Hands-on Activity 9.1 Data Visualization using Pandas and Matplotlib

Name: Cuadra, Audrick Zander G.

Section: CPE22S3

Date: April 1, 2024

Submitted to: Engr. Roman Richard

Instructions:

- Create a Python notebook to answer all shown procedures, exercises and analysis in this section.

Resources:

- Download the following datasets: earthquakes-1.csv Download earthquakes-1.csv, fb_stock_prices_2018.csv

Procedures:

- 9.1 Introduction to Matplotlib
- 9.2 Plotting with Pandas
- 9.3 Pandas Plotting Subpackage

Data Analysis:

- Matplotlib is Python library that assists with visualizing data through plot generation. Scatter boxplots, histograms, and sub-plotting are just a few examples of what the user can do with Matplotlib. Lastly, it can have a wide range of customization which can accomodate the user's preference by utilizing the Pandas library. Pandas' plotting subpackages allows for a more advance plotting generation, examples of these includes lag plots, bootstrap plots, scatter matrix, and auto-correlation plots. These subpackages enables more functionalities in plotting allowing users to perform advanced analyses.

Supplementary Activity:

Using the CSV files provided and what we have learned so far in this module complete the following exercises:

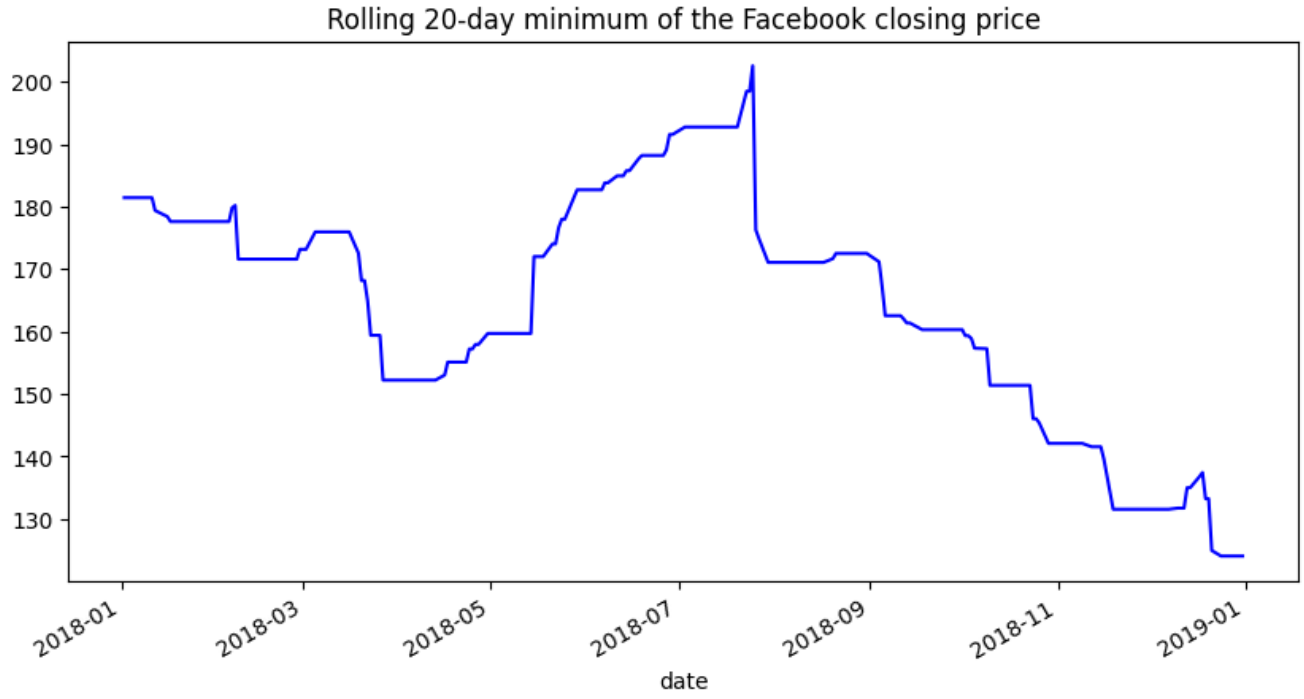
1. Plot the rolling 20-day minimum of the Facebook closing price with the pandas plot() method.

```
import pandas as p

fb = p.read_csv('/content/fb_stock_prices_2018.csv', index_col='date', parse_dates=True)
fb['fb_roll_20'] = fb['close'].rolling('20D').min()
```

```
fb.plot(
    kind='line',
    y='fb_roll_20',
    figsize=(10, 5),
    style='b-',
    legend=False,
    title='Rolling 20-day minimum of the Facebook closing price'
)
```

```
<Axes: title={'center': 'Rolling 20-day minimum of the Facebook closing price'}, xlabel='date'>
```



2. Create a histogram and KDE of the change from open to close in the price of Facebook stock.

```
1 fb['open_to_close'] = fb['open'] - fb['close']
2 fb
```

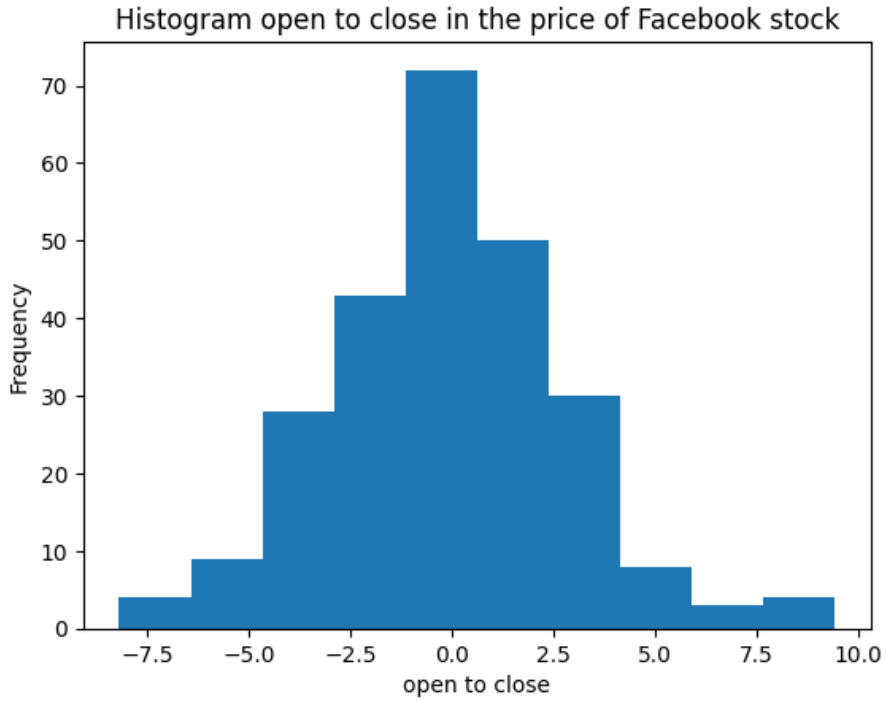
date	open	high	low	close	volume	fb_roll_20	open_to_close
2018-01-02	177.68	181.58	177.5500	181.42	18151903	181.42	-3.74
2018-01-03	181.88	184.78	181.3300	184.67	16886563	181.42	-2.79
2018-01-04	184.90	186.21	184.0996	184.33	13880896	181.42	0.57
2018-01-05	185.59	186.90	184.9300	186.85	13574535	181.42	-1.26
2018-01-08	187.20	188.90	186.3300	188.28	17994726	181.42	-1.08
...	...	...	...	...	...	...	...
2018-12-24	123.10	129.74	123.0200	124.06	22066002	124.06	-0.96
2018-12-26	126.00	134.24	125.8900	134.18	39723370	124.06	-8.18
2018-12-27	132.44	134.99	129.6700	134.52	31202509	124.06	-2.08
2018-12-28	135.34	135.92	132.2000	133.20	22627569	124.06	2.14
2018-12-31	134.45	134.64	129.9500	131.09	24625308	124.06	3.36

251 rows × 7 columns

Next steps: View recommended plots

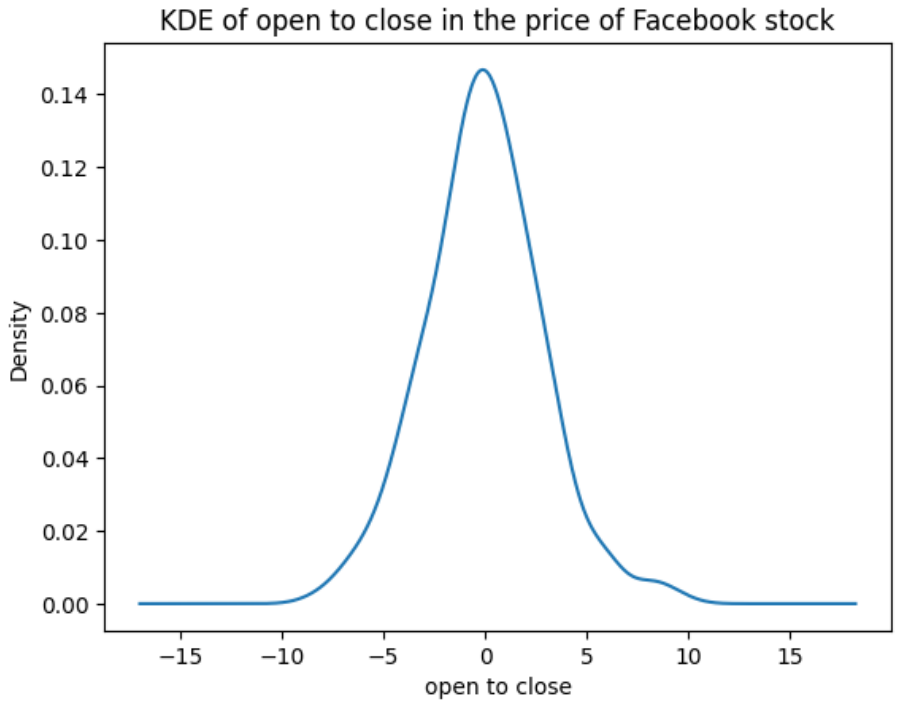
```
1 import matplotlib.pyplot as mpl
2
3 fb.open_to_close.plot(
4     kind='hist',
5     title='Histogram open to close in the price of Facebook stock'
6 )
7 mpl.xlabel('open to close')
```

```
Text(0.5, 0, 'open to close')
```




```
fb.open_to_close.plot(
    kind='kde',
    title='KDE of open to close in the price of Facebook stock'
)
mpl.xlabel('open to close') # label the x-axis (discussed in chapter 6)
```

```
Text(0.5, 0, 'open to close')
```



3. Using the earthquake data, create box plots for the magnitudes of each magType used in Indonesia.

```
earthquakes = p.read_csv('/content/earthquakes-1.csv')
earthquakes
```

	mag	magType	time	place	tsunami	parsed_place
0	1.35	ml	1539475168010	9km NE of Aguanga, CA	0	California
1	1.29	ml	1539475129610	9km NE of Aguanga, CA	0	California
2	3.42	ml	1539475062610	8km NE of Aguanga, CA	0	California
3	0.44	ml	1539474978070	9km NE of Aguanga, CA	0	California
4	2.16	md	1539474716050	10km NW of Avenal, CA	0	California
...	...	...	...	...	...	...
9327	0.62	md	1537230228060	9km ENE of Mammoth Lakes, CA	0	California
9328	1.00	ml	1537230135130	3km W of Julian, CA	0	California
9329	2.40	md	1537229908180	35km NNE of Hatillo, Puerto Rico	0	Puerto Rico
9330	1.10	ml	1537229545350	9km NE of Aguanga, CA	0	California
9331	0.66	ml	1537228864470	9km NE of Aguanga, CA	0	California

9332 rows × 6 columns

Next steps: View recommended plots

```
1 mag_indo = earthquakes.query('parsed_place == "Indonesia"')
2 mag_indo
```

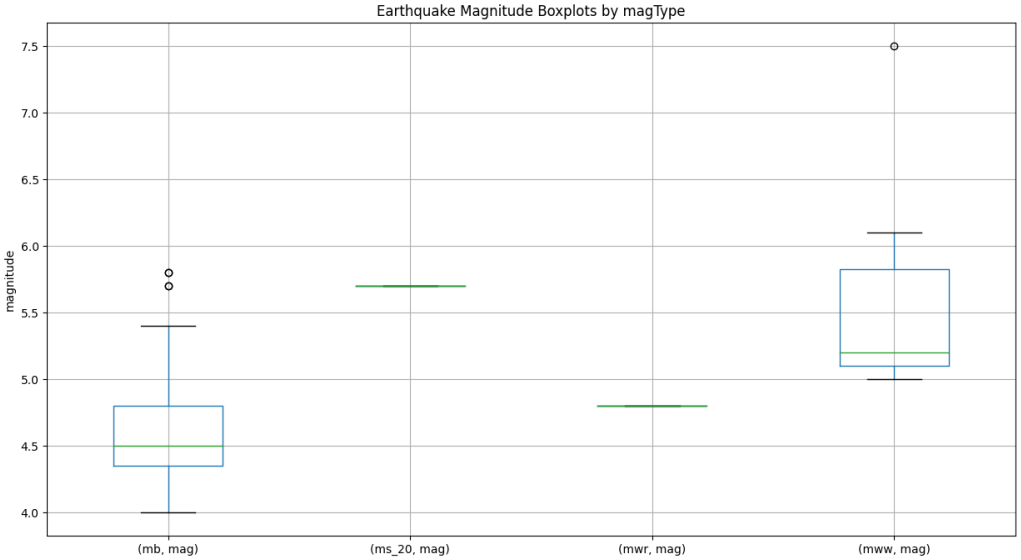
	mag	magType	time	place	tsunami	parsed_place
9	4.7	mb	1539472814760	219km SSE of Saparua, Indonesia	0	Indonesia
13	4.5	mb	1539470898340	120km SSW of Banda Aceh, Indonesia	0	Indonesia
180	5.2	mww	1539405255580	25km E of Bitung, Indonesia	0	Indonesia
421	4.7	mb	1539331098920	38km SSW of Nggongi Satu, Indonesia	0	Indonesia
660	4.4	mb	1539258833830	51km WSW of Kasiguncu, Indonesia	0	Indonesia
...	...	...	...	...	...	...
9041	4.3	mb	1537296305750	7km WSW of Karangsubagan, Indonesia	0	Indonesia
9075	4.4	mb	1537288723310	103km W of Kuripan, Indonesia	0	Indonesia
9108	4.0	mb	1537280181100	123km NE of Bitung, Indonesia	0	Indonesia
9209	4.7	mb	1537256021950	18km NE of Reuleuet, Indonesia	0	Indonesia
9212	4.7	mb	1537255636260	2km ESE of Lokokrangan, Indonesia	0	Indonesia

147 rows × 6 columns

Next steps: View recommended plots

```
1 mag_indo[['mag', 'magType']].groupby('magType').boxplot(
2     figsize=(15, 8), subplots=False
3 )
4 mpl.title('Earthquake Magnitude Boxplots by magType')
5 mpl.ylabel('magnitude')
```

```
Text(0, 0.5, 'magnitude')
```



4. Make a line plot of the difference between the weekly maximum high price and the weekly minimum low price for Facebook. This should be a single line.

```
print("Maximum high price:", max(fb.high))
```

```
Maximum high price: 218.62
```

```
print("Minimum low price:", min(fb.low))
```

```
Minimum low price: 123.02
```

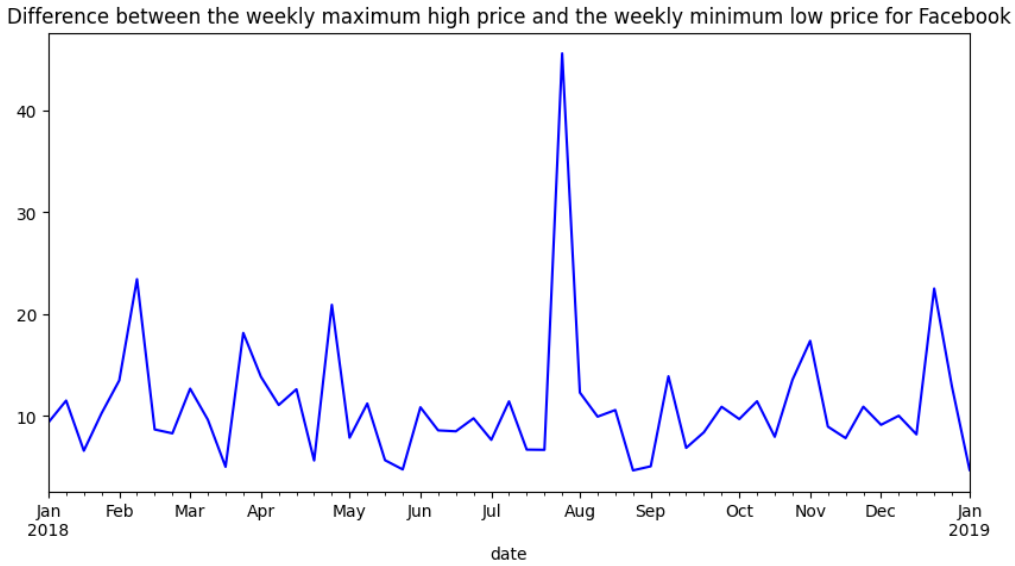
```
fb_diff = fb.resample('W').apply(
    lambda x: x['high'].max() - x['low'].min()
)
fb_diff
```

```
date
2018-01-07     9.3500
2018-01-14    11.5000
2018-01-21     6.5700
2018-01-28    10.2500
2018-02-04    13.4800
2018-02-11    23.4300
2018-02-18     8.6600
2018-02-25     8.2800
2018-03-04    12.6700
2018-03-11     9.6200
2018-03-18     4.9900
2018-03-25    18.1500
2018-04-01    13.8300
2018-04-08    11.0650
2018-04-15    12.6100
2018-04-22     5.6100
2018-04-29    20.9100
2018-05-06     7.8500
2018-05-13    11.2100
2018-05-20     5.6400
2018-05-27     4.7300
2018-06-03    10.8392
```

```
2018-06-10     8.5700
2018-06-17     8.4800
2018-06-24     9.7600
2018-07-01     7.6400
2018-07-08    11.4200
2018-07-15     6.6800
2018-07-22     6.6600
2018-07-29    45.6200
2018-08-05    12.2900
2018-08-12     9.9200
2018-08-19    10.5700
2018-08-26     4.6400
2018-09-02     5.0401
2018-09-09    13.8900
2018-09-16     6.8500
2018-09-23     8.3844
2018-09-30    10.8900
2018-10-07     9.6800
2018-10-14    11.4300
2018-10-21     7.9400
2018-10-28    13.5400
2018-11-04    17.3700
2018-11-11     8.9400
2018-11-18     7.8100
2018-11-25    10.9000
2018-12-02     9.1160
2018-12-09    10.0099
2018-12-16     8.1800
2018-12-23    22.5100
2018-12-30    12.9000
2019-01-06     4.6900
Freq: W-SUN, dtype: float64
```

```
fb_diff.plot(
    kind='line',
    y='max_high_min_low_diff',
    figsize=(10, 5),
    style='b-',
    legend=False,
    title='Difference between the weekly maximum high price and the weekly minimum low price for Facebook'
)
```

```
<Axes: title={'center': 'Difference between the weekly maximum high price and the
weekly minimum low price for Facebook'}, xlabel='date'>
```



5. Using matplotlib and pandas, create two subplots side-by-side showing the effect that after-hours trading has had on Facebook's stock price:

- The first subplot will contain a line plot of the daily difference between that day's opening price and the prior day's closing price (be sure to review the Time series section of Aggregating Pandas DataFrames for an easy way to do this).
- The second subplot will be a bar plot showing the net effect this had monthly, using resample().
- Bonus #1: Color the bars according to whether they are gains in the stock price (green) or drops in the stock price (red).
- Bonus #2: Modify the x-axis of the bar plot to show the threeletter abbreviation for the month.

```
1 fb.head()
```

date	open	high	low	close	volume	fb_roll_20	open_to_close
2018-01-02	177.68	181.58	177.5500	181.42	18151903	181.42	-3.74
2018-01-03	181.88	184.78	181.3300	184.67	16886563	181.42	-2.79
2018-01-04	184.90	186.21	184.0996	184.33	13880896	181.42	0.57
2018-01-							

Next steps: View recommended plots

```
1 fb['daily_diff_open_close'] = fb['open'] - fb['close'].shift(1)
2
3 fb
```

date	open	high	low	close	volume	fb_roll_20	open_to_close	daily_dif
2018-01-02	177.68	181.58	177.5500	181.42	18151903	181.42	-3.74	
2018-01-03	181.88	184.78	181.3300	184.67	16886563	181.42	-2.79	
2018-01-04	184.90	186.21	184.0996	184.33	13880896	181.42	0.57	
2018-01-05	185.59	186.90	184.9300	186.85	13574535	181.42	-1.26	
2018-01-08	187.20	188.90	186.3300	188.28	17994726	181.42	-1.08	
...	...	...	...	...	...	...	...	
2018-12-24	123.10	129.74	123.0200	124.06	22066002	124.06	-0.96	
2018-12-26	126.00	134.24	125.8900	134.18	39723370	124.06	-8.18	
2018-								

Next steps: View recommended plots

```
fig, axs = mpl.subplots(1, 2, figsize=(12, 6))

axs[0].plot(fb.index, fb['daily_diff_open_close'], color='green')
axs[0].set_title('Daily difference of that day\'s opening price and yesterday\'s closing price')
axs[0].set_xlabel('Date')
axs[0].set_ylabel('Price Difference')
axs[0].axhline(0, color='black', linestyle='-')

monthly_net_eff = fb.resample('M')['daily_diff_open_close'].sum()

monthly_net_eff.plot(kind='bar', ax=axs[1], color=['midnightblue' if val >= 0 else 'orange' for val in monthly
axs[1].set_title('Monthly Net Effect of After-Hours Trading')
axs[1].set_xlabel('Month')
axs[1].set_ylabel('Net Effect')
axs[1].set_xticklabels([month.strftime('%b') for month in monthly_net_eff.index], rotation=45)
axs[1].axhline(0, color='black', linestyle='-')
```

```
<matplotlib.lines.Line2D at 0x79a7b204e920>
```