

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

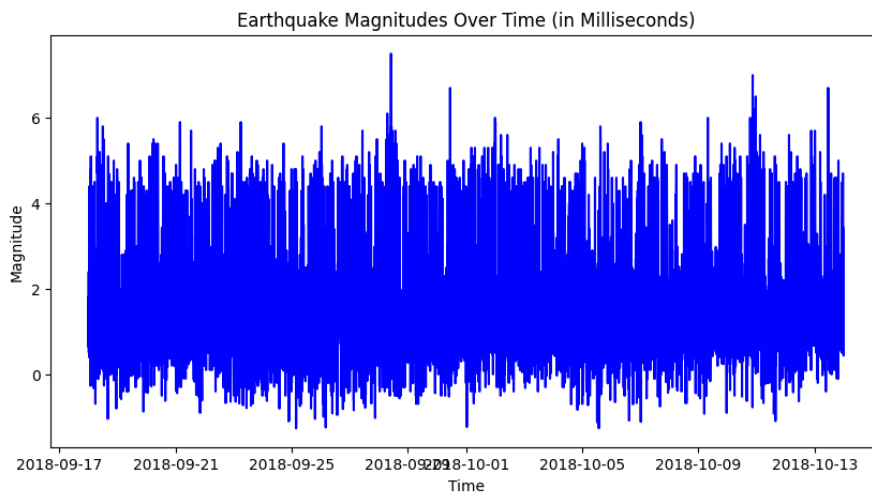
✓ Procedures:

✓ 9.1 Introduction to Matplotlib

```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3
4 earthquakes = pd.read_csv('/content/data/earthquakes.csv')
5 earthquakes['time'] = pd.to_datetime(earthquakes['time'], unit='ms')
```

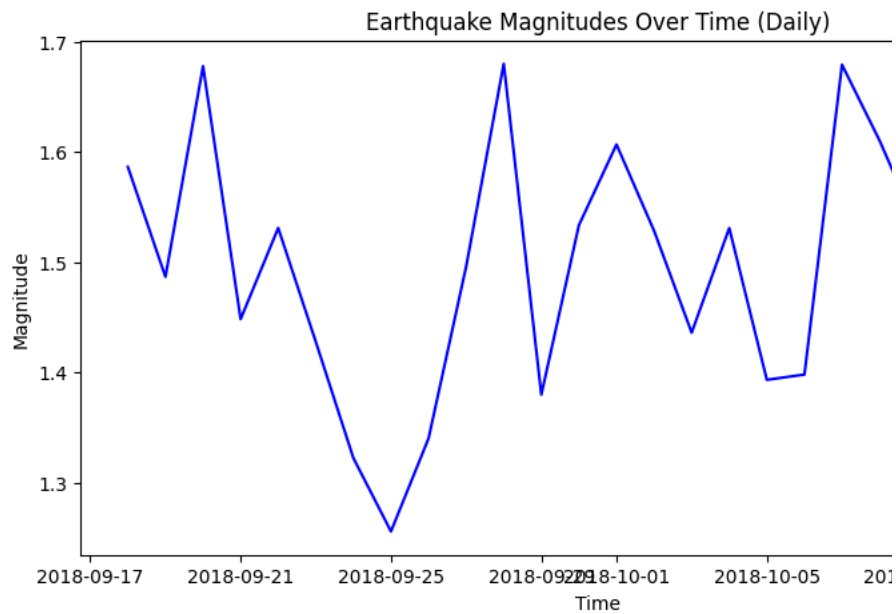
✓ Line Plot

```
1 plt.figure(figsize=(10, 5))
2 plt.plot(earthquakes['time'], earthquakes['mag'], color='blue')
3 plt.xlabel('Time')
4 plt.ylabel('Magnitude')
5 plt.title('Earthquake Magnitudes Over Time (in Milliseconds)')
6 plt.show()
```



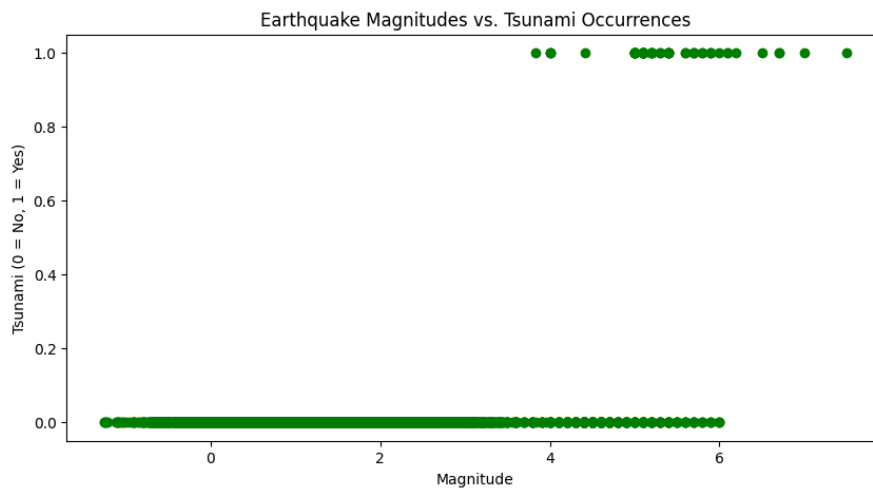
```
1 downsampled_line_plot = earthquakes.resample('D', on='time').mean()
2 plt.figure(figsize=(10, 5))
3 plt.plot(downsampled_line_plot.index, downsampled_line_plot['mag'], color='blue')
4 plt.xlabel('Time')
5 plt.ylabel('Magnitude')
6 plt.title('Earthquake Magnitudes Over Time (Daily)')
7 plt.show()
```

```
<ipython-input-34-e3e4343e960d>:1: FutureWarning: The default value of numeric_only in
downsampled_line_plot = earthquakes.resample('D', on='time').mean()
```



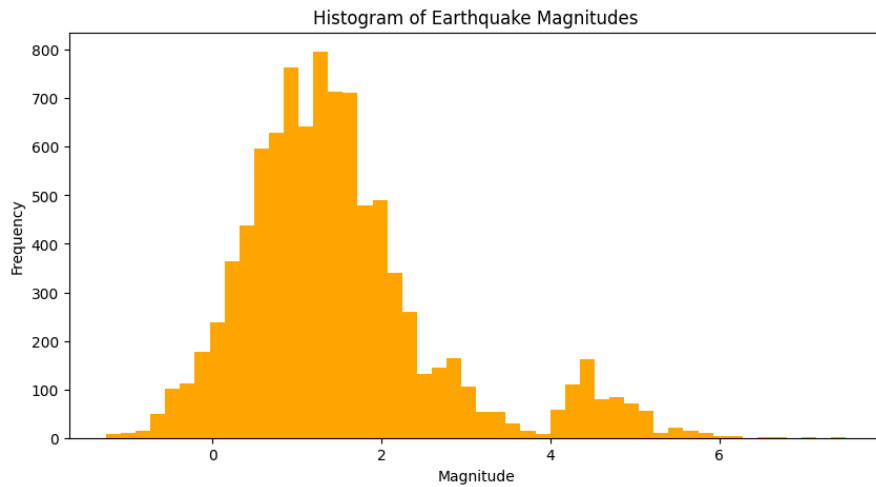
✓ Earthquake Scatter Plot

```
1 plt.figure(figsize=(10, 5))
2 plt.scatter(earthquakes['mag'], earthquakes['tsunami'], color='green')
3 plt.xlabel('Magnitude')
4 plt.ylabel('Tsunami (0 = No, 1 = Yes)')
5 plt.title('Earthquake Magnitudes vs. Tsunami Occurrences')
6 plt.show()
```



✓ Histogram

```
1 plt.figure(figsize=(10, 5))
2 plt.hist(earthquakes['mag'], bins=50, color='orange')
3 plt.xlabel('Magnitude')
4 plt.ylabel('Frequency')
5 plt.title('Histogram of Earthquake Magnitudes')
6 plt.show()
```



```
1 plt.savefig('Histogram_of_Earthquake_Magnitudes.png')
2 plt.show
```

```
matplotlib.pyplot.show
def show(*args, **kwargs)
```

```
/usr/local/lib/python3.10/dist-packages/matplotlib/pyplot.py
Display all open figures.
```

```
Parameters
```

```
-----
block : bool, optional
```

```
<Figure size 640x480 with 0 Axes>
```

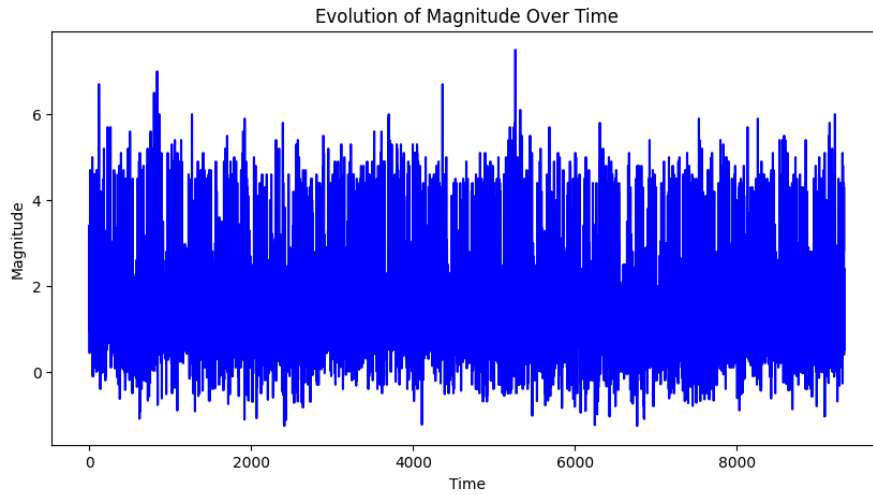
✓ 9.2 Plotting with Pandas

```
1 %matplotlib inline
2 import matplotlib.pyplot as plt
3 import numpy as np
4 import pandas as pd
5 from statsmodels.distributions.empirical_distribution import ECDF
6
7 earthquakes = pd.read_csv('/content/data/earthquakes.csv')
8 earthquakes['time'] = pd.to_datetime(earthquakes['time'], unit='ms')
```

✓ Line Plot

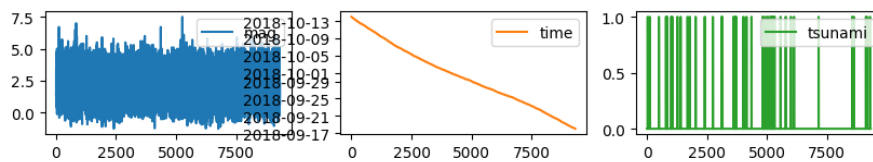
```
1 earthquakes.plot(kind='line',
2                 y='mag',
3                 figsize=(10, 5),
4                 style='b-',
5                 legend=False,
6                 title='Evolution of Magnitude Over Time')
7 plt.xlabel('Time')
8 plt.ylabel('Magnitude')
```

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



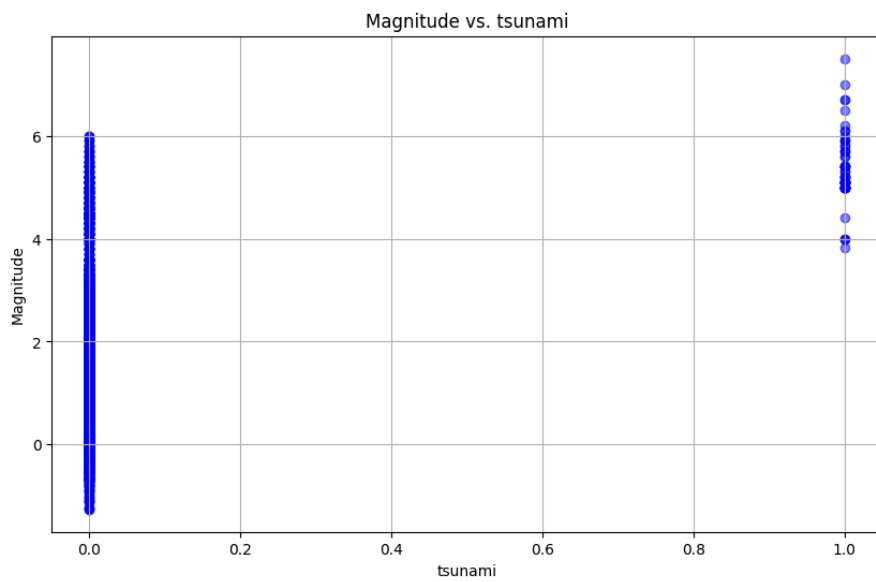
```
1 earthquakes.plot(kind='line',
2                   subplots = True,
3                   layout = (3, 3),
4                   figsize = (10, 5),
5                   title = 'Earthquakes')
```

```
array([[<Axes: >, <Axes: >, <Axes: >],
       [<Axes: >, <Axes: >, <Axes: >],
       [<Axes: >, <Axes: >, <Axes: >]], dtype=object)
Earthquakes
```



✓ Scatter Plot

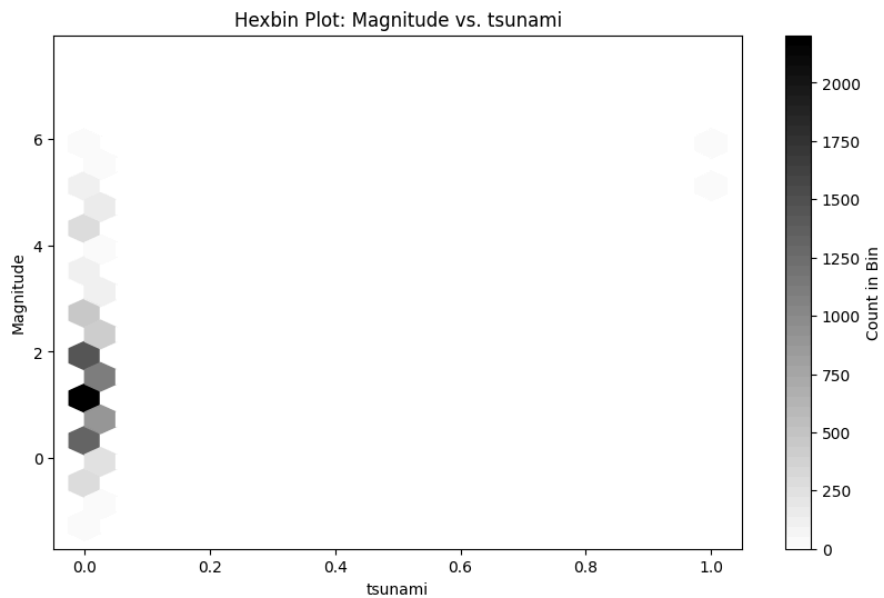
```
1 plt.figure(figsize=(10, 6))
2 plt.scatter(earthquakes['tsunami'], earthquakes['mag'], c='blue', alpha=0.5)
3 plt.title('Magnitude vs. tsunami')
4 plt.xlabel('tsunami')
5 plt.ylabel('Magnitude')
6 plt.grid(True)
```



✓ Hexbins

```
1 plt.figure(figsize=(10, 6))
2 plt.hexbin(earthquakes['tsunami'], earthquakes['mag'], gridsize=20, cmap='gray_r')
3 plt.colorbar(label='Count in Bin')
4 plt.title('Hexbin Plot: Magnitude vs. tsunami')
5 plt.xlabel('tsunami')
6 plt.ylabel('Magnitude')
```

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



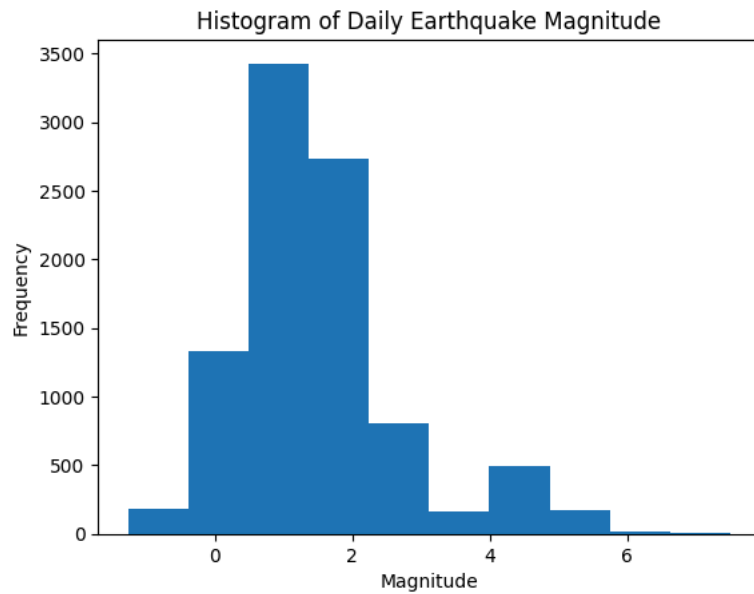
✓ Histograms

```

1 earthquakes.mag.plot(kind='hist',
2                       title='Histogram of Daily Earthquake Magnitude')
3 plt.xlabel('Magnitude')

```

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



✓ 9.3 Pandas Plotting Subpackage

```

1 %matplotlib inline
2 import matplotlib.pyplot as plt
3 import numpy as np
4 import pandas as pd
5
6 earthquakes = pd.read_csv('/content/data/earthquakes.csv', index_col='time', parse_dates=True)
7 earthquakes.index = pd.to_datetime(earthquakes.index, unit='ms')

```

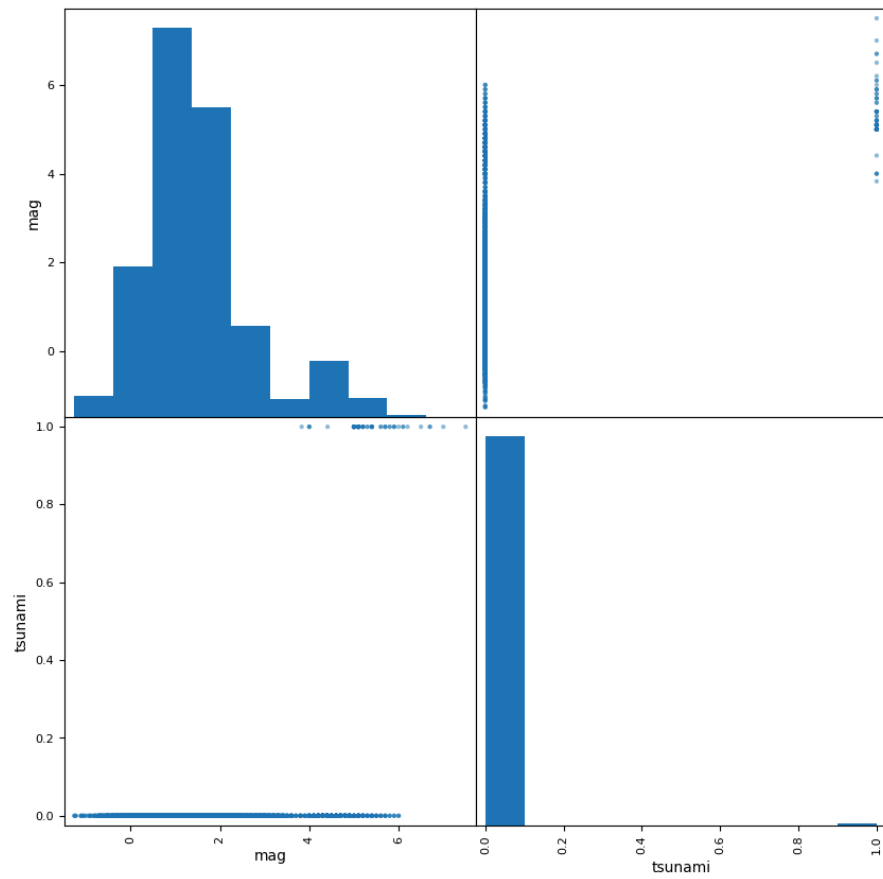
✓ Scatter Matrix

```

1 from pandas.plotting import scatter_matrix
2 scatter_matrix(earthquakes, figsize=(10, 10))

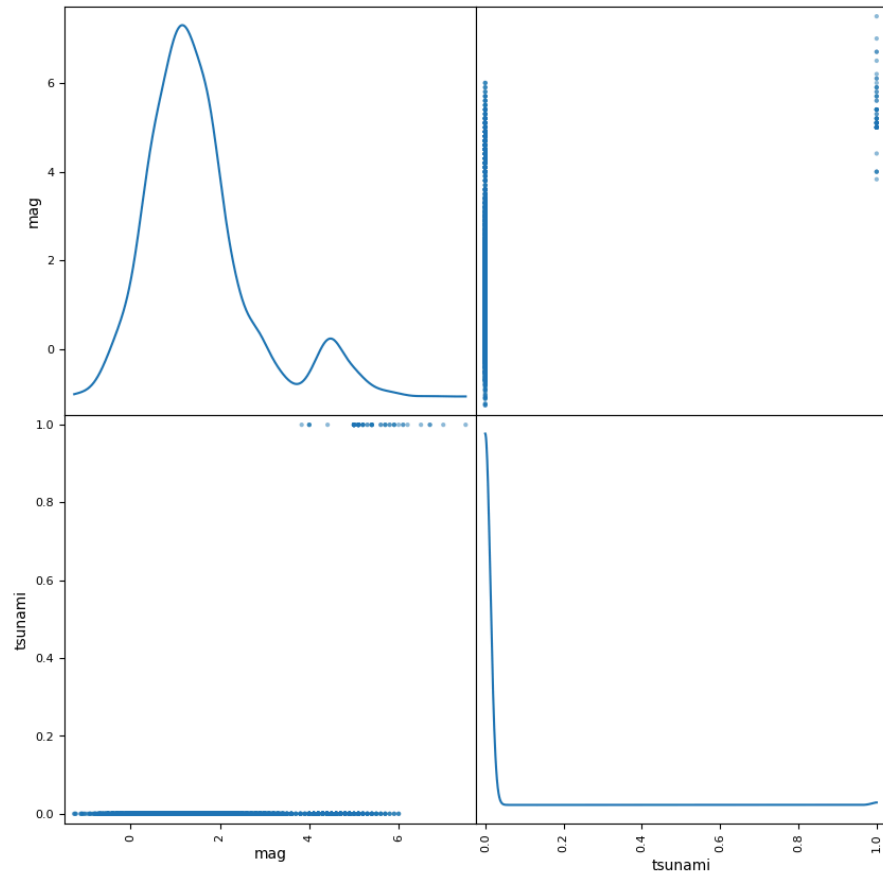
```

```
array([[<Axes: xlabel='mag', ylabel='mag'>,
        <Axes: xlabel='tsunami', ylabel='mag'>],
       [<Axes: xlabel='mag', ylabel='tsunami'>,
        <Axes: xlabel='tsunami', ylabel='tsunami'>]], dtype=object)
```



```
1 scatter_matrix(earthquakes, figsize=(10, 10), diagonal='kde')
```

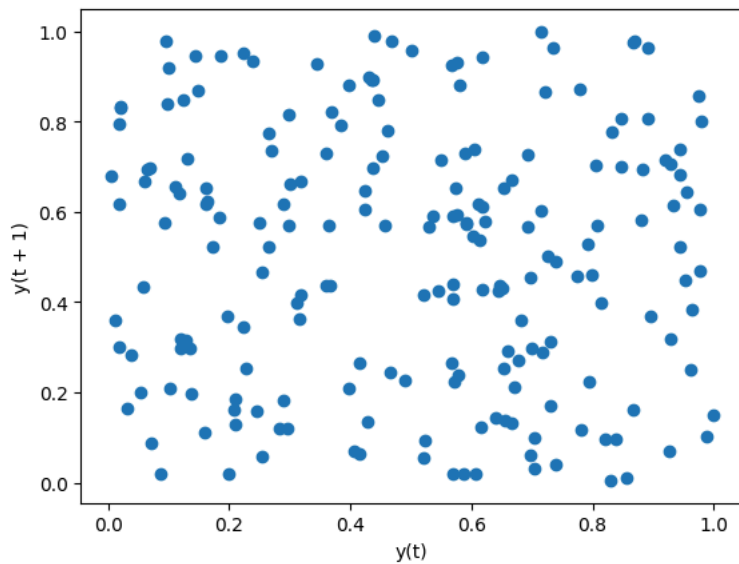
```
array([[<Axes: xlabel='mag', ylabel='mag'>,
      <Axes: xlabel='tsunami', ylabel='mag'>],
      [<Axes: xlabel='mag', ylabel='tsunami'>,
      <Axes: xlabel='tsunami', ylabel='tsunami'>]], dtype=object)
```



✓ Lag Plot

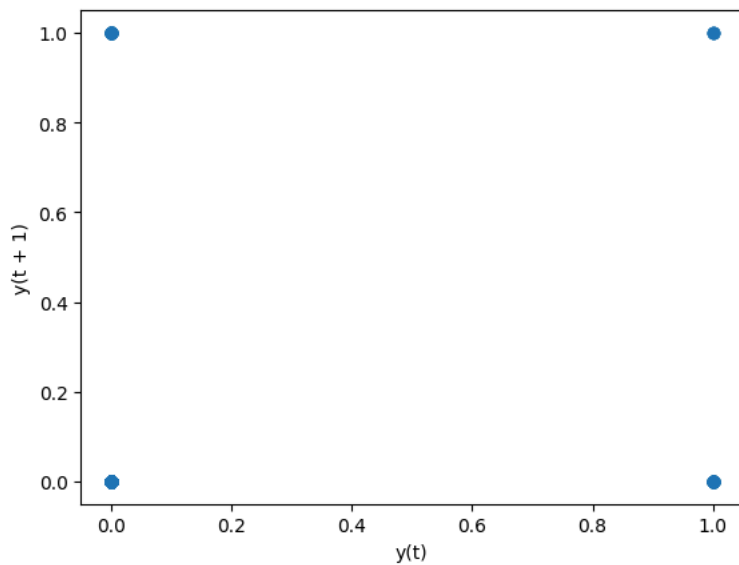
```
1 from pandas.plotting import lag_plot
2 np.random.seed(0)
3 lag_plot(pd.Series(np.random.random(size=200)))
```


<Axes: xlabel='y(t)', ylabel='y(t + 1)'>



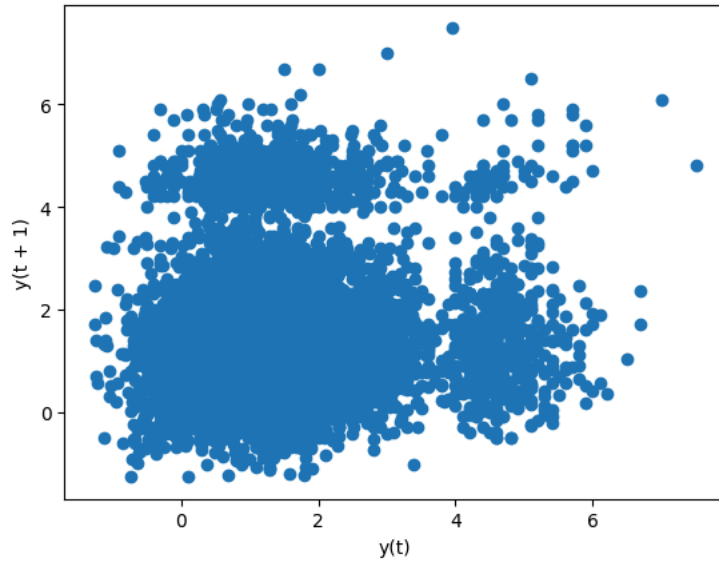
```
1 lag_plot(earthquakes.tsunami)
```

<Axes: xlabel='y(t)', ylabel='y(t + 1)'>



```
1 lag_plot(earthquakes.mag)
```

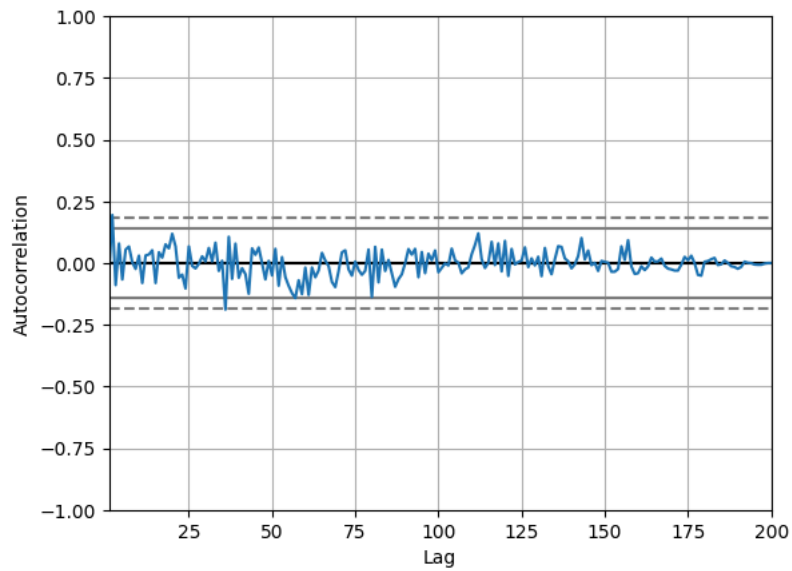
<Axes: xlabel='y(t)', ylabel='y(t + 1)'>



✓ Autocorrelation plots

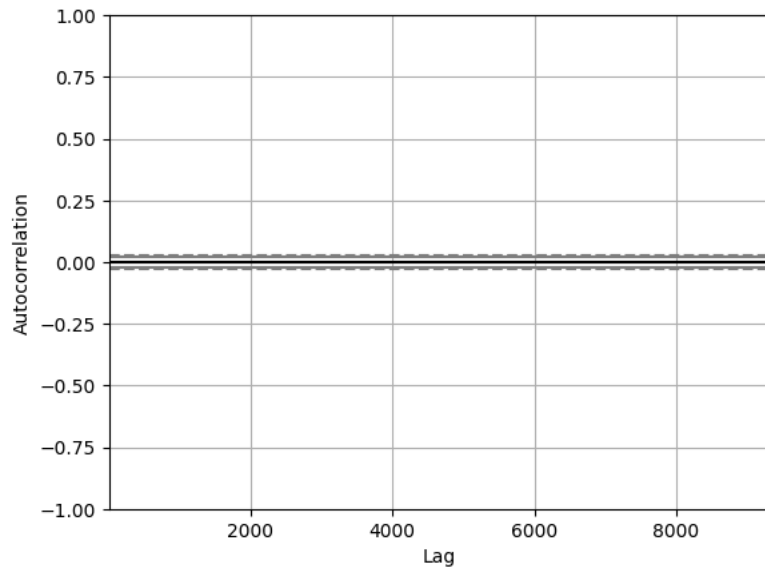
```
1 from pandas.plotting import autocorrelation_plot
2 np.random.seed(0)
3 autocorrelation_plot(pd.Series(np.random.random(size=200)))
```

<Axes: xlabel='Lag', ylabel='Autocorrelation'>



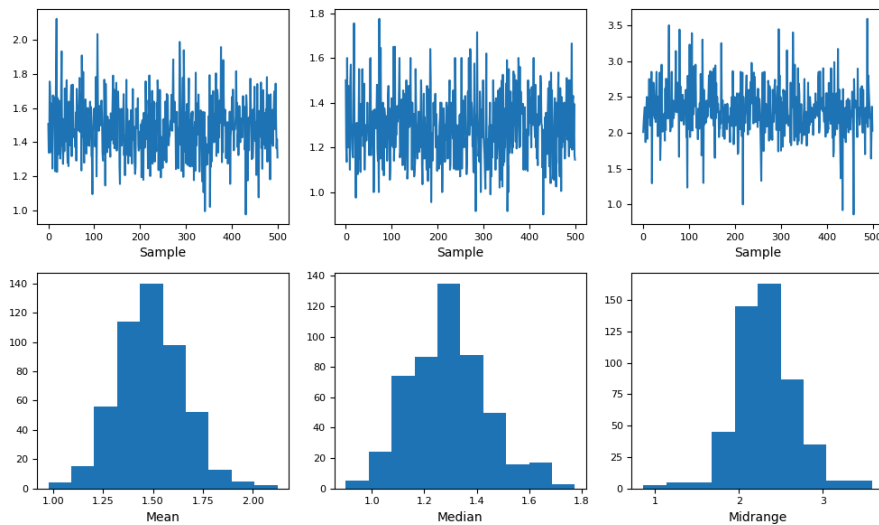
```
1 autocorrelation_plot(earthquakes.mag)
```

<Axes: xlabel='Lag', ylabel='Autocorrelation'>



✓ Bootstrap Plot

```
1 from pandas.plotting import bootstrap_plot
2 fig = bootstrap_plot(earthquakes.mag, fig=plt.figure(figsize=(10, 6)))
```



Data Analysis:

Provide comments on output from the procedures above.

1. 9.1

- From the output shown above, we can see plots using matplotlib for the earthquake.csv file, revealing to us the patterns and trends which would have otherwise taken a long time to identify without the use of these tools. We can use the line plot to see the date when the highest earthquake magnitude happened, the scatter plot to see the magnitude that which a tsunami occurred – which in this case is clearly on above the magnitude of 3 – and the histogram to identify the frequency of the earthquakes' magnitudes.

2. 9.2

- The same values are shown here, but this time we utilized the functions and methods in pandas.

3. 9.3

- From the output shown above, we can immediately infer that the scatter matrix is a very useful tool in viewing the relationship between the magnitude and occurrences of tsunami, which is the same as in the previous plots. We can then see in the autocorrelation plot that the plot looks straight near zero, suggesting that there is periodic behavior in the data.

✓ Supplementary Activity:

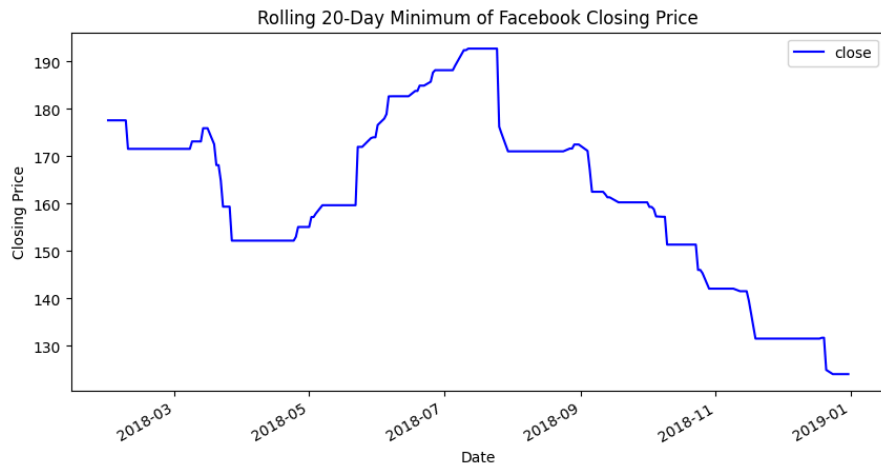
Using the CSV files provided and what we have learned so far in this module complete the following exercises:

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.

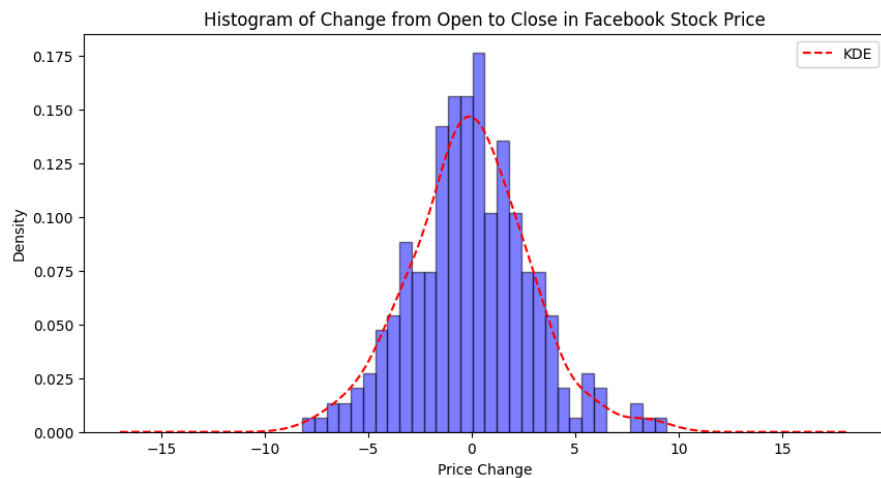
```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3
4 fb = pd.read_csv('/content/data/fb_stock_prices_2018.csv', parse_dates=['date'], index_col='date')
5 earthquakes = pd.read_csv('/content/data/earthquakes.csv', parse_dates=['time'], index_col='time')
```

```
1 #1.
2 roling_20 = fb['close'].rolling(window=20).min()
3 roling_20.plot(kind='line',
4               y='close',
5               figsize=(10, 5),
6               style='b-',
7               legend=True,
8               title='Rolling 20-Day Minimum of Facebook Closing Price')
9 plt.xlabel('Date')
10 plt.ylabel('Closing Price')
```

```
Text(0, 0.5, 'Closing Price')
```



```
1 #2.
2 fb['change'] = fb['open'] - fb['close']
3 plt.figure(figsize=(10, 5))
4 plt.hist(fb['change'],
5         bins=30,
6         alpha=0.5,
7         color='blue',
8         edgecolor='black',
9         density=True)
10 plt.title('Histogram of Change from Open to Close in Facebook Stock Price')
11 plt.xlabel('Price Change')
12 plt.ylabel('Density')
13
14 fb['change'].plot(kind='kde',
15                 color='red',
16                 linestyle='--',
17                 label='KDE')
18 plt.legend()
19 plt.show()
```

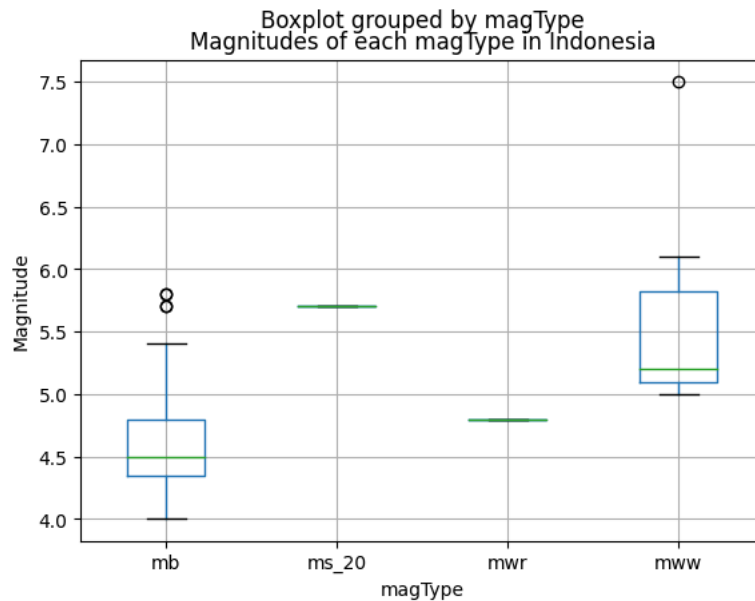


```

1 #3.
2 indonesia_earthquakes = earthquakes[earthquakes['parsed_place'] == 'Indonesia']
3 plt.figure(figsize=(10, 5))
4 indonesia_earthquakes.boxplot(column='mag', by='magType')
5 plt.title('Magnitudes of each magType in Indonesia')
6 plt.xlabel('magType')
7 plt.ylabel('Magnitude')

```

Text(0, 0.5, 'Magnitude')
<Figure size 1000x500 with 0 Axes>

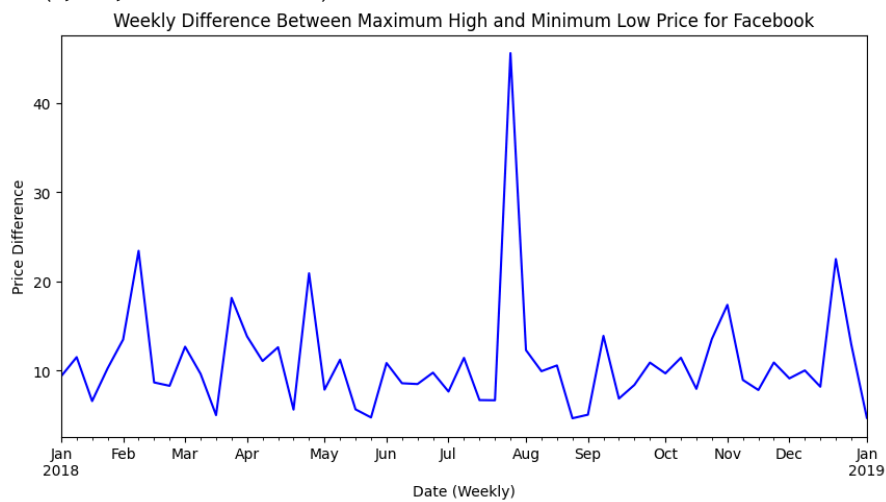


```

1 #4.
2 weekly_max_highPrice = fb['high'].resample('W').max()
3 weekly_min_lowPrice = fb['low'].resample('W').min()
4 difference = weekly_max_highPrice - weekly_min_lowPrice
5
6 plt.figure(figsize=(10, 5))
7 difference.plot(color='blue')
8 plt.title('Weekly Difference Between Maximum High and Minimum Low Price for Facebook')
9 plt.xlabel('Date (Weekly)')
10 plt.ylabel('Price Difference')

```

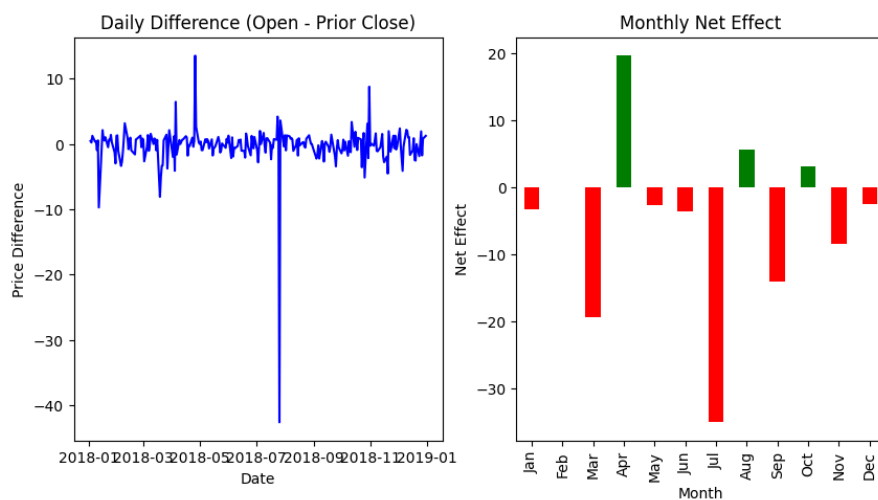
Text(0, 0.5, 'Price Difference')



```

1 #5.
2 dailyPrior_difference = fb['open'] - fb['close'].shift(1)
3 difference_netEffect = dailyPrior_difference.resample('M').sum()
4 fig, axes = plt.subplots(1, 2, figsize=(10, 5))
5
6 def daily_linePlot():
7     axes[0].plot(dailyPrior_difference.index, dailyPrior_difference, color='blue')
8     axes[0].set_title('Daily Difference (Open - Prior Close)')
9     axes[0].set_xlabel('Date')
10    axes[0].set_ylabel('Price Difference')
11
12 def netEffect_barPlot():
13     colors = ['green' if val >= 0 else 'red' for val in difference_netEffect]
14     difference_netEffect.plot(kind='bar', ax=axes[1], color=colors)
15     axes[1].set_title('Monthly Net Effect')
16     axes[1].set_xlabel('Month')
17     axes[1].set_ylabel('Net Effect')
18     axes[1].set_xticklabels(difference_netEffect.index.strftime('%b'))
19
20 x = daily_linePlot()
21 y = netEffect_barPlot()

```



✓ Summary/Conclusion:

Provide a summary of your learnings and the conclusion for this activity.

- Through this activity, I have learned a lot in the field of visualizing data using the matplotlib and pandas libraries. I have learned how to create a diversity of different plots like line plots for analyzing trends, bar plots for comparing them, box plots for understanding how data are distributed, and KDE for visualizing the density of data. The activities in the procedure and supplementary not only enhanced my