**Task: Interday Swing Trade Visualization**

1. Import the following modules at the start of your algorithm

A black and orange text

Description automatically generated

2. Create the following variables:



3. Create a list of all the tickers from task 21

A close-up of a computer code

Description automatically generated

4. Obtain the day by day volume (over the past 3 months) for each stock in “tickers” by making a request to the Polygon.io (<https://polygon.io/docs/stocks/get_v2_aggs_ticker__stocksticker__range__multiplier___timespan___from___to>) and you will get a JSON object as a return value. The response will also contain the day by day closing prices (over the past 3 months).

A screenshot of a computer

Description automatically generated

Hint: Use an f-string with the two variables from step 2 for the “from” and “to” parameters

4. Use the statistics module to find the average volume of each stock over the past 5 days

5. Use the statistics module to find the average volume of each stock over the past 3 months

6. Create a new empty list called “highvolume”

7. For each ticker in “tickers”, create a function that checks to see if the average volume over the past 5 days is greater than the average volume over the past 3 months. If so add the ticker to the “highvolume” list.

8. Create a new empty list called “trendup”

9. Create a new empty list called “trenddown”

10. For each ticker in “highvolume”, create a function that checks to see if a stock has had an increase in closing price five days in a row over the past five days. If so, add the ticker to the “trendup” list

11. For each ticker in “highvolume”, create a function that checks to see if a stock has had a decrease in closing price five days in a row over the past five days, If so, add the ticker to the “trenddown” list

12. Print out the “trendup” and “trenddown” lists

13. For each stock in both “trendup” and “trendown” obtain the day by day data by making a request to the Polygon.io (<https://polygon.io/docs/stocks/get_v2_aggs_ticker__stocksticker__range__multiplier___timespan___from___to>) and you will get a JSON object as a return value.

A screenshot of a computer

Description automatically generated

Hint: Use an f-string with the two variables from step 2 for the “from” and “to” parameters

14. For each stock in both “trendup” and “trendown” create a list of the stock’s day by day open price

15. For each stock in both “trendup” and “trendown” create a list of the stock’s day by day close price

16. For each stock in both “trendup” and “trendown” create a list of the stock’s day by day high price

17. For each stock in both “trendup” and “trendown” create a list of the stock’s day by day low price

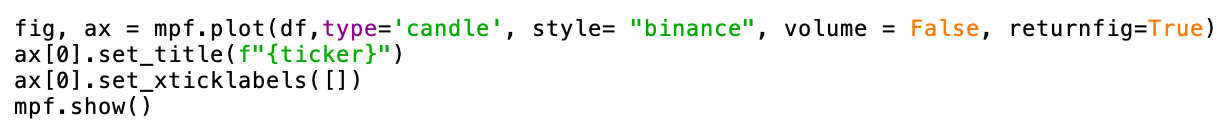
18. For each stock in both “trendup” and “trendown” create a list of the stock’s day by day volume

19. For each stock in both “trendup” and “trendown” create an empty dataframe using pandas

20. For each stock in both “trendup” and “trendown” add 5 columns to the data frame (one for each of the following: open, close, high, low, and volume.

21. For each stock in both “trendup” and “trendown” manipulate the first column in the dataframe so that it displays the time. It doesn’t matter which time you use since we will be omitting the x-axis labels.

22. For each stock in both “trendup” and “trendown” display the dataframe using the following code:



23. Save your algorithm as a “.py” file in your documents folder:

A screenshot of a computer

Description automatically generated

24. Run your algorithm from the terminal (change directory to documents before running code)

A screenshot of a computer

Description automatically generated