Econometrics, Python and VBA

Python project

Summary :

The principle of this project is to build a simulation of an ideal trading bot which could take the best trading decisions on cryptocurrency historical time series.

The inputs are the name of the chosen cryptocurrency pairs (example: BTCEUR, ETHBTC, etc...), and the amount of money available in your customer wallet. A dedicated API exchange platform will be used for this project.

The idea is to take the price curve of the cryptocurrency over a certain period, to split it in tables corresponding to each day, to find minimum and maximum price of the current day. If the minimum price arrives before the maximum, calculate the equivalent trading gain considering that a buy operation has been done at the minimum price and a sell one at the maximum price. If the minimum arrives after the maximum, no trading operation is done for this day.

The outputs are a list all the trading operations done over the period. The information of each trading are stored in a dictionary :

```
[
  {‘buy_date’: ‘2022-03-30 10:00’, ‘sell_date’: ‘2022-03-30 23:14’, ‘gain’: 0,74BTC, ‘relative gain’: 1%},
  {‘buy_date’: ‘2022-03-29 05:39’, ‘sell_date’: ‘2022-03-29 07:17’, ‘gain’: 0,01BTC, ‘relative gain’: 0,15%}
]
```

Project breakdown:

Enter the inputs (3 points)

Enter the name of the currency source (the currency contained inside the wallet: EUR, BTC, ETH...) and currency target (the currency to trade: ADA, BNB...), and the available amount in your customer wallet. You can use “input” python function and make the good type conversion.

The start and end date of the period to consider will also be precised.

Connect to the platform exchange and fetch the price curve history (3 points)

For the purpose of this project, a dedicated exchange platform will be used. The url of the API will be http://51.38.188.218/kedge/api/. The allowed requests will be :

- all_currencies : get the list of all currencies

- get_order_book/{symbol}?start={start_day}&end={end_day} : returns a list of pairs of prices (the bids and the asks) for the currency pair from start to end with a time step of 1 min. The data will formatted this way :

```
[
  {
    "time": 1648594559.606206,                    # timestamp in s
    "bids": [["4.00000000", "431.00000000"]],      # price, quantity
    "asks": [["4.00000200", "12.00000000"]]
  }, …
]
```

To access this API, you can use the python library request. An example of request :

```
>>> from requests import get
>>> url = 'http://51.38.188.218/kedge/api/get_order_book/ETHBTC?start=2022-01-01&end=2022-02-01'
>>> response = get(url)
>>> for line in response.json():
>>>           print(line)
```

Build buy and sell DataFrame (3 points)

Parse the data received from the exchange and format it to build a pandas DataFrame with a column of prices to buy (from the “asks” part of the order book) and a column of prices to sell (from the “bids” part of the order book). Format the time value expressed in seconds to a date with the format “2022-01-01 22:10:03” and use it as an index column in the DataFrame.

Split according to the index of the global DataFrame to obtain a DataFrame for each day.

Calculate the best trade of each day (8 points)

Make a loop to treat each DataFrame for each day. Find the minimum and maximum price of the day. If the minimum arrives before the maximum, it means that for this day, an ideal trader could have bought the currency at the minimum price and sold it at the maximum price. In that case the trade can recorded in a dictionary. You can calculate the amount gained and the relative gain. For instance :

```
{‘buy_date’: ‘2022-03-30 10:00’, ‘sell_date’: ‘2022-03-30 23:14’, ‘gain’: 0,74BTC, ‘relative gain’: 1.07%}
```

If the minimum arrives after the maximum, no trade are done for this day.

Print the result (3 points)

Present the result in a DataFrame which can be printed on screen. Display a bar graph to show the curve of the gain and another one for the relative gain over the chosen period.

Bonus (5 points)

An interesting feature could be to be able to enter a date and to to display the curve of the prices for this day and mark a dot to show when the buy and sell operation were done during that day.

Another bonus could be the possibility to search for the best trade of the day even if the minimum arrive after the maximum. It implies to find a minimum value, which is not the minimum of the day, followed by a maximum.