Project 3
Cryptocurrency Trading

MAFS6010Z, Fall 2023

Market data

Order book data

Bid/ask… for market to decide stock prices

Order book is a representation for trading intention on the market. An actively traded financial instrument always has a dense order book.

Trade data

Reflects how active the market is. You can get technical signals of financial market from it.

Three transaction actions:
- Long
- Short
- Do nothing

Data

You are provided with historical minute level data of four major cryptocurrencies 'BTC(比特币)', 'EOS(柚子币)', 'ETH(以太坊)', and 'TRX(波场币)' traverses from '2021-09-14 00:00:00' to '2023-04-19 00:00:00'

OHLCV (open price, high price, low price, close price, volume)

Get preprocessed data via:
- h5 format (recommend)
  - google driver link
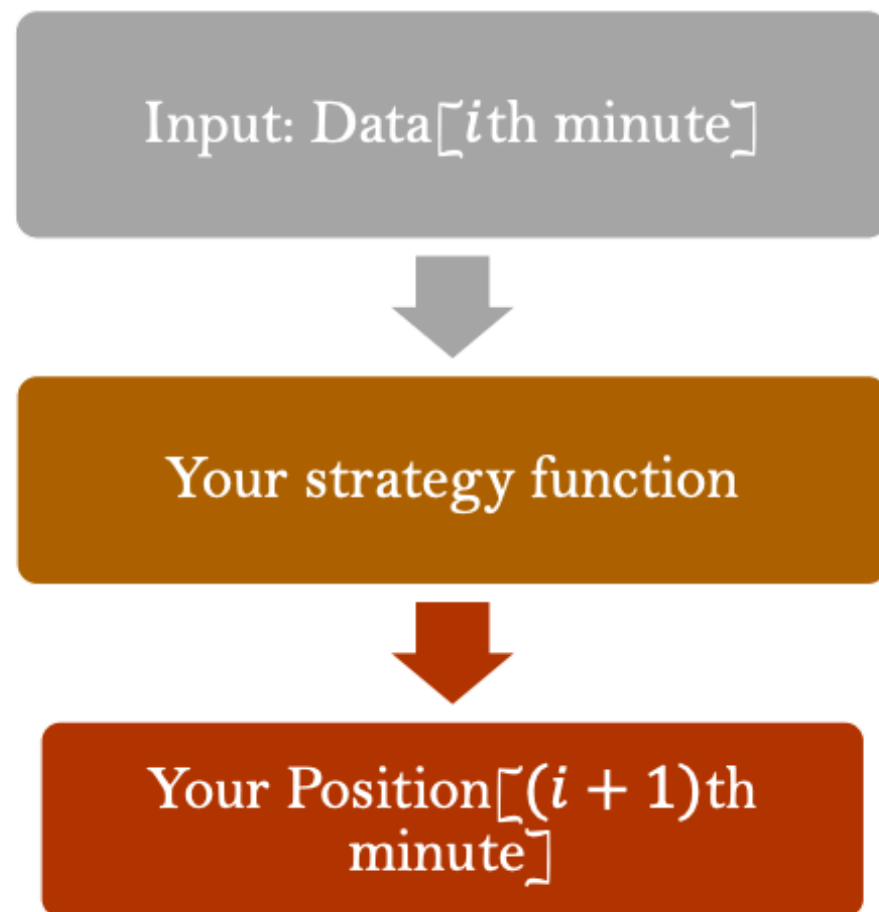- Non-splitted csv format:
  - google driver link

Your job

Develop your minute level trading strategy to earn money!

Given data from one minute, your strategy should output its desired position at next minute, which implies how will you trade (long/short/do nothing) different cryptocurrencies at next minute. **Your strategy need to generalize to various kinds of cryptocurrencies.**

We provide you with **demos** and backtesting scripts, and you need to follow it to write your strategy:
https://github.com/yao-lab/yao-lab.github.io/tree/master/aifin/2021/project3/demo

Input: Data$[i$th minute$]$

⬇

Your strategy function

⬇

Your Position$[(i + 1)$th minute$]$

Trading details

- **Initial asset: $100, 000.**

- **Transaction cost: 0.0005 for each trade.**
e.g., short 5 BTC next minute when the average price is $9000, transaction cost is:

$$5 * 9000 * 0.0005 = \$22.5$$

After after one hour, the average price is $9500, and you want to close your position, transaction cost:

$$5 * 9500 * 0.0005 = \$23.75$$

Trading details

● **Transaction volume restrict.**

Our test program will check your position change to make sure that it does not exceed **0.25*(Volume next minute)**. For example, if you want to buy 10 BTCs next minute but the market trading volume of BTC is just 5.6, then our test program will adjust your position change only by 0.25*5.6 = 1.4.

● **Transaction price.**

At next minute, if your positions of some assets change, we assume the trading price is always the **average price** of the asset in this minute. Here average price is calculated by **(open+high+low+close)/4**.

● **Final total return**

Cash balance + crypto balance (last day price)

Performance evaluation

- **Total return**

- **Pnl (profit & loss) curve**

- **Max drawdown**

- **Sharpe ratio**

$$Sharpe\ Ratio = \frac{R_p - R_f}{\sigma_p},$$

where R_p= return of portfolio, R_f=risk-free rate, σ_p=standard deviation of the portfolio's excess return.

Notes

- Build your strategy based on your observation, experience, data analysis, and models form papers and financial reports.

- **Try not to speculate or gamble extravagantly** (for example, use all your money buy BTC at the very beginning and pray BTC price rockets up next week). Even if you get good performance score by luck, don't forget the **final report & method novelty** counts heavily in the total marks.

- You can make use of the "**memory**" variable flexibly, since it is a list (or Python class) which will be read and write cyclically during the whole testing period. For example, if you want to calculate the current 5-minute moving average for close price, you should keep the recent 5 close price in your memory variable.

Notes

- **Read the demo carefully**. You can use the eval data to assess your strategy. Make sure that your strategy.py can pass the backTest.py directly since we will assess your strategy on our test data (not published), and the format of test data is the same as the eval data.

For those who wants to try more: We will provide additional historical OHLCV data with different frequencies (including 'BTCUSD', 'BCHBTC', 'ETHUSD', 'LTCUSD' transaction data in the minute /15- minute/1- hour/1-day bar). Contact with us!

Acknowledge to Cyril DL and [Bastian Kälin](#) for providing relevant materials and data download API.

1. You can get list of available books from their API https://api.bitso.com/v3/available_books/
2. Get historical data from Bitso (https://bitso.com) OHLC csv format.