

Submitted by: Angelo Luis C. Cu

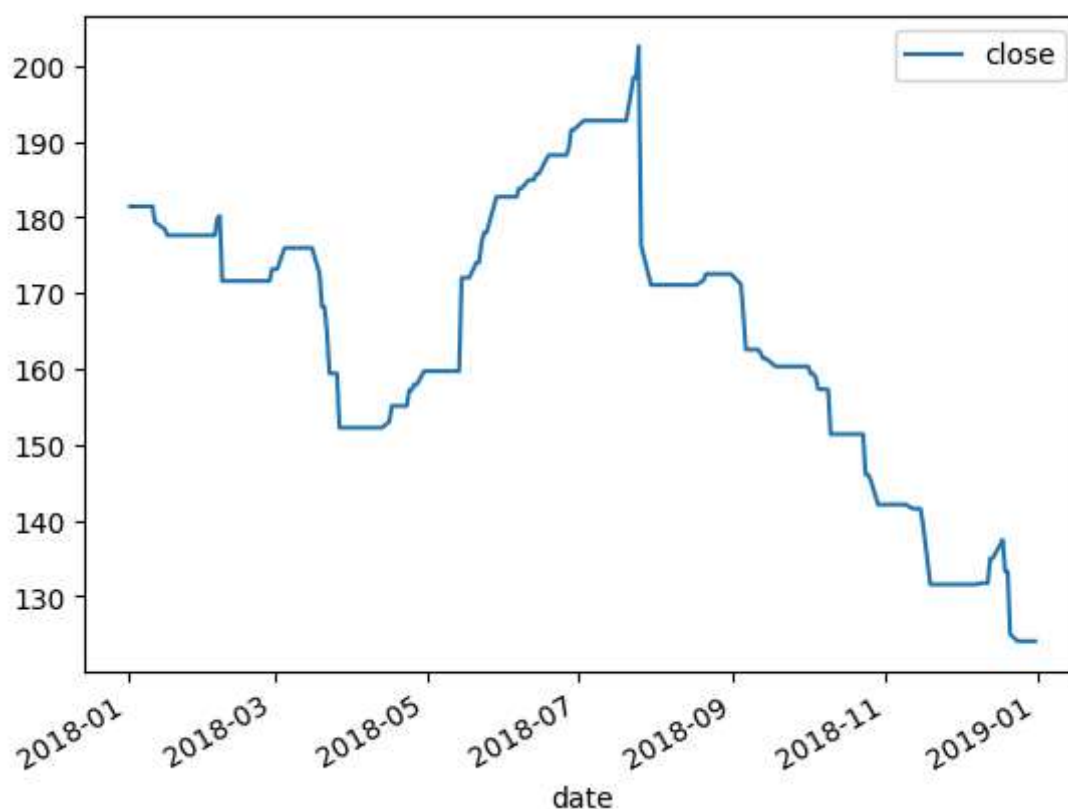
```
# basic setup
%matplotlib inline
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd

fb = pd.read_csv(
    'data/fb_stock_prices_2018.csv', index_col='date', parse_dates=True
)
quakes = pd.read_csv('data/earthquakes-1.csv')
```

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

```
fb.rolling('20D').agg({'close':'min'}).plot()
```

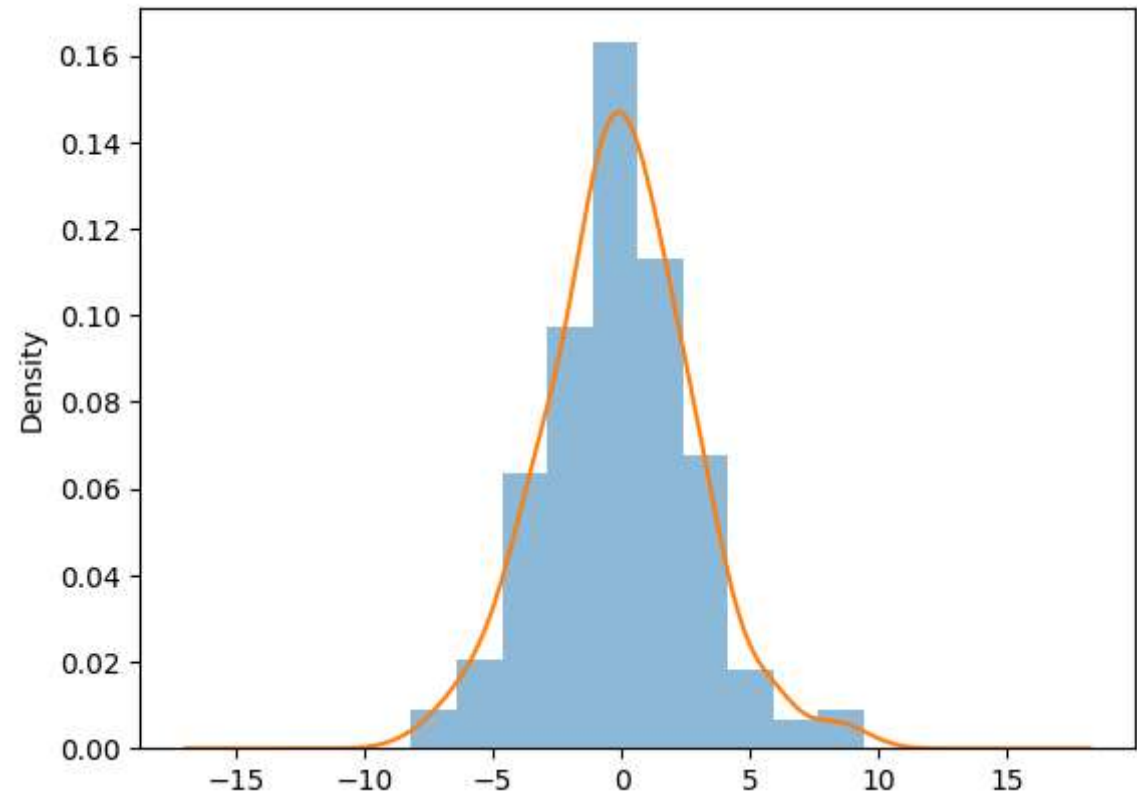
<Axes: xlabel='date'>



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

```
fb_change = fb.assign(
    change = lambda x : x.open-x.close
)
ax = fb_change.change.plot(kind = 'hist', density = True, alpha = 0.5)
fb_change.change.plot(ax = ax, kind = 'kde')
```

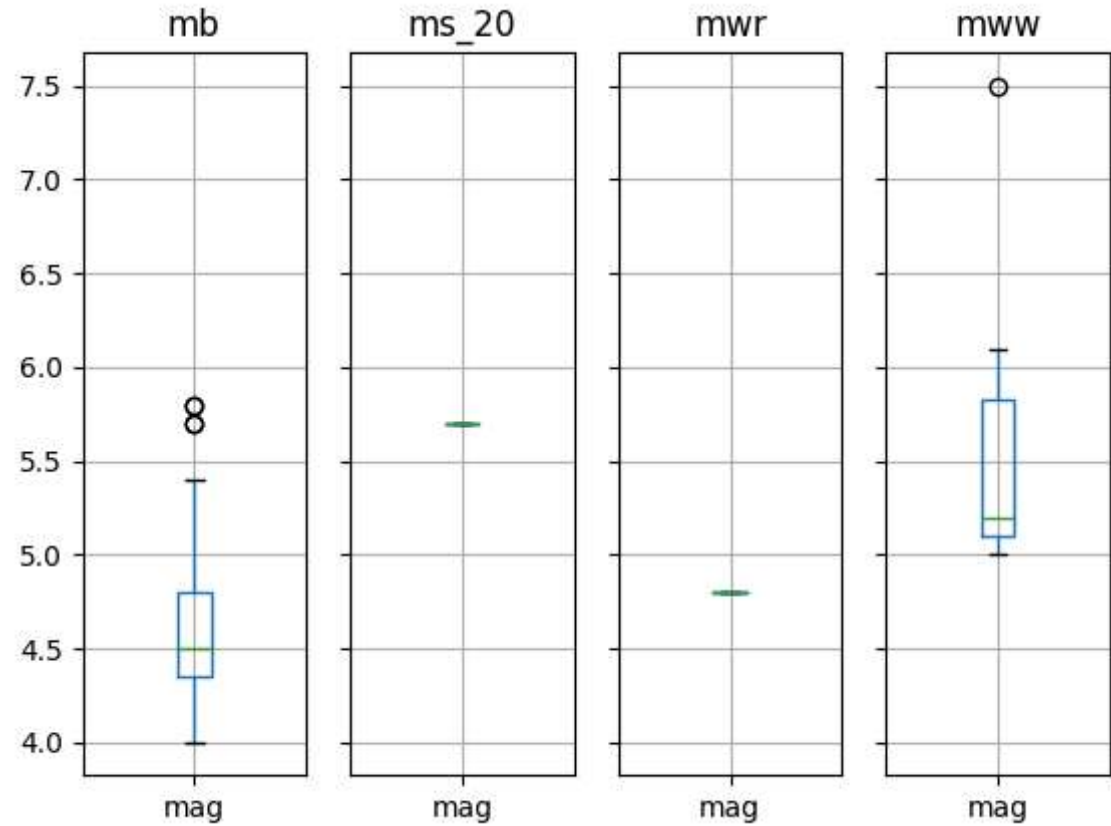
<Axes: ylabel='Density'>



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

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

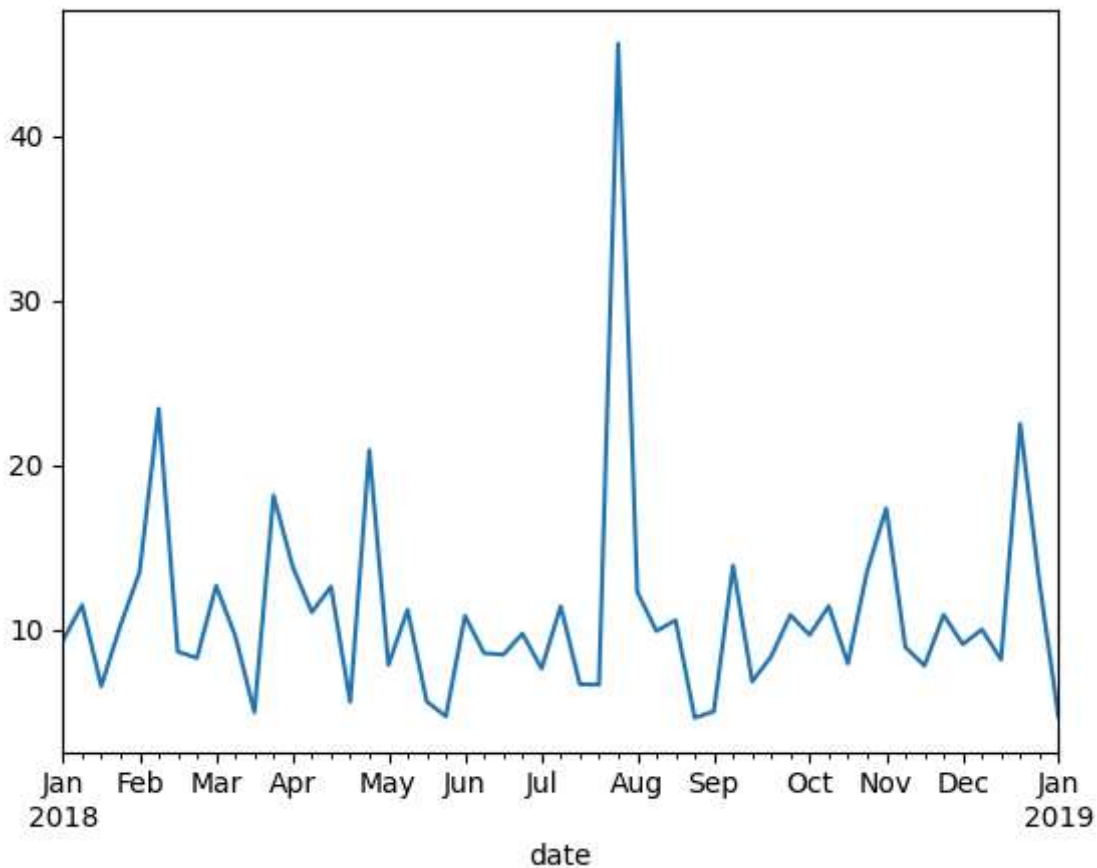
mb Axes(0.1,0.15;0.173913x0.75)  
ms\_20 Axes(0.308696,0.15;0.173913x0.75)  
mwr Axes(0.517391,0.15;0.173913x0.75)  
mww Axes(0.726087,0.15;0.173913x0.75)  
dtype: object



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.

```
weekly_high = fb.high.resample('W').max()
weekly_low = fb.low.resample('W').min()
diff = weekly_high-weekly_low
diff.plot()
```

<Axes: xlabel='date'>



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.

```

fb = fb.assign( # code is from time series notebook
    prior_close=lambda x: x.close.shift(),
    after_hours_change_in_price=lambda x: x.open - x.prior_close,
    abs_change=lambda x: x.after_hours_change_in_price.abs()
)

fig = plt.figure(figsize = (14,7))
gs = fig.add_gridspec(1, 2)
left = fig.add_subplot(gs[0,0])
right = fig.add_subplot(gs[0,1])

left.plot(fb.index, fb.after_hours_change_in_price)
monthly_net_effect = fb.after_hours_change_in_price.resample('M').sum()

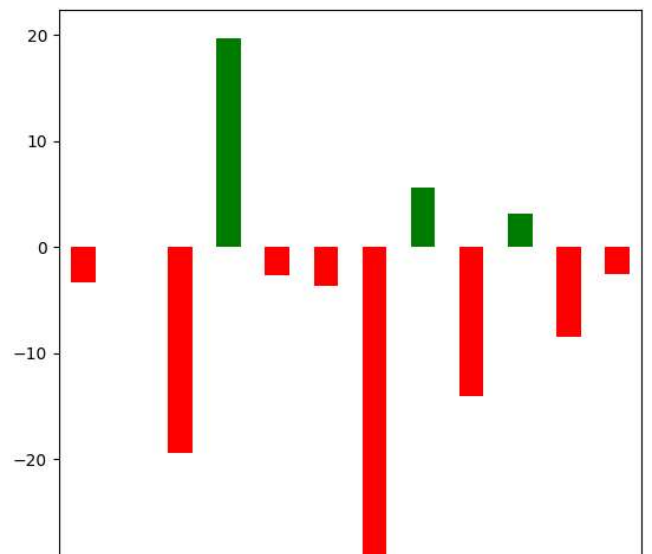
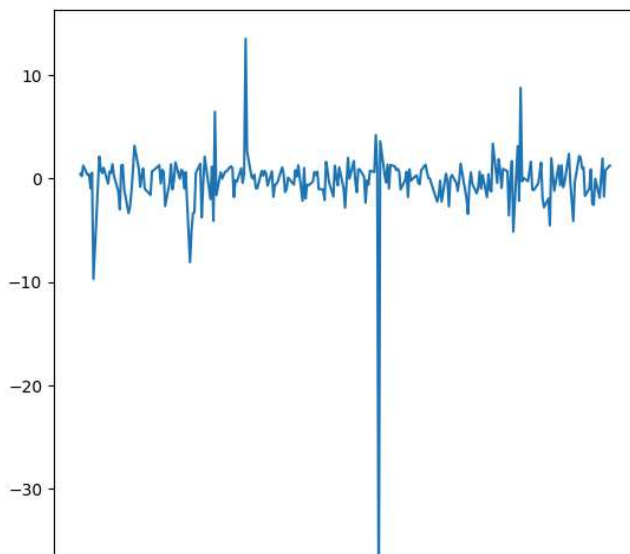
# bonus 1
colors = ['r' if value < 0 else 'g' for value in monthly_net_effect]

monthly_net_effect.plot(kind = 'bar', ax = right, color=colors)

# bonus 2
right.set_xticklabels(monthly_net_effect.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')]

```



Summary:

I have learned how to plot and graph dataframes using matplotlib and pandas. I also learned the different types of graphs available and their different parameters.