TASK 3 REPORT

COS30018

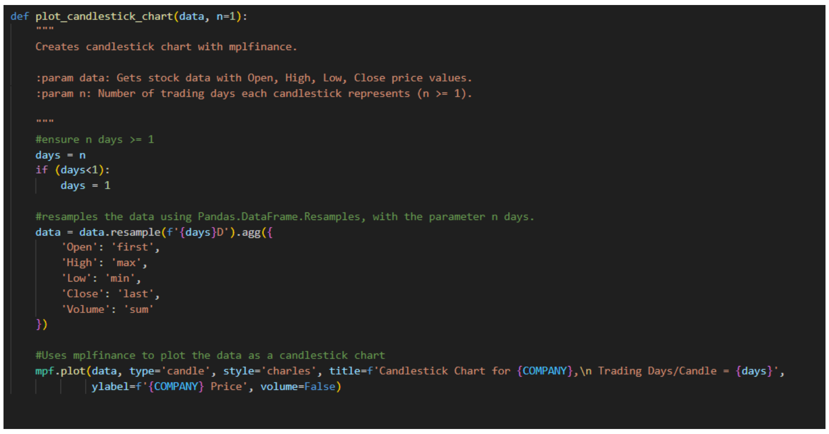
INTELLIGENT SYSTEMS

Halim Vlahos (104015943)

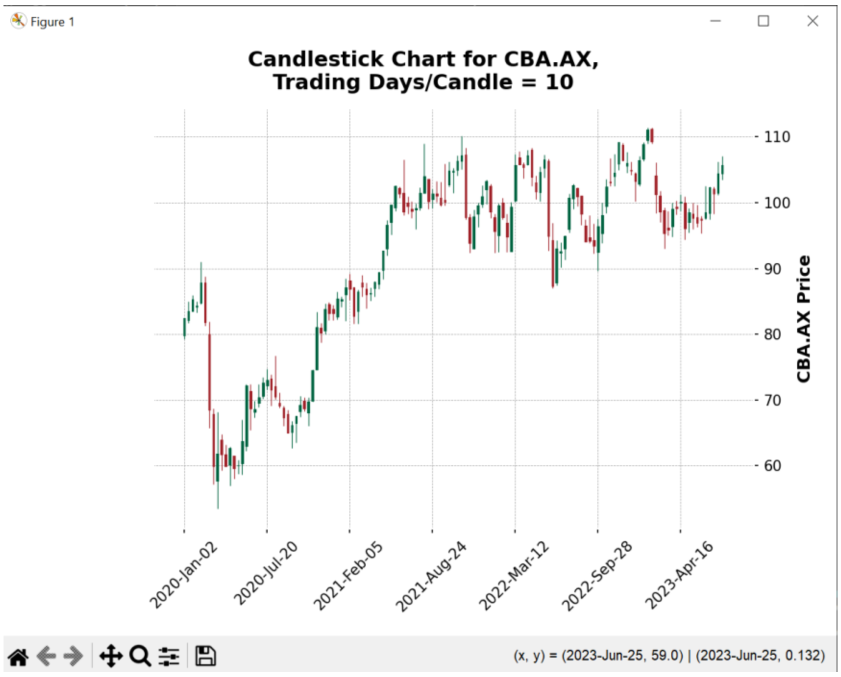
CANDLESTICK CHART SUMMARY

The candlestick chart was simply implemented using mplfinance, with the plot type set as 'candle'. To add the ability to set the parameter n as the days represented per candlestick, I had to use pandas.resample (combined with the agg function) with the parentheses as n days ('D').

https://www.alpharithms.com/aggregating-time-series-data-with-pandas-resampling-411212/



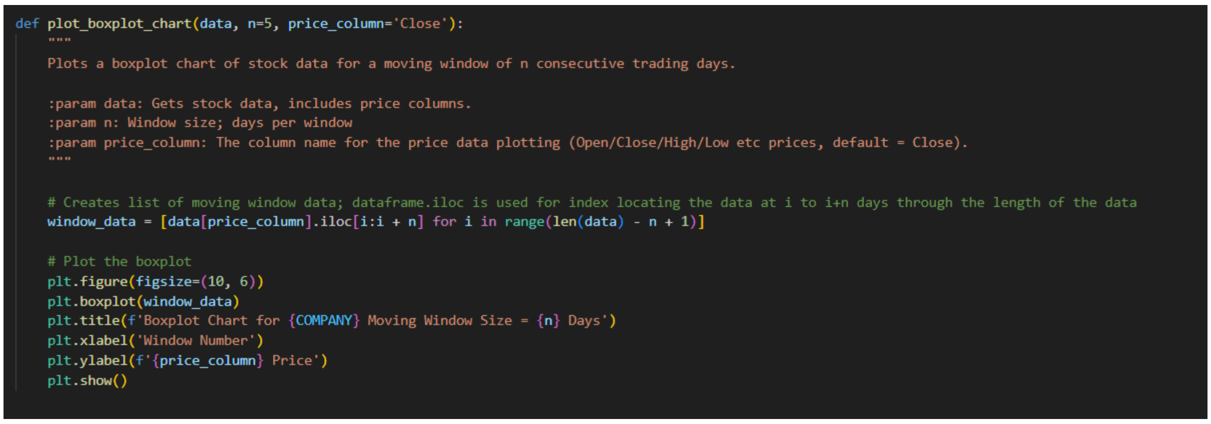
```python
def plot_candlestick_chart(data, n=1):
    """
    Creates candlestick chart with mplfinance.

    :param data: Gets stock data with Open, High, Low, Close price values.
    :param n: Number of trading days each candlestick represents (n >= 1).

    """
    #ensure n days >= 1
    days = n
    if (days<1):
        days = 1

    #resamples the data using Pandas.DataFrame.Resamples, with the parameter n days.
    data = data.resample(f'{days}D').agg({
        'Open': 'first',
        'High': 'max',
        'Low': 'min',
        'Close': 'last',
        'Volume': 'sum'
    })

    #Uses mplfinance to plot the data as a candlestick chart
    mpf.plot(data, type='candle', style='charles', title=f'Candlestick Chart for {COMPANY},\n Trading Days/Candle = {days}',
            ylabel=f'{COMPANY} Price', volume=False)
```



BOXPLOT CHART SUMMARY

The Boxplot Chart was plotted using matplotlib.pyplot (as plt). The data for the moving window was to be stored in a list, which was created from (price column) values taken from looping through the dataset, each index of the list being a window of the data at index days up to n days ahead.

```
window_data = [data[price_column].iloc[i:i + n] for i in range(len(data) - n + 1)]
```



```python
def plot_boxplot_chart(data, n=5, price_column='Close'):
    """
    Plots a boxplot chart of stock data for a moving window of n consecutive trading days.

    :param data: Gets stock data, includes price columns.
    :param n: Window size; days per window
    :param price_column: The column name for the price data plotting (Open/Close/High/Low etc prices, default = Close).
    """

    # Creates list of moving window data; dataframe.iloc is used for index locating the data at i to i+n days through the length of the data
    window_data = [data[price_column].iloc[i:i + n] for i in range(len(data) - n + 1)]

    # Plot the boxplot
    plt.figure(figsize=(10, 6))
    plt.boxplot(window_data)
    plt.title(f'Boxplot Chart for {COMPANY} Moving Window Size = {n} Days')
    plt.xlabel('Window Number')
    plt.ylabel(f'{price_column} Price')
    plt.show()
```

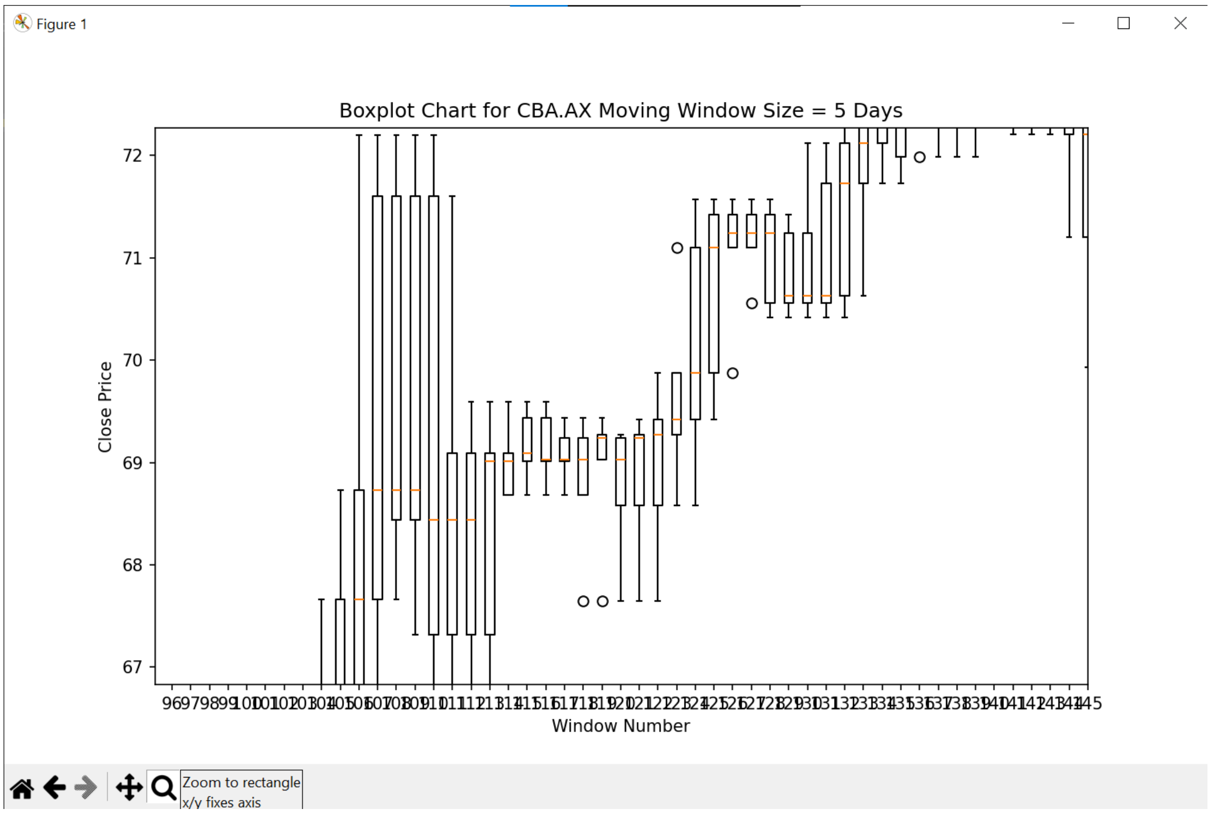


Figure 1
Boxplot Chart for CBA.AX Moving Window Size = 5 Days
Close Price
Window Number
Zoom to rectangle
x/y fixes axis