

Final Project: Stock Exchange Performance Analysis

Ava De La Cruz, Chanhui (Daniel) Lee, Ian Wall

1. A summary overview

Volatility is a crucial metric for evaluating stock exchange performance as it measures the frequency and magnitude of price movements over time. The analysis of volatility provides essential insights into the stock exchange's behavior and risk assessment. There are multiple ways to calculate volatility, but for this project, we will focus on only two measurements: "daily price range" and "percent change." The daily price range could be calculated by subtracting the lowest price from the highest price. The percent change could be calculated by dividing the difference between the current close price and the previous close price by the previous close price, then multiplying the result by 100.

2. Description of the project's functionality

This project should read the CSV file that was edited from a public CSV file¹ containing daily index prices for multiple stock exchanges and allow user-selection for a specific stock exchange and typing the user's desired file name for the volatility result file. Then, through the separate Python file that runs the calculation for extracting "Daily Price Range" and "Percent Change," the project should write a new file with columns including "Daily Price Range" and "Percent Change." Also, the project should plot graphs of the selected stock exchange's performance based on three subplots constituted with the actual price movement, "Daily Price Range," and "Percent Change." As mentioned above, the user can decide one stock exchange from 5 different stock exchanges for countries around the world—the US, Hong Kong, Japan, Germany, and South Africa.

3. Instructions

Below you will find a folder containing all pertinent information, which you will need to download onto your computer to properly run the code.

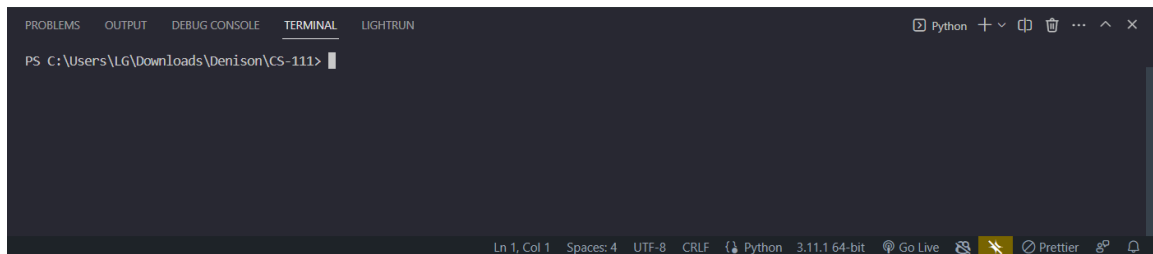
<https://drive.google.com/drive/folders/1CMENmUTj5b2btOgCopnpA3MPcSCO8seP?usp=sharing>

¹ <https://www.kaggle.com/datasets/mattiuzc/stock-exchange-data?select=indexData.csv>

To run this project, download indexData.csv and all four Python files, including main.py, fileUtils.py, analysis.py, and plot.py. Ensure you downloaded all these files into the same folder and you should check your current working directory in the terminal to see if you are in this folder.

For example, we have downloaded the files to the following path, so the terminal should show the path ending with '\CS-111'.

C:\Users\LG\Downloads\Denison\CS-111



Additionally, if you haven't installed the Matplotlib package and Pandas library before, successively run the following commands in the terminal.

```
pip install matplotlib
pip install pandas
```

If you have downloaded and located all the files in the appropriate path, you can run this project by running main.py, and it will ask you to select one stock exchange by typing the index (IXIC, HSI, GDAXI, J203.JO, N225). It will only proceed to the next step if you have typed one of the five indexes in the parenthesis in the exact format (e.g., it will not proceed if you have typed "ixic" since it is case-sensitive.). The same question will appear if you have entered something that is not one of the five indexes in the parentheses, or if you enter one of the five indexes in the lower case.

The next question will ask you to type your desired file name for the volatility result file. As mentioned in the question, you do not need to specify the file format, as this project will output the result file only in CSV format. If you specify a file format, for example, typing test.pdf, this program will simply regard the typed text as a whole file name and output the CSV file without causing an error.

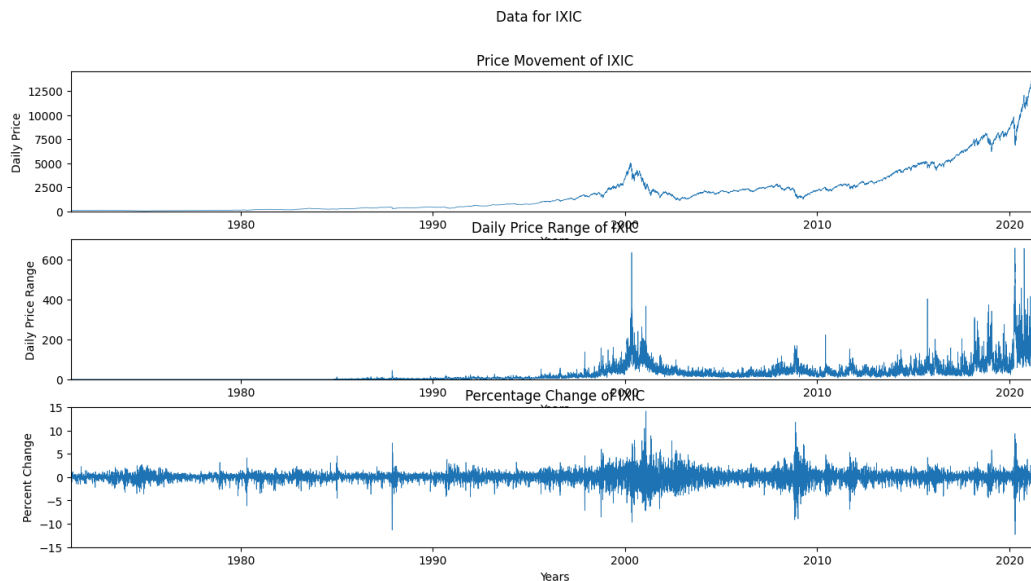
If you have successfully answered the two questions, the program will show you the graph of your selected stock exchange based on three subplots constituted with the actual price movement, "Daily Price Range," and "Percent Change." For the clearest view,

ensure that the window with the graphs is full screen. Also, the volatility result file would have been created in the same folder that you are located in.

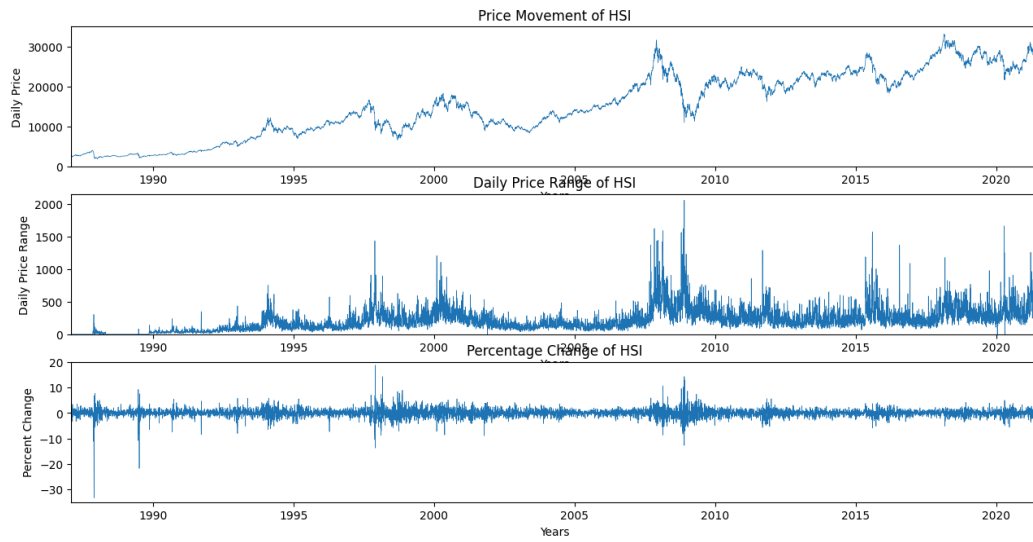
You can run unit test functions for fileUtils.py and analysis.py but they should not be operated while the main.py is running. For running unit testing, you can simply navigate to either fileUtils.py or analysis.py and run the file. If the unit tests are successfully finished, the terminal will show multiple statements that describe each function in the file having passed all tests. Unit test functions for plot.py were not created for this project because it only plots the data.

4. Results

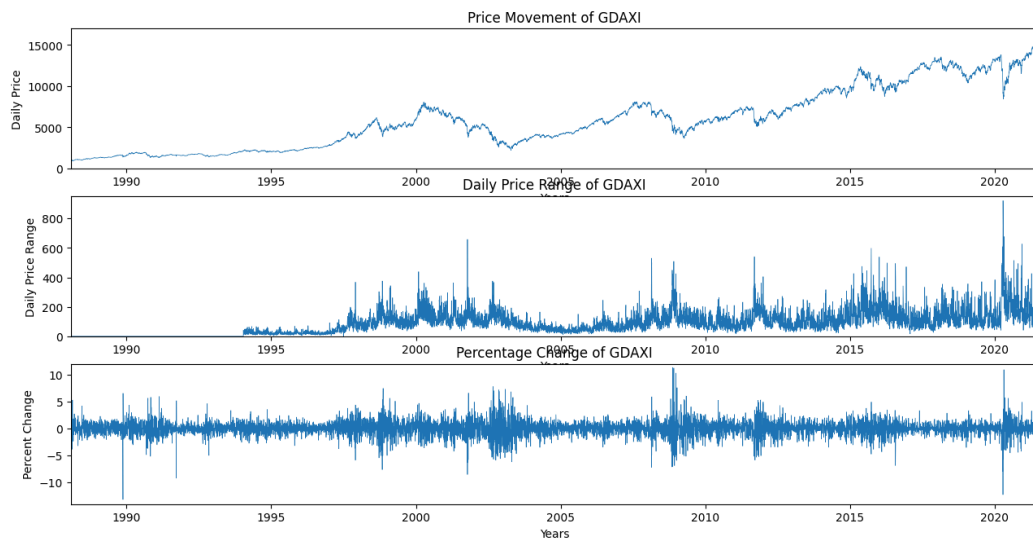
These are the graphs the file produces when the user selects a stock exchange. The name of the specific stock exchange is at the top of each screenshot, and each graph is labeled according to the data it reflects. The top graph shows the price movement based on the closing data at the end of each day. The middle graph shows the price range, which we found by subtracting the lowest price from the highest price. The bottom graph shows the percent change from day to day of the stock.

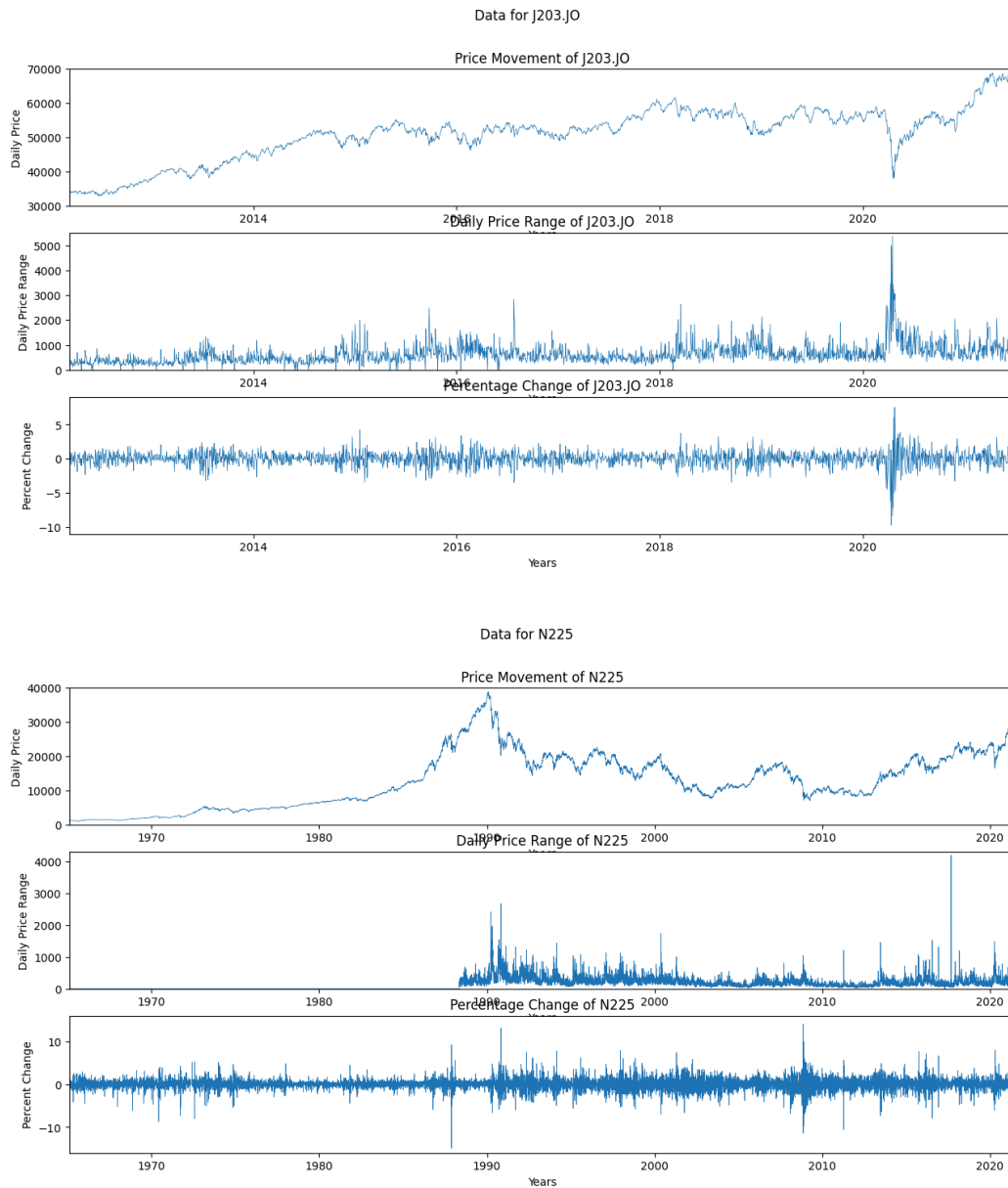


Data for HSI

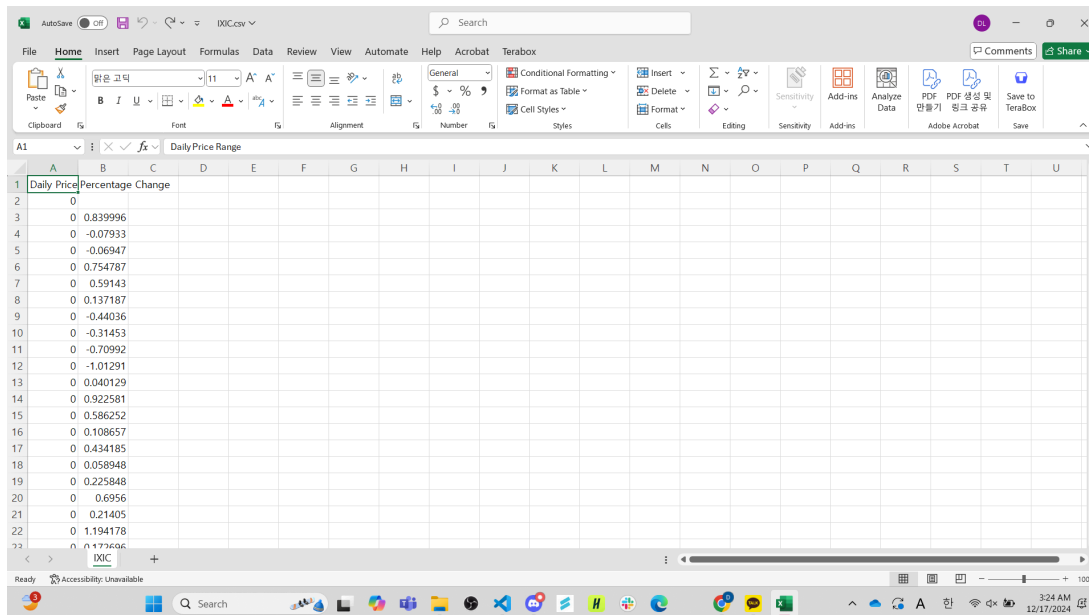


Data for GDAXI





The following image shows the example volatility result file named IXIC.csv created by selecting IXIC for the input.



5. Commentary on the project

Our project offers a way for users to look at data trends using select stock exchanges over the course of many years. It is representative of countries around the world and reflects their different major events. For example, we see that each country suffered from a dip in their stock exchange right after 2020, likely due to their varying responses to COVID-19. In the case of South Africa, one can easily see that these drops were rapid with slower recoveries in the coming years. This is also easily seen in Hong Kong's graph, as their recovery process was slow at first and then ended in an upward trend towards the end of 2021. In graphs with more data, such as America and Japan, the spike is far more drastic, and in the case of America specifically, it is continuing to climb by late 2021.

Throughout the process of creating the project, we learned more about working with CSV files and the Pandas library. If others were to build upon it in the future, they could offer the user more options or data to interact with, such as the stock exchanges of other countries, and more options to customize the CSV file the code returns.

6. Incorporated the necessary criteria

- **File reading:** Read the CSV file containing daily price data for indexes tracking stock exchanges worldwide (United States, Hong Kong, Japan, Germany, South Africa). This edited file is based on the public CSV file that was downloaded from Kaggle.

- **File writing:** Write a new file in CSV of selected stock exchanges by adding data based on two columns, "Daily Price Range" and "Percent Change." If the user runs the code again, it should overwrite the file.
- **Some form of visual display:** Plot graphs of the selected stock exchanges' performance based on three subplots constituted with the actual price movement, "Daily Price Range," and "Percent Change." This plot should include all available timeframes in the selected stock exchange, from the beginning to the end, from the edited CSV file.
- **Meaningful organization over multiple .py files:** Structure the project design based on four Python files with main.py for user interaction and orchestration, plot.py for plotting graphs, fileUtils.py for reading and writing files, and analysis.py for performing the calculations to extract and return "Daily Price Range" and "Percent Change" from the edited CSV file.
- **Unit tests:** Create unit test functions for each function created during the construction of this project, except the plot-related functions.

7. Citations

<https://www.kaggle.com/datasets/mattiuzc/stock-exchange-data?select=indexData.csv>

<https://www.geeksforgeeks.org/how-to-set-a-single-main-title-for-all-the-subplots-in-matplotlib/>

https://www.w3schools.com/python/matplotlib_line.asp

https://www.w3schools.com/python/pandas/pandas_dataframes.asp

<https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.html>

https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.to_csv.html

<https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.insert.html>

https://www.w3schools.com/python/python_user_input.asp

<https://docs.python.org/3/library/os.path.html#os.path.exists>

<https://docs.python.org/3/tutorial/inputoutput.html#formatted-string-literals>