Data Analysis

9.1 Introduction to Matplotlib

I learned that Matplotlib is a powerful library for creating a variety of plots in Python. It helps visualize data through different types of charts, like line plots, bar plots, scatter plots, histograms, and box plots. I can customize these plots by adding titles, axis labels, and grid lines to make them clearer and more informative. Matplotlib really seems useful for tracking trends or comparing data, and I now understand how to make the most of it in my projects.

9.2 Plotting with Pandas

What's great about Pandas is that it makes plotting super simple by allowing me to plot data directly from DataFrames. With just a single command (df.plot()), I can create line plots, bar plots, and more. I also discovered that Pandas handles time-based data well, so I can easily plot trends and even resample data to different time frequencies, like weekly or monthly, which really helps when I need to see patterns over time.

9.3 Pandas Plotting Subpackage

I learned that Pandas has its own plotting functionality through the df.plot() function, which simplifies data visualization. It's pretty straightforward, and I can make different plot types by just specifying the kind argument. Plus, I can easily customize the plots with titles, labels, and legends. The best part is being able to work with time series data, where I can resample the data to different intervals (like weekly or monthly), which is perfect for tracking changes over time.

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

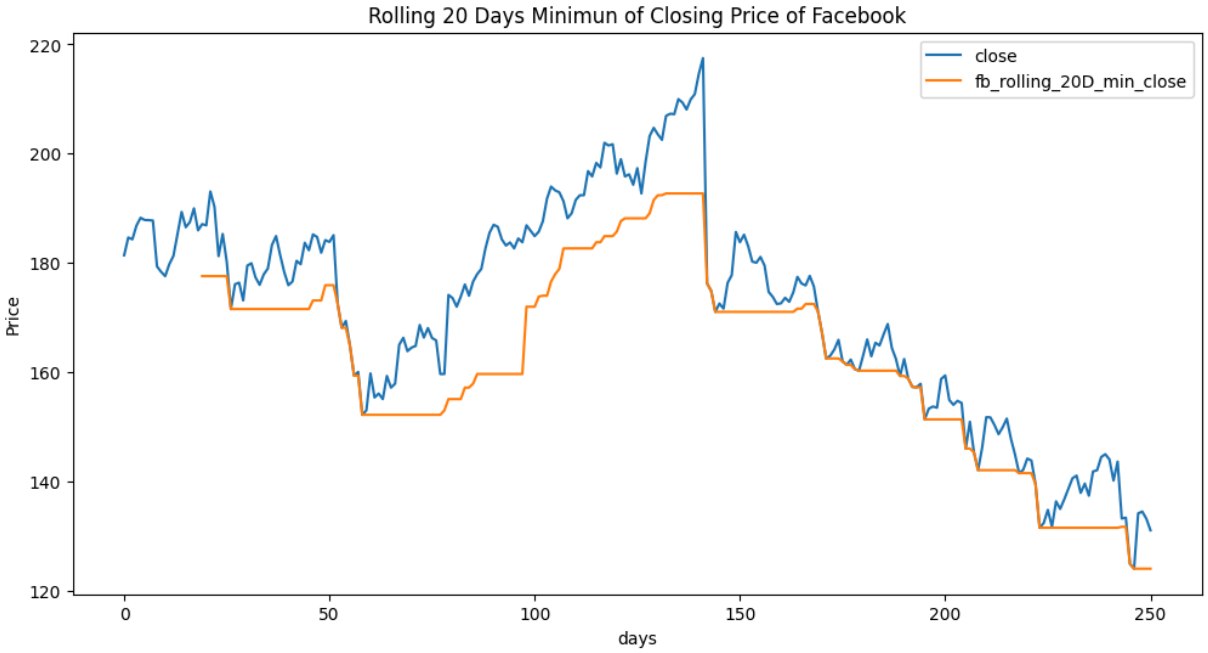
line.

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.

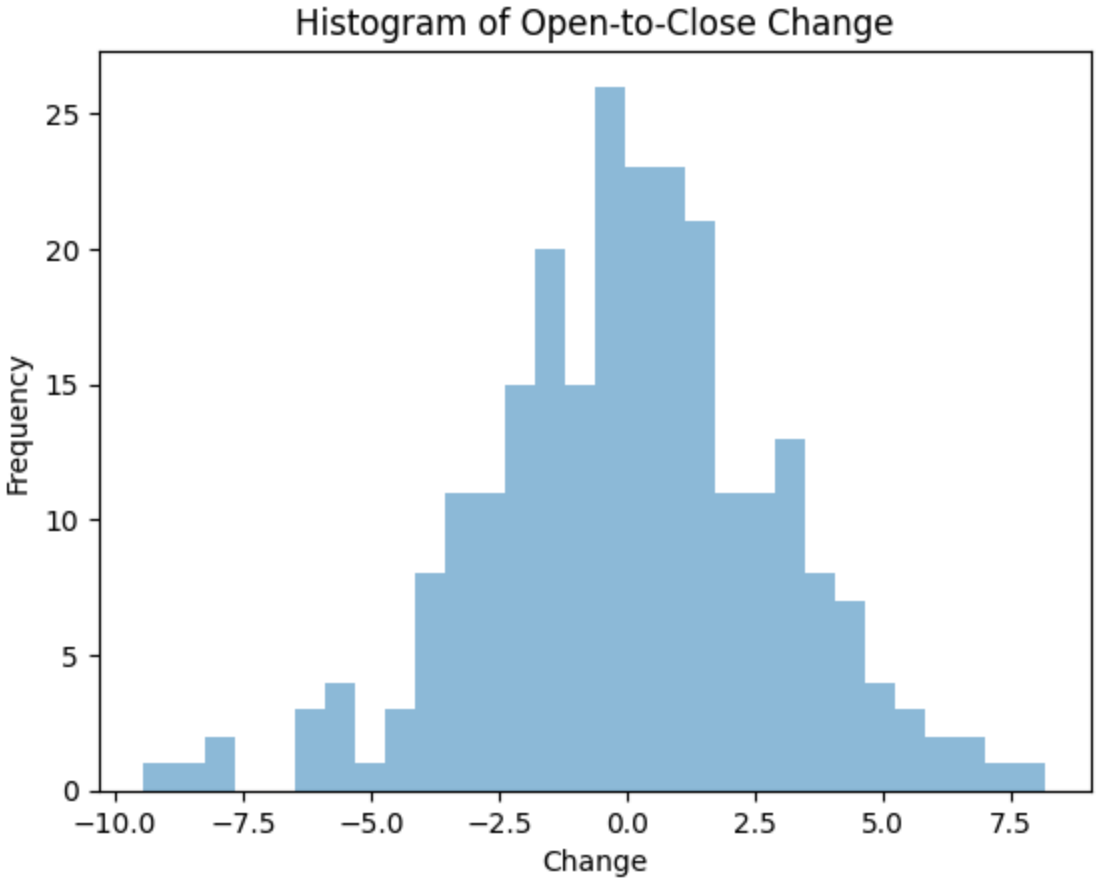
In [29]:
```
 # I did this so that I will now type plt.show() anymore
%matplotlib inline
import pandas as pd
import matplotlib.pyplot as plt

fb = pd.read_csv('data/fb_stock_prices_2018.csv')

#Plot the rolling 20-day minimum of the Facebook closing price with the pandas plot

fb['fb_rolling_20D_min_close'] = fb['close'].rolling(20).min()

fb[['close','fb_rolling_20D_min_close']].plot(figsize=(12,6), title = "Rolling 20 D
plt.xlabel('days')
plt.ylabel('Price')
```

Out[29]:
```
Text(0, 0.5, 'Price')
```



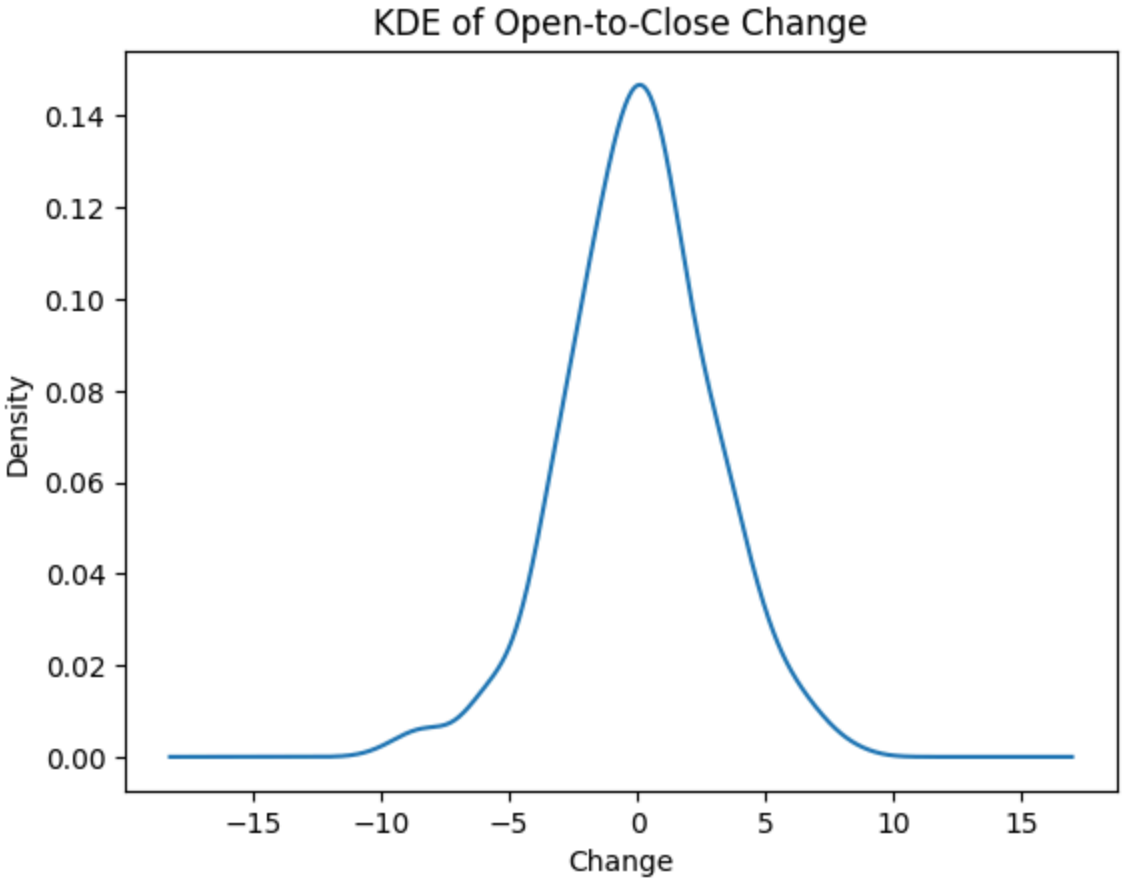
In [30]:
```
#Create a histogram and KDE of the change from open to close in the price of Facebo

#Creating the new column formula and showing the histogram
fb['Open_to_Close_Change'] = fb['close'] - fb['open']

fb['Open_to_Close_Change'].plot(kind='hist', bins=30, alpha=0.5, title="Histogram o
plt.xlabel("Change")
```

Out[30]:
```
Text(0.5, 0, 'Change')
```



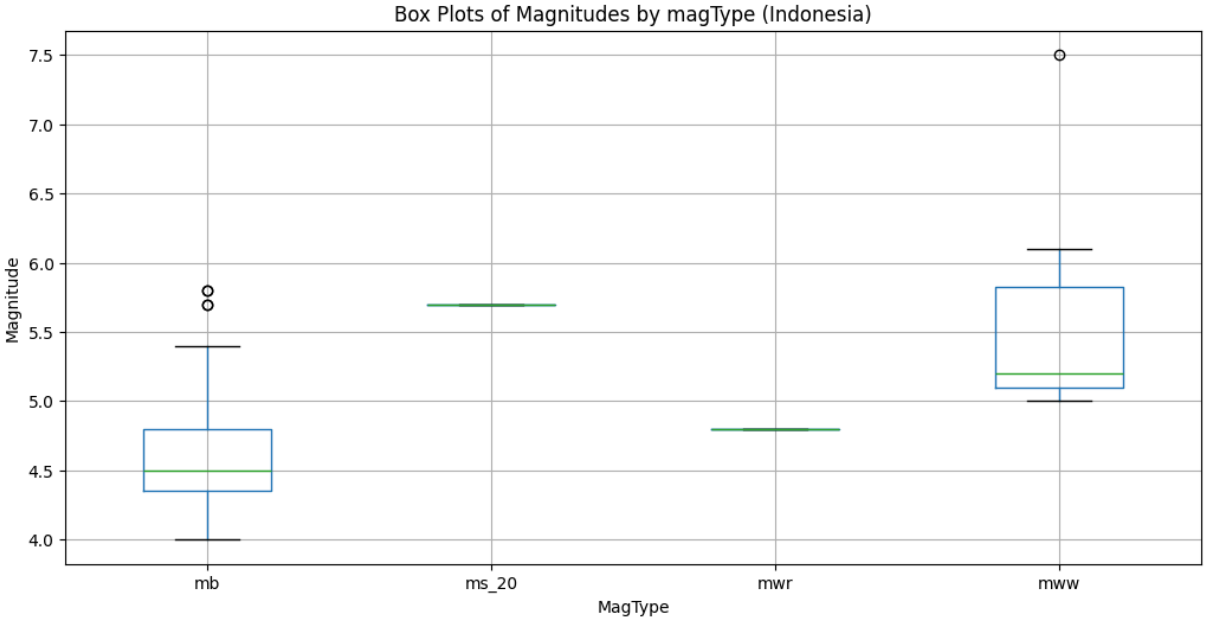
In [31]:
```
#this is the KDE
fb['Open_to_Close_Change'].plot(kind='kde', title="KDE of Open-to-Close Change")
plt.xlabel("Change")
```

Out[31]:
```
Text(0.5, 0, 'Change')
```



In [32]:
```
# Using the earthquake data, create box plots for the magnitudes of each magType us

# First Load earthquake data

eq = pd.read_csv('data/earthquakes.csv')

eqfilter = eq.query('parsed_place == "Indonesia"') #I made a filter that specifical

# Create box plots for each magType
eqfilter.boxplot(column='mag', by='magType', figsize=(12, 6))
plt.title("Box Plots of Magnitudes by magType (Indonesia)")
plt.suptitle("")
plt.xlabel("MagType")
plt.ylabel("Magnitude")
```

Out[32]:
```
Text(0, 0.5, 'Magnitude')
```



In [33]:
```
#let's check if there's missing values to ensure that data is accurate

eqfilter.isna().any() #it turned out that there are no missing values
```

Out[33]:

	0
mag	False
magType	False
time	False
place	False
tsunami	False
parsed_place	False

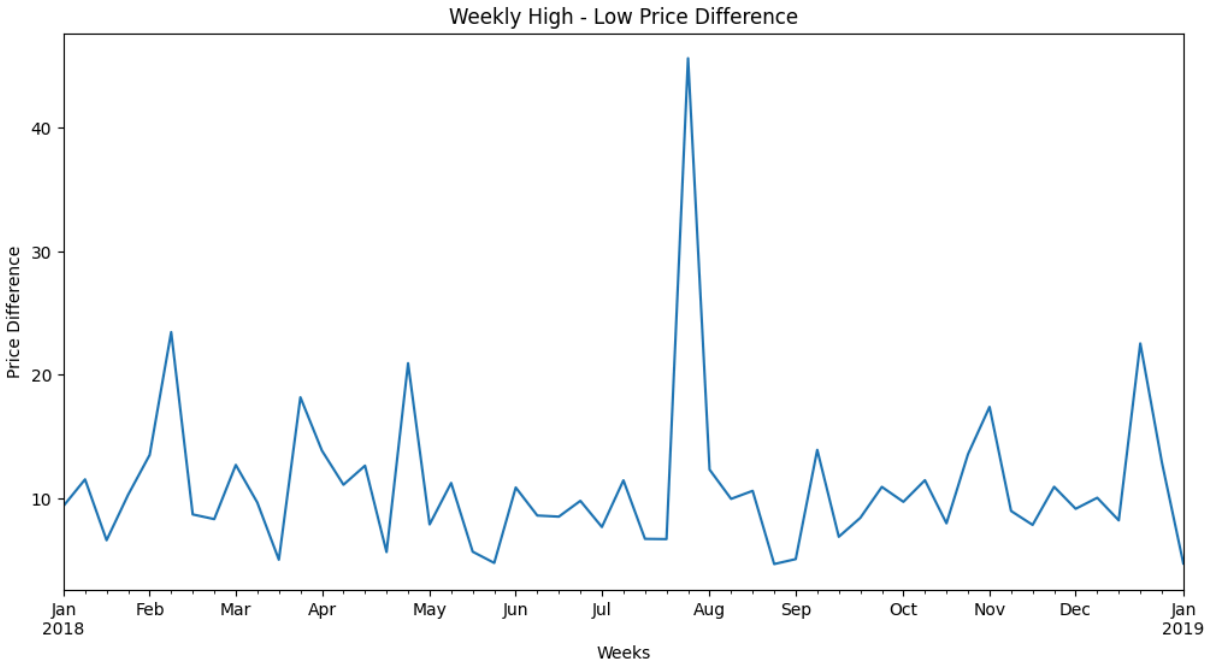
dtype: bool

In [34]:
```
# Make a line plot of the difference between the weekly maximum high price and the

fb.info() #let's first correct the dtype for fb
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 251 entries, 0 to 250
Data columns (total 8 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   date                     251 non-null    object
 1   open                     251 non-null    float64
 2   high                     251 non-null    float64
 3   low                      251 non-null    float64
 4   close                    251 non-null    float64
 5   volume                   251 non-null    int64
 6   fb_rolling_20D_min_close 232 non-null    float64
 7   Open_to_Close_Change     251 non-null    float64
dtypes: float64(6), int64(1), object(1)
memory usage: 15.8+ KB
```

In [35]:
```
fb['date'] = pd.to_datetime(fb['date']) # to make sure that 'date' in datetime

# Set the datetime column as the index
fb.set_index('date',inplace = True)

# Resample to weekly and calculate the difference
weekly_high_low_diff = fb.resample('W').agg({'high': 'max', 'low': 'min'})
weekly_high_low_diff['High_Low_Diff'] = weekly_high_low_diff['high'] - weekly_high_

# Plot the difference
weekly_high_low_diff['High_Low_Diff'].plot(figsize=(12, 6), title="Weekly High - Lo
plt.xlabel("Weeks")
plt.ylabel("Price Difference")
```

Out[35]:
```
Text(0, 0.5, 'Price Difference')
```

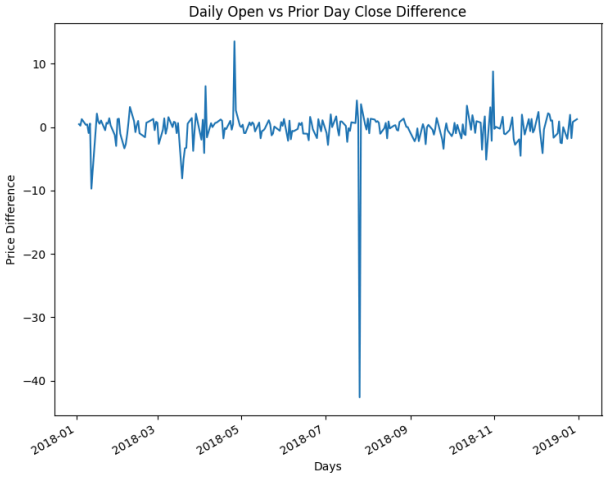


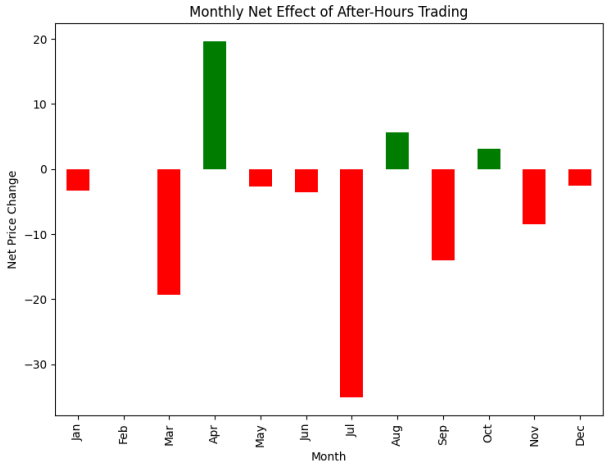
5. Using matplotlib and pandas, create two subplots side-by-side showing the effect that after-hours trading has had on Facebook's stock price:

- 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.

In [37]:
```
# Reset the index
fb.reset_index().set_index('date', inplace=True)


# Calculate the daily opening price difference compared to the previous day's close
fb['Prior_Day_Close'] = fb['close'].shift(1)
fb['Daily_Open_Close_Diff'] = fb['open'] - fb['Prior_Day_Close']
```

In [46]:
```
# The first subplot which plots the daily difference between opening and prior day'
fig, axes = plt.subplots(1, 2, figsize=(15, 6))

fb['Daily_Open_Close_Diff'].plot(ax=axes[0], title="Daily Open vs Prior Day Close D
axes[0].set_xlabel("Days")
axes[0].set_ylabel("Price Difference")

# The second subplot will be a bar plot showing the net effect this had monthly, us

monthly_effect = fb['Daily_Open_Close_Diff'].resample('ME').sum()

#color according to their stock price

colors = ['green' if val > 0 else 'red' for val in monthly_effect]

monthly_effect.plot(kind='bar', ax=axes[1], color=colors, title="Monthly Net Effect

axes[1].set_xlabel("Month")
axes[1].set_ylabel("Net Price Change")

# Modify x-axis to show month abbreviations
axes[1].set_xticklabels([pd.Timestamp(idx).strftime('%b') for idx in monthly_effect

plt.tight_layout()
```


Daily Open vs Prior Day Close Difference
Price Difference
10
0
−10
−20
−30
−40
2018-01
2018-03
2018-05
2018-07
2018-09
2018-11
2019-01
Days


Monthly Net Effect of After-Hours Trading
Net Price Change
20
10
0
−10
−20
−30
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Month