Torrodjae Somerville

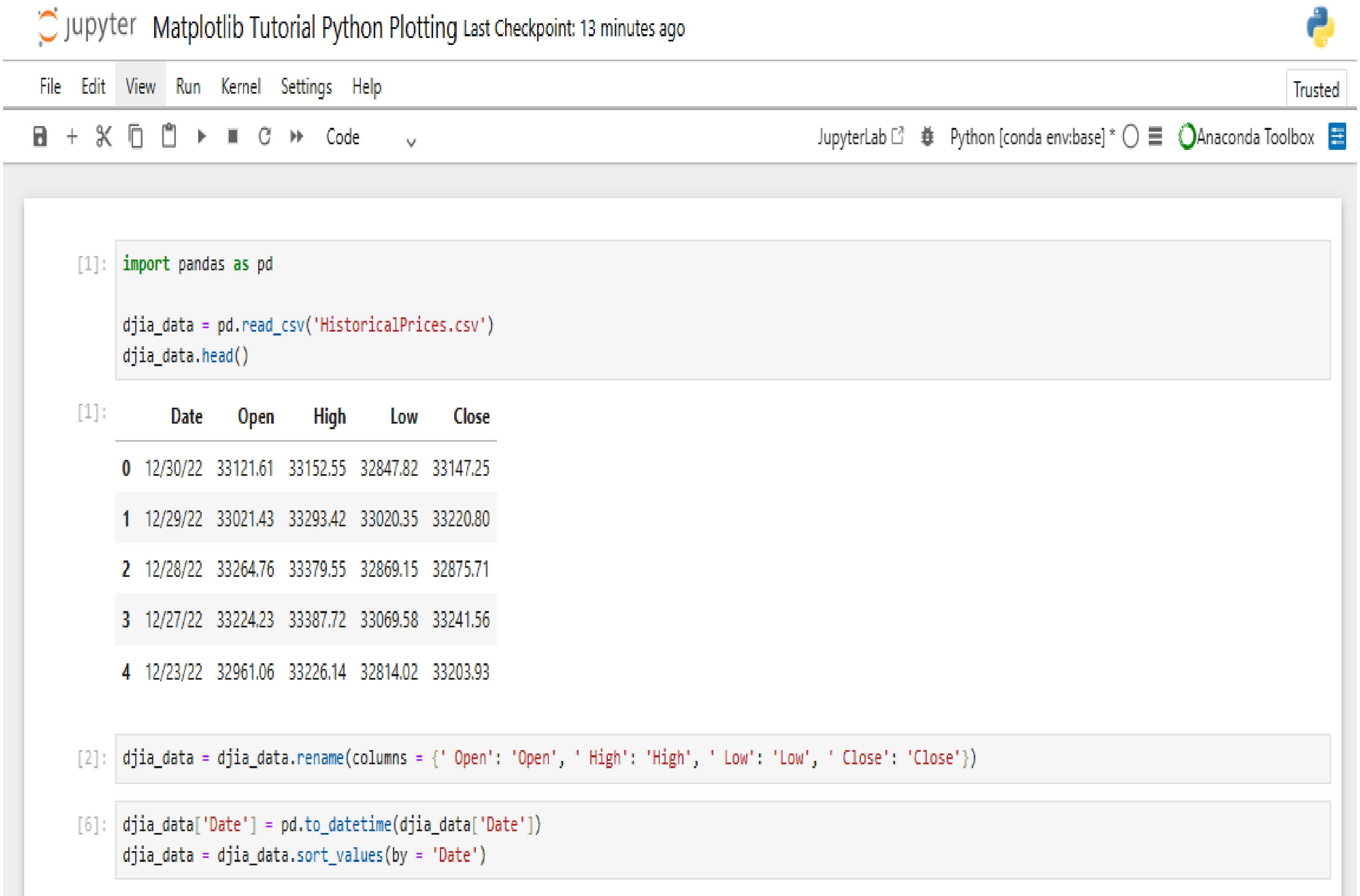
CTEC 298-101

11/15/2025

Dr. Bemley

Matplotlib Tutorial: Python Plotting

The Dataset:


jupyter
Matplotlib Tutorial Python Plotting Last Checkpoint: 13 minutes ago
File
Edit
View
Run
Kernel
Settings
Help
Trusted
Code
JupyterLab
Python [conda env:base] *
Anaconda Toolbox

[1]:

```
import pandas as pd

djia_data = pd.read_csv('HistoricalPrices.csv')
djia_data.head()
```

[1]:

	Date	Open	High	Low	Close
0	12/30/22	33121.61	33152.55	32847.82	33147.25
1	12/29/22	33021.43	33293.42	33020.35	33220.80
2	12/28/22	33264.76	33379.55	32869.15	32875.71
3	12/27/22	33224.23	33387.72	33069.58	33241.56
4	12/23/22	32961.06	33226.14	32814.02	33203.93

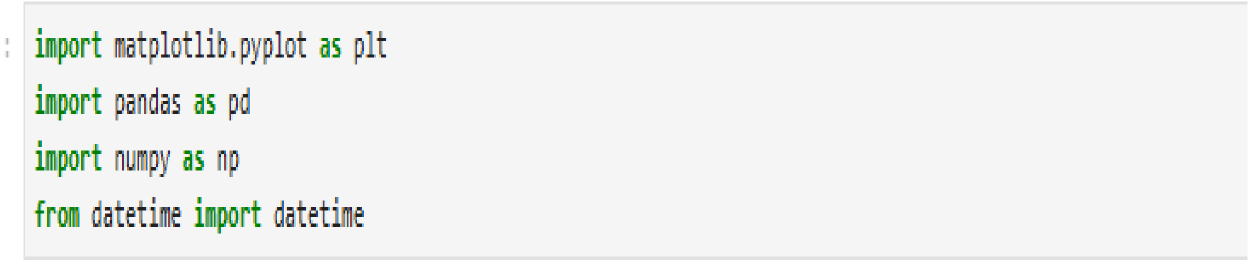
[2]:

```
djia_data = djia_data.rename(columns = {' Open': 'Open', ' High': 'High', ' Low': 'Low', ' Close': 'Close'})
```

[6]:

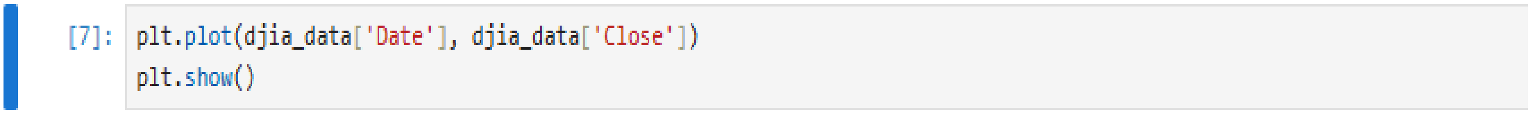
```
djia_data['Date'] = pd.to_datetime(djia_data['Date'])
djia_data = djia_data.sort_values(by = 'Date')
```

Importing matplotlib:

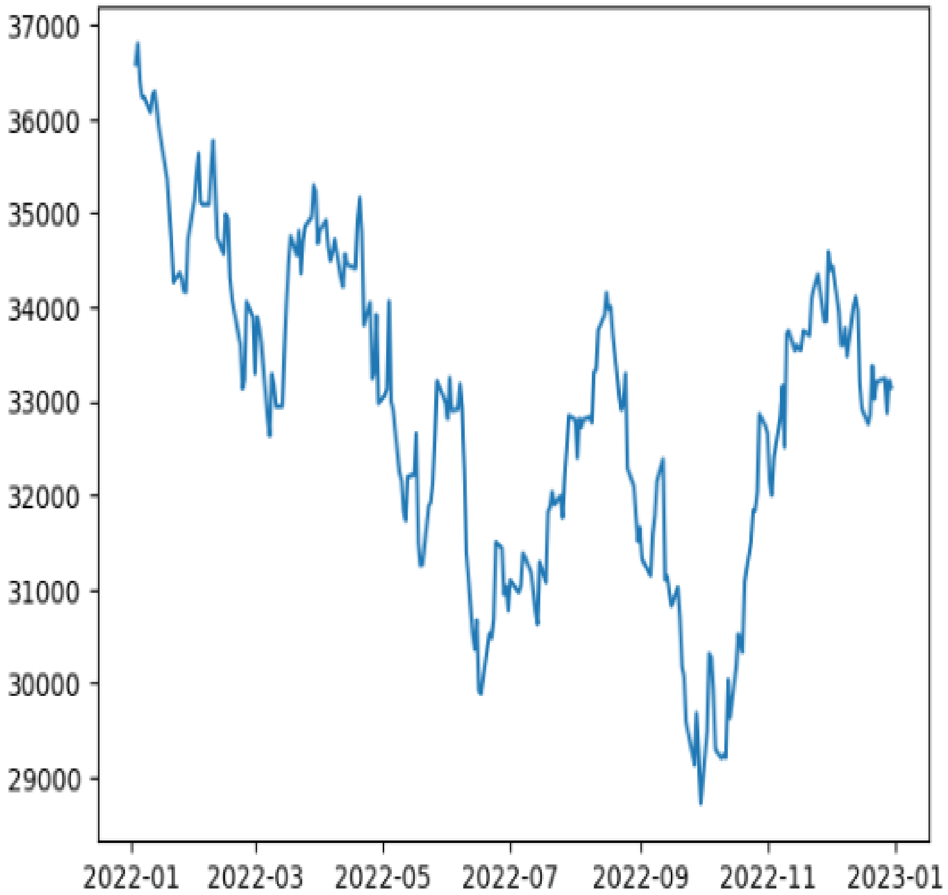


```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
from datetime import datetime
```

Line Plots With a Single Line:

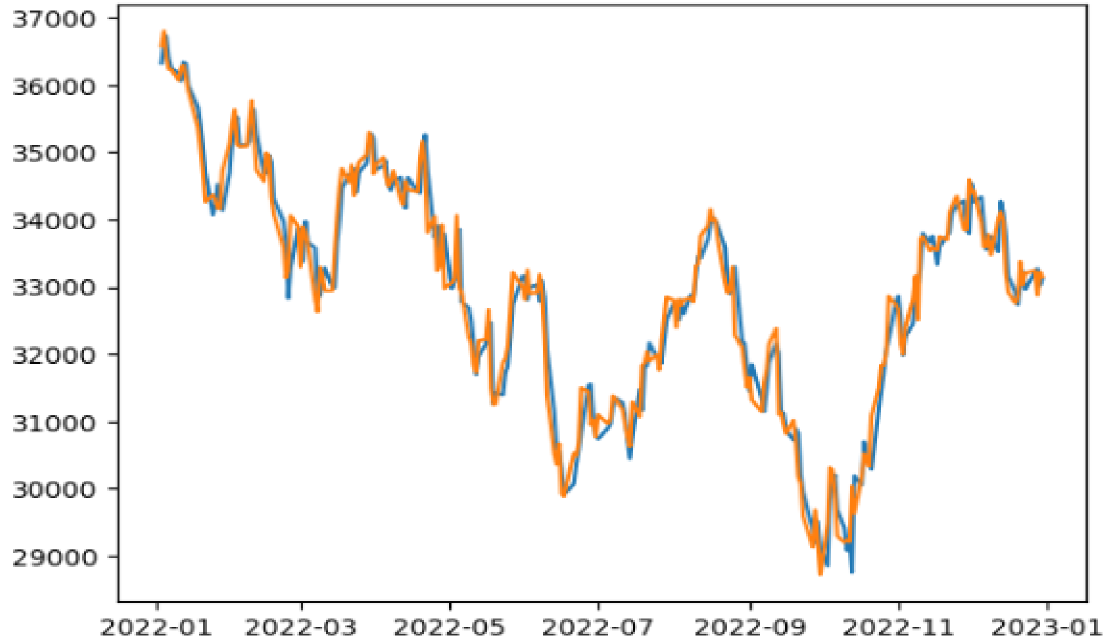


```
[7]: plt.plot(djia_data['Date'], djia_data['Close'])
     plt.show()
```

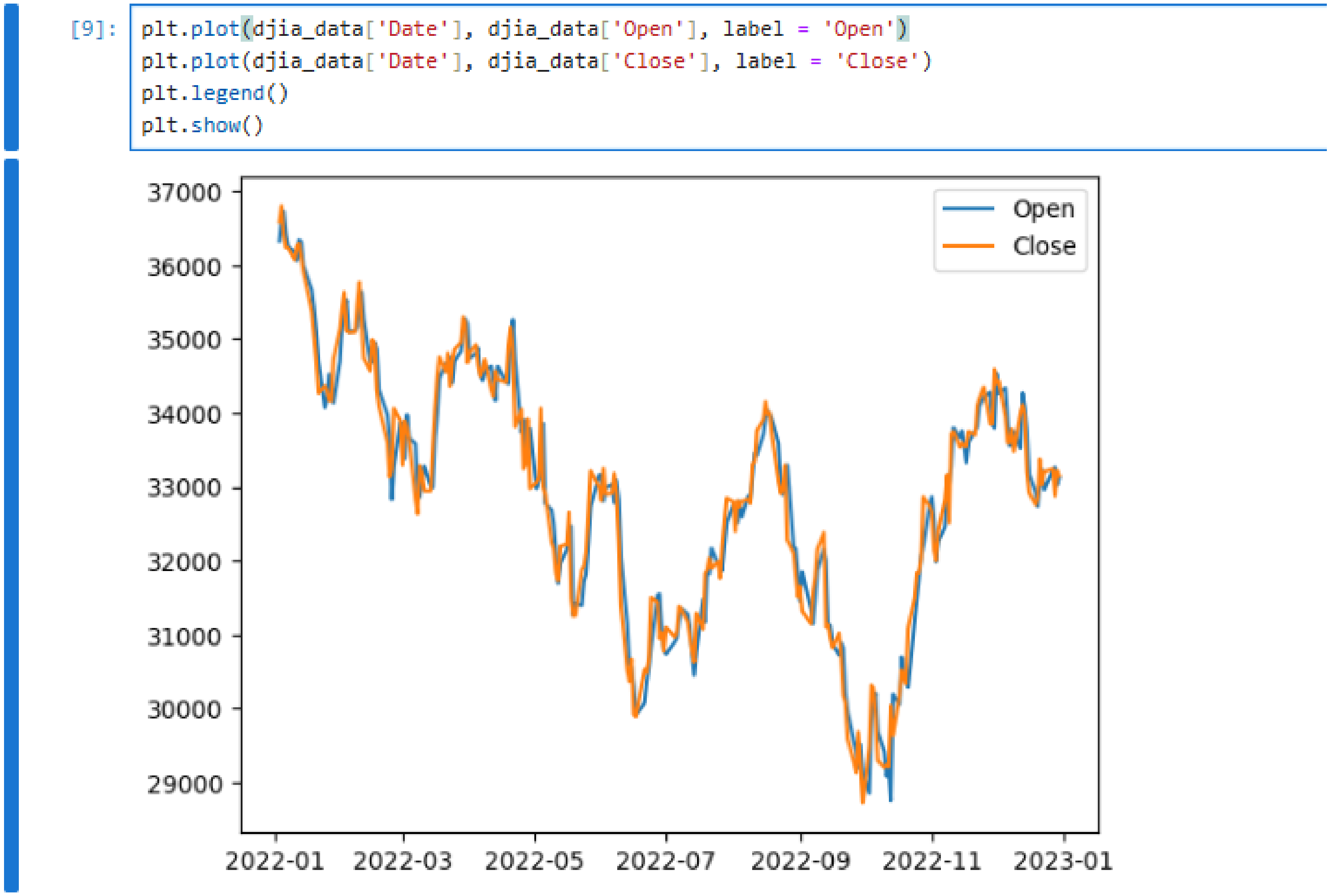


Line Plots with Multiple Lines:

```
[8]: plt.plot(djia_data['Date'], djia_data['Open'])
     plt.plot(djia_data['Date'], djia_data['Close'])


     plt.show()
```



Adding a Legend:

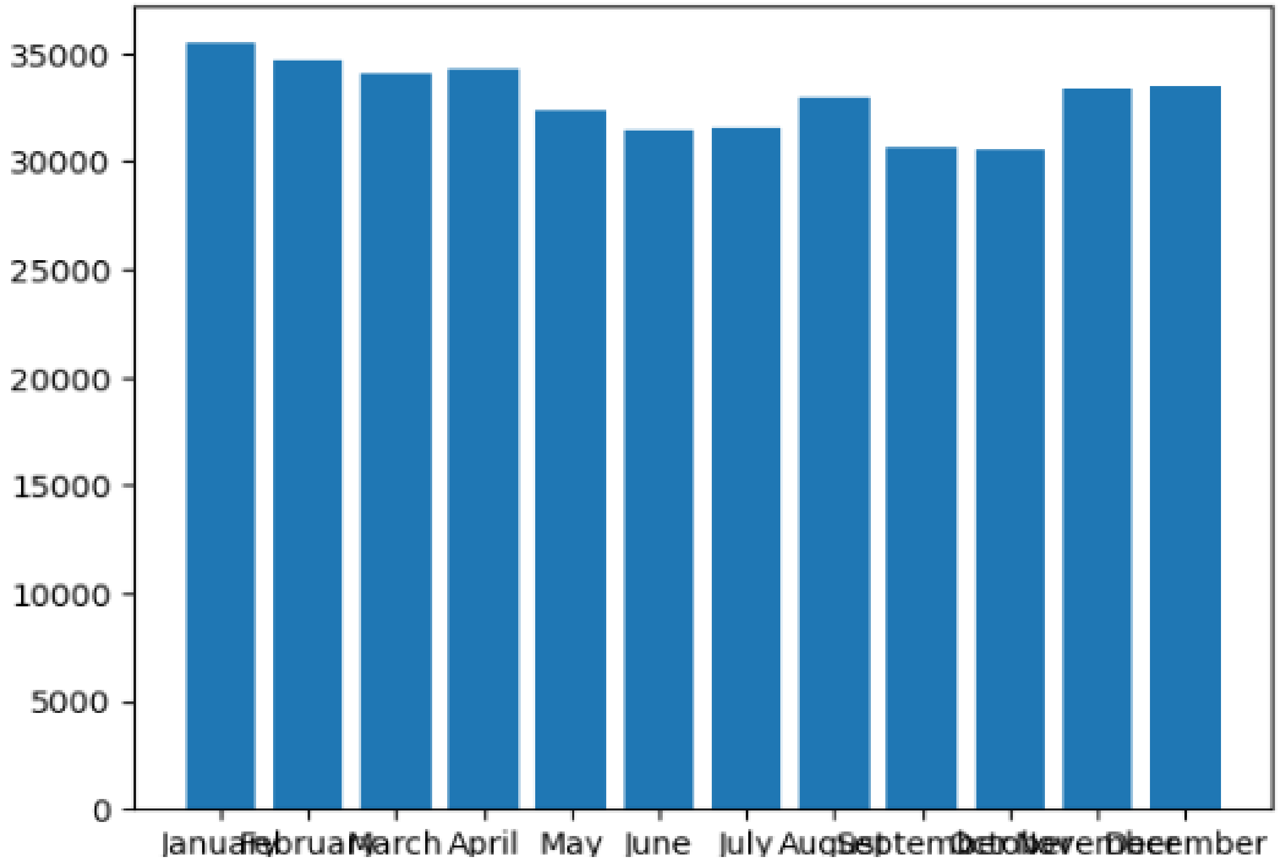


```
[9]: plt.plot(djia_data['Date'], djia_data['Open'], label = 'Open')
     plt.plot(djia_data['Date'], djia_data['Close'], label = 'Close')
     plt.legend()
     plt.show()
```

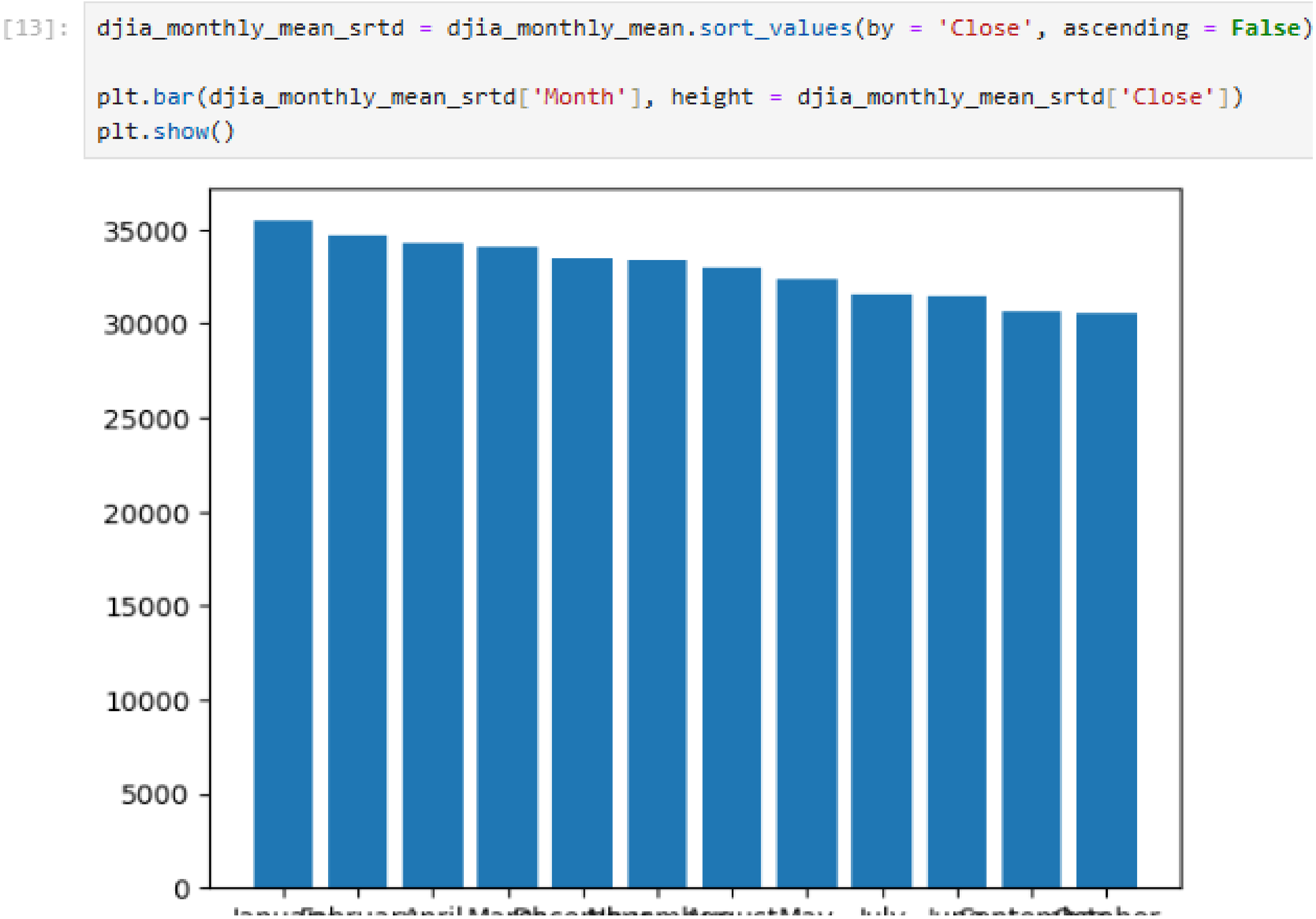
Drawing Bar Plots:

[12]:

```
plt.bar(djia_monthly_mean['Month'], height = djia_monthly_mean['Close'])
plt.show()
```

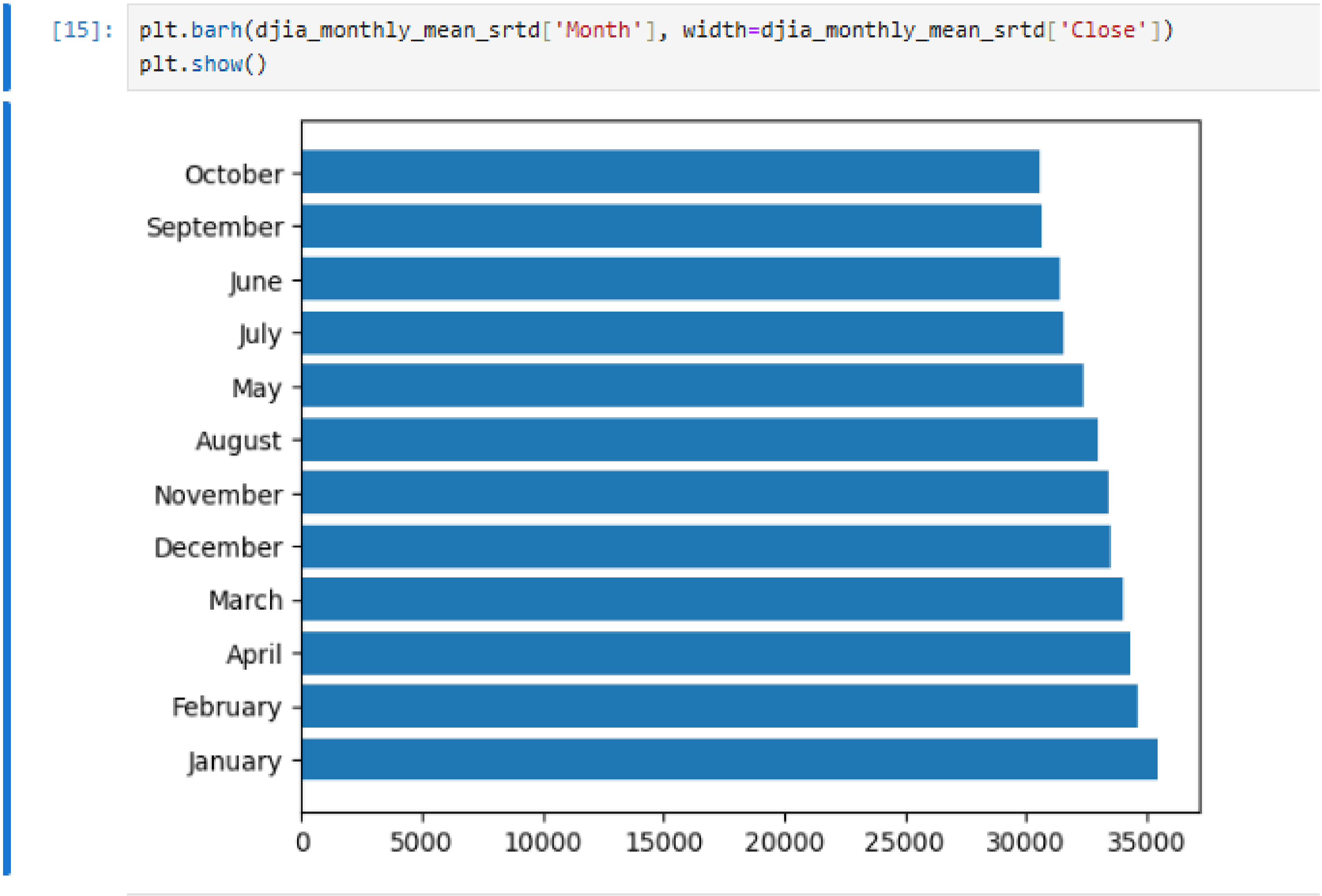


Vertical Bar Plots:



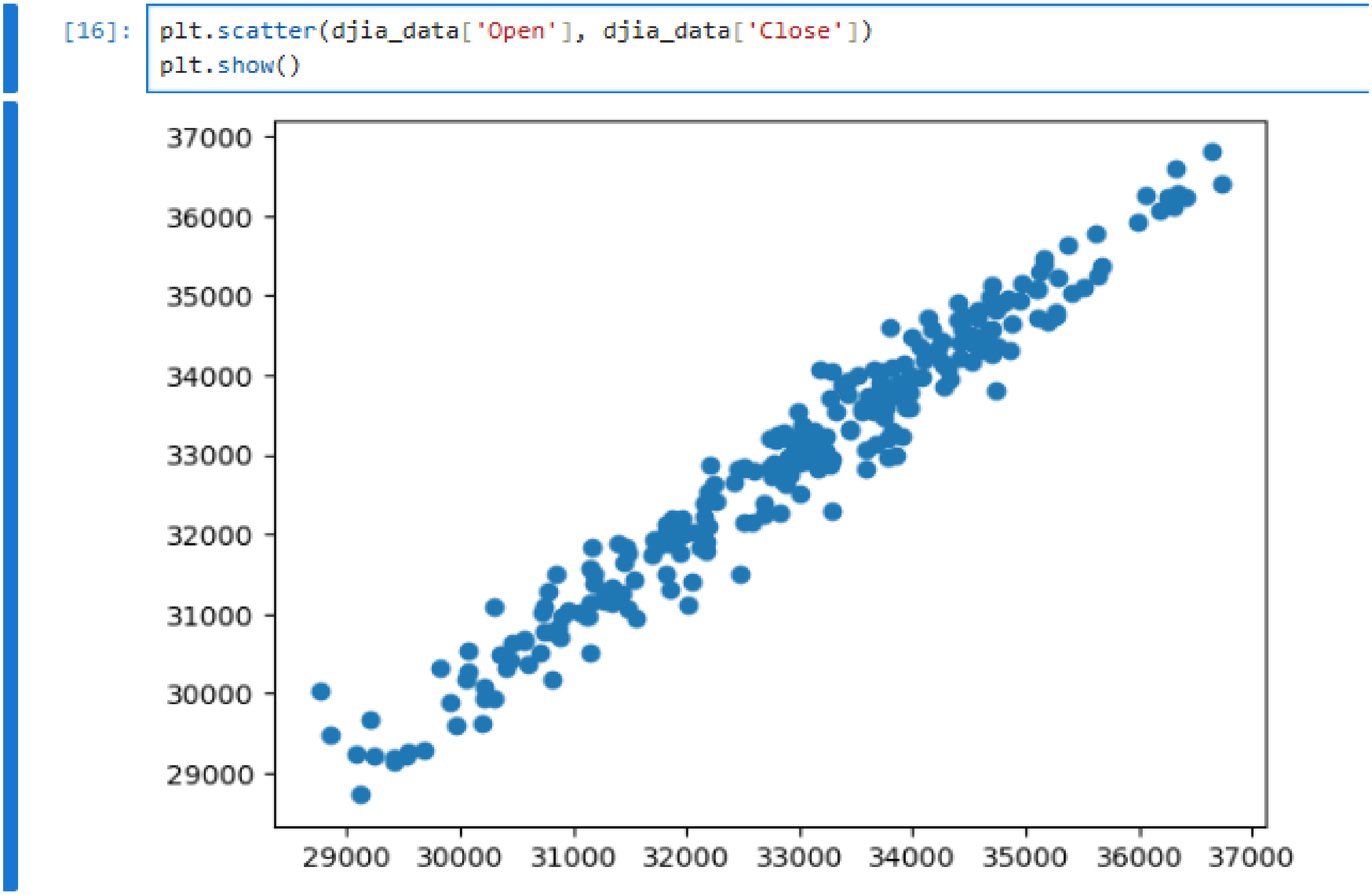
```
[13]: djia_monthly_mean_srtd = djia_monthly_mean.sort_values(by = 'Close', ascending = False)

      plt.bar(djia_monthly_mean_srtd['Month'], height = djia_monthly_mean_srtd['Close'])
      plt.show()
```

Horizontal Bar Plots:



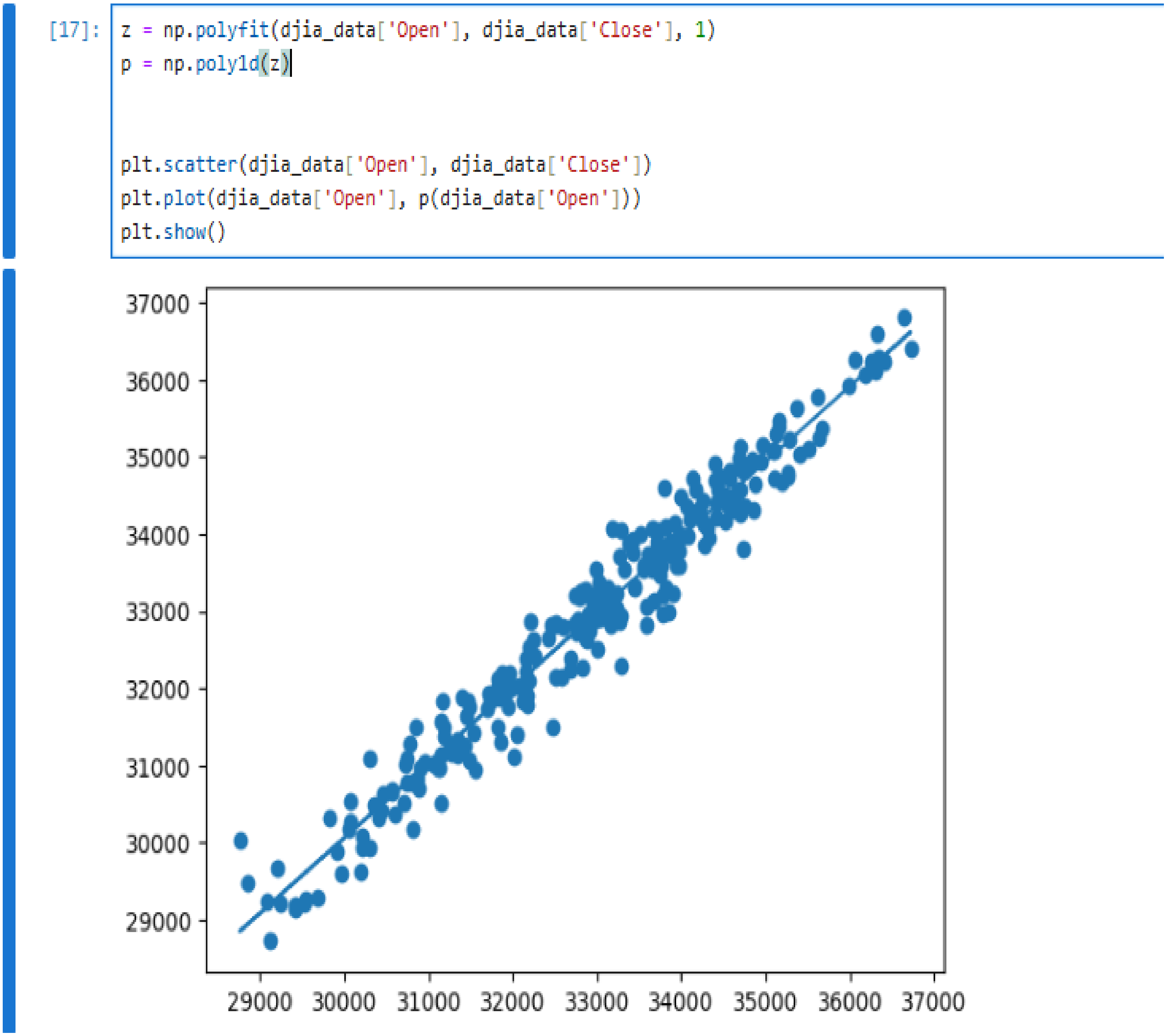
[15]:
```
plt.barh(djia_monthly_mean_srtd['Month'], width=djia_monthly_mean_srtd['Close'])
plt.show()
```

Drawing Scatter Plots:



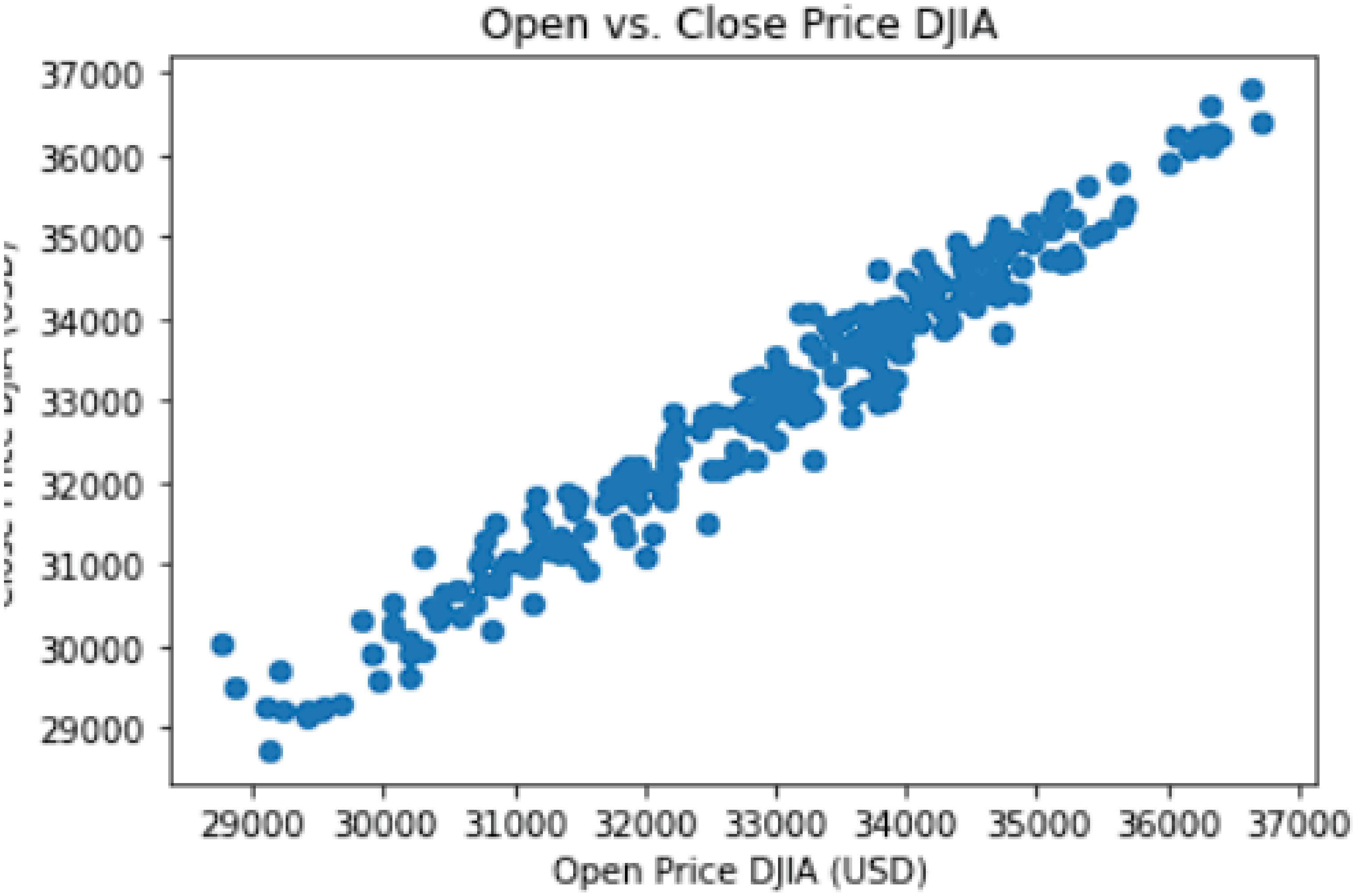
```
plt.scatter(djia_data['Open'], djia_data['Close'])
plt.show()
```

Scatter Plots with a Trend Line:

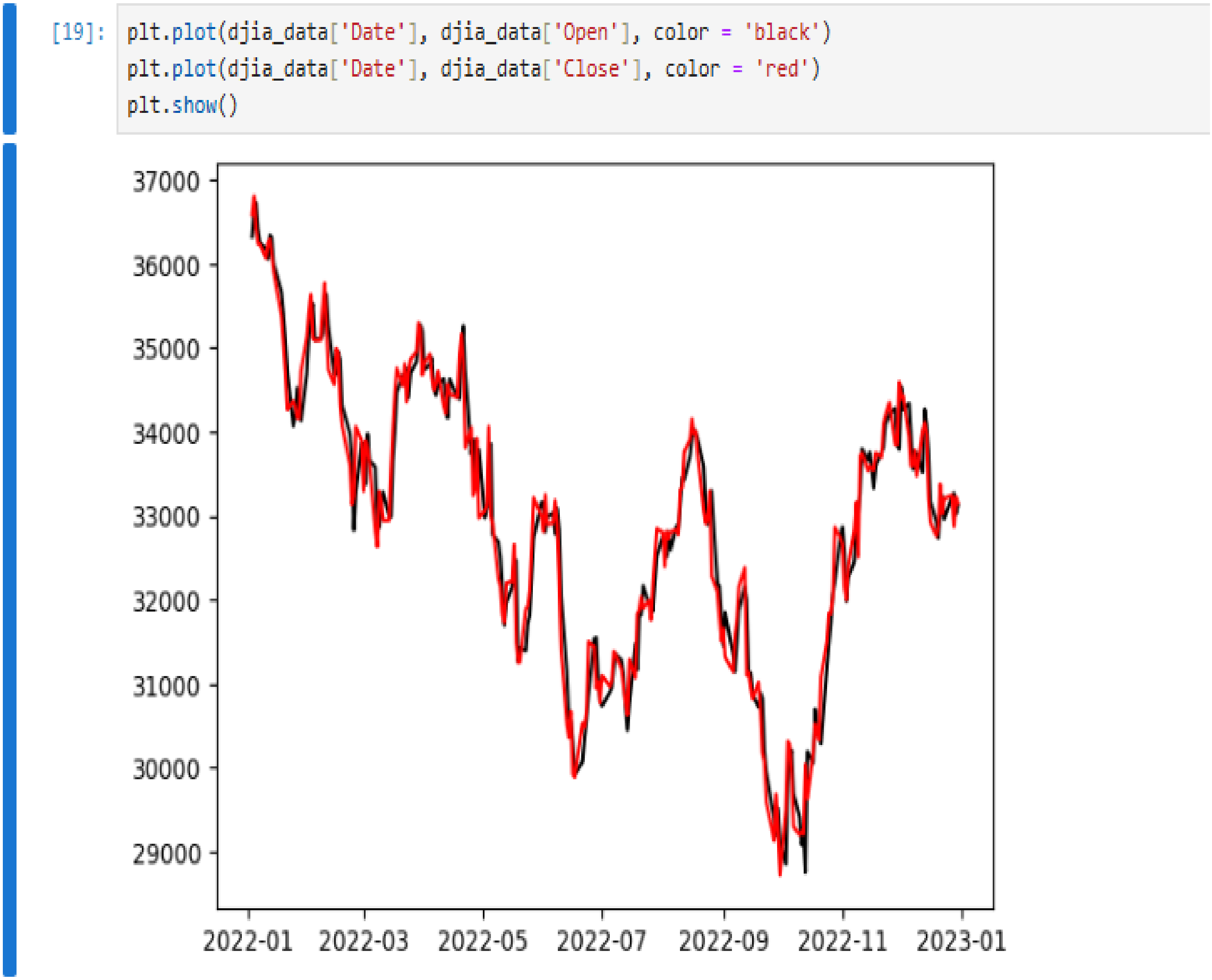


```
[17]: z = np.polyfit(djia_data['Open'], djia_data['Close'], 1)
      p = np.poly1d(z)


      plt.scatter(djia_data['Open'], djia_data['Close'])
      plt.plot(djia_data['Open'], p(djia_data['Open']))
      plt.show()
```

Setting the Plot Title and Axis Labels:

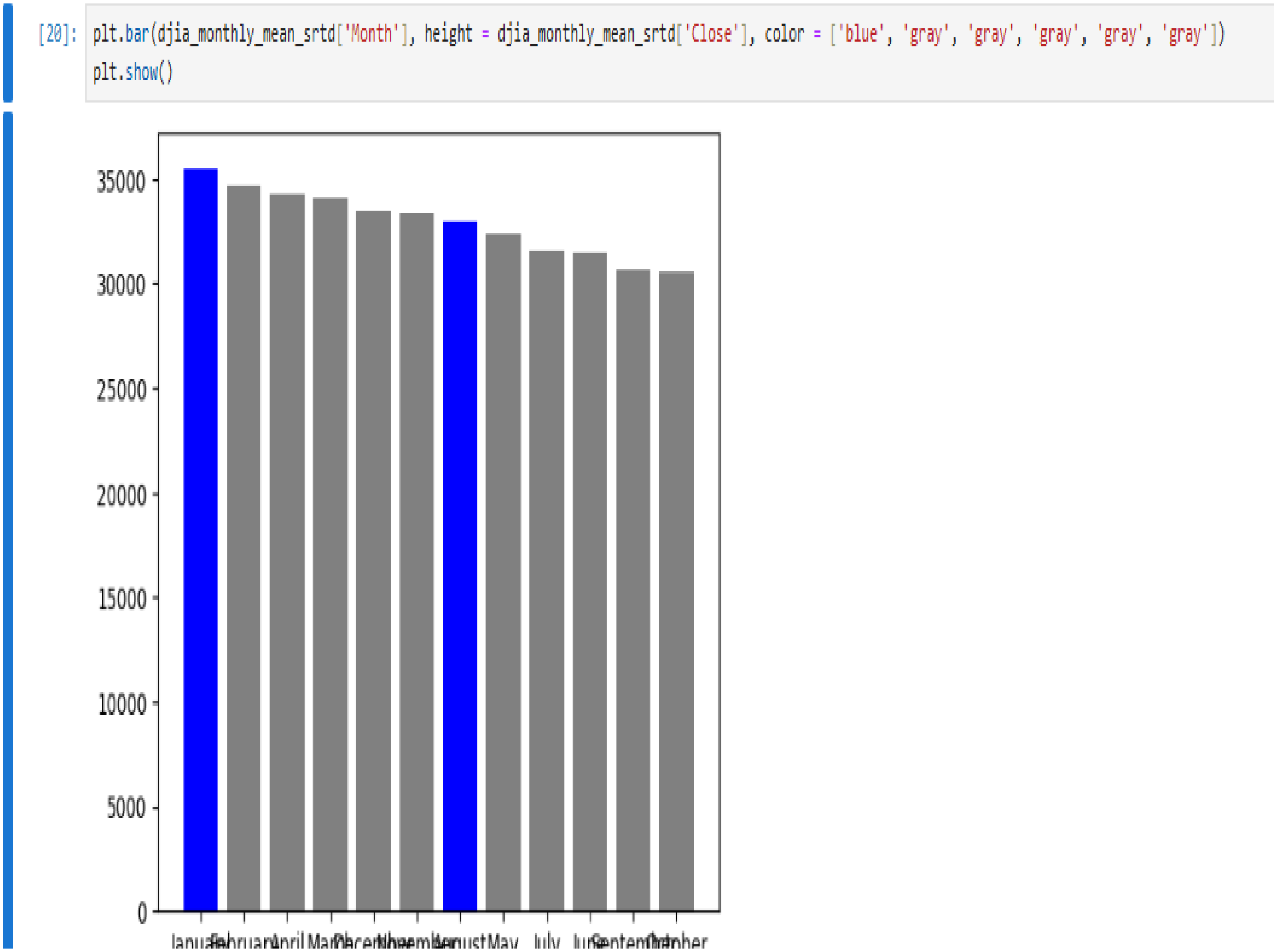


Changing colors: Lines



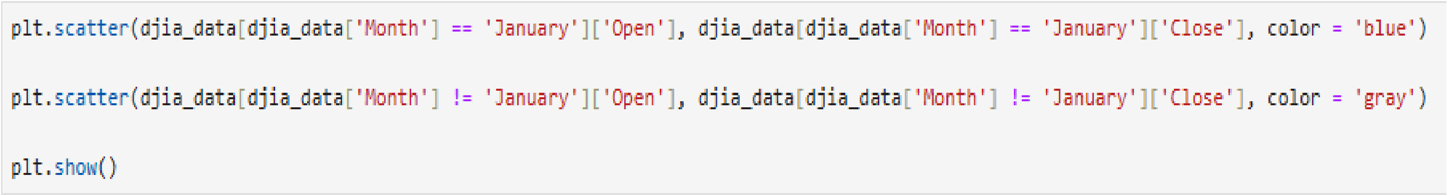
```
[19]: plt.plot(djia_data['Date'], djia_data['Open'], color = 'black')
      plt.plot(djia_data['Date'], djia_data['Close'], color = 'red')
      plt.show()
```

Changing Colors: Bars

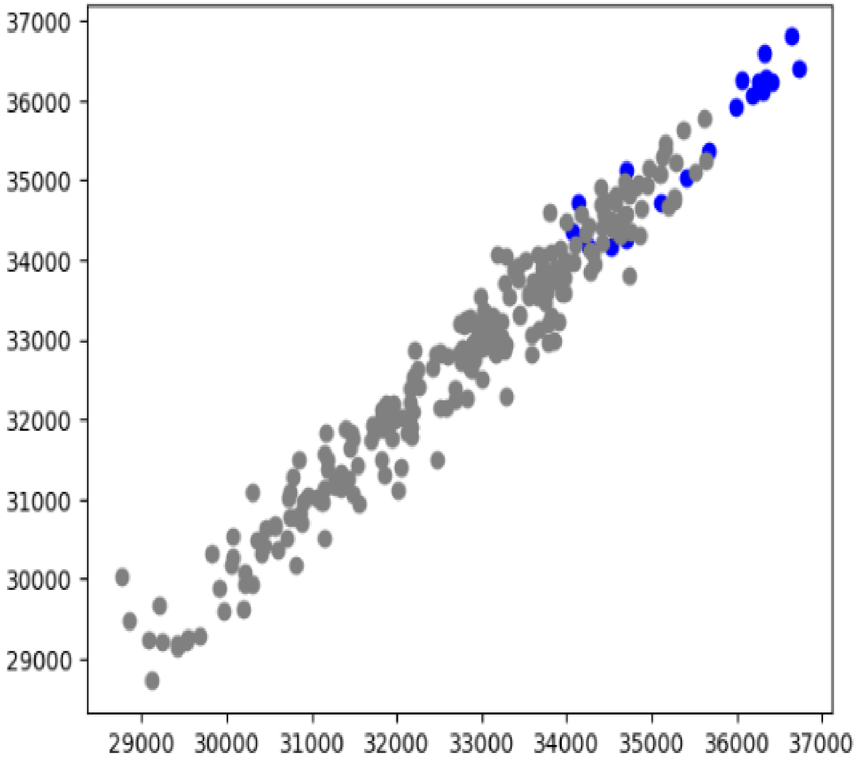


```
[20]: plt.bar(djia_monthly_mean_srtd['Month'], height = djia_monthly_mean_srtd['Close'], color = ['blue', 'gray', 'gray', 'gray', 'gray', 'gray'])
      plt.show()
```

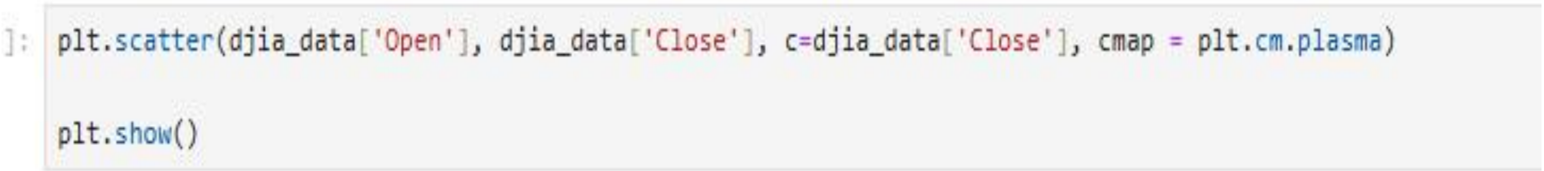
Changing Colors: Points



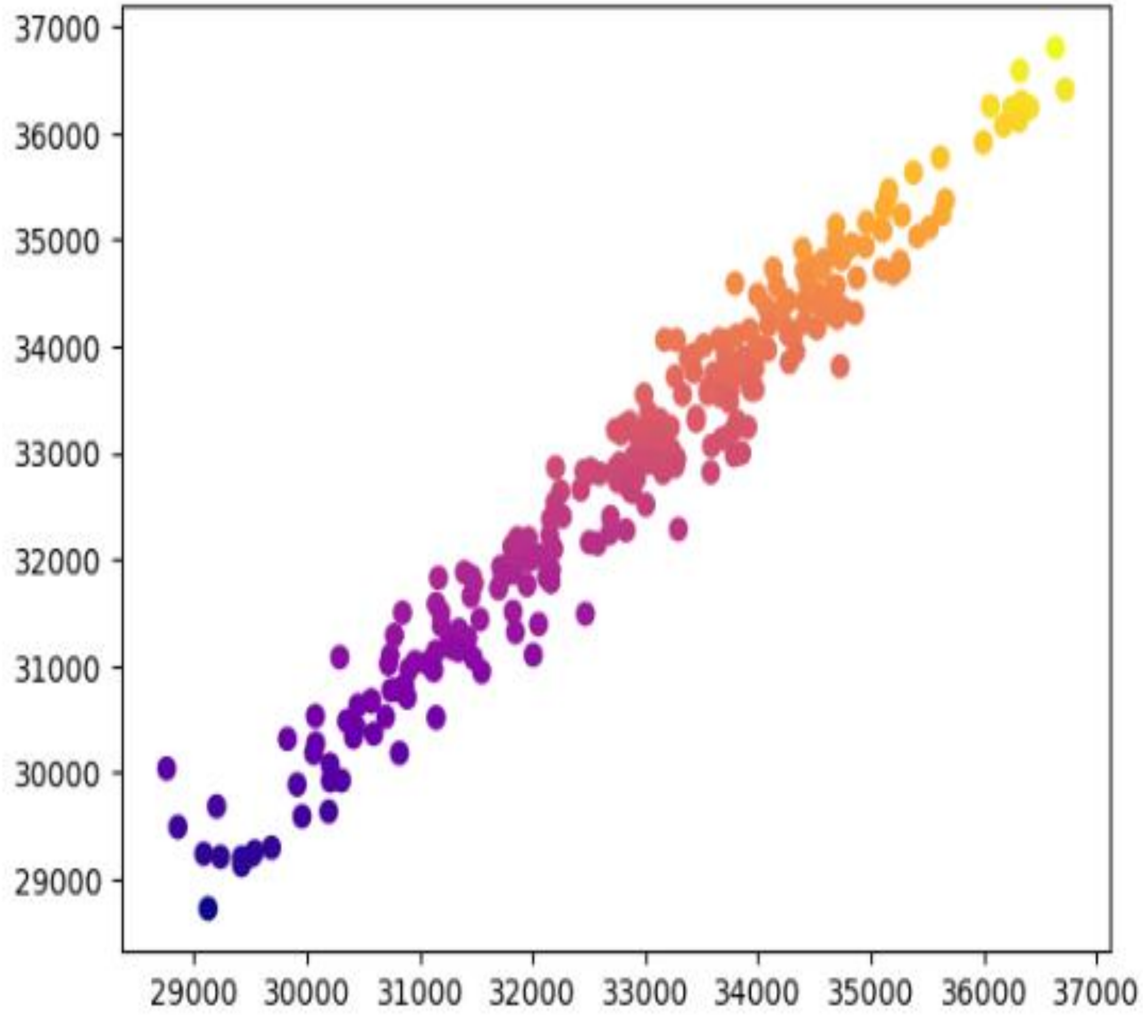
```
[21]: plt.scatter(djia_data[djia_data['Month'] == 'January']['Open'], djia_data[djia_data['Month'] == 'January']['Close'], color = 'blue')

      plt.scatter(djia_data[djia_data['Month'] != 'January']['Open'], djia_data[djia_data['Month'] != 'January']['Close'], color = 'gray')

      plt.show()
```



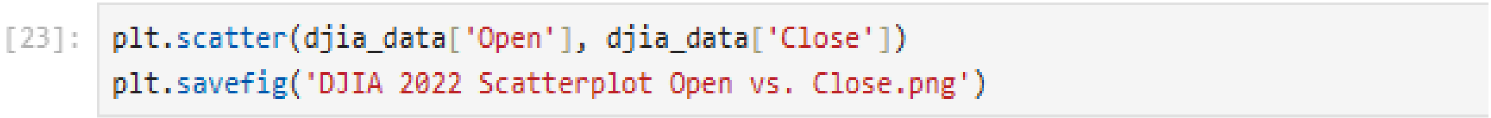
Using Color Maps:



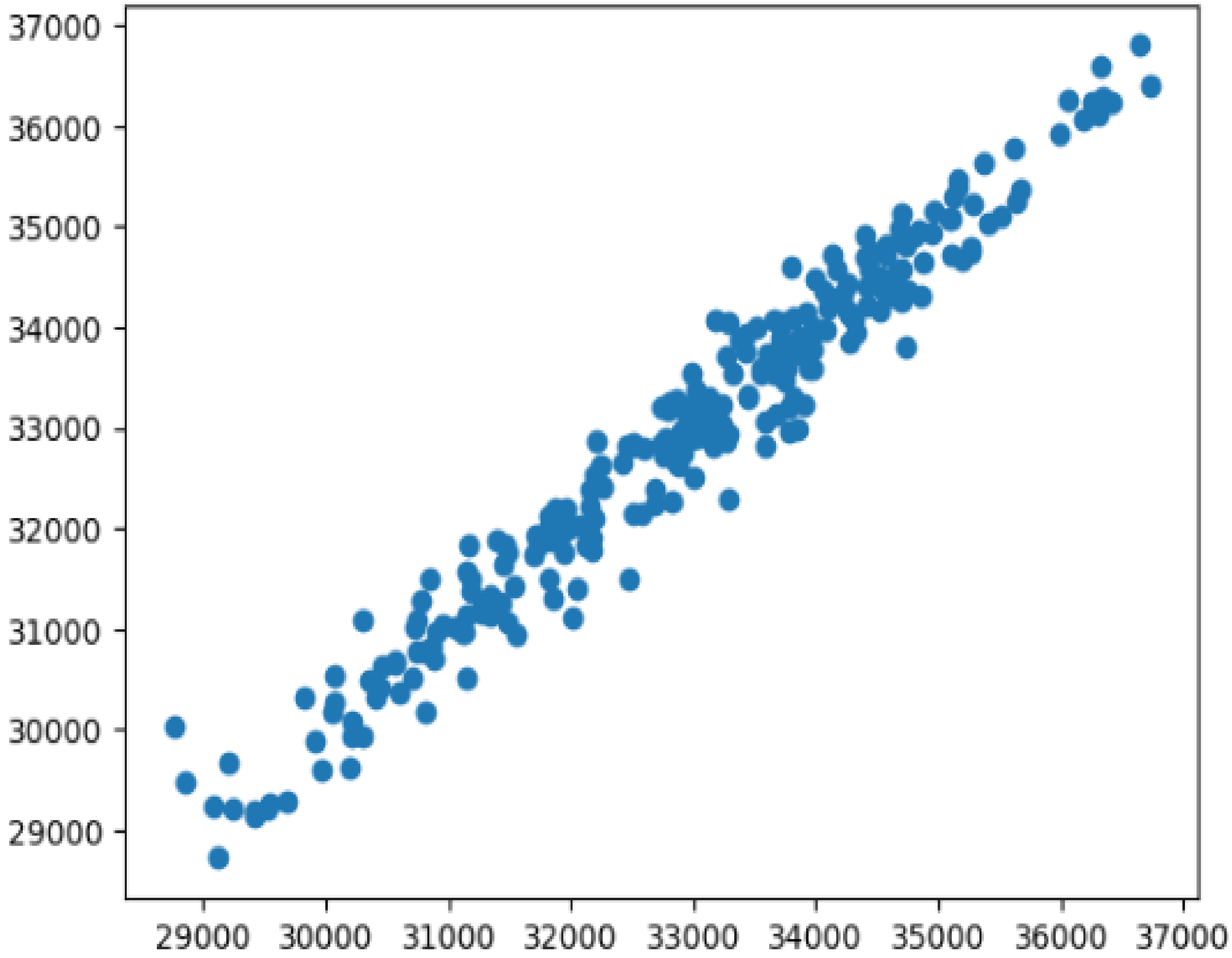
```
]: plt.scatter(djia_data['Open'], djia_data['Close'], c=djia_data['Close'], cmap = plt.cm.plasma)

   plt.show()
```



Setting Axis Limits & Saving a Plot:



```
[23]: plt.scatter(djia_data['Open'], djia_data['Close'])
      plt.savefig('DJIA 2022 Scatterplot Open vs. Close.png')
```


37000
36000
35000
34000
33000
32000
31000
30000
29000
29000
30000
31000
32000
33000
34000
35000
36000
37000