

COS30018

Intelligent Systems

Task B.3

Saynab Ismail

103063323@student.swin.edu.au

Table of Contents

Introduction.....	3
Candle Stick Chart.....	3
Boxplot Chart	4
Visual Output.....	4
Issues.....	5

Link to repository: <https://github.com/SaynabIsmail/Task-B3>

Introduction

This task aims to build two distinct functions: one for boxplot charts and another for candlestick charts, which are used to display stock market data using the provided Python script, stock-prediction.py

This report aims to document the methodology used to create these functions, including establishing the required settings, incorporating the functions into the current codebase, and extensively evaluating their results.

Candlestick chart

I started this assignment by working on enhancing the current stock-prediction.py file and writing up the candlestick function based on the YouTube video linked in the assignment requirements alongside the actual assignment requirements.

The function below works to create a candlestick chart for stock data by first resampling the data based on a specified number of trading days (`n_days`) which in this case the default value is 1 day. It then plots the resampled data using the `mplfinance` library, which creates a candlestick chart visualizing the stock's open, high, low, and close prices over the specified periods.

```
# Plot the test predictions
## To do:
# 1) Candle stick charts
# 2) Chart showing High & Lows of the day
# 3) Show chart of next few days (predicted)
# -----
def plot_candlestick(data, n_days=1): 1 usage

    resampled_data = data.resample(f'{n_days}D').agg({
        'Open': 'first',
        'High': 'max',
        'Low': 'min',
        'Close': 'last'
    })

    resampled_data.dropna()

    mpf.plot(resampled_data, type='candle', style='charles', title=f'Candlestick Chart',)
```

Figure 1: Candlestick Function

Boxplot chart

Next, I moved on to working on the boxplot function which creates a boxplot to visualize the distribution of prices over a specified period of time.

This function works by first resampling the data based on the specified number of days and then calculating the median closing price for each period. The function then uses Matplotlib to create and customize a boxplot, showing the spread and central tendency of closing prices for each period, with appropriate labels and formatting.

```
def plot_boxplot(data, n_days=1): 1 usage

    resampled_data = data.resample(f'{n_days}D').agg({
        'Close': 'median'
    })

    resampled_data.dropna(inplace=True)

    plt.figure(figsize=(12, 6))
    plt.boxplot(
        [resampled_data['Close'][resampled_data['Period'] == period] for period in resampled_data['Period'].unique()],
        labels=resampled_data['Period'].unique())
    plt.title(f'Boxplot Chart')
    plt.xlabel('Period')
    plt.ylabel('Closing Price')
    plt.xticks(rotation=45)
    plt.show()
```

Figure 2: BoxPlot Function

Visual Output

Once I had completed the code the results were as follows:

The candlestick chart, as seen below, aggregates data over a number of trading days that are customizable.

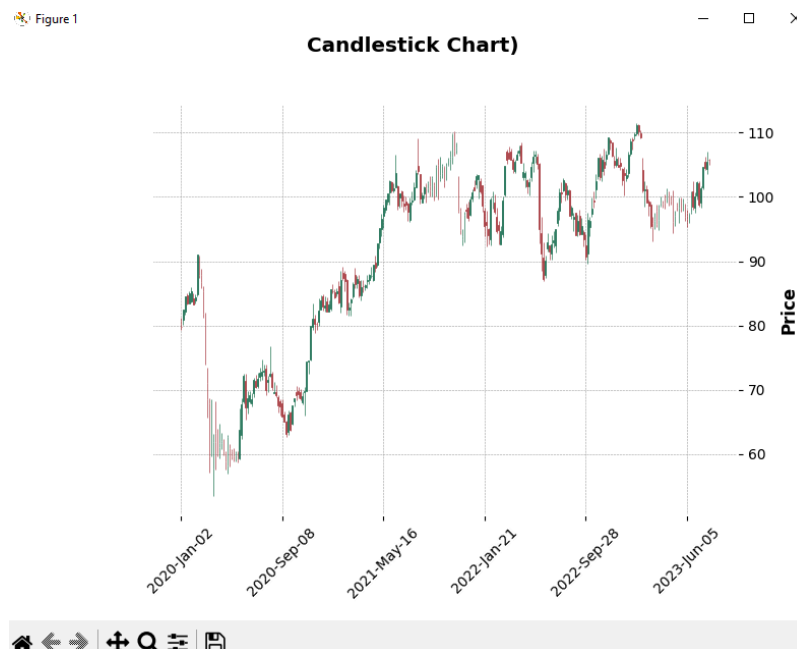


Figure 3: Candlestick chart

While the boxplot chart shows data over a sliding window of successive trading days,

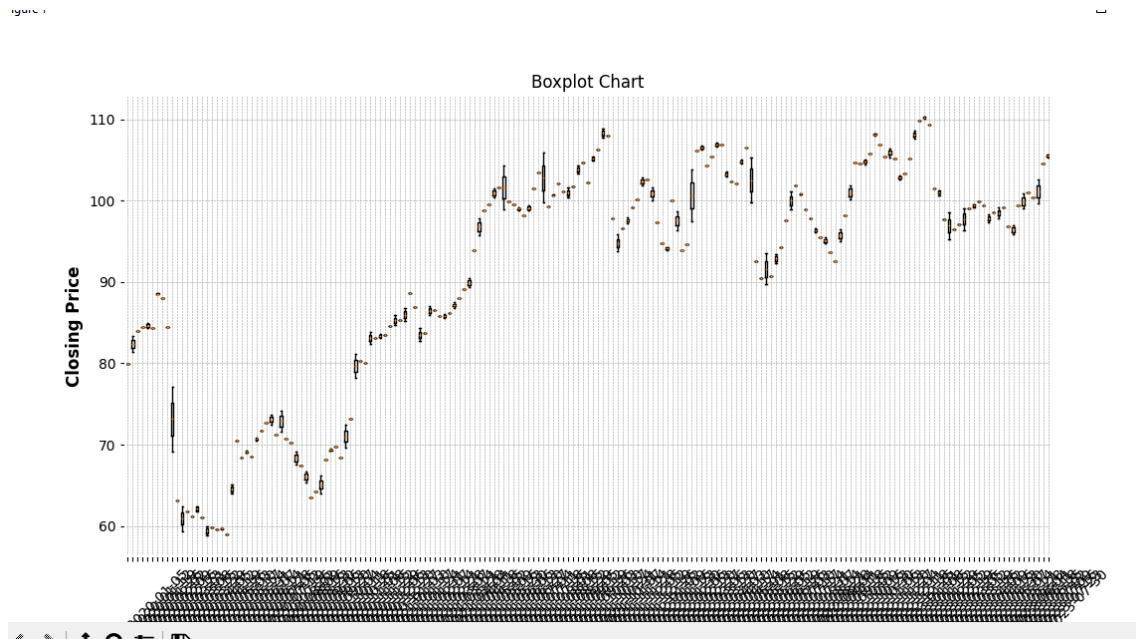


Figure 4: Boxplot chart

Issues:

I couldn't find the specific file code we were supposed to work from called v.01 so I'm not sure if this was incorrect or if there was another way to go about it but I edited the week 1 v.01 and made it my v.02?