Project name –

Stock Price Analysis: Use web scraping to collect stock price data and analyze it using

Python's built-in statistical libraries

Date –

08.04.2023

Project members –

Y. Sachith Nimesh

Description –

This project aims to collect historical stock price data of a specific company using web scraping techniques and analyze it using Python's built-in statistical libraries to gain insights into the stock's performance.

Methodology –

The project will be divided into the following two parts:

Part 1: Data Collection

In this part, you will collect historical stock price data from a financial website such as Yahoo Finance, Google Finance, or Alpha Vantage. You will use web scraping techniques with Python libraries such as BeautifulSoup, Requests, and Pandas to extract the required data. The data should include the stock's opening price, closing price, highest price, lowest price, and volume for a specific period, say, for the last 5 years.

Part 2: Data Analysis

In this part, you will analyze the collected data using Python's built-in statistical libraries such as NumPy, Pandas, and Matplotlib. You will perform the following analysis:

- Plotting the stock's closing price over time to see its overall trend.
- Calculating the moving average of the stock's closing price to smooth out the fluctuations and see the long-term trend.

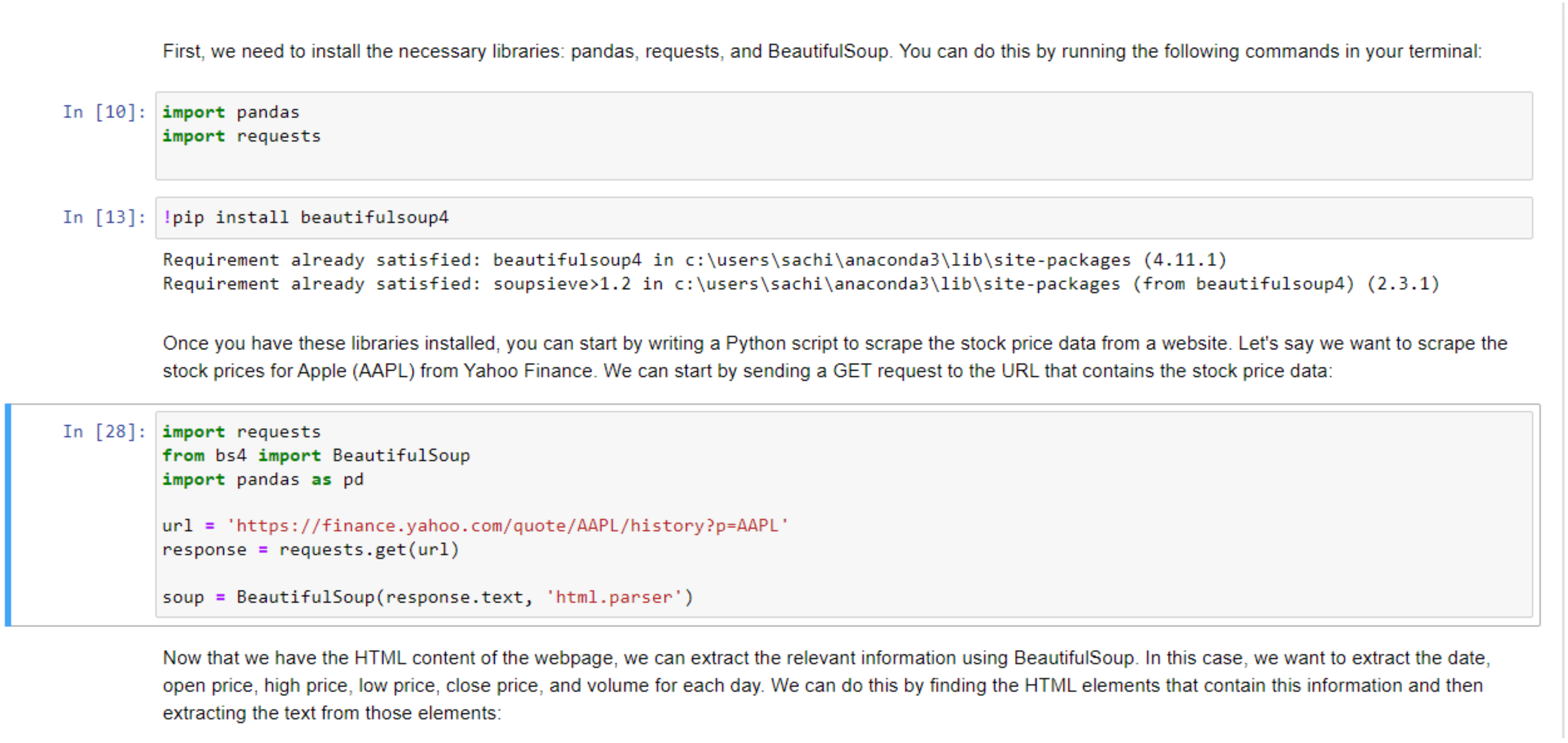
- Calculating the daily returns of the stock to see how much it fluctuates on a daily basis.
- Analyzing the distribution of daily returns and calculating basic statistical metrics such as mean, standard deviation, and variance.
- Calculating the correlation between the stock's returns and the returns of other stocks or market indices to see how it is affected by external factors.

Deliverables:

- Python script for data collection and analysis
- Visualization of the stock's performance over time and other relevant insights
- Report summarizing the findings of the analysis and insights gained

Additional Ideas:

- Use machine learning algorithms to predict the stock's future performance based on its historical data.
- Compare the stock's performance with its competitors to see how it fares in the market.
- Collect news articles related to the stock and analyze their sentiment to see how it affects the stock's price



First, we need to install the necessary libraries: pandas, requests, and BeautifulSoup. You can do this by running the following commands in your terminal:

```
In [10]: import pandas
         import requests
```

```
In [13]: !pip install beautifulsoup4
```

```
Requirement already satisfied: beautifulsoup4 in c:\users\sachi\anaconda3\lib\site-packages (4.11.1)
Requirement already satisfied: soupsieve>1.2 in c:\users\sachi\anaconda3\lib\site-packages (from beautifulsoup4) (2.3.1)
```

Once you have these libraries installed, you can start by writing a Python script to scrape the stock price data from a website. Let's say we want to scrape the stock prices for Apple (AAPL) from Yahoo Finance. We can start by sending a GET request to the URL that contains the stock price data:

```
In [28]: import requests
         from bs4 import BeautifulSoup
         import pandas as pd

         url = 'https://finance.yahoo.com/quote/AAPL/history?p=AAPL'
         response = requests.get(url)

         soup = BeautifulSoup(response.text, 'html.parser')
```

Now that we have the HTML content of the webpage, we can extract the relevant information using BeautifulSoup. In this case, we want to extract the date, open price, high price, low price, close price, and volume for each day. We can do this by finding the HTML elements that contain this information and then extracting the text from those elements:

In [24]:
```python
table = soup.find_all('table')[0]

data = []
for row in table.find_all('tr'):
    cols = row.find_all('td')
    if len(cols) == 7:
        date = cols[0].text
        open_price = cols[1].text
        high_price = cols[2].text
        low_price = cols[3].text
        close_price = cols[4].text
        volume = cols[6].text
        data.append([date, open_price, high_price, low_price, close_price, volume])
```

Out[24]:
```
[]
```

In [21]:
```python
print(response.text)
```

```
</title><meta name="viewport" content="width=device-width" /><meta name="robots" content="index,follow"/>
<meta name="Description" content="Clases particulares de inglés en Valencia. Encuentra gratis profesor particular de Inglés e
n Valencia, lee comentarios y elige tu profesor ideal."/>
<meta name="facebook-domain-verification" content="kc3b74kfvtl1c1c759ki5v5ges3jca"/>
<link rel="canonical" href="https://www.tusclasesparticulares.com/profesores-ingles/valencia.aspx"/>
<link rel="next" href="?pagina=2"/>
<meta property="og:title" content="Clases particulares de inglés en Valencia"/>
<meta property="og:description" content="Clases particulares de inglés en Valencia. Encuentra gratis profesor particular de I
nglés en Valencia, lee comentarios y elige tu profesor ideal."/>
<meta property="og:type" content="website" /><meta property="og:url" content="https://www.tusclasesparticulares.com/profesore
s-ingles/valencia.aspx"/>
<meta property="og:image" content="https://d1reana485161v.cloudfront.net/i/tusclasesparticulares_1200x630_fff.png"/>
<link id="lnkcss" type="text/css" rel="stylesheet" href="https://d1reana485161v.cloudfront.net/tces.min.css?v=1437" /><meta h
ttp-equiv="X-UA-Compatible" content="IE=edge" /><link rel="icon" type="image/png" href="https://d1reana485161v.cloudfront.ne
t/i/favicon.png">
</head>
<body id="body" class="desk footer2 country_es nob search-page" data-portalid="0" data-wp-req-per="1">
    <div id="pp"></div>
    <div id="wrapper" class="flex">
        <header id="header" class="header2">
```

In [17]:
```python
df = pd.DataFrame(data, columns=['Date', 'Open', 'High', 'Low', 'Close', 'Volume'])
df['Return'] = df['Close'].pct_change()

mean_return = df['Return'].mean()
std_return = df['Return'].std()

print(f"Mean daily return: {mean_return:.4f}")
print(f"Standard deviation of daily return: {std_return:.4f}")
```