

Project B

Task 3 – Data Processing 2

A data aggregation function was created and used in both the candlestick function and the boxplot function.

```
def aggregate_data(data, n_days=1):
    if n_days > 1:
        data_resampled = data.resample(f'{n_days}D').agg({
            'Open': 'first', # Set open to be the first data point of this set
            'High': 'max', # Set high to be the highest data point of this set
            'Low': 'min', # Set low to be the lowest data point of this set
            'Close': 'last', # Set close to be the last data point of this set
            'Adj Close': 'last', # Set adj close to be the last data point of this set
            'Volume': 'sum' # Set volume to be the sum of all data points in this set
        })
    else:
        data_resampled = data
    return data_resampled
```

Candlestick chart function with parameter for grouping of days.

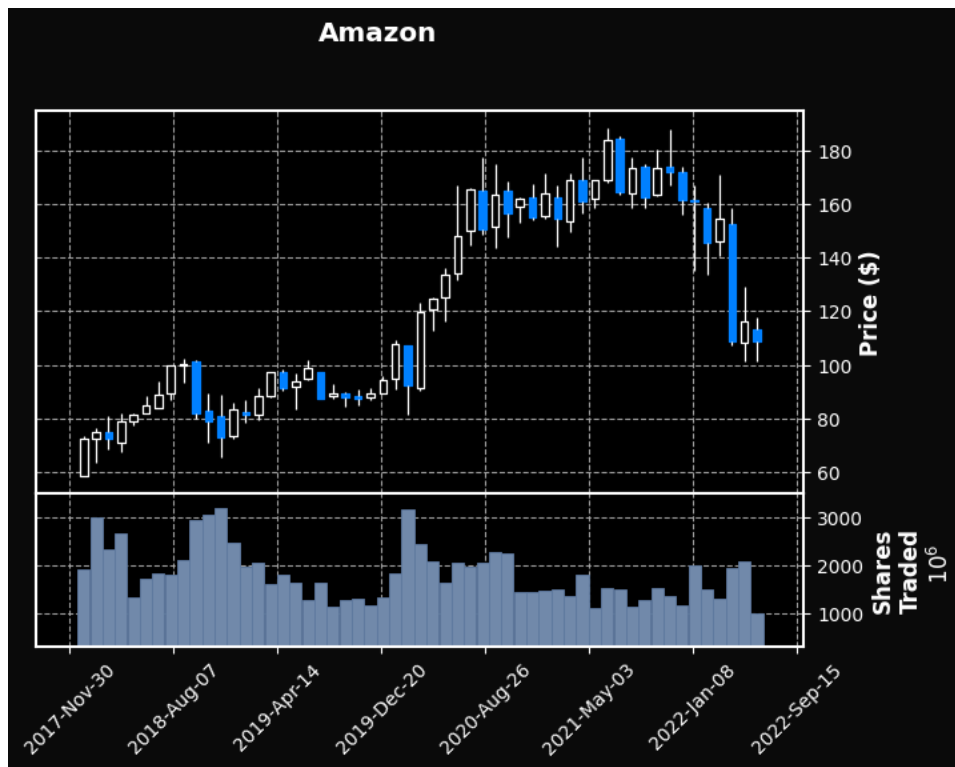
```
####
# Plots a candlestick chart according to the parameters
# Parameters:
# - data: the dataset to plot
# - volume: Bool to determine if secondary chart is plotted below main candlestick chart
# - show_nontrading: Bool to determine if NaN days are shown
# - title: String for the title of the chart
# - n_days: How many days each candlestick represents, default is 1
####

def plot_candlestick_chart(data, volume, show_nontrading, title, n_days=1):
    new_data = aggregate_data(data, n_days)

    fplt.plot(
        new_data,
        type="candle", # Type of chart
        volume=volume,
        show_nontrading=show_nontrading,
        title=title,
        style="mike", # Visual style of chart
        ylabel='Price ($)', # Y candle chart Label
        ylabel_lower="Shares\nTraded" # Y secondary chart label
    )

plot_candlestick_chart(train_data, True, True, "Amazon", 30)
```

Output of Candlestick chart function.



Boxplot chart function with parameter for grouping of days.

```
####  
# Plots a boxplot chart according to the parameters  
# Parameters:  
# - data: pandas dataframe, the dataset to plot  
# - features: Each feature to be plotted as a string in a list  
# - n_days: How many days each candlestick represents, default is 1  
####  
  
#### Why not just do all features?  
# If we do, the volume values dwarf the other values  
# Therefore we should only do it for the non volume features  
# Scaling them is also typically not good for box plots  
  
def plot_boxplot_chart(data, features, n_days=1):  
    new_data = aggregate_data(data, n_days)  
  
    # Extract the data for the selected features  
    box_data = [new_data[feature] for feature in features]  
  
    # Create figure  
    fig = plt.figure()  
  
    # Create box plots for each feature  
    plt.boxplot(box_data)  
  
    # X axis labels
```

```
plt.xlabel(features)

# Show the chart
plt.show()

plot_boxplot_chart(train_data,["Open", "High", "Low", "Close"], 30)
```

The inspiration for the boxplot chart code came from this website:

<https://coderzcolumn.com/tutorials/data-science/interactive-plotting-in-python-jupyter-notebook-using-bqplot>

Output of boxplot chart function.

