



OPTION B - DATA PROCESSING 2

TRAN DUC ANH DANG

Student ID: 103995439



Requirements

Virtual Environment

- Google Colab
- Jupyter Notebook

Dependencies

- numpy
- matplotlib
- mplfinance
- pandas
- scikit-learn
- pandas-datareader
- yfinance
- pandas_ta

Installation

*Note: Anaconda is required unless Google Colab is being used

Anaconda/Virtual Environment

1. Download Anaconda: Go to the Anaconda website (<https://www.anaconda.com/products/distribution>) and download the appropriate version for your operating system.
2. Install Anaconda: Follow the installation instructions for the OS from the Anaconda website.
3. Open Anaconda Navigator: Launch Anaconda Navigator from your installed applications.
4. Create a New Environment (Required): Create a new environment to isolate Jupyter installation on each project. Click on "Environments" in Navigator and then "Create" to make a new environment.
5. Install Jupyter Notebook: In the selected environment, click on the environment name and select "Open Terminal". In the terminal, type: `conda install jupyter`.

Dependencies

In Google Colab or Jupyter Notebook, it can directly install the required dependencies using the `!pip` command in code cells. Here's an example of how to install the dependencies:

`!pip install <package> or !pip install -r <text file>`

Data Processing 2

Candlestick Plotting Function

Summary

The `'plot_candlestick'` function is designed to plot candlestick charts of stock market financial data. The function allows for the visualization of Open, High, Low, Close, and Volume data over time. Additionally, it provides an option to represent each candlestick with data from 'n' trading days.

Function Signature and Parameters

The function accepts the following parameters:

- 'input_df': A DataFrame containing stock market financial data with columns like 'Open', 'High', 'Low', 'Close', and 'Volume'.
- 'n': An optional parameter specifying the number of trading days each candlestick should represent. The default value is 1.

Code Explanation

Creating a Copy of DataFrame

- A copy of the input DataFrame is made to avoid any warnings related to modifying the DataFrame in-place.

Data Resampling for n Trading Days

- If $n > 1$, the data is resampled to represent n trading days. The 'Open' price is the first price, 'High' is the maximum, 'Low' is the minimum, 'Close' is the last, and 'Volume' is the sum for these n days.

Calculating Moving Averages

- Calculating 50, 100, and 200 days moving averages for the 'Close' prices and storing them in new columns 'MA50', 'MA100', 'MA200'.

Additional Plot Configurations

- Configuration for additional plots of moving averages with specific colors is set.

Plotting the Candlestick Chart

- The candlestick chart is plotted using the 'mpf.plot()' function from the mplfinance library, with additional settings like chart type, style, titles, and labels.