Stock Market Data Analysis for Selected Companies

Introduction

This project involves the analysis of historical stock market data for selected companies. The primary objective is to visualize and understand stock price trends over time using Python data analysis libraries.

Libraries Used, Loading the Dataset & Dataset Structure

[8]:
```
import pandas as pd
import numpy as np
import matplotlib as mlb
import matplotlib.pyplot as plt
import seaborn as sns
```

[9]:
```
stock_data=pd.read_csv('Stock_Market_Data.csv')
stock_data.head()
```

[9]:

	Date	Name	Open	High	Low	Close	Volume
0	02-01-2022	01.Bank	22.83	23.20	22.59	22.93	1842350.41
1	03-01-2022	01.Bank	23.03	23.29	22.74	22.90	1664989.63
2	04-01-2022	01.Bank	22.85	23.13	22.64	22.84	1354510.97
3	05-01-2022	01.Bank	22.91	23.20	22.70	22.98	1564334.81
4	06-01-2022	01.Bank	23.12	23.65	23.00	23.37	2586344.19

[10]:
```
stock_data.shape
```

[10]:
```
(49158, 7)
```

Converting Date Format

[11]:
```
stock_data.dtypes
```

[11]:
```
Date       object
Name       object
Open      float64
High      float64
Low       float64
Close     float64
Volume    float64
dtype: object
```

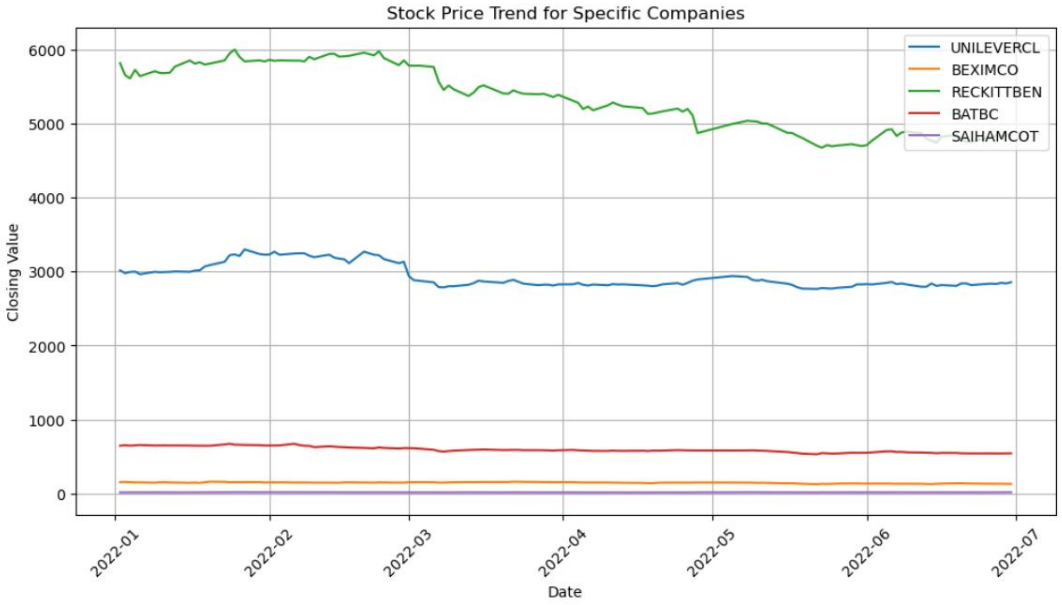
[15]:
```
stock_data["Date"]=pd.to_datetime(stock_data["Date"],dayfirst=True)
```

[17]:
```
stock_data["Date"].max()
```

[17]:
```
Timestamp('2022-06-30 00:00:00')
```

Stock Trend Visualization for Selected Companies

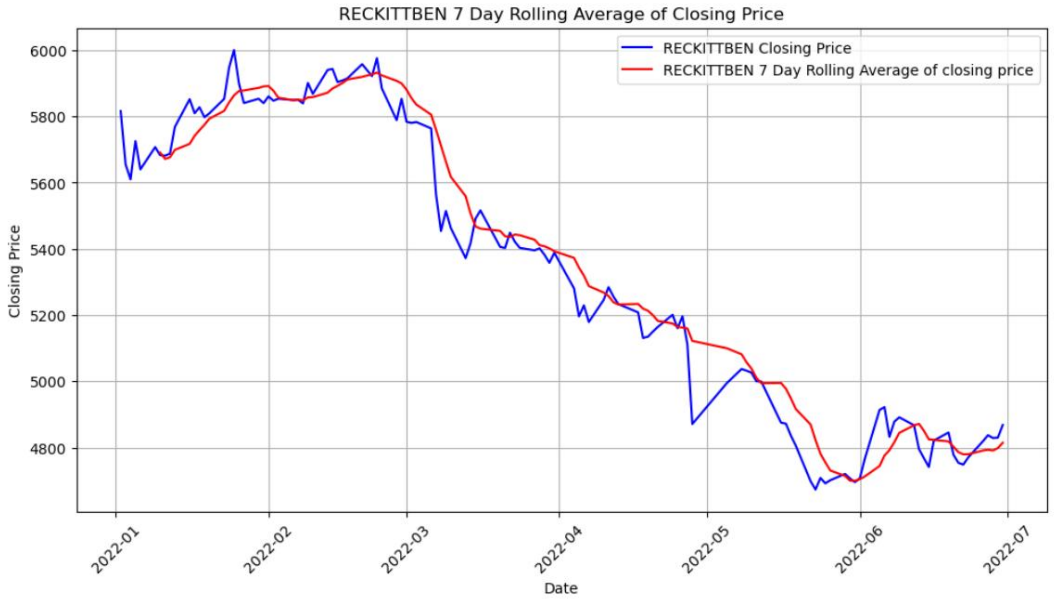
[23]:
```
# Creating a list of specific comapnies
specific_companies=['UNILEVERCL', 'BEXIMCO', 'RECKITTBEN', 'BATBC', 'SAIHAMCOT']
# Filter out your desired companies' data and put it to another dataframe
specific_data=stock_data[stock_data['Name'].isin(specific_companies) ]
# Creating Line Graph for Each Companies
plt.figure(figsize= (12, 6))
for company in specific_companies:
    company_data = specific_data[specific_data['Name'] == company]
    plt.plot(company_data['Date'], company_data['Close'], label=company)
# Adding Labels & Titles
plt.xlabel('Date')
plt.ylabel('Closing Value')
plt.title('Stock Price Trend for Specific Companies')
plt.legend()
plt.grid( )
# Improving readability
plt.xticks(rotation=45)
plt.show()
```



This code filters stock market data to extract records for five selected companies: UNILEVERCL, BEKXICO, RECKITTBEN, BATBC, and SAHAMACOT. It then plots the closing stock prices over time for each company on a single line chart. This visualization allows for easy comparison of stock performance trends across different companies, helping to identify patterns such as growth, stability, or decline over time.

Analyzing Stock Price Trends with a 7-Day Rolling Average

[45]:
```
import matplotlib.pyplot as plt
import pandas as pd

specific_company = 'RECKITTBEN'
specific_data = stock_data[stock_data['Name'] == specific_company].copy()  # Use copy() to avoid SettingWithCopyWarning

# Convert 'Date' to datetime if not done already
specific_data['Date'] = pd.to_datetime(specific_data['Date'])

# Calculate 7-day rolling average
specific_data['7_Day_Rolling_Avg'] = specific_data['Close'].rolling(window=7).mean()

plt.figure(figsize=(12, 6))
plt.plot(specific_data['Date'], specific_data['Close'], label=f'{specific_company} Closing Price', color='blue')
plt.plot(specific_data['Date'], specific_data['7_Day_Rolling_Avg'], label=f'{specific_company} 7 Day Rolling Average of closing price', color='red')

plt.xlabel('Date')
plt.ylabel('Closing Price')
plt.title(f'{specific_company} 7 Day Rolling Average of Closing Price')
plt.grid()
plt.legend()
plt.xticks(rotation=45)
plt.show()
```



This code extracts the stock data for RECKITTBEN and calculates a 7-day rolling average of its closing prices. It then plots both the daily closing prices and the 7-day rolling average on the same graph. This visualization smooths out daily fluctuations, making it easier to identify the underlying trend in stock price movements over time. By reducing short-term volatility, the rolling average helps highlight sustained upward or downward trends, offering a clearer picture of the stock's performance.