

Name : Siddhi Parekh

Reg no. : 221071047

Batch C

SY CE

EXPERIMENT 9

AIM:

Write an application to draw plots using Matplotlib library

THEORY:

Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack.

One of the greatest benefits of visualization is that it allows us visual access to huge amounts of data in easily digestible visuals.

Matplotlib consists of several plots like line, bar, scatter, histogram etc.

Matplotlib comes with a wide variety of plots. Plots help to understand trends, and patterns, and to make correlations. They're typically instruments for reasoning about quantitative information.

Python is the most used language for Matplotlib is a plotting library for creating static, animated, and interactive visualizations in Python.

Pyplot is a Matplotlib module that provides a MATLAB-like interface. Matplotlib is designed to be as usable as MATLAB, with the ability to use Python and the advantage of being free and open-source. Each function makes some changes to a figure: e.g., creates a figure, creates a plotting area in a figure, plots some lines in a plotting area, decorates the plot with labels, etc.

CODE:

```
import pandas as pd
import numpy as np
from datetime import datetime

data = pd.read_csv('/content/HistoricalPrices.csv')
data.head()

data['Date'] = pd.to_datetime(data['Date'])
data = data.sort_values(by = 'Date')

plt.plot(data['Date'], data['Close'])
plt.show()

plt.plot(data['Date'], data['Open'], label = 'Open')
plt.plot(data['Date'], data['Close'], label = 'Close')
plt.legend()
plt.show()

plt.scatter(data['Open'], data['Close'])
plt.show()

data['Date'] = pd.to_datetime(data['Date'])
data = data.sort_values(by='Date')

data['Month'] = data['Date'].dt.month_name()
average_close_by_month = data.groupby('Month')['Close'].mean()
plt.bar(average_close_by_month.index, average_close_by_month)

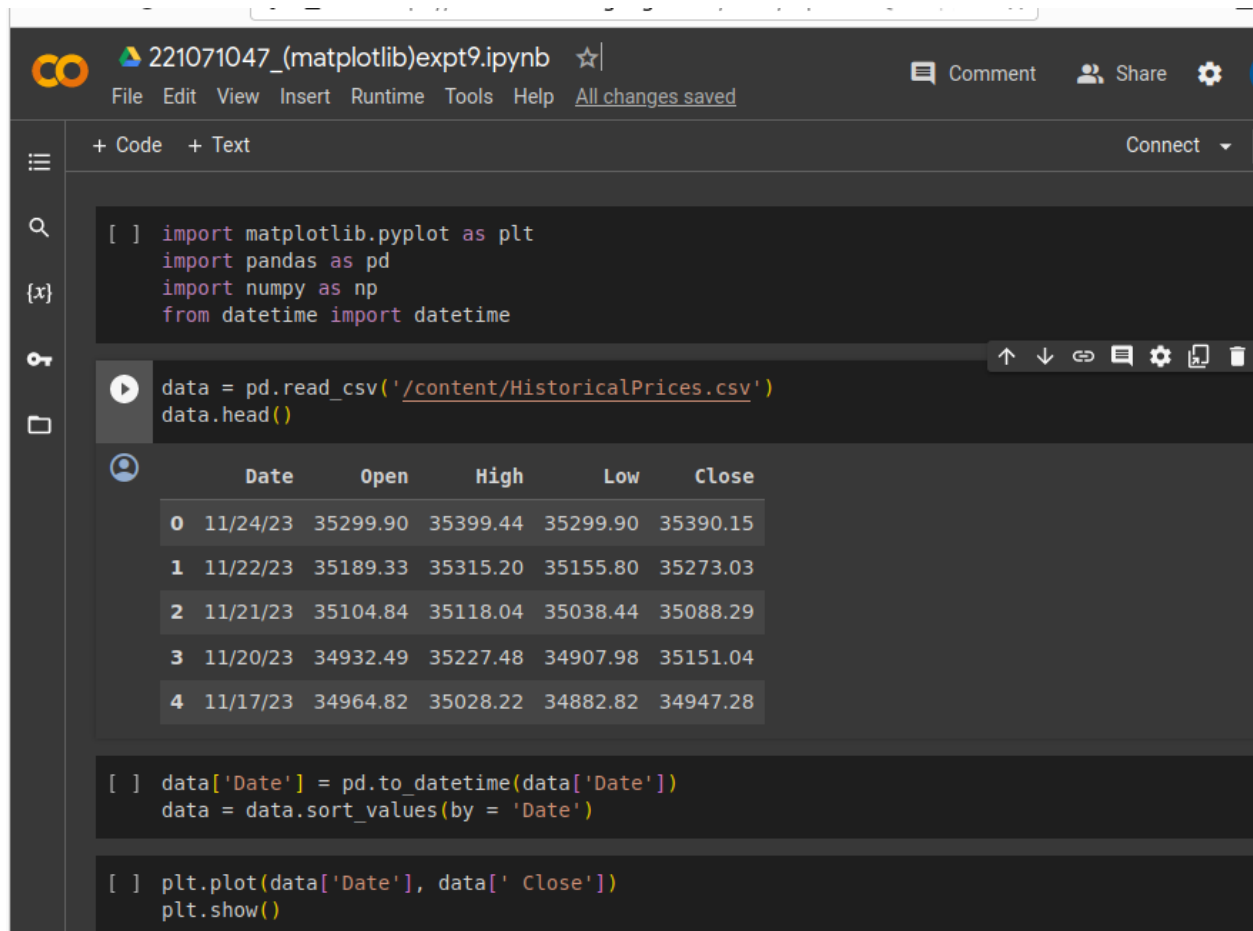
plt.xlabel('Month')
plt.ylabel('Average Close Price')plt.title('Average Close Price by Month')

plt.title('Average Close Price by Month')
plt.show()

plt.hist(data['Close'], bins=20, color='blue', alpha=0.7)
plt.xlabel('Close Price')
```

```
plt.ylabel('Frequency')
plt.title('Histogram of Close Prices')
plt.show()
```

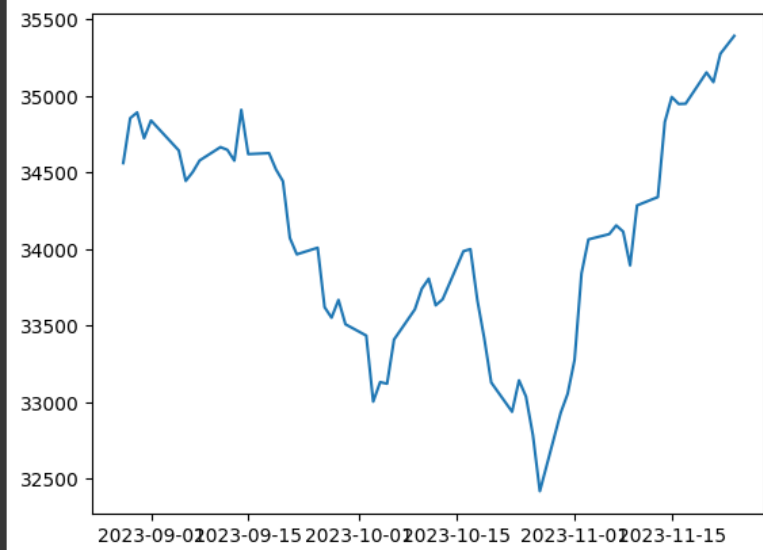
OUTPUT:



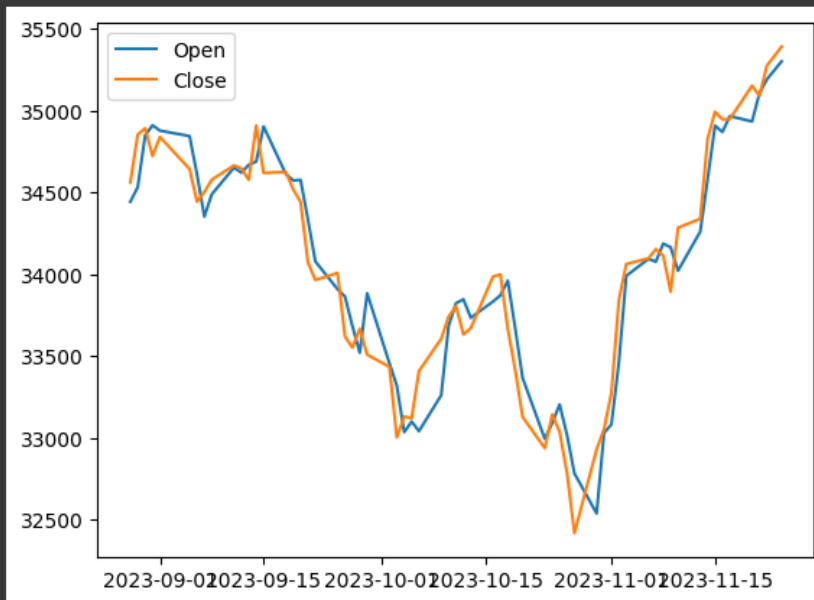
The screenshot shows a Jupyter Notebook interface with the following components:

- Header:** 221071047_(matplotlib)expt9.ipynb, with icons for Comment, Share, and settings.
- Menu:** File, Edit, View, Insert, Runtime, Tools, Help, All changes saved.
- Left Sidebar:** Contains icons for file explorer, search, and other notebook functions.
- Code Cells:**
 - Cell 1: Imports matplotlib.pyplot as plt, pandas as pd, numpy as np, and datetime module.
 - Cell 2: Executes data = pd.read_csv('/content/HistoricalPrices.csv') and data.head().
 - Cell 3: Executes data['Date'] = pd.to_datetime(data['Date']) and data = data.sort_values(by = 'Date').
 - Cell 4: Executes plt.plot(data['Date'], data['Close']) and plt.show().
- Output:** A table preview of the first 5 rows of the data, showing columns: Date, Open, High, Low, Close.

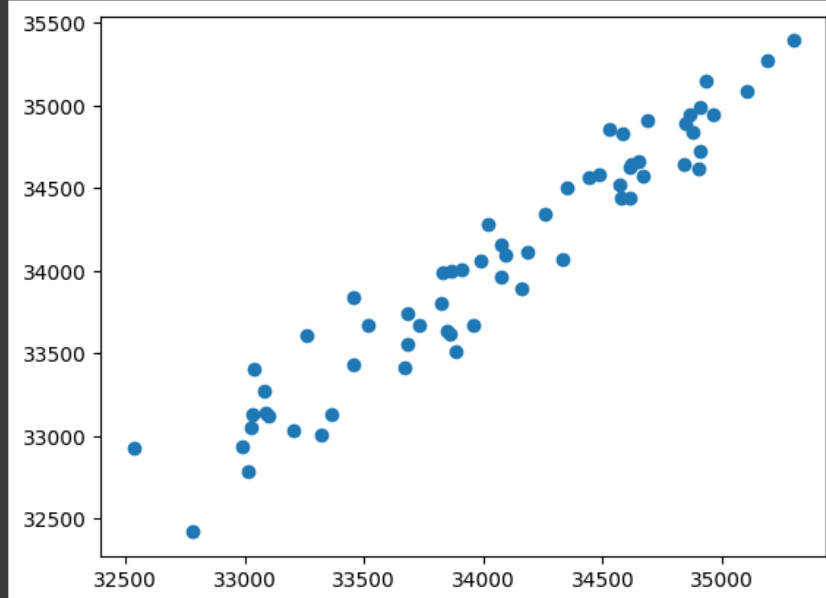
	Date	Open	High	Low	Close
0	11/24/23	35299.90	35399.44	35299.90	35390.15
1	11/22/23	35189.33	35315.20	35155.80	35273.03
2	11/21/23	35104.84	35118.04	35038.44	35088.29
3	11/20/23	34932.49	35227.48	34907.98	35151.04
4	11/17/23	34964.82	35028.22	34882.82	34947.28



```
plt.plot(data['Date'], data['Open'], label = 'Open')
plt.plot(data['Date'], data['Close'], label = 'Close')
plt.legend()
plt.show()
```



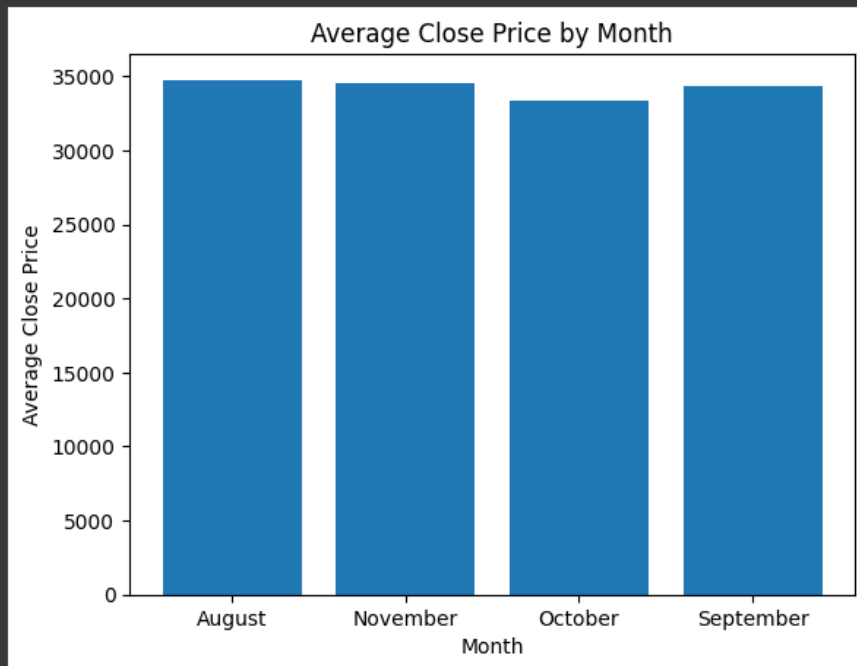
```
[ ] plt.scatter(data['Open'], data['Close'])
plt.show()
```



```
data['Date'] = pd.to_datetime(data['Date'])
data = data.sort_values(by='Date')

data['Month'] = data['Date'].dt.month_name()

average_close_by_month = data.groupby('Month')['Close'].mean()
plt.bar(average_close_by_month.index, average_close_by_month)
plt.xlabel('Month')
plt.ylabel('Average Close Price')
plt.title('Average Close Price by Month')
plt.show()
```



```
data['Date'] = pd.to_datetime(data['Date'])
data = data.sort_values(by='Date')

# Plot histogram of 'Close' prices
plt.hist(data['Close'], bins=20, color='blue', alpha=0.7)
plt.xlabel('Close Price')
plt.ylabel('Frequency')
plt.title('Histogram of Close Prices')
plt.show()
```



```
[ ]
```

CONCLUSION :

Thus, from this experiment matplotlib was learnt, along with how to plot graphs with the given dataset.