

Adam_Syed_02_matplotlib_exercises

September 10, 2023

1 Matplotlib Exercises

- The goal of the exercises
 - Get comfortable with Matplotlib
 - Establish workflows with Matplotlib
- At each step, there are cells after the instructions where you should write your code. Just below that, there is a cell showing the output you should get.

1.0.1 Step 1

- Import pandas as pd
- Then import matplotlib.pyplot as plt
- Then set the %matplotlib inline
- Set the figure(figsize parameter to (5,2) so that it fits nicely on the screen.

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

```
[2]: %matplotlib inline
plt.rcParams['figure.figsize'] = (5, 2)
```

1.0.2 Step 2

- Read the data from CSV file stock-data.csv
- Use the index_col=0
- Also, it is convenient to use parse_dates=True
- Take head of data

```
[5]: stock_data = pd.read_csv('stock_data.csv', index_col=0, parse_dates=True)
```

```
[ ]:
```

```
[ ]:
```

```
[6]: stock_data.head()
```

```
[6]:
```

	High	Low	Open	Close	Volume	Adj Close
Date						
2020-01-02	86.139999	84.342003	84.900002	86.052002	47660500.0	86.052002

2020-01-03	90.800003	87.384003	88.099998	88.601997	88892500.0	88.601997
2020-01-06	90.311996	88.000000	88.094002	90.307999	50665000.0	90.307999
2020-01-07	94.325996	90.671997	92.279999	93.811996	89410500.0	93.811996
2020-01-08	99.697998	93.646004	94.739998	98.428001	155721500.0	98.428001

1.0.3 Step 3

- make a plot of Close price.

[]:

```
[11]: plt.figure(figsize=(10, 6))
plt.plot(stock_data.index, stock_data['Close'], label='Close Price', color='blue')
plt.title('Close Price Over Time')
plt.xlabel('Date')
plt.ylabel('Close Price')
plt.legend()
plt.grid(True)
plt.show()
```



1.0.4 Step 5

- Make a figure and axes by calling matplotlib.pyplot from plt
- Now make a chart with Closing price

```
[ ]:
```

```
[14]: fig, ax = plt.subplots(figsize=(10, 6))
ax.plot(stock_data.index, stock_data['Close'], label='Close Price',
        color='blue')
ax.set_title('Close Price Over Time')
ax.set_xlabel('Date')
ax.set_ylabel('Close Price')
ax.legend()
ax.grid(True)
plt.show()
```



1.0.5 Step 6

- Make new figure of 4 subplots in 2 rows and 2 columns
- Make charts of Open, High, Low, and Close in each axes

```
[ ]:
```

```
[ ]:
```

```
[ ]:
```

```
[17]: fig, axes = plt.subplots(2, 2, figsize=(12, 8))

axes[0, 0].plot(stock_data.index, stock_data['Open'], label='Open Price',
               ↪color='blue')
axes[0, 0].set_title('Open Price')

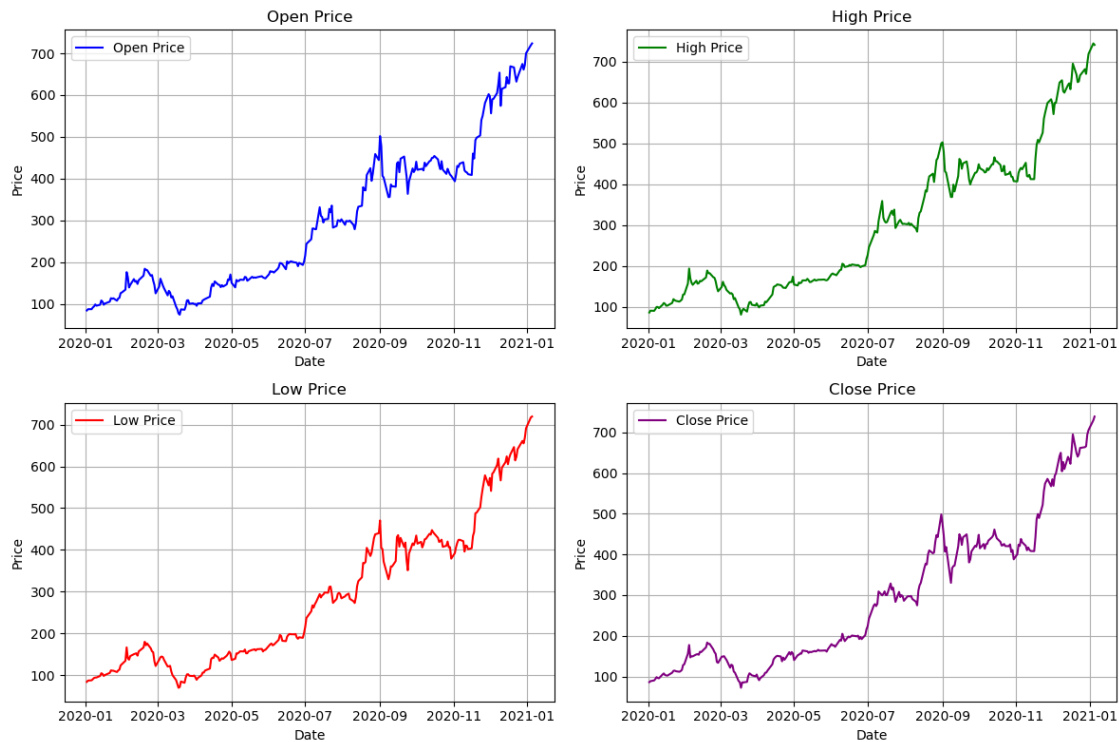
axes[0, 1].plot(stock_data.index, stock_data['High'], label='High Price',
               ↪color='green')
axes[0, 1].set_title('High Price')

axes[1, 0].plot(stock_data.index, stock_data['Low'], label='Low Price',
               ↪color='red')
axes[1, 0].set_title('Low Price')

axes[1, 1].plot(stock_data.index, stock_data['Close'], label='Close Price',
               ↪color='purple')
axes[1, 1].set_title('Close Price')

for ax in axes.flat:
    ax.set_xlabel('Date')
    ax.set_ylabel('Price')
    ax.legend()
    ax.grid(True)

plt.tight_layout()
plt.show()
```



1.0.6 Step 7

- Make a new figure
- Make a bar chart with volume for January 2020

```
[47]: data.head()
```

```
[47]:
```

Date	High	Low	Open	Close	Volume	Adj Close
2020-01-02	86.139999	84.342003	84.900002	86.052002	47660500.0	86.052002
2020-01-03	90.800003	87.384003	88.099998	88.601997	88892500.0	88.601997
2020-01-06	90.311996	88.000000	88.094002	90.307999	50665000.0	90.307999
2020-01-07	94.325996	90.671997	92.279999	93.811996	89410500.0	93.811996
2020-01-08	99.697998	93.646004	94.739998	98.428001	155721500.0	98.428001

```
[48]:
```

```
[ ]:
```

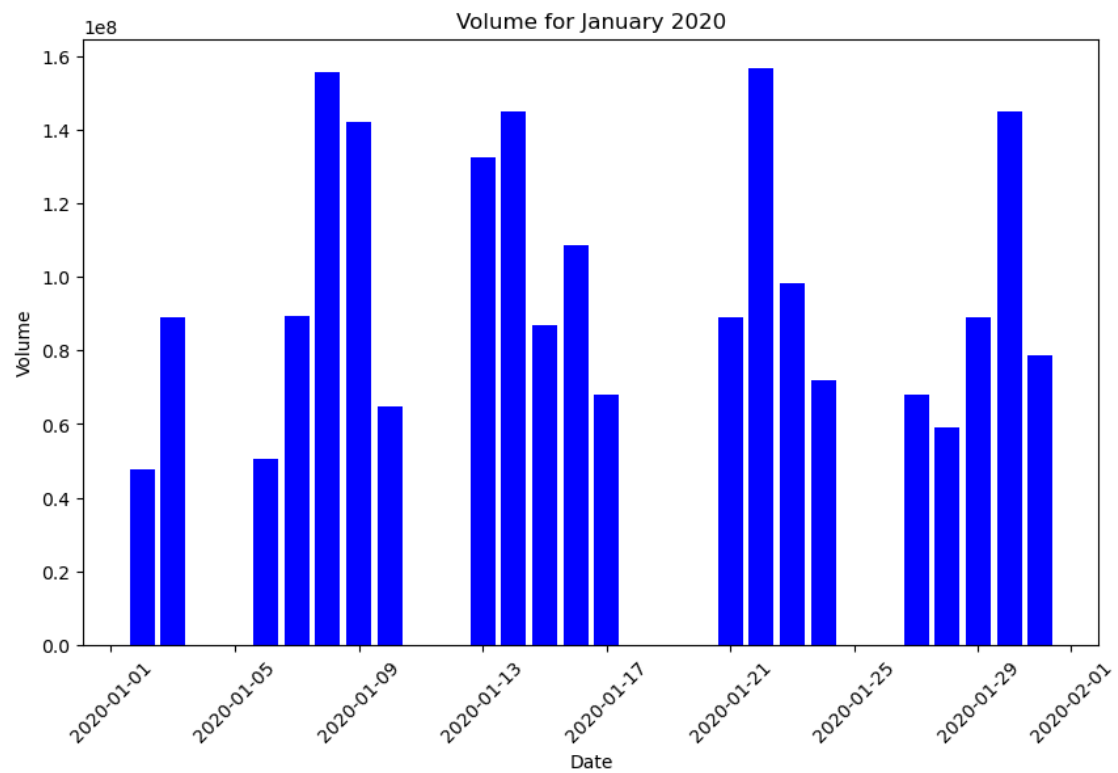
```
[19]: january_2020_data = stock_data.loc['2020-01']

plt.figure(figsize=(10, 6))

plt.bar(january_2020_data.index, january_2020_data['Volume'], color='blue')
```

```
plt.title('Volume for January 2020')
plt.xlabel('Date')
plt.ylabel('Volume')
plt.xticks(rotation=45)

plt.show()
```



[]: