SUBMITTED BY- MAZIZ YAMIN RAHMAN(102008721)

COS30018 - Option B - Task 3: Data processing 2

In this task, we undertake the process to plot the data using desired arguments

Setting the environment for use of plotly module. The following commands are used.

1. Installing the 'plotly' module
   $pip install plitly
2. Importing the module to our project
   $import plotly.graph_objects as gp

Understanding to implement traces

1. *'trace' is an option of plotting graphs. We can use multiple traces consecutively within the same graph*
2. *We will set up the trace for the candlestick graph*

```
trace1 = {
        'x': data.index,
        'open': data["Open"],
        'close': data["Close"],
        'high': data['High'],
        'low': data['Low'],
        'type': 'candlestick',
        'name': company,
        'showlegend': True
    }
```

3. *X represent the data which is to be plotted on the x-axis. 'Open', 'high' and 'close' denote the stock prices. 'type' represents the type of graph and 'name' will be the name of the graph.*
4. *The requirements specify that we need to express the data for variable trading days. It can be done by plotting the average for a specified days.*

```
# avg_window is the input the argument to the function
  avg = data['Close'].rolling(window=avg_window, min_periods=1).mean()
```

5. *We can plot the 'avg' in the y-axis.*

```
trace2 = {
        'x': data.index,
        'y': avg,
        'type': 'scatter',
        'mode': 'lines',
        'line': {
           'width': 1,
           'color': 'blue'
        },
        'name': company,
        'showlegend': True
    }
```

6. *Plotting the graph*

```
fig = go.Figure(data=[trace1, trace2])
  fig.show()
```

We will see the output for the both the traces

7. *We need to wrap it in a function*

```
def plot_graph(avg_window):
    ...


  plot_graph(45)
```

8. *The function is called with a variable which will be no.of days and the average will be plotted.*

We can notice the difference between 45 and 90 days.