NEOMA
BUSINESS SCHOOL
REIMS · ROUEN · PARIS

Introduction to Python with Applications to Finance

Instructions for your group project

OBJECTIVES

NEOMA BUSINESS SCHOOL
REIMS · ROUEN · PARIS

Use Python to construct a limit order book and simulate the dynamic trading process.

INSTRUCTIONS FOR STUDENTS (1)

NEOMA BUSINESS SCHOOL
REIMS · ROUEN · PARIS

1. Read the material provided. Understand what the market order and the limit order are and how the trading process dynamically affects the limit order book. You will need to simulate a trading process with the above two types of orders and display the limit order book.

2. Create your Python program. The Python program will need to have the following features.

3. Your Python program needs to have functions that can process both type of orders (market order and limit order). For the buy/sell market orders, the function does not need to specify the price since they will be extecuted at the best possible prices. For the buy/sell limit orders, the function needs to specify the price limit (upper bound for the buy limit order, lower bound for the sell limit order). Both functions will need to specify the quantity of stocks intended to buy/sell.

4. For simplicity, we assume that once the orders are created, they cannot be cancelled.

INSTRUCTIONS FOR STUDENTS (2)

NEOMA BUSINESS SCHOOL
REIMS · ROUEN · PARIS

5. To process orders, your Python program needs to have functions that can match an incoming order with the ones in the limit order book. The matching process is explained in the material provided. The matching is based on the price and quantity specified in each order. If two orders have the same price, we will adopt the time priority rule, which means that the order first submitted will be matched first.

6. Your Python program needs to dynamically maintain a limit order book. In the limit order book that stores the unexecuted limit orders for each price, you should have at least the following features of the order: the price of the order, the direction of order, the quantity of the order, the submitted time of the order, and the order ID (you can create one for each order). The limit order book should be sorted based on the execution likelihood. That is, the orders that are more likely to be executed should be placed on top, and vice versa. For example, buy limit orders with the highest price limit should be placed on top. Similarly, sell limit orders with the lowest price limit should be placed on top. The limit order book shall dynamically change with the trading process (order submission and matching).

7. You python program should have a function that can display the top 10 best outstanding buy and sell limit orders in the limit order book when called.

INSTRUCTIONS FOR STUDENTS (3)

NEOMA BUSINESS SCHOOL
REIMS · ROUEN · PARIS

8. Create 5000 random buy and sell orders (both limit orders and market orders). You can choose the random quantity from a normal distribution (you can set your own mean and variance). The quantity needs to be an integer (i.e. we do not allow an odd lot). In addition, for limit orders, you can also set the price limits in the random limit orders based on a process defined by you.

9. Everytime a trade happens (buy and sell orders are succesfully matched), output the execution price to an external text file.

10. Plot your price process in the Excel. How does your price fluctuation depend on the parameters you set?

11. Create a function to compute the bid-ask spread as the difference between the price of the best outstanding buy limit order and sell limit order in the order book.

12. Everytime an order is submitted, regardless of whether it triggers an execution or not, compute the bid-ask spread and output the computed bid-ask spead to an external text file.

INSTRUCTIONS FOR STUDENTS (4)

13. Plot your bid-ask spreads in the Excel. How does your bid-ask spreads depend on the parameters you set?

INSTRUCTIONS FOR STUDENTS (5)

NEOMA BUSINESS SCHOOL
REIMS · ROUEN · PARIS

- Use the print instructions to display in the IPython console the results.
- The professor should be able to generate all your results by just clicking the « run » Button.
- Coding clarity is strongly recommended.
- Comments describing your calculation steps are welcome.
- You MUST not share you code with other groups. All groups whose Python programs are highly similar will suffer heavy grade reduction or even fail.
- You MUST not share this document with any other person.