z

Stock Correlation and Visualization Tool

Overview

This Python script allows users to analyze stock data from multiple companies, visualize stock price correlations in a heatmap, and generate candlestick charts for selected companies. The tool fetches historical stock data using the ***yfinance*** *library, processes the data, and provides insights into the relationships between selected stocks.*

Features

1. *Stock Data Fetching : Fetches historical stock data for a list of pre-defined companies (e.g., Apple, Microsoft, Tesla) over a specified time range.*

2. *Correlation Heatmap : Calculates the correlation of stock prices between multiple companies and visualizes the relationship using a heatmap.*

3. *Candlestick Charts : Generates a visual representation of the stock prices using candlestick-like charts with indicators for positive and negative price changes.*
4. *User Input : Users can select multiple companies from a pre-defined list and visualize their stock data.*
5. *Visualizations : Uses matplotlib and seaborn to generate high-quality visualizations such as heatmaps and line plots.*

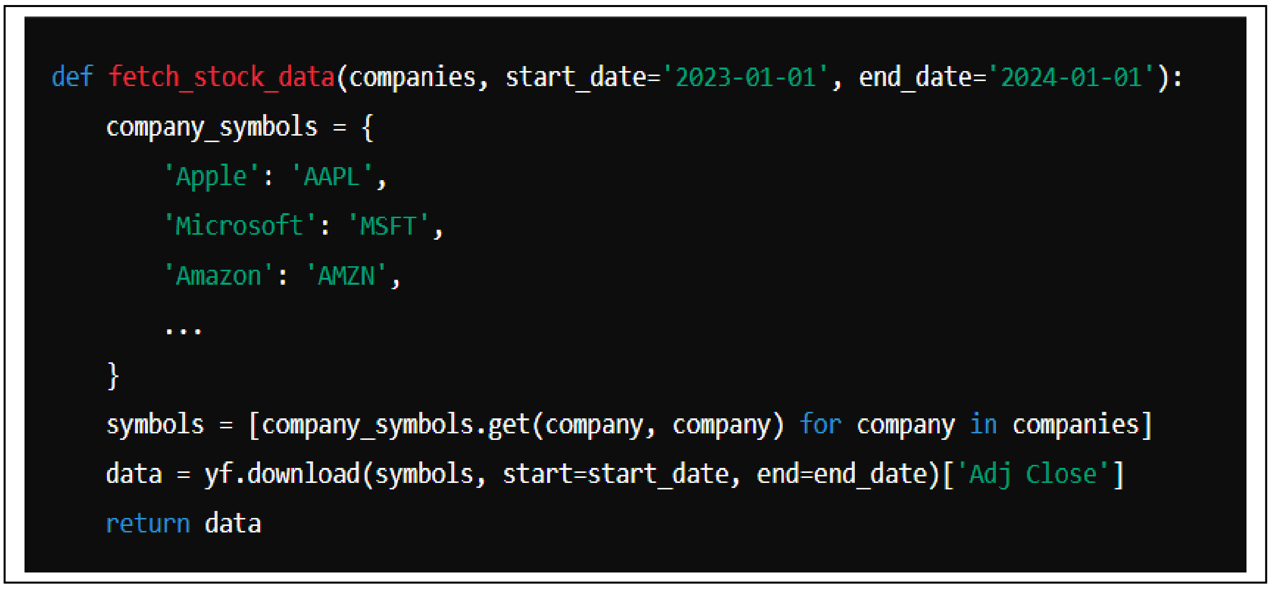
Libraries Used

- ***yfinance****: To fetch historical stock data.*
- ***numpy and pandas****: For data manipulation and analysis.*
- ***matplotlib and seaborn****: For creating visualizations (heatmaps, line charts, etc.).*

How the Script Works

1. Fetching Stock Data

The fetch_stock_data function fetches adjusted closing prices of the specified companies between the given start_date and end_date. It uses Yahoo Finance API via the yfinance library to download historical stock data. The script includes a dictionary that maps company names to their respective stock symbols (e.g., "Apple" to "AAPL").



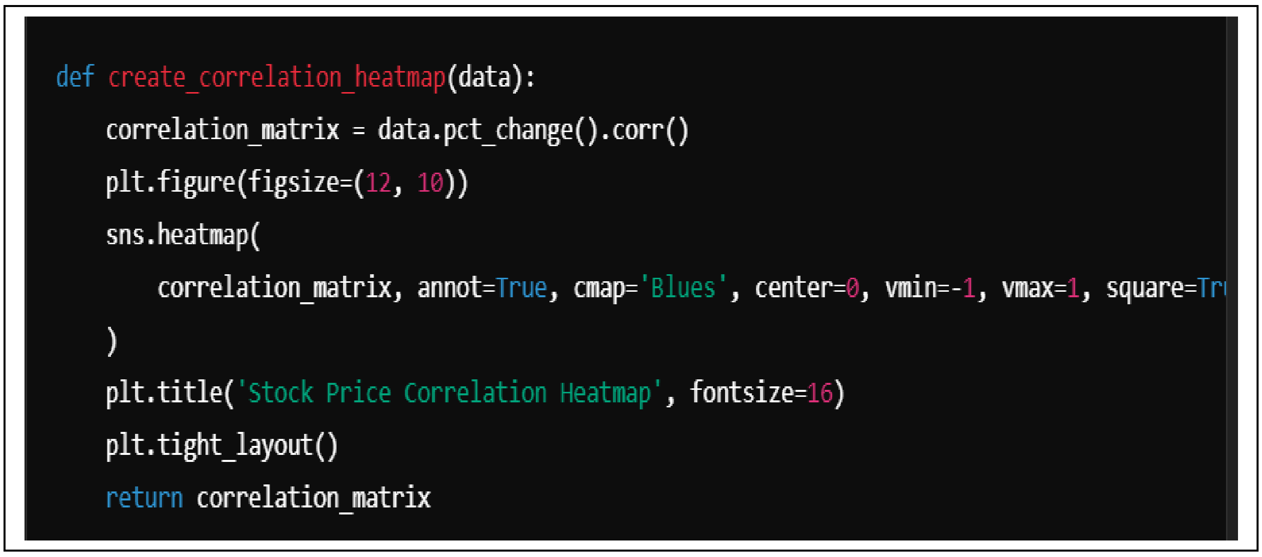
```python
def fetch_stock_data(companies, start_date='2023-01-01', end_date='2024-01-01'):
    company_symbols = {
        'Apple': 'AAPL',
        'Microsoft': 'MSFT',
        'Amazon': 'AMZN',
        ...
    }
    symbols = [company_symbols.get(company, company) for company in companies]
    data = yf.download(symbols, start=start_date, end=end_date)['Adj Close']
    return data
```

***Input**: List of company names (e.g., ['Apple', 'Microsoft'])*

***Output**: Pandas DataFrame with adjusted closing prices for each company.*

2. Creating a Correlation Heatmap

The create_correlation_heatmap function calculates the correlation between the percentage change in stock prices of the selected companies and generates a heatmap. A correlation matrix is calculated using the daily percentage changes in stock prices.

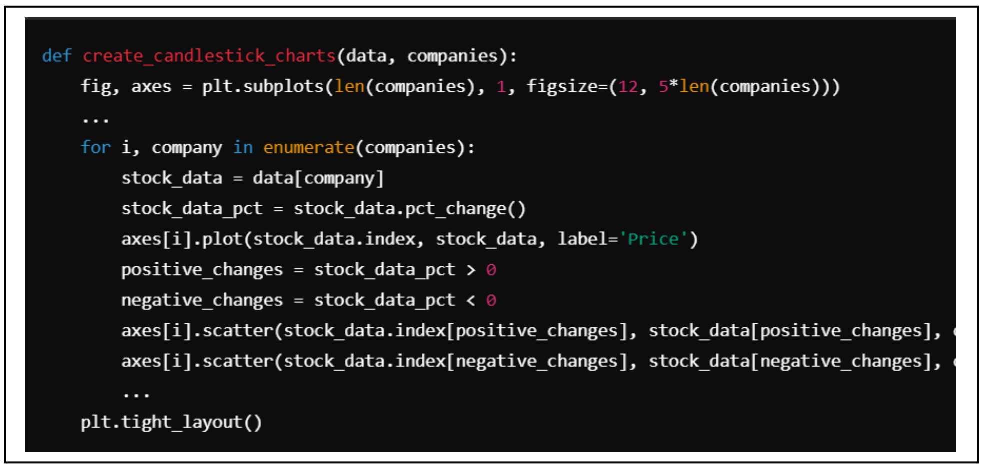


```python
def create_correlation_heatmap(data):
    correlation_matrix = data.pct_change().corr()
    plt.figure(figsize=(12, 10))
    sns.heatmap(
        correlation_matrix, annot=True, cmap='Blues', center=0, vmin=-1, vmax=1, square=Tr
    )
    plt.title('Stock Price Correlation Heatmap', fontsize=16)
    plt.tight_layout()
    return correlation_matrix
```

***Input**: DataFrame of stock price data.*
***Output**: A heatmap displaying the correlation between stock prices of different companies.*

3. Creating Candlestick-like Charts

*The **create_candlestick_charts** function generates line charts with markers for price changes. Green upward-pointing triangles (^) indicate positive price changes, and red downward-pointing triangles (v) indicate negative price changes.*



```python
def create_candlestick_charts(data, companies):
    fig, axes = plt.subplots(len(companies), 1, figsize=(12, 5*len(companies)))
    ...
    for i, company in enumerate(companies):
        stock_data = data[company]
        stock_data_pct = stock_data.pct_change()
        axes[i].plot(stock_data.index, stock_data, label='Price')
        positive_changes = stock_data_pct > 0
        negative_changes = stock_data_pct < 0
        axes[i].scatter(stock_data.index[positive_changes], stock_data[positive_changes],
        axes[i].scatter(stock_data.index[negative_changes], stock_data[negative_changes],
        ...
    plt.tight_layout()
```

***Input**: DataFrame of stock data, list of selected companies.*
***Output**: Candlestick-like charts showing the price trends for each company.*

4. Main Functionality

The main function orchestrates the entire process. It prompts the user to select at least two companies from the pre-defined list, fetches the stock data, generates the correlation heatmap, and creates candlestick charts.

Output:

- *The script will fetch the stock data for the selected companies.*
- *It will then display a correlation heatmap, showing how similar the price movements are between the selected companies.*
- *The script will also generate candlestick-like charts for each of the selected companies.*

Sample Output

When the user enters "Apple" and "Microsoft", the following outputs will be generated:

1. ***Correlation Heatmap****: A graphical heatmap displaying the correlation coefficients between the stock prices of Apple, Microsoft, and any other selected companies.*
2. ***Candlestick Charts****: Line charts showing the price movements of Apple and Microsoft, with visual markers indicating positive and negative price changes.*

Error Handling

The script includes basic error handling to ensure that:

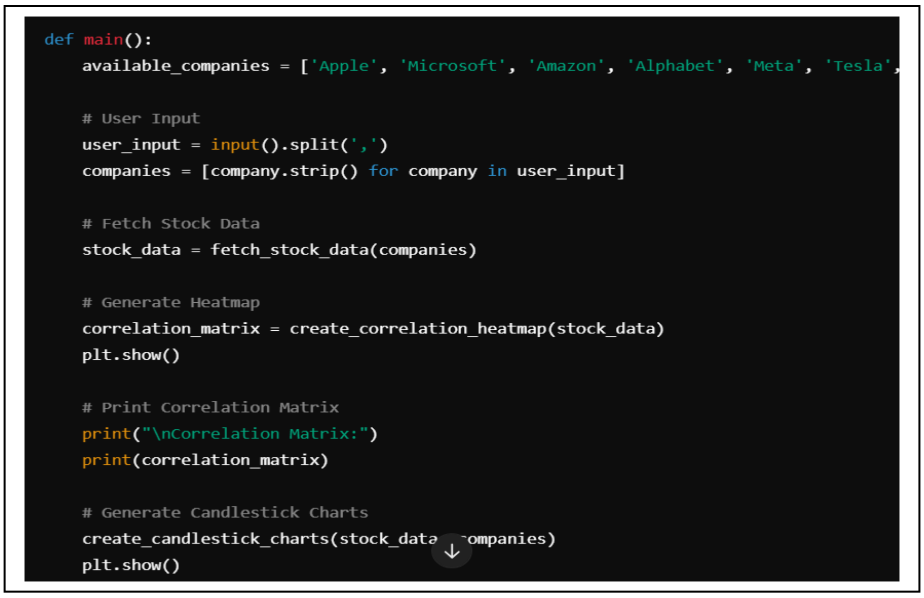
- *The user inputs at least two valid companies.*
- *The stock data is fetched successfully without issues.*
- *Proper error messages are shown if something goes wrong, such as a network issue or invalid company name.*

Possible Improvements

- ***Extended Data Analysis****: You can add more advanced analytics such as moving averages, volatility measures, or machine learning models for predicting stock prices.*
- ***UI Enhancement****: Implement a graphical user interface (GUI) to make it more user-friendly.*
- ***Real-time Data****: Add functionality to get real-time stock data instead of just historical data.*

Conclusion

This script provides a simple but powerful way to visualize and analyze stock price correlations. By comparing stock movements between various companies, users can gain insights into how different companies are related in terms of thei stock price behavior. This can be useful for making informed investment decisions or understanding market trends.



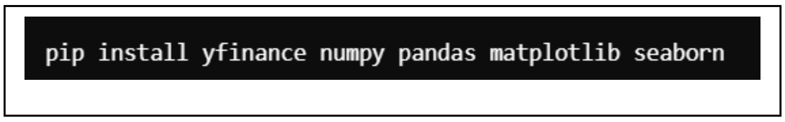
```python
def main():
    available_companies = ['Apple', 'Microsoft', 'Amazon', 'Alphabet', 'Meta', 'Tesla',

    # User Input
    user_input = input().split(',')
    companies = [company.strip() for company in user_input]

    # Fetch Stock Data
    stock_data = fetch_stock_data(companies)

    # Generate Heatmap
    correlation_matrix = create_correlation_heatmap(stock_data)
    plt.show()

    # Print Correlation Matrix
    print("\nCorrelation Matrix:")
    print(correlation_matrix)

    # Generate Candlestick Charts
    create_candlestick_charts(stock_data, companies)
    plt.show()
```

How To Use The Script

***pandasRequirements**: Ensure you have the necessary Python libraries installed:*

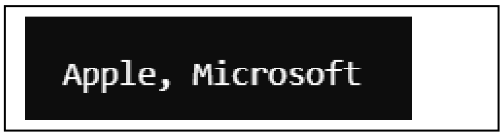
1. *yfinance*
2. *numpy*
3. *pandas*
4. *matplotlib*
5. *seaborn*

You can install them using pip:



```
pip install yfinance numpy pandas matplotlib seaborn
```

***Running the Script**: Run the script in a Python environment. When prompted, enter at least two company names (separated by commas). For example:*



```
Apple, Microsoft
```