⌄ Hands-on Activity 9.1 Data Visualization using Pandas and Matplotlib

CPE311 - Computational Thinking with Python

Name: Delloson, Angelo Dan Renz G.

Section: CPE22S3

Performed on:

Submitted on:

Submitted to: Engr. Roman M. 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, fb_stock_prices_2018.csv

Procedures:

- 9.1 Introduction to Matplotlib: https://colab.research.google.com/drive/1I234xOS7EzjBXicP5dcpAvQDh9c_Y36U?usp=sharing
- 9.2 Plotting with Pandas: https://colab.research.google.com/drive/1I-qTrQwv5PBSy_vQ-K7U7Eygaiye6ueO?usp=sharing
- 9.3 Pandas Plotting Subpackage: https://colab.research.google.com/drive/1q9JUv7PT9RF5TueeJw2enh91qnNg446Q?usp=sharing

Data Analysis:

- Provide comments on output from the procedures above.

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.
2. Create a histogram and KDE of the change from open to close in the price of Facebook stock.
3. Using the earthquake data, create box plots for the magnitudes of each magType used in Indonesia.
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.
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 %matplotlib inline
2 import matplotlib.pyplot as plt
3 import pandas as pd
4 import numpy as np
5
6 fb = pd.read_csv('/content/fb_stock_prices_2018 (1).csv', index_col='date', parse_dates=True)
7 quakes = pd.read_csv('/content/earthquakes-1 (1).csv')
```
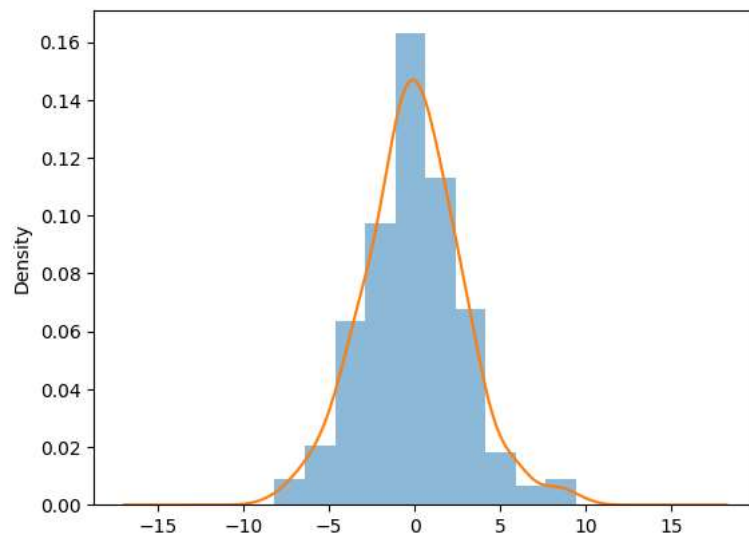
```
1 fb.rolling('20D').agg({'close':'min'}).plot()
```

⇥ `<Axes: xlabel='date'>`
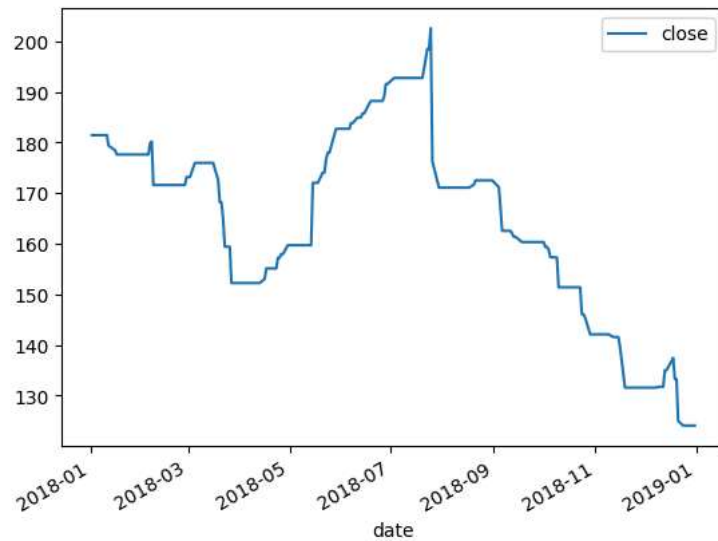

```
1 changeFB = fb.assign(change = lambda x : x.open-x.close)
2 ax = changeFB.change.plot(kind = 'hist', density = True, alpha = 0.5)
3 changeFB.change.plot(ax = ax, kind = 'kde')
```
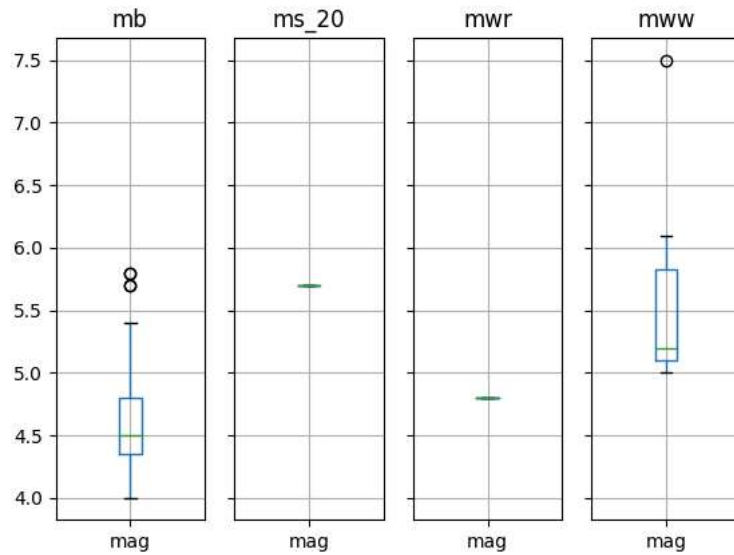
⇥ `<Axes: ylabel='Density'>`


```
1 quakes.query('parsed_place == "Indonesia"').groupby('magType').boxplot(column = 'mag', layout = (1,4))
```

```
mb          Axes(0.1,0.15;0.173913x0.75)
ms_20    Axes(0.308696,0.15;0.173913x0.75)
mwr      Axes(0.517391,0.15;0.173913x0.75)
mww      Axes(0.726087,0.15;0.173913x0.75)
dtype: object
```
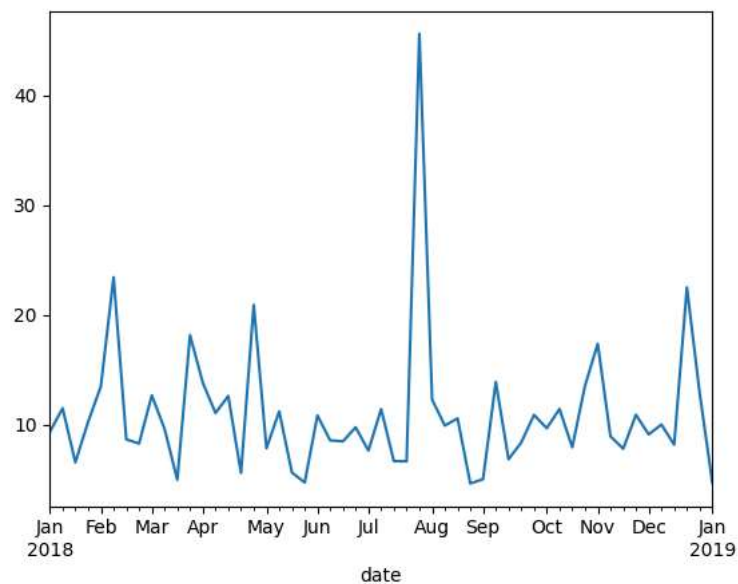


```
1 weeklyMax = fb.high.resample('W').max()
2 weeklyMin = fb.low.resample('W').min()
3 difference = weeklyMax - weeklyMin
4 difference.plot()
```

```
<Axes: xlabel='date'>
```



```
 1 fb = fb.assign(beforeClose= lambda x: x.close.shift(), after=lambda x: x.open - x.beforeClose, change= lambda x: x.after.abs())
 2
 3 figure = plt.figure(figsize = (14,7))
 4 grid = figure.add_gridspec(1,2)
 5 left = figure.add_subplot(grid[0,0])
 6 right = figure.add_subplot(grid[0,1])
 7
 8 left.plot(fb.index, fb.after)
 9 showEffect = fb.after.resample('M').sum()
10
11 colors = ['red' if value < 0 else 'green' for value in showEffect]
12 showEffect.plot(kind = 'bar', ax = right, color = colors)
13
14 right.set_xticklabels(showEffect.index.strftime('%b'))
```

```
[Text(0, 0, 'Jan'),
 Text(1, 0, 'Feb'),
 Text(2, 0, 'Mar'),
 Text(3, 0, 'Apr'),
 Text(4, 0, 'May'),
 Text(5, 0, 'Jun'),
 Text(6, 0, 'Jul'),
 Text(7, 0, 'Aug'),
 Text(8, 0, 'Sep'),
 Text(9, 0, 'Oct'),
 Text(10, 0, 'Nov'),
 Text(11, 0, 'Dec')]
```