Q1. An asset manager is carrying out the back-testing of a rule-based trading strategy before he launches it to the market. The manager believes that stock prices should revert to its long-term average prices and whenever the stock prices deviate from the long-term average price, these are opportunities to buy/sell the stocks, due to under- or overvaluation. Given a set of stock price data, the first N years(s) is the observation period and the daily prices in the observation period is used to determine the volatility s of the stock. If the stock price ≥ S0 + ns where S0 is the reference price for a stock and n is the volatility multiplier, the stock is over-valued. If the stock price < S0 - ns, the stock is under-valued. S0 is the prices of the stock on the last day of the observation period. (S0 + ns) and (S0 - ns) are the respective sell and buy thresholds for this strategy. In this ECA we will examine the use of these rules to determine the right times to enter and exit a position in a stock.

Given the following Python inputs, determine the buy/sell dates and prices of a chosen stock

```
api_key = '' # alpha vantage api key
stock = 'SPY'
obs_period = 1 # year
vol_multiplier = 2
start_dt = np.datetime64('2019-01-01')
end_dt = np.datetime64('2022-04-30')
```

Figure 1: Python inputs to the trading strategy

The trader will observe the stock prices for obs_period years, and price data are available from from start_dt to end_dt on stock. The trader will use her api_key to access stock prices from a remote database. Prepare data for your work, by making use of the alpha vantage module to download the prices of stock from start_dt to end_dt.

Q2. Evaluate the reference price, buy, and sell thresholds for stock, given the duration of the observation period and vol-multiplier in Figure 1. Print the reference price, buy, and sell thresholds (to two decimal points) of stock clearly in your IPYNB.

Q3. Evaluate all the instances where the stock crossed the buy and sell thresholds. Print all the dates and prices where the crossovers happened in your IPYNB. Indicate clearly if a crossover is a buy or sell opportunity.

Q4. When the manager examines all the buy/sell opportunities that were presented in Question 3, she realized that not all the buy/sell opportunities were valid opportunities, because an investor has finite wealth, and the investor is not allowed to short stock. These constraints lead to:

- No consecutive buy transactions, due to finite wealth
- No consecutive sell transactions, due to no shorting constraint
- The first transaction in the strategy must be a buy operation, due to no shorting constraint

In addition, the manager wishes to conclude all outstanding trades on the end_dt. This means that if there is any stock that is still held by the manager by end_dt, the stock will be sold on end_dt, regardless of the value of the stock.

Given the above constraints and assumptions, evaluate the subset of all valid buy/sell opportunities from Question 3.

Print all the dates and prices of all valid buy/sell opportunities in your IPYNB. Indicate clearly if it is a buy or sell opportunity.

Q5. Create a figure to summarize and present the results of your implementation, as shown in **Appendix A.**

Q6. Submission Requirements

Besides assessing your programming competency, your submission will be graded based on the following criteria:

- Clear documentation. Suitable use of inline comments and markdown is strongly recommended especially if you intend to provide long description for your code.
- Code management. Any code that is not required should be deleted and not commented out.
- Programming to specifications. You are required to formulate, document, and explain clearly of any assumptions you have made in your implementation.

Appendix:

Appendix A

The following plot is used to summarise the implementation results.

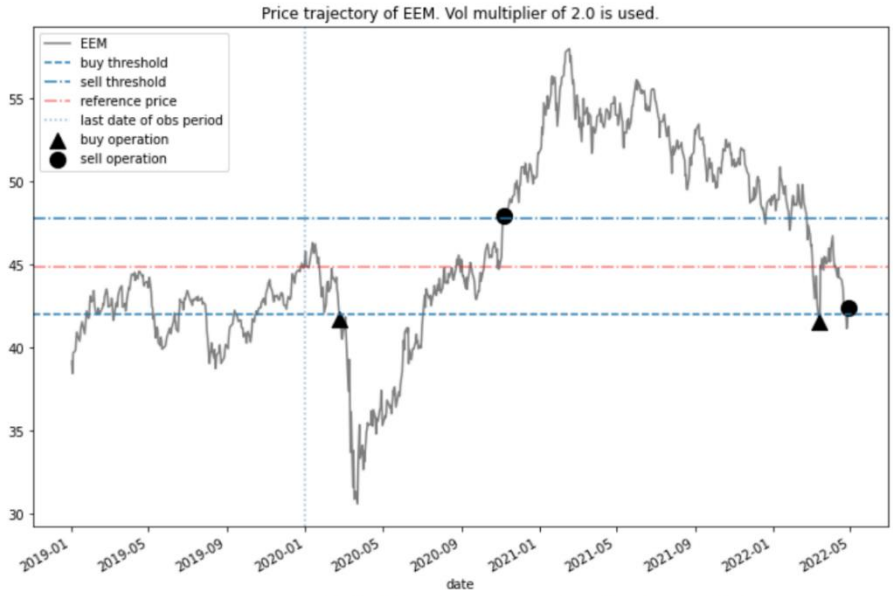

Figure A1: Valid buy/sell opportunities for EEM.