HW2: University Comparison Visualizer

Jeeyoun Jung (jjung83@wisc.edu)

About the Visualization

This visualization explores the volatility of stock prices spanning from 2000 to 2021. The dataset contains comprehensive information about stock price events, including dates, symbols, opening and closing prices, as well as the highest and lowest prices. The dataset was obtained from Kaggle, by Vopani (https://kaggle.com/datasets/rohanrao/nifty50-stock-market-data). The main objective of this visualization is to illustrate the daily price fluctuation between the highest and lowest prices for various stock events. Users are empowered to select a specific stock and define a year range to filter the data according to their preferences. Through interactive plotting capabilities, users can gain insights into the volatility trends of their chosen stocks over time.

Interesting Findings

Upon analysis, it's intriguing to note the absence of a consistent pattern or discernible trend in the upper and lower price limits across all stocks. While analyzing specific periods or subsets, occasional instances of patterns such as sharp inclines or declines were observed. However, when observing the entire graph, significant volatility and lack of stability are evident.

Furthermore, it's noteworthy that certain stocks were not present at the beginning of the dataset in 2000 and only emerged several years later. This observation underscores the dynamic nature of the stock market, characterized by the introduction of new companies and evolving market conditions over time.

Interface Creation Process

I utilized Kaggle's dataset 'NIFTY-50 Stock Market Data (2000 - 2021)'. I randomly selected 10 stocks from the dataset. Although there was extensive information available about each stock's daily prices, I used filters for dates, symbols (names of stocks), and their highest and lowest prices. I performed two different filtering operations: by stock and by year. I designed the interface to be coherent, user-friendly, and complete with useful information. The original dataset contained daily stock data, which made visualizing it challenging. Thus, I transformed the data into monthly data and used it plot_ly. It made my graph zooming in and showing the date yearly, monthly and even daily. I noticed that using vibrant colors in the dynamic graph caused eye strain. Therefore, I opted to change the graph's colors to sky blue and pink. I referenced actual stock graphs, marking high points in pink and low points in sky blue. While examining the entire graph, these differences may not be immediately noticeable, but I aimed to provide a sense of the overall directional movement of the graph.

Reactive Graph

The reactive graph is designed to be intuitive and user-friendly. Users can easily navigate through the data by selecting different stocks and adjusting the year range. The graph dynamically updates based on user input, providing real-time insights into the volatility of stock prices. The application's responsive architecture incorporates a year-filtering slider, a dropdown menu for filtering stock symbols, and a brush object to facilitate precise "zooming in" on specific segments of the line chart. These reactive features enable effortless filtering of stock data based on selected year ranges and stock symbols, facilitating the extraction of pertinent information. Such elements effectively enhance interactivity within the application, empowering users to draw insights and delve deeper into the dataset with ease. By incorporating this reactive graph, users can explore the data more effectively, gaining valuable insights into the behavior of various stocks over the specified time period.