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

---

**Name :** Jann Moises Nyll B. De los Reyes

**Section :** CPE22S3

**Submitted to:** Engr. Roman Richard

**Date :** March 28, 2024

---

Instructions:

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

**Resources :**

- Download the following datasets: 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:**

Provide comments on output from the procedures above.

Data visualization is a powerful tool that transforms raw data into a visual context, such as a map or graph, making the data easier to understand and interpret. It allows us to spot trends, patterns, and outliers in groups of data. Using libraries like Matplotlib, a popular plotting library for Python, enhances this process by providing a range of plotting functions that are both versatile and user-friendly. From line graphs that can reveal trends over time, to histograms that show frequency distributions, Matplotlib helps in crafting insightful visual narratives. Scatter plots can highlight correlations between variables, and with the right bin sizing, histograms can effectively summarize large datasets in an intuitive format.

Learning to manipulate plot components like figures, axes, and subplots is crucial, as these are the building blocks of any data visualization in Matplotlib. By mastering these elements, one can create complex layouts with multiple subplots, customize figure sizes, and fine-tune the appearance using rcParams to make the visualizations more informative and appealing. Indeed, the ability to represent data graphically using Matplotlib is an invaluable skill for anyone looking to analyze data and communicate their findings effectively.

**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.

**Summary/Conclusion:**

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

# Setup

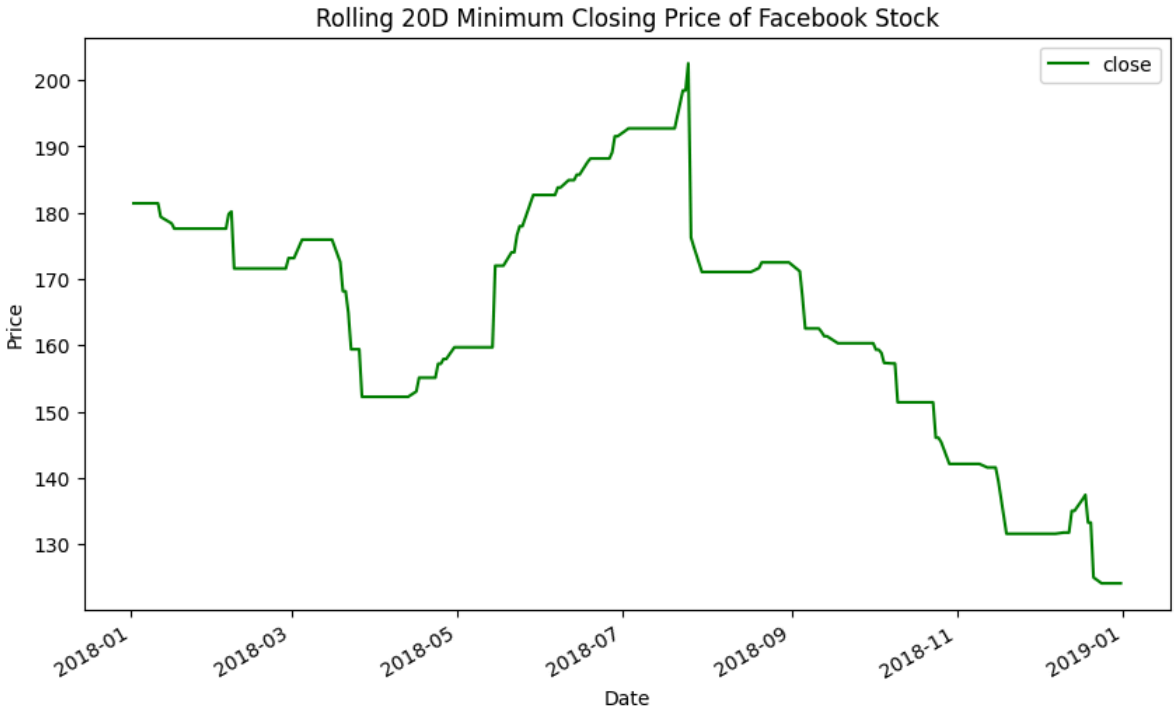
In [ ]:
```
%matplotlib inline
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd

fb = pd.read_csv('/content/drive/MyDrive/Module 9: Data Visualization using Pandas, Matplotlib and Seaborn/data hoa9.1/fb_stoc
                 index_col = 'date', parse_dates =True)

quakes = pd.read_csv('/content/drive/MyDrive/Module 9: Data Visualization using Pandas, Matplotlib and Seaborn/data hoa9.1/ear
```

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

First, we need to calculate the 20 day rolling minimum:

In [ ]:
```
fb.close.rolling('20D').min().plot(
    style = 'g',
    title='Rolling 20D Minimum Closing Price of Facebook Stock',
    figsize = (10,6),
    )
plt.xlabel('Date')
plt.ylabel('Price')
plt.legend()
```

Out[ ]:
```
<matplotlib.legend.Legend at 0x7a80fb0a5f60>
```

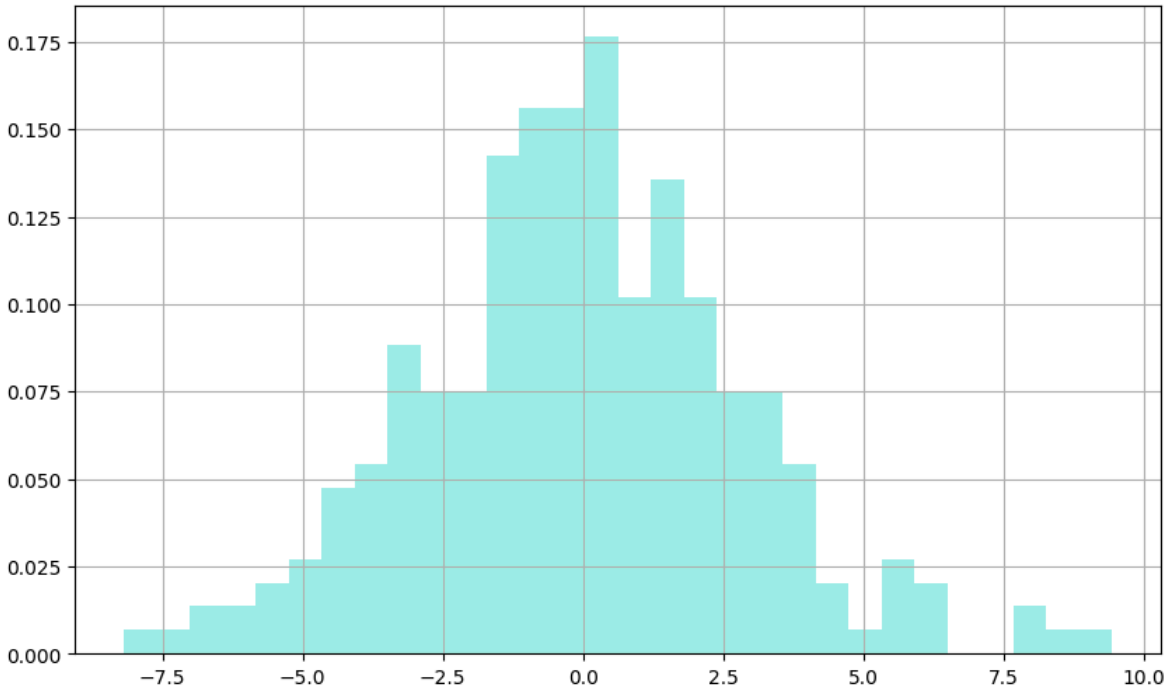



2. Create a histogram and KDE of the change from open to close in the price of Facebook stock.

In [ ]:
```
# to calculate  open to close
fb['overnight_change'] = fb['open'] - fb['close']
```

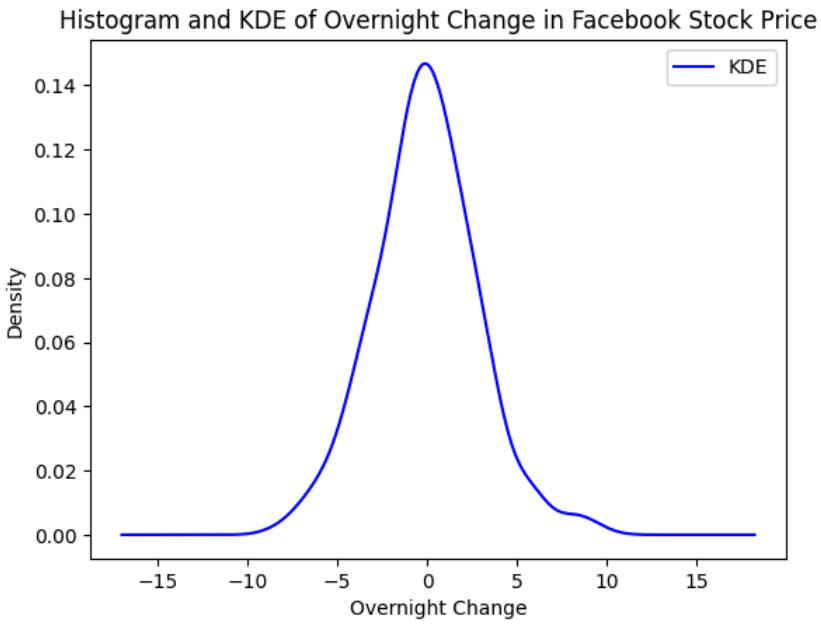
In [ ]:
```
 # plot the histogram

plt.figure(figsize=(10,6))
fb['overnight_change'].hist(bins=30,alpha=0.5,color='turquoise',density =True, label = 'histogram')
```

Out[ ]: <Axes: >




In [ ]:
```
# plotting KDE
fb['overnight_change'].plot(
    kind='kde', color = 'blue', label = 'KDE')

plt.xlabel('Overnight Change')
plt.ylabel('Density')
plt.title('Histogram and KDE of Overnight Change in Facebook Stock Price')
plt.legend()
```

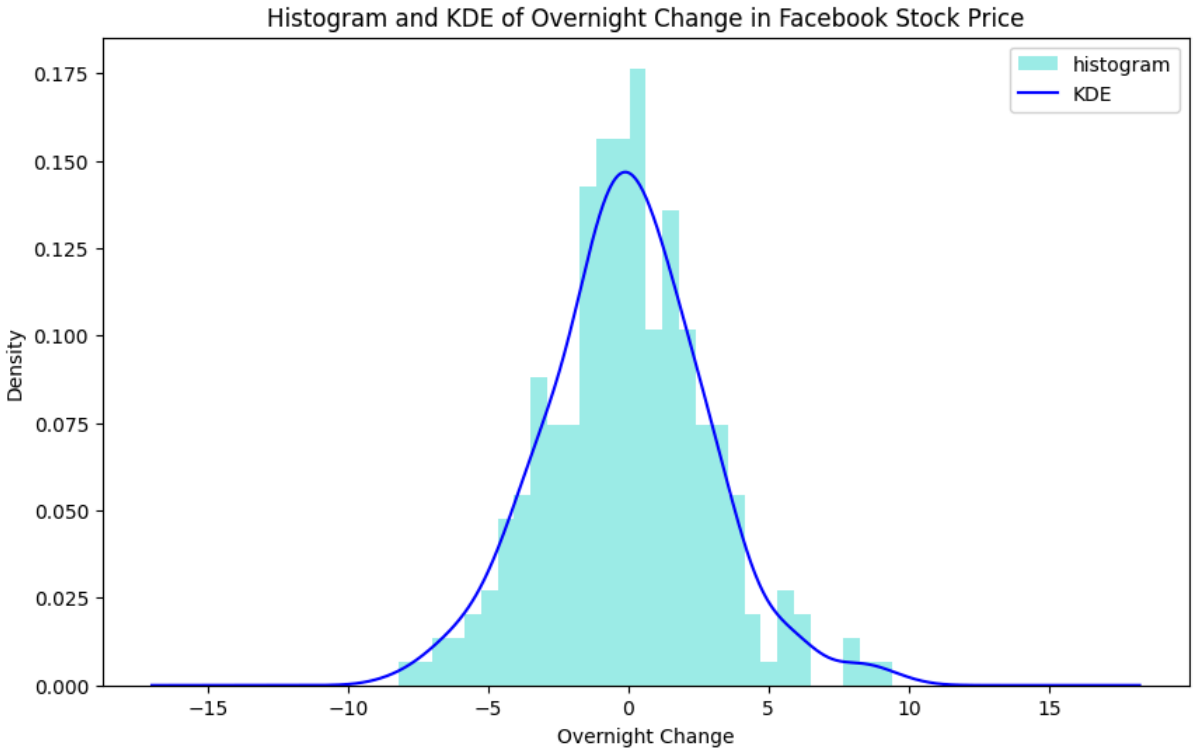
Out[ ]: <matplotlib.legend.Legend at 0x7a80cbc836d0>




In [ ]:
```
# join together
plt.figure(figsize=(10,6))
fb['overnight_change'].hist(bins=30,alpha=0.5,color='turquoise',density =True, label = 'histogram')

# plotting KDE
fb['overnight_change'].plot(
    kind='kde', color = 'blue', label = 'KDE')

plt.xlabel('Overnight Change')
```

```
plt.ylabel('Density')
plt.title('Histogram and KDE of Overnight Change in Facebook Stock Price')
plt.legend()
```

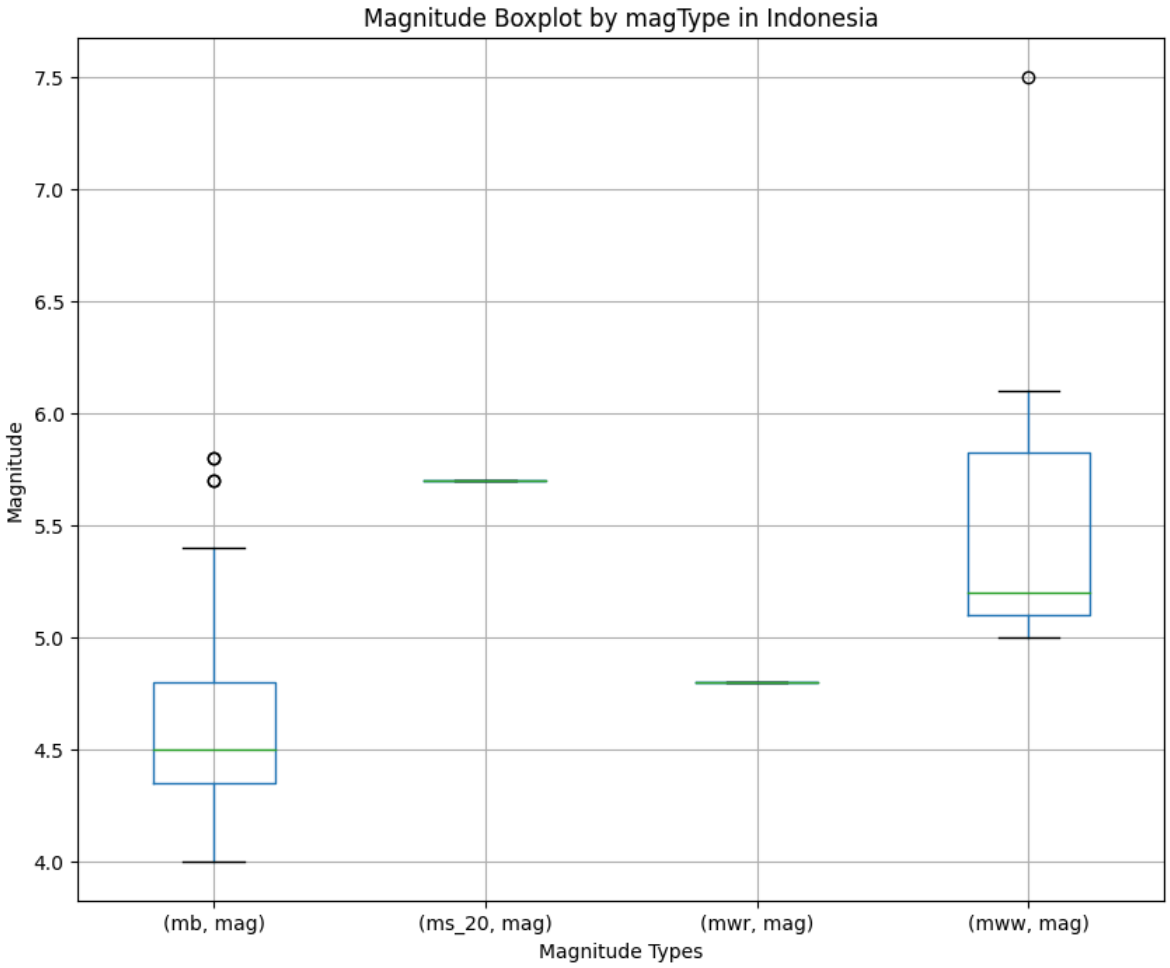
Out[ ]: <matplotlib.legend.Legend at 0x7a80cbc83640>




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

In [ ]:
```
indo_quakes = quakes.query('parsed_place == "Indonesia"')

indo_quakes[['mag']].groupby(indo_quakes['magType']).boxplot(
    figsize =(10,8), subplots=False
)

plt.title("Magnitude Boxplot by magType in Indonesia")
plt.xlabel('Magnitude Types')
plt.ylabel('Magnitude')
```

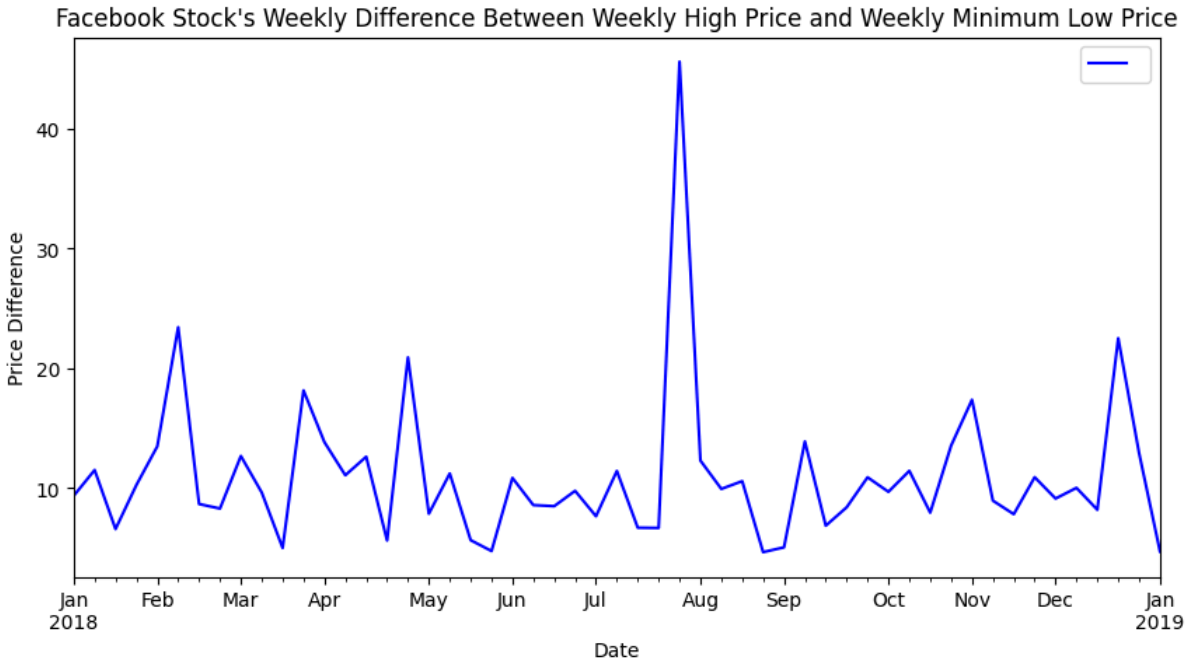
Out[ ]: Text(0, 0.5, 'Magnitude')




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.

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

weekly_price_change.plot(
    kind = 'line',
    figsize = (10,5),
    style = 'b-',
    legend = True,
    title = 'Facebook Stock\'s Weekly Difference Between Weekly High Price and Weekly Minimum Low Price'
)
plt.xlabel('Date')
plt.ylabel('Price Difference')
```

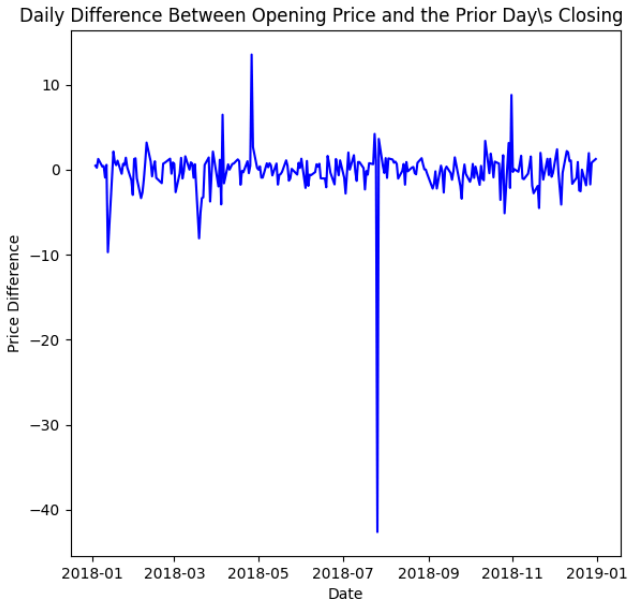
Out[ ]:
```
Text(0, 0.5, 'Price Difference')
```

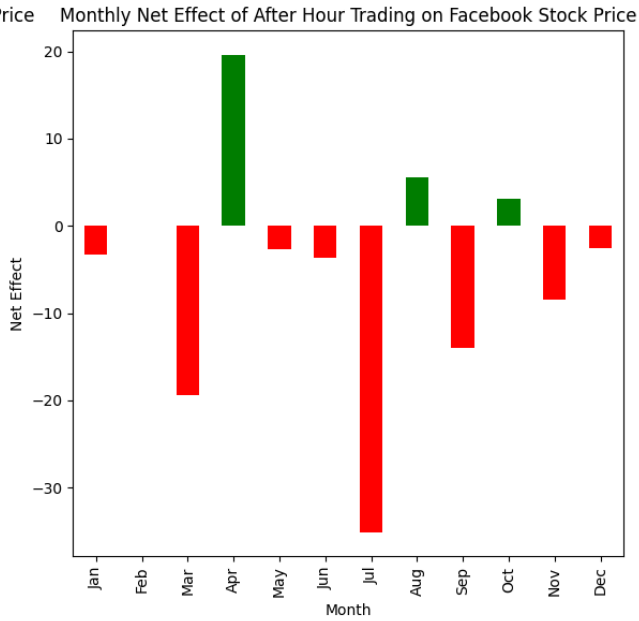



Using matplotlib and pandas, create two subplots side-by-side showing the effect that after-hours trading has had on Facebook's stock price:

- 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.

In [ ]:
```
# calculate the daily difference
daily_difference = fb['open'] - fb['close'].shift(1)

#subplots
fig,axes = plt.subplots(1,2, figsize = (12,6))

#plot subplot 1 (Daily Difference)
axes[0].plot(daily_difference,color ='blue')
axes[0].set_title('Daily Difference Between Opening Price and the Prior Day\s Closing Price')
axes[0].set_xlabel('Date')
axes[0].set_ylabel('Price Difference')

#calculate monthly net effect using resample
monthly_net_effect = daily_difference.resample('M').sum()

#plotting subplot 2 (Monthly Net Effect)
colors = ['green' if value >= 0 else 'red' for value in monthly_net_effect]
monthly_net_effect.plot(
    kind = 'bar',
    ax = axes[1],
    color = colors
)
axes[1].set_title('Monthly Net Effect of After Hour Trading on Facebook Stock Price')
axes[1].set_xlabel('Month')
axes[1].set_ylabel('Net Effect')
axes[1].set_xticklabels(monthly_net_effect.index.strftime('%b'))

plt.tight_layout()
```




In [ ]:
```
monthly_net_effect
```

Out[ ]:
```
date
2018-01-31    -3.3500
2018-02-28     0.0200
2018-03-31   -19.3700
2018-04-30    19.6247
2018-05-31    -2.6488
2018-06-30    -3.6246
2018-07-31   -35.0750
2018-08-31     5.5992
2018-09-30   -14.0150
2018-10-31     3.0950
2018-11-30    -8.5100
2018-12-31    -2.5200
Freq: M, dtype: float64
```

Sumary/Conclusion:

Learning data visualization using matplotlib is a powerful skill that allows us to interpret and analyze data effectively. In this activity, various techniques are employed to visualize stocks and earthquake data. For instance, using pandas to plot the rolling 20-day minimum of Facebook's closing price can reveal trends and patterns over time. Creating histograms and Kernel Density Estimates (KDE) can help understand the distribution and density of stock price changes. Box plots for earthquake magnitudes in Indonesia can provide insights into the range and outliers of the data. Line plots of the weekly price range of Facebook stock offer a clear view of volatility over time. Lastly, side-by-side subplots using matplotlib and pandas can illustrate the impact of after-hours trading on stock prices. These methods are essential for making informed decisions in finance and other fields that rely on data-driven strategies.

In [ ]: