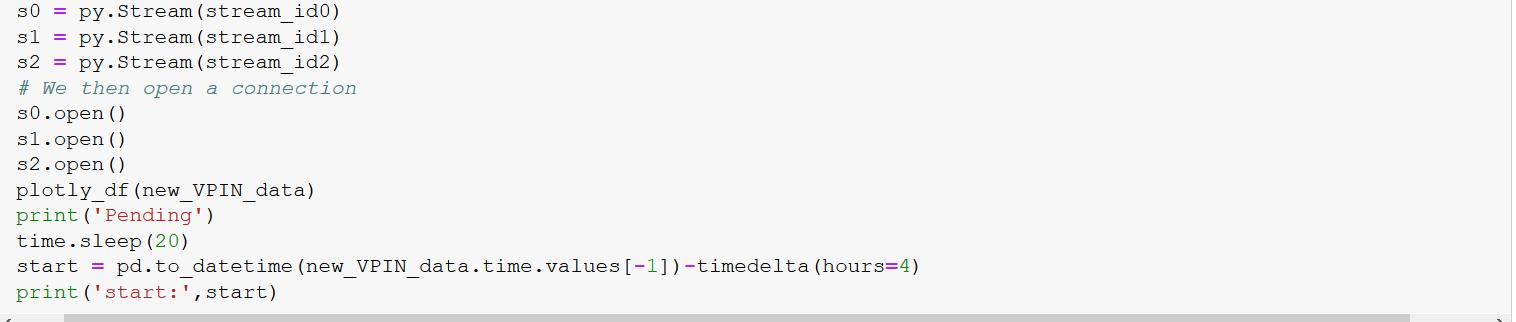






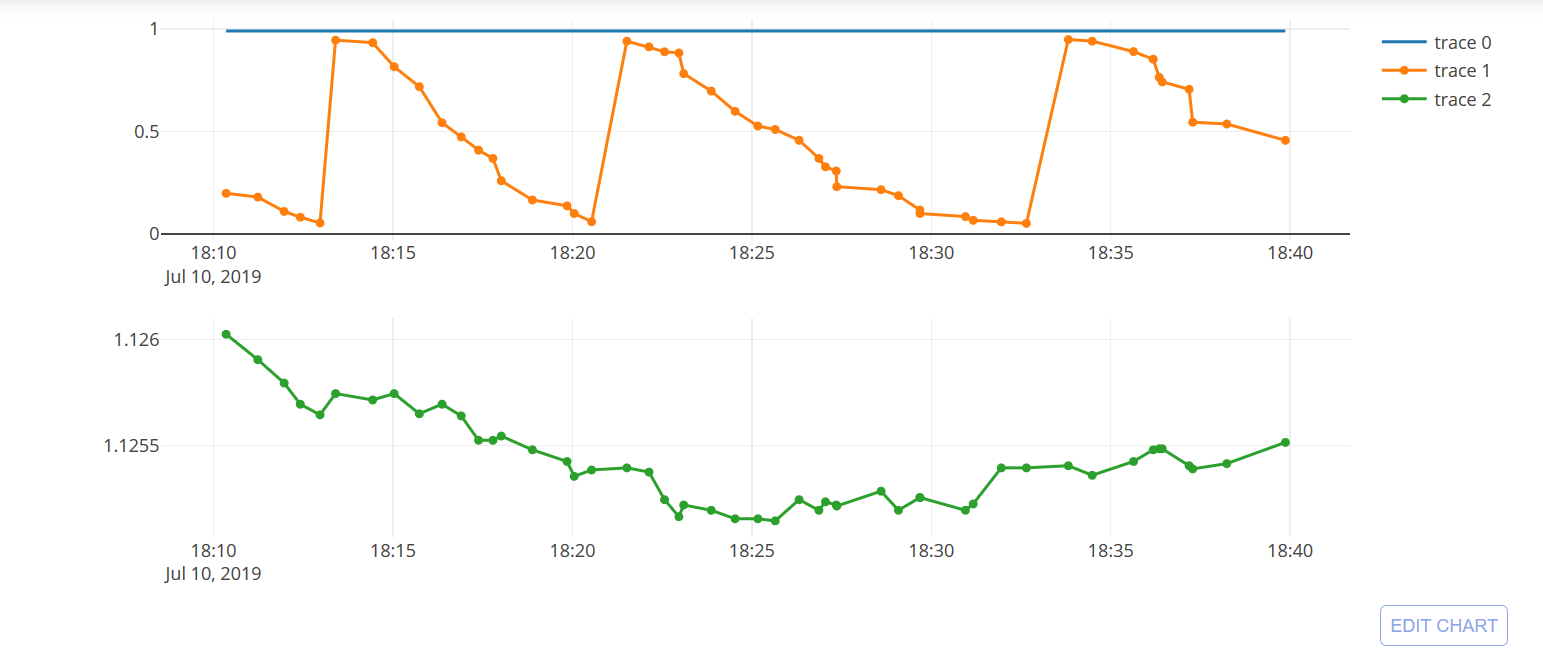








The final plot is look like this:



This project could be used to calculate and reflect VPIN of different instruments such as stock, futures, foreign exchange and so on, while the optimal parameters need to be adjusted under different circumstances.