

# Analyzing the Casablanca Stock Exchange data project

*Supervisor: Pr. A. Ratnani*

*Co-Supervisors: J. Maaouni & N. Ouhammadou*

The stock exchange is a regulated market on which securities of listed companies are traded. Founded in 1929 as a clearing house, the status and organization of the stock exchange have known several structural reforms. In 1995, a management company was established and was granted the management of the stock exchange, pursuant to a set of specifications approved by the Ministry of Finance. This late overhaul of the status of the stock exchange was undertaken as part of a comprehensive modernization of the financial market, initiated in 1993, which had as an objective to endow the stock exchange with a modern organizational structure and means to strengthen its role in the financing economy.



The Casablanca Stock Exchange is a société Anonyme with a Board of Directors and a Supervisory Board. It holds the public service concession for managing the stock market. Its share capital is jointly owned by brokerage firms which have sole authority to execute stock market trades

## Objectives:

In this guided project you will gain first-hand experience with a real problem, apply the skills that you learned before, and be proficient in:

- Data collection and cleaning.
- Data Analysis and visualization with Pandas and Matplotlib.
- Python object-oriented programming.
- Problem-Solving and learning new algorithms used in Finance
- Code optimization and acceleration.

The project is guided and split into several tasks to be easy for you, you will find at the end a section for resources that you may need. don't forget to search on the internet you may find a better and easy solution for some tasks.

**Estimation time:** 8 hours

## Part I: Data Collection

In this section, we will collect the data that we will work on in the following sections. For this project, we will be working with stock data from Bourse de Casablanca.

- Head over [here](#) and select a sector with at least 5 instruments. (ie Assurances, Banques, Bâtiments et matériaux de construction...).
- From this sector, select 5 instruments that you will work on.
- For each instrument, head over to its data page by clicking on it. For example, [this](#) is the page for Aluminium du Maroc. Choose the historique tab, and toggle the selection to Par période, then, choose 01/01/2020 for Date de début and 01/12/2022 for Date de fin..



Click on the Télécharger **les données** icon to download the data. The data is stored in a .aspx file.

### Task 1:

- Create a function that takes the file you just downloaded and returns a CSV file with the following columns:

```
date,closing,adjusted,evolution,quantity,volume
01/12/2022,1449.00,1449.00,0.00,0,0
30/11/2022,1449.00,1449.00,-1.43,10,14522
29/11/2022,1470.00,1470.00,0.00,0,0
28/11/2022,1470.00,1470.00,0.00,0,0
25/11/2022,1470.00,1470.00,-2.33,140,201861
```

**Be careful:** The values in the original format use a comma instead of a point for the decimal point. Make sure to take this into account.

## Part II: Data Processing

Now that we have collected the necessary data, it is time to make use of it.

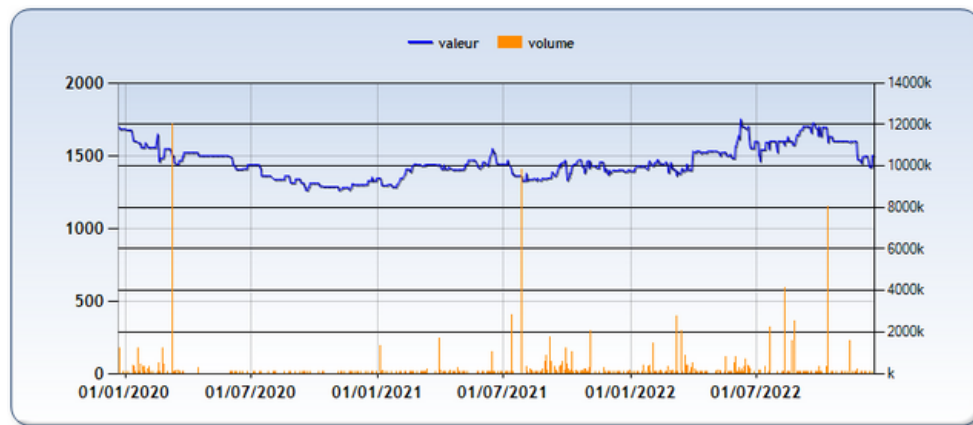
### Task 2:

- Create a class Stock that will represent stock in this project. This class should have at least the following instance attributes:
  - **name:** The name of the stock it represents.
  - **data:** A Pandas Dataframe generated from the CSV of the previous section.

**Be careful:** The values in the original format use a comma instead of a point for the decimal point. Make sure to take this into account

### Task 3:

- Inside Stock, create a method responsible for visualization. It should be able to create something similar to the graph present on the website.



Feel free to add extra functionality for extra credit.

Make sure to add a docstring to your method to summarize its inner workings to the user.

### Task 4:

- Inside Stock, create the following instance methods:
  - a method that returns the **maximum value** in the Dataframe.
  - a method that returns the **minimum value** in the Dataframe.
  - a method that returns the **maximum volume** in the Dataframe.
  - a method that returns the **maximum quantity** in the Dataframe.

These methods will be called a lot. Make sure to take this into consideration while implementing them.

### Task 5:

- Inside Stock, create a method responsible for visualization. It should be able to create something similar to the graph on the website.

### Task 6:

- Inside Stock, create a method that calculates the momentum for the given stock for any value of N. Your method should also accept a list of values for N; ie, it should be able to compute the momentum for various values of N at once. This behavior is tantamount to calling the method for each N in the list.

**Be careful:** Expensive computation requires optimization.

### Task 7:

- Inside Stock, create a method that calculates the simple moving average for the given stock.

**Be careful:** Expensive computation requires optimization.

### Task 8:

- Extend the method you implemented previously for visualization to include the two indicators you just implemented.

### Resources:

- Money flow index: Wikipedia article [https://en.wikipedia.org/wiki/Money\\_flow\\_index](https://en.wikipedia.org/wiki/Money_flow_index)
- Momentum : [https://en.wikipedia.org/wiki/Momentum\\_\(technical\\_analysis\)](https://en.wikipedia.org/wiki/Momentum_(technical_analysis))
- Simple moving average [https://en.wikipedia.org/wiki/Moving\\_average](https://en.wikipedia.org/wiki/Moving_average)