Portfolio Programming Assignment- Improving the Stock Problem with Additional Functionality for

Master of Science

Information and Communications Technology

Anna Sheffield

University of Denver University College

[June 5th, 2022]

Faculty: Abeer Khaleq

Dean: Michael J. McGuire, MLS

Reflection Component:

- *Describe the updates you included in your work based on the instructor feedback given in previous submissions. Are there areas you would like to still improve it in?*

The updates that I added to my code include creating more detailed classes to be called through my module file. Cleaning up the classes and adding some additional classes (such as load_stocks, load_bonds, add_data) allow my main coding file to be more organized and clean. I also removed the index assignments for my Stocks and Bonds files as Pandas automatically assigns indexes. I also created a new column for my JSON databases to include "New Close" values for each individual stock log (determined by quantity X close) so that graphing the Y Axis would be easier as the math would completed earlier in the code and my loops wouldn't be as confusing. Finally, I included additional comments for the areas of my code that performed functions and explained what the functions did to easily see steps I took for each outcome.

I think my code could be improved still by combining some of the functions into existing classes to shorten the main code, however when I tried to do this- my IDE was having issues calling the last function of a Class, so I kept them separated so that it would run for our purposes. I also think my exceptions could be improved upon and enhanced to test for additional possible errors.

- *Can you think of other functionality that would be beneficial?*

I think adding more data for the Stocks is beneficial. I decided to add a bar graph showing the overall number of individual stocks purchased per Symbol so users can see which stocks they purchased the most of overall.

- *What was your experience implementing the new functionality?*

I had a lot of trouble implementing this new graph and it caused me to go over the due date time. I couldn't figure out how to get the number of items for each stock and then put that into a format in which the bar graph could read. I wanted to combine two of the options for adding new functionality: using pygal and also graphing new data points that weren't calculated before.

- *Was the functionality hard to implement?*

Yes because I am new to pygal and data visualizations overall.

- *Was the documentation easy or difficult to find?*

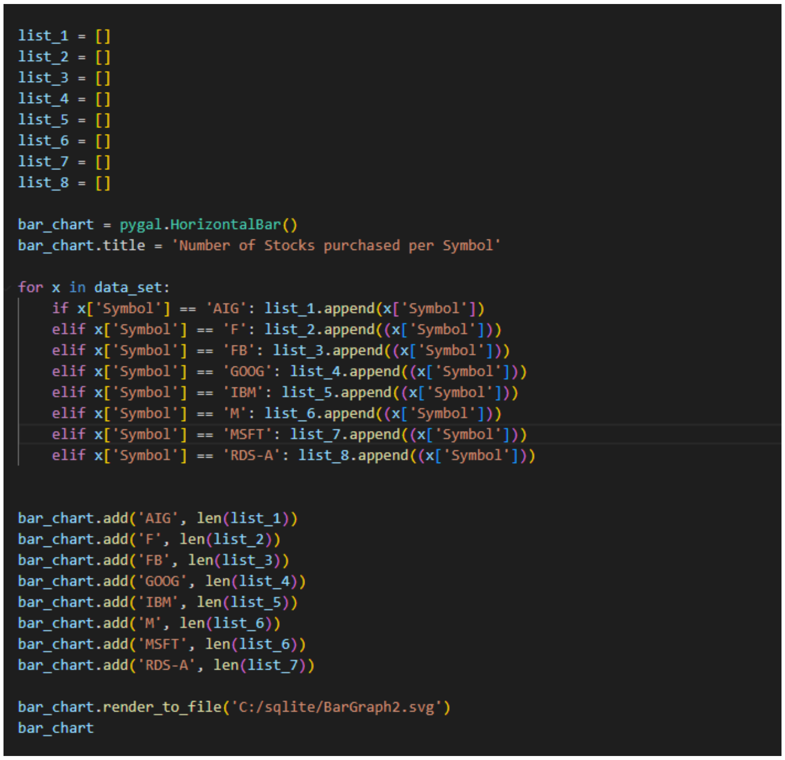
The documentation for the overall concept was easy to find but I couldn't find an example of a loop to import data points into a graph individually using pygal.

- *Was the documentation difficult to interpret?*

Yes, see above

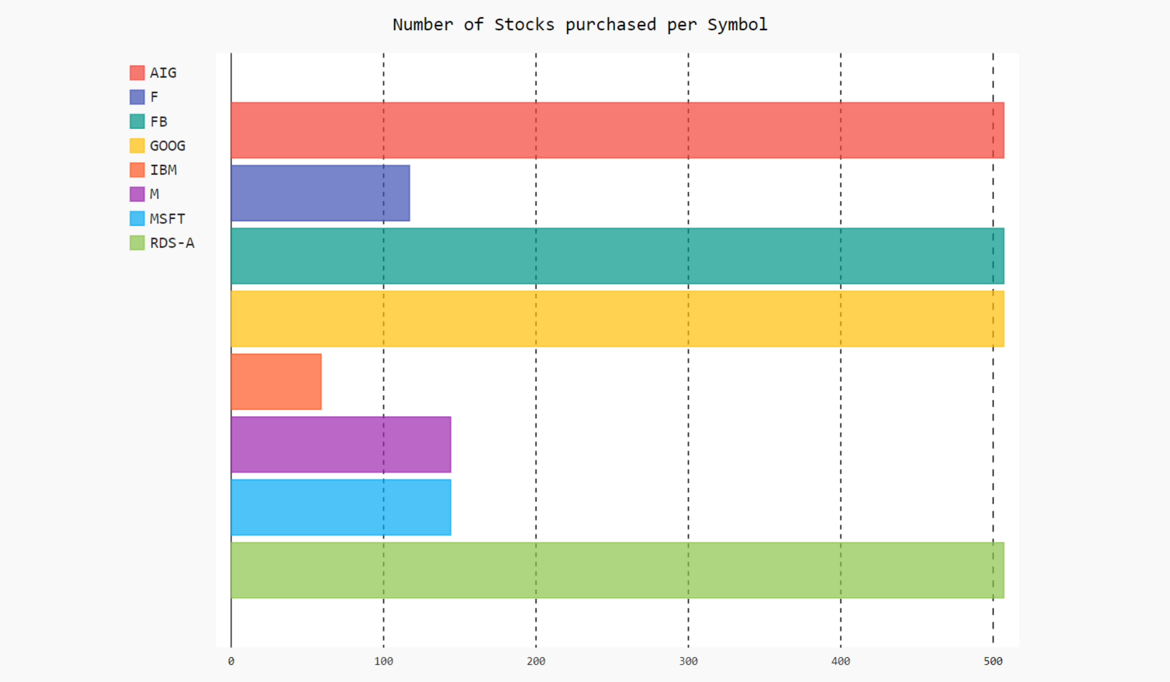
- *Show screenshots of your added functionality at work.*

Screenshot of the added code:



```
list_1 = []
list_2 = []
list_3 = []
list_4 = []
list_5 = []
list_6 = []
list_7 = []
list_8 = []

bar_chart = pygal.HorizontalBar()
bar_chart.title = 'Number of Stocks purchased per Symbol'

for x in data_set:
    if x['Symbol'] == 'AIG': list_1.append(x['Symbol'])
    elif x['Symbol'] == 'F': list_2.append((x['Symbol']))
    elif x['Symbol'] == 'FB': list_3.append((x['Symbol']))
    elif x['Symbol'] == 'GOOG': list_4.append((x['Symbol']))
    elif x['Symbol'] == 'IBM': list_5.append((x['Symbol']))
    elif x['Symbol'] == 'M': list_6.append((x['Symbol']))
    elif x['Symbol'] == 'MSFT': list_7.append((x['Symbol']))
    elif x['Symbol'] == 'RDS-A': list_8.append((x['Symbol']))


bar_chart.add('AIG', len(list_1))
bar_chart.add('F', len(list_2))
bar_chart.add('FB', len(list_3))
bar_chart.add('GOOG', len(list_4))
bar_chart.add('IBM', len(list_5))
bar_chart.add('M', len(list_6))
bar_chart.add('MSFT', len(list_6))
bar_chart.add('RDS-A', len(list_7))

bar_chart.render_to_file('C:/sqlite/BarGraph2.svg')
bar_chart
```

Screenshot of the bar graph output from JSON file metrics:


Number of Stocks purchased per Symbol
AIG
F
FB
GOOG
IBM
M
MSFT
RDS-A