

Assignment 3: Time series analysis and visualization

Submission deadline: 23:59, Monday, Oct. 21, 2019

Assignment 3

In this assignment, you will analyse and plot time series data. The dataset that you will be working with contains daily stock information of products or services over different periods. You need to pull them from 'Yahoo' and store them locally on your computer. The assignment contains two parts. In Part 1 you will analyse on a single product while in Part 2, you will analyse on multiple products.

The dataset:

In each part, you are required to load and save information from certain stock products (names). Use 'pandas_reader.data.DataReader' to load the dataset and set 'yahoo' as the 'data_source'. You should store the information about **each** requested stock product in a data frame. The information for each stock product includes six attributes as follows:

Open (price), High (price), Low (price), Close (price), Adj Close and Volume.
'Date' information will be read as the index of the stored data frame, by default.

Part 1:

In this part, you will need to write four functions to analyse the data from one stock variable. In the following, the specifications of each function along with the required **sample** to show the validity of your resulting output, are provided:

- 1.1. The function (show_correaltion) should receive the stock name and a period as the input arguments; reads 'Close' and 'Open' prices. Then shows the correlation between Open and Close prices over **the last business day of each month** (in the specified period).

Hint: You should use **scatter** plot to show the correlation.

Sample: stock name= 'AAPL' and period= 2017-01-03 to 2018-01-01.

- 1.2. The function (show_daily_features) should receive the stock name and a period as the input arguments, and outputs four (daily) **subplots** with the same size for each feature: line plots for 'Low' and 'High' prices, histogram plots for 'Close' and 'Open' values.

Sample: stock name= 'AMZN' and period= 2017-01-01 to 2017-04-01.

- 1.3. The function (show_monthly_average) should receive the stock name and a period as the input arguments, calculates the **monthly average** price for each feature ('Open', 'High', 'Low' and 'Close'), then outputs four **line**

plots in one graph representing changes in **average** prices over the given period. (Label in X axis: **Month** (not Day), Label in y axis: Montly average price).

Sample: stock name= 'HPQ' and period: 2018-01-01 to 2019-01-01.

- 1.4. The function (show_stock_statistics) should receive the stock name and a period as the input arguments, calculates the maximum, minimum and average values of 'Close' price per month, and outputs these monthly statistics in terms of **bar plots** in one graph over the given period. (Label in X axis: Month, Label in y axis: Close price)

Sample: stock name= 'AMZN' and period: 2018-01-01 to 2019-01-01.

- All the graphs should have labels in x-axis and y-axis; besides, they should have legends.
- Generally, you may use the 'interpolation' method of Pandas data frame for filling in missing dates. You should also drop the columns with all zero values.
- For clarification among the lines, you can either use colouring or use different line styles and legend command.

Part 2:

In this part, you provide and plot multiple stock data in one graph for the given list of stock names and dates. The function that you will write for this part should have these specifications:

- The function (show_multiple_stocks), should receive the list of stock names and the period. Then for each stock variable, it should choose and store the daily 'Close' price information (in a data frame). Then it should plot the variations of 'Close' prices of all given stock names over the given period (by line plots).

Also, the function should provide answers to the following question (as the output):

- 2.1. In which month in 2016, stock number 2 (GM) has had the highest closing price and in which month in 2016, stock number 3 (TSLA) has had the lowest Close value?

You should use pointers on the graph to show the **answers (the month** where stock number 2 has the highest price and **the month** when stock number 3 has the lowest price.

2.2 Find the year and month where each stock has had the highest average value. (Mention year, month, the stock name and the highest value in the answer, no need to use pointers here)

- To have a fair comparison among the prices of different stocks, you can normalize 'Close' values of stocks using the base value of each stock from 2016.01.04.
- Again, you may need interpolation to fill in the missing values.
- The label of 'x' axis of the resulting plot should show the **months** of the year and the label of 'y' axis should show the **normalized 'Close'** values.
- You should check and show the validity of your resulting output by the following **sample**:

show multiple_stocks (stock list, period), where the stock list is ('F', 'GM' and 'TSLA') representing Ford, General Motors and Tesla stocks and the period is from 2016.01.04 to 2017.12.30 (a date range with daily interval)

Grading:

- The result of grading will be either 'Passed' or 'Failed'. Your grade will be assessed on the performance and result of your code. The accepted performance means that you should at least provide 80 per cent of the solutions in each part.

Submission guideline:

- Deadline: 23:59, Monday, Oct. 21, 2019 (submit your assignment through Canvas).
- Source code submitted for the assignments should be your own code. If you use the codes from the internet or if you use someone's code without referencing, that will be treated as plagiarism and you will fail the assignment.
- Source code should be written in one Jupyter notebook files. Filenames could be the student's first name, last name and name of the assignment (firstname.lastname.timeseries).
- The assignment is individual and can NOT be solved in groups.
- The deadline cannot be extended.
- If you have any questions, please send an email to this address: aida.mehdipourpirbazari@uis.no.