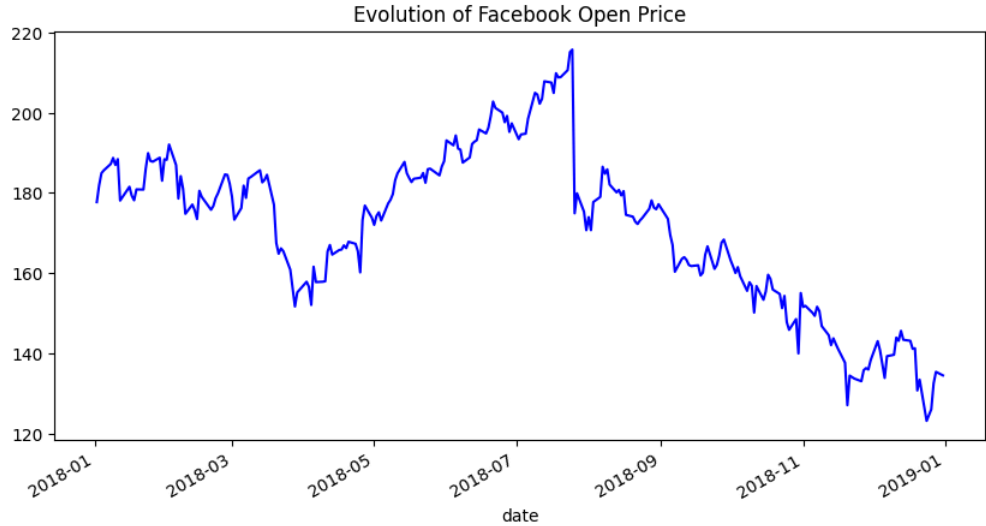


```
%matplotlib inline
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd

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

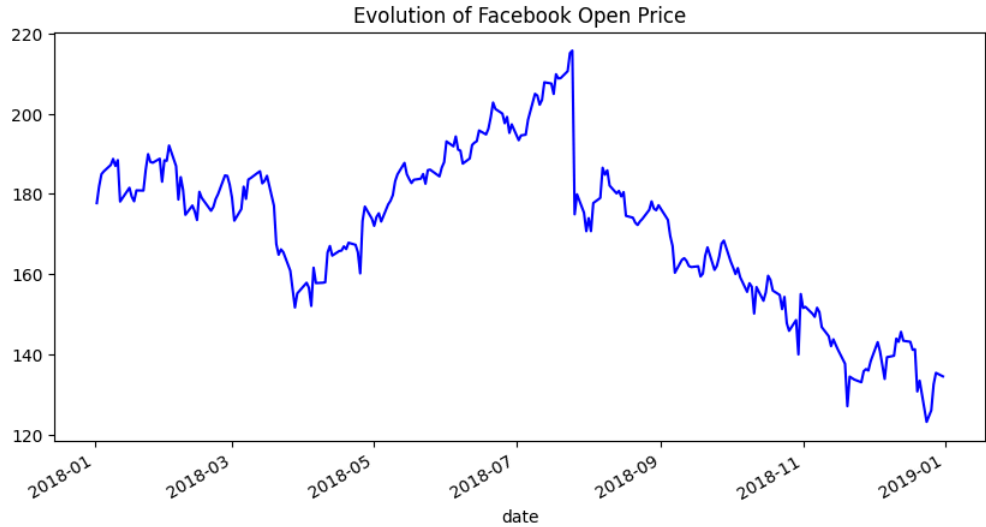
```
fb.plot( # using pandas to plot
    kind='line', # line graph
    y='open', # sets open at y axis
    # by default, x axis is the index which is date
    figsize=(10, 5), # 10x5 inches
    style='b-', # styles available are at matplotlib.style.available
    legend=False, # no legend, default is True
    title='Evolution of Facebook Open Price'
)
```

<Axes: title={'center': 'Evolution of Facebook Open Price'}, xlabel='date'>



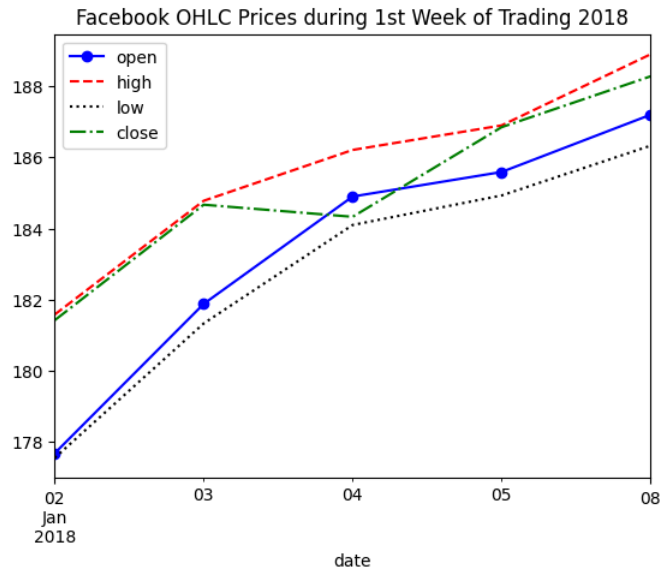
```
fb.plot(
    kind='line',
    y='open',
    figsize=(10, 5),
    color='blue', # uses color param instead of style
    linestyle='solid', # linestyle instead of style
    legend=False,
    title='Evolution of Facebook Open Price'
) # results in a similar graph as above
```

<Axes: title={'center': 'Evolution of Facebook Open Price'}, xlabel='date'>



```
fb.iloc[:5,].plot(
    y=['open', 'high', 'low', 'close'], # many lines are possible
    style=['b-o', 'r--', 'k:', 'g-.'], # different styles per line
    title='Facebook OHLC Prices during 1st Week of Trading 2018'
)
```

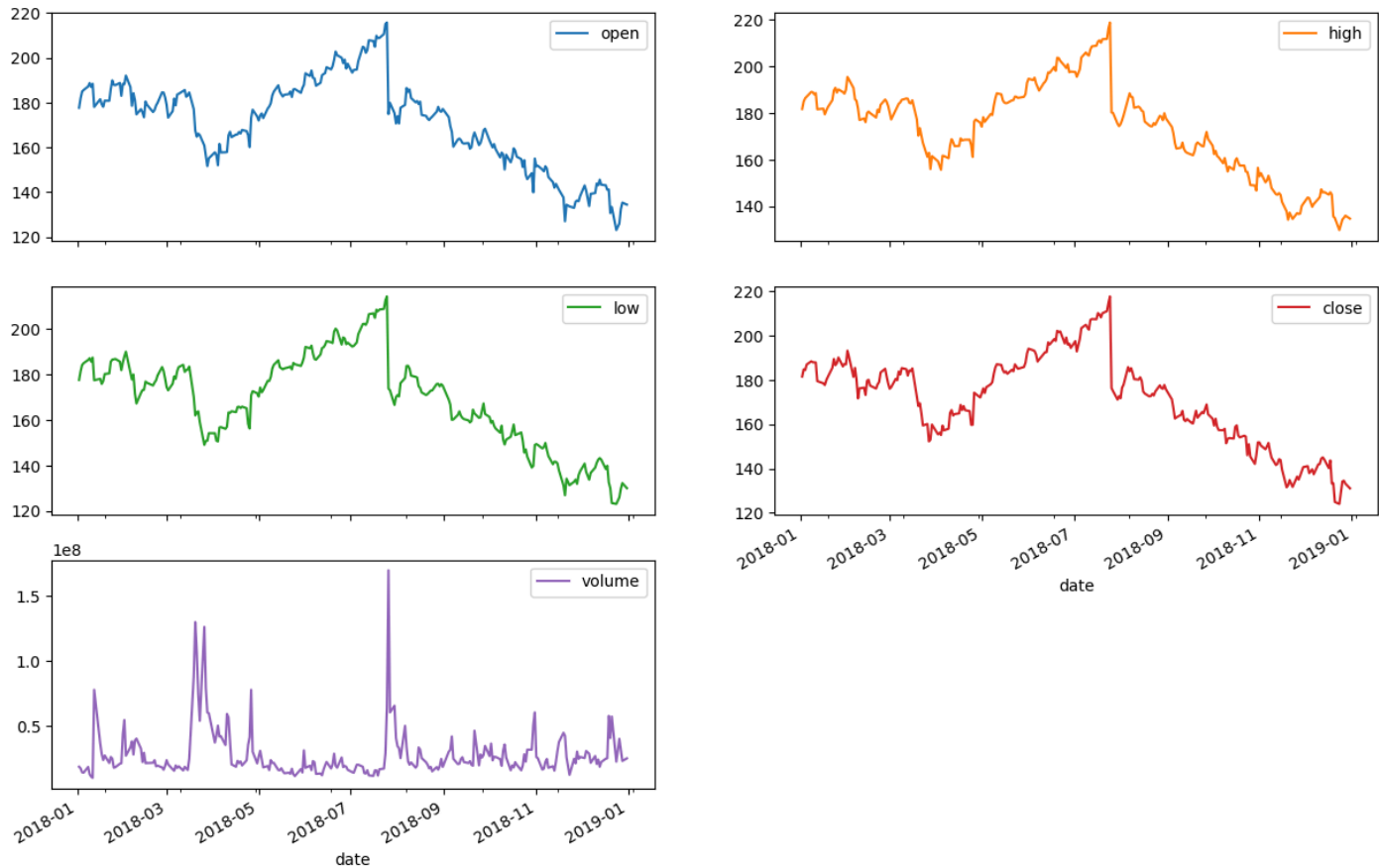
<Axes: title={'center': 'Facebook OHLC Prices during 1st Week of Trading 2018'}, xlabel='date'>



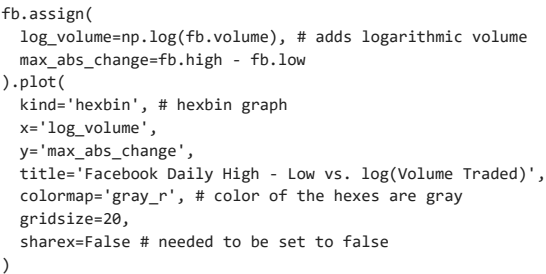
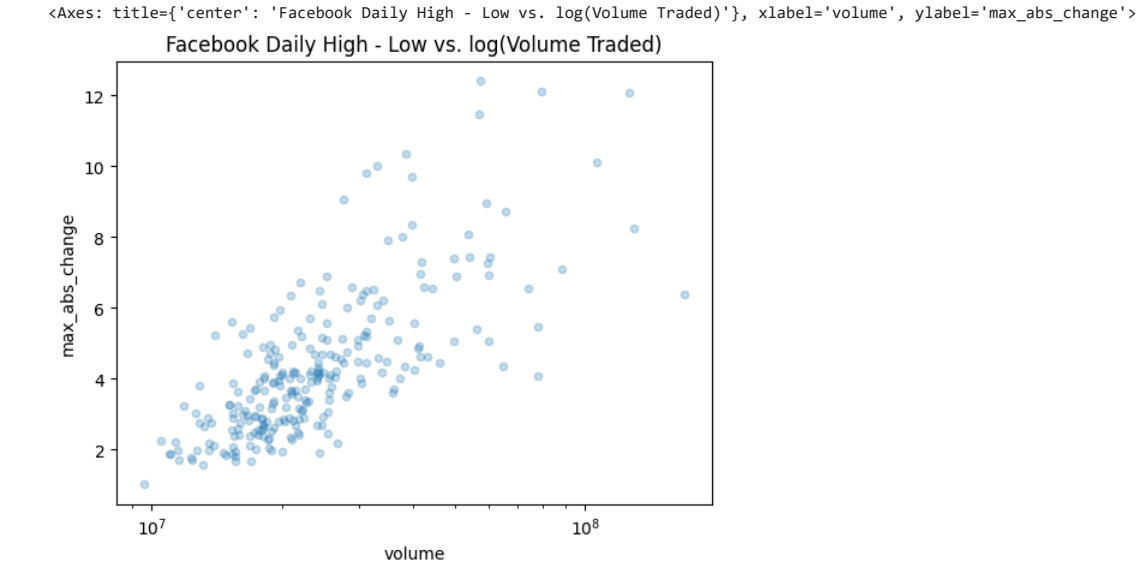
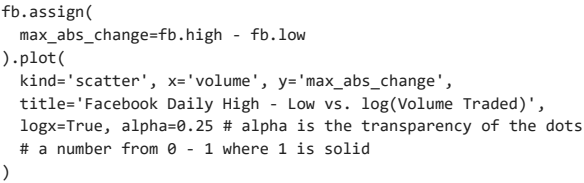
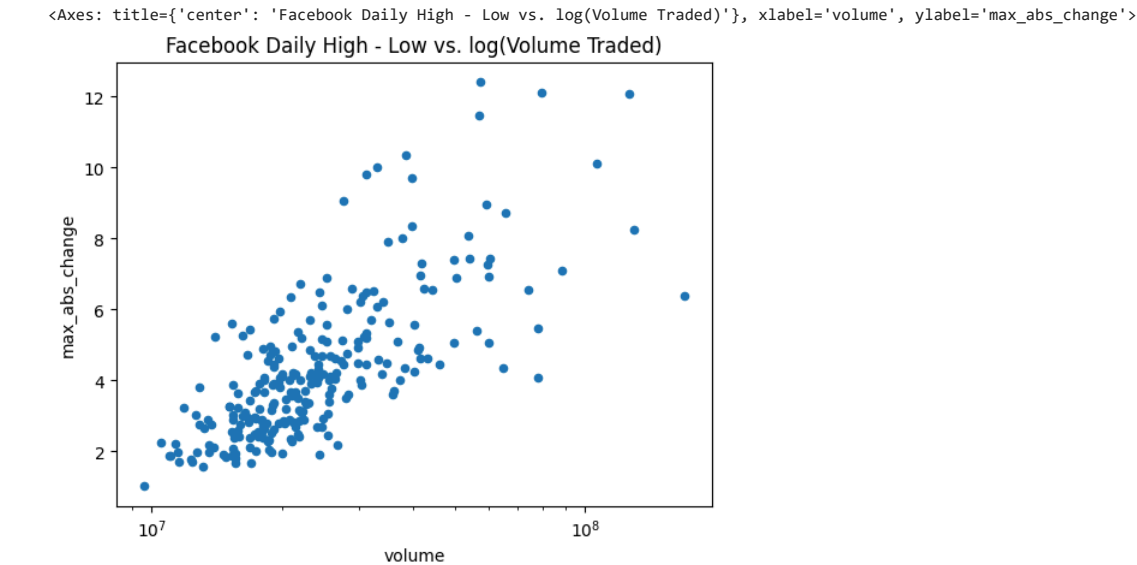
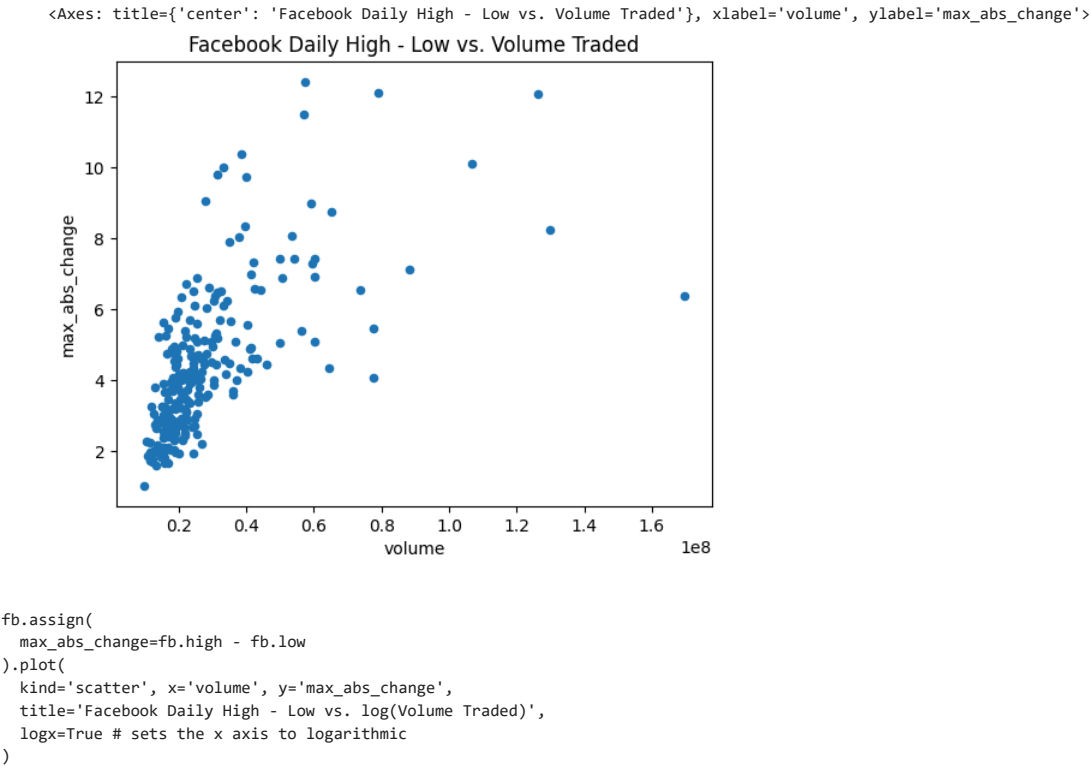
```
fb.plot(
    kind='line',
    subplots=True, # allow subplots
    layout=(3,2), # specifies the layout of the subplots
    figsize=(15,10),
    title='Facebook Stock 2018'
)
```

array([[<Axes: xlabel='date'>, <Axes: xlabel='date'>],
 [<Axes: xlabel='date'>, <Axes: xlabel='date'>],
 [<Axes: xlabel='date'>, <Axes: xlabel='date'>]], dtype=object)

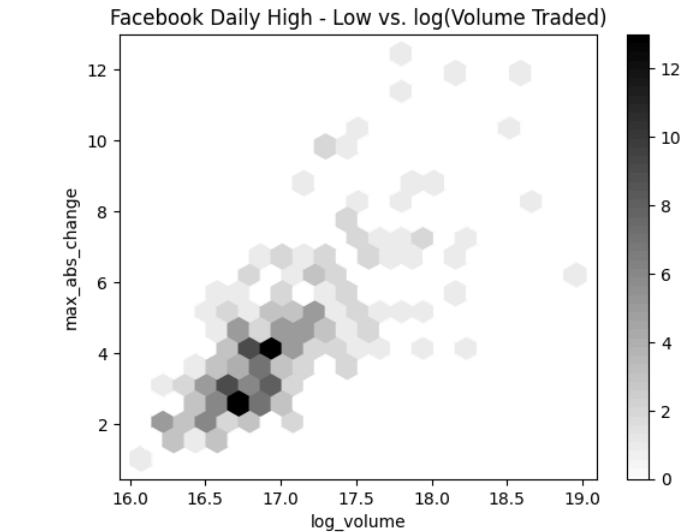
Facebook Stock 2018



```
fb.assign(
    max_abs_change=fb.high - fb.low # adds absolute change column
).plot(
    kind='scatter', x='volume', y='max_abs_change', # scatter plot
    title='Facebook Daily High - Low vs. Volume Traded'
)
```



```
<Axes: title={'center': 'Facebook Daily High - Low vs. log(Volume Traded)'}, xlabel='log_volume', ylabel='max_abs_change'>
```

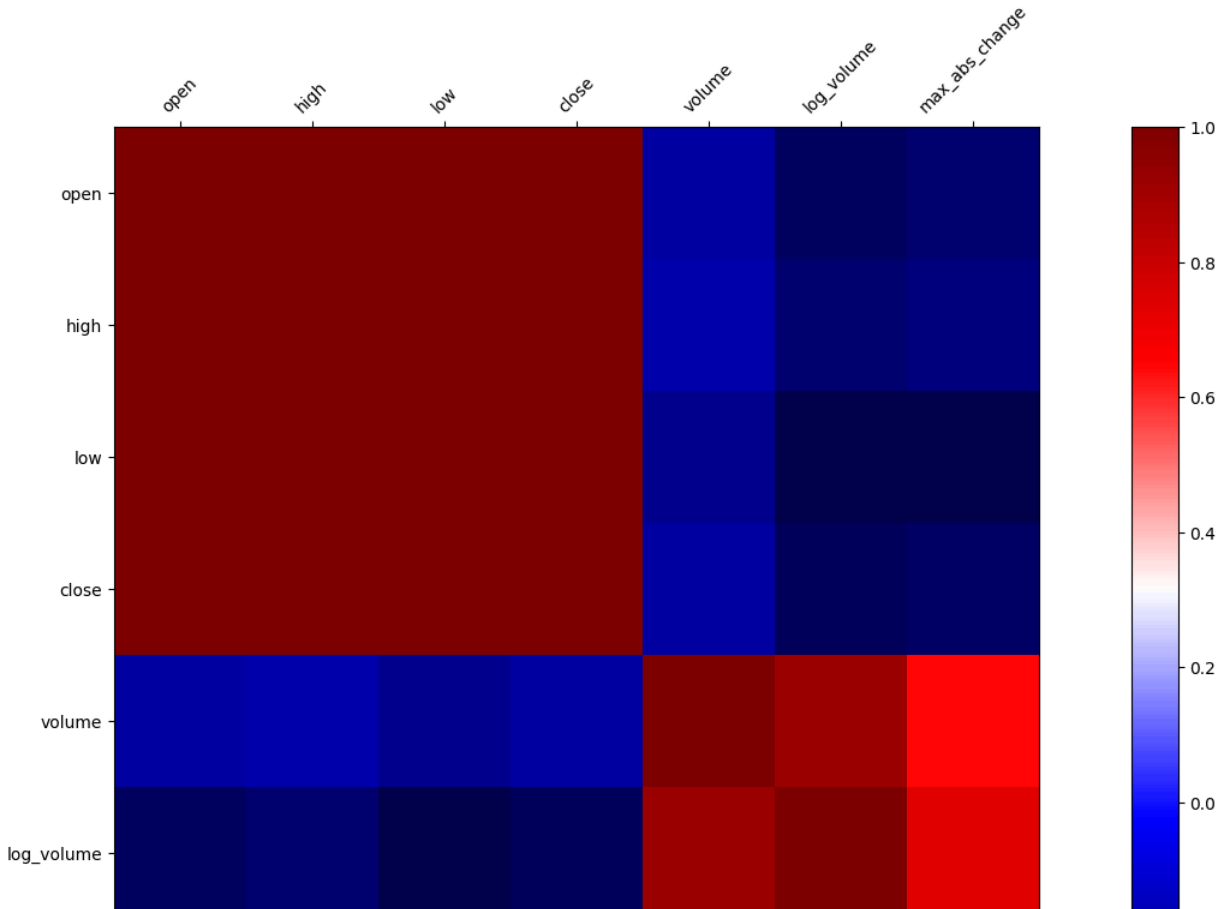


```
fig, ax = plt.subplots(figsize=(20, 10))
fb_corr = fb.assign(
    log_volume=np.log(fb.volume),
    max_abs_change=fb.high - fb.low
).corr() # gets the correlation between every column
```

```
im = ax.matshow(fb_corr, cmap='seismic')
fig.colorbar(im) # creates the heatmap
labels = [col.lower() for col in fb_corr.columns] # gets the column names
ax.set_xticklabels([''] + labels, rotation=45) # sets the column names for x, rotated 45 degrees
ax.set_yticklabels([''] + labels) # sets the column names for y
```

```
# removes the .set_clim() method as it results in an error
```

```
<ipython-input-11-83890aefe672>:10: UserWarning: FixedFormatter should only be used together with FixedLocator
ax.set_xticklabels([''] + labels, rotation=45) # sets the column names for x, rotated 45 degrees
<ipython-input-11-83890aefe672>:11: UserWarning: FixedFormatter should only be used together with FixedLocator
ax.set_yticklabels([''] + labels) # sets the column names for y
[Text(0, -1.0, ''),
 Text(0, 0.0, 'open'),
 Text(0, 1.0, 'high'),
 Text(0, 2.0, 'low'),
 Text(0, 3.0, 'close'),
 Text(0, 4.0, 'volume'),
 Text(0, 5.0, 'log_volume'),
 Text(0, 6.0, 'max_abs_change'),
 Text(0, 7.0, '')]
```



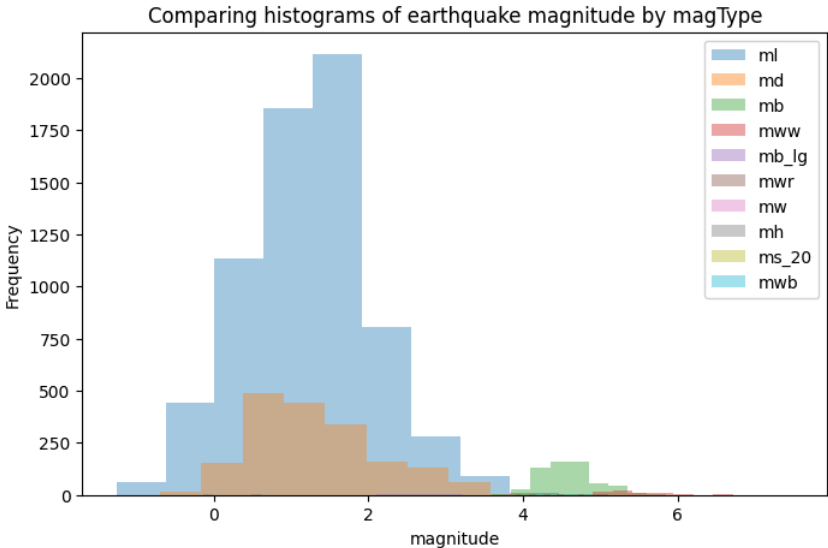
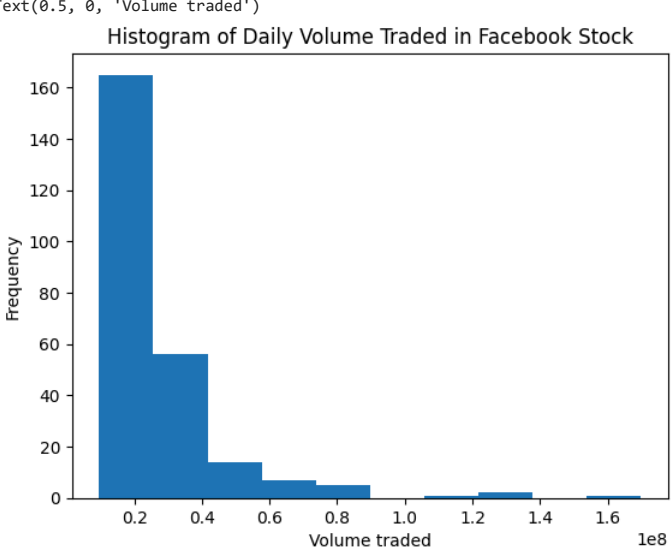
```
fb_corr.loc['max_abs_change', ['volume', 'log_volume']]
# gets the r value between max_abs_change and volume and log_volume
```

```
volume      0.642027
log_volume   0.731542
Name: max_abs_change, dtype: float64
```

```
fb.volume.plot(
    kind='hist', # histogram
    title='Histogram of Daily Volume Traded in Facebook Stock'
)
plt.xlabel('Volume traded')

Text(0.5, 0, 'Volume traded')

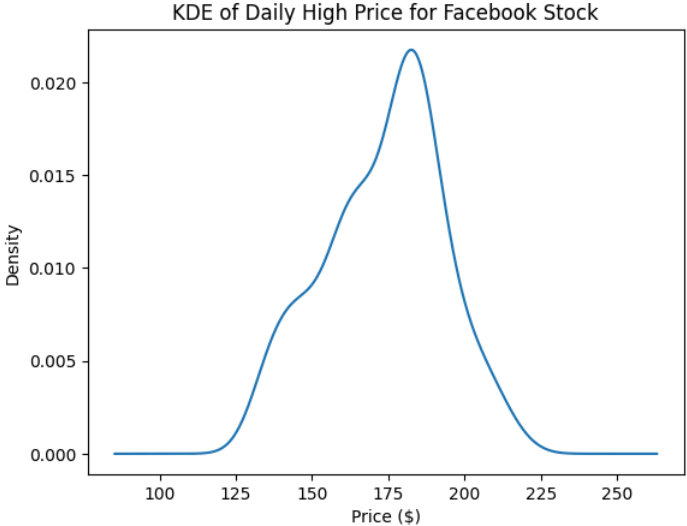
fig, axes = plt.subplots(figsize=(8, 5))
for magtype in quakes.magType.unique(): # iterates thru every magType
    data = quakes.query(f'magType == "{magtype}"').mag # gets the magnitudes for the certain magType
    if not data.empty:
        data.plot(
            kind='hist', ax=axes, alpha=0.4, # creates the histogram, recall alpha
            label=magtype, legend=True,
            title='Comparing histograms of earthquake magnitude by magType'
        )
plt.xlabel('magnitude')
```

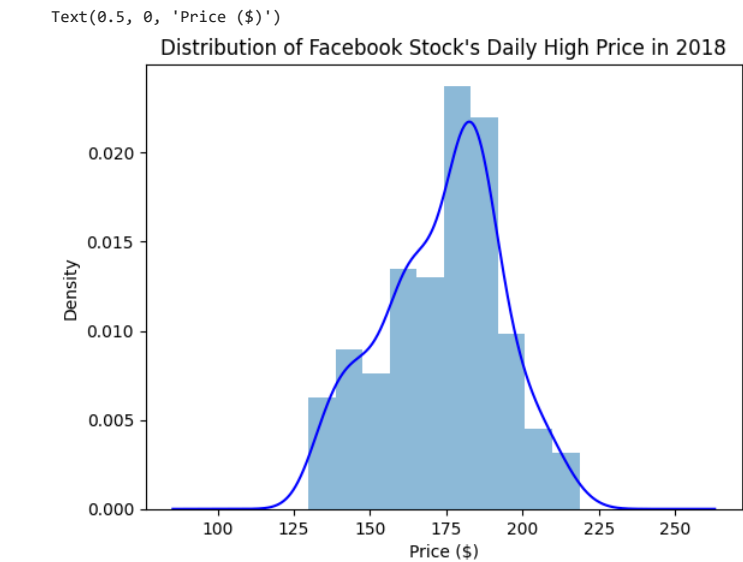


```
fb.high.plot( # uses the high column of fb datagram
    kind='kde', # kernel density estimation graph
    title='KDE of Daily High Price for Facebook Stock'
)
plt.xlabel('Price ($)')

Text(0.5, 0, 'Price ($)')

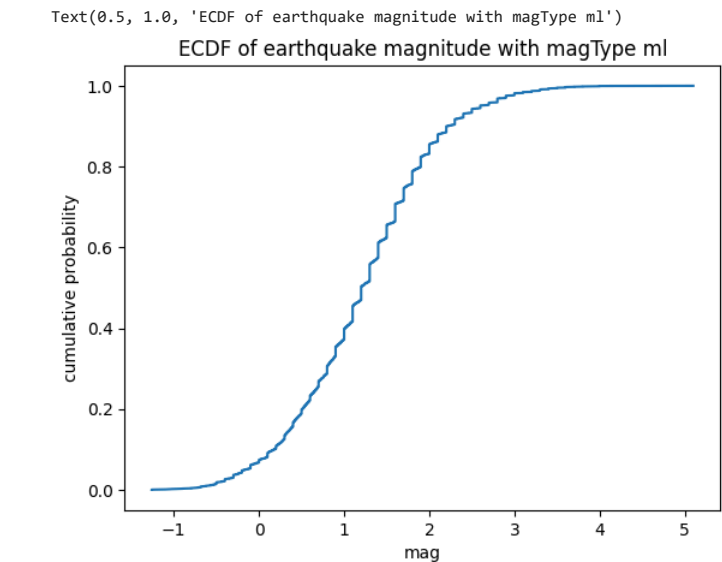
ax = fb.high.plot(kind='hist', density=True, alpha=0.5) # creates a histogram
fb.high.plot(
    ax=ax, kind='kde', color='blue', # creates a kde
    title='Distribution of Facebook Stock\'s Daily High Price in 2018'
)
plt.xlabel('Price ($)') # these two plots would overlap
```





```
from statsmodels.distributions.empirical_distribution import ECDF
# creates the empirical cumulative distribution function

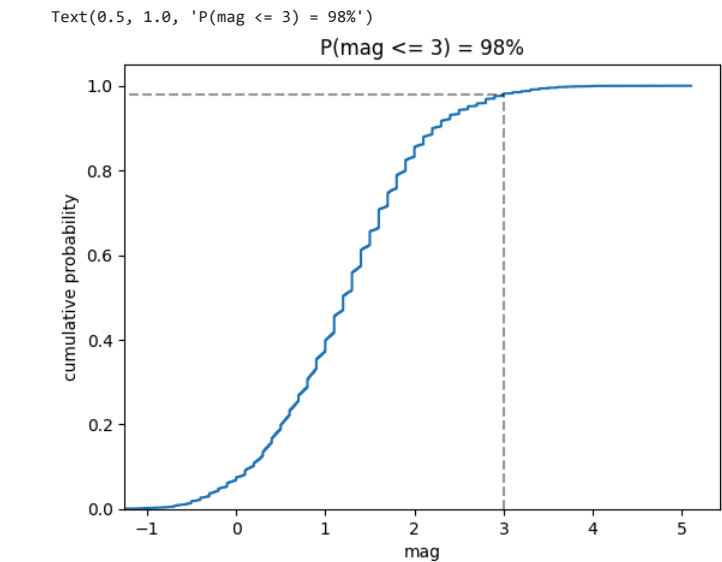
ecdf = ECDF(quakes.query('magType == "ml").mag) # gets magnitudes with magType ml
plt.plot(ecdf.x, ecdf.y) # plots the ecdf result
# labels and title
plt.xlabel('mag')
plt.ylabel('cumulative probability')
plt.title('ECDF of earthquake magnitude with magType ml')
```



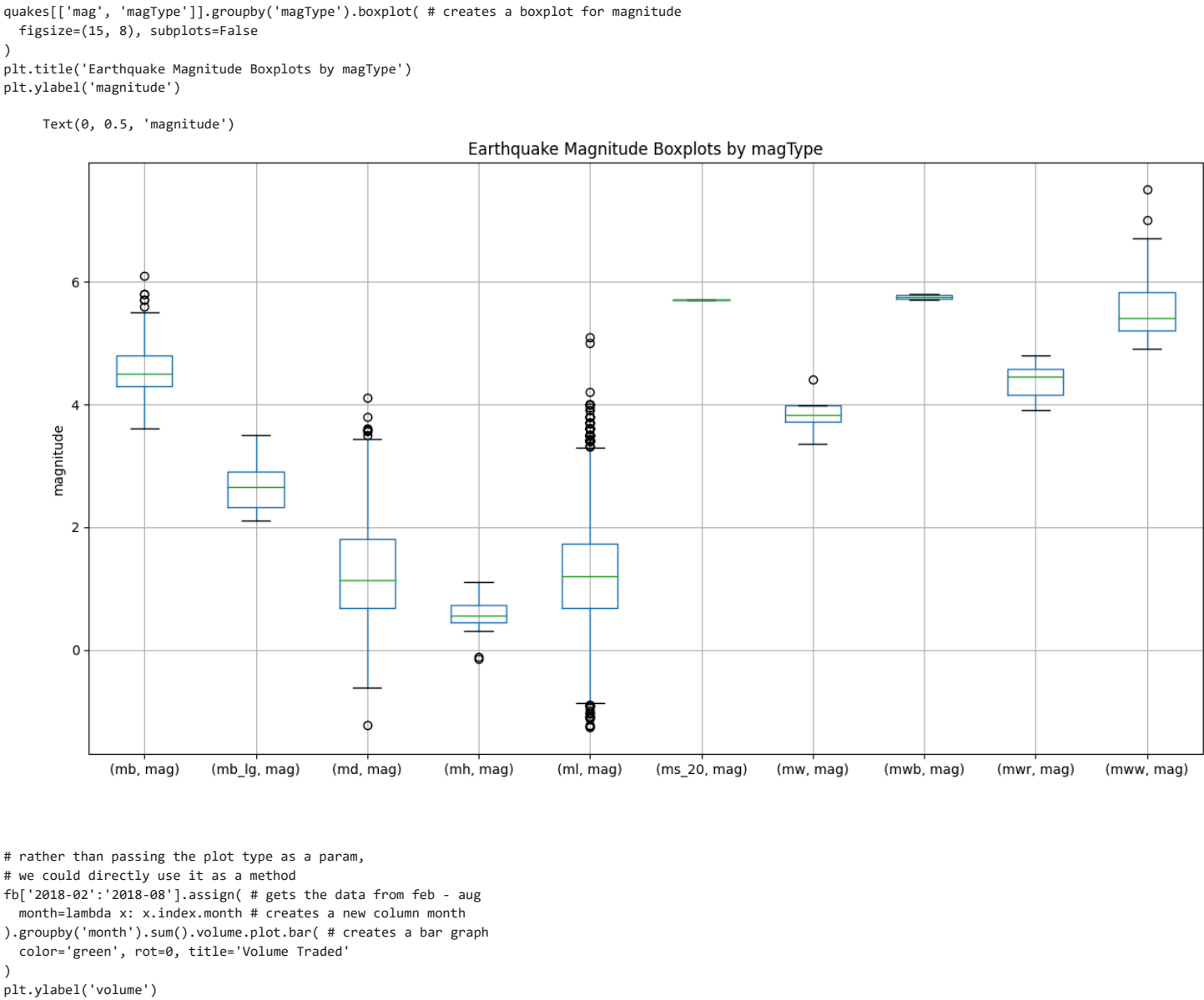
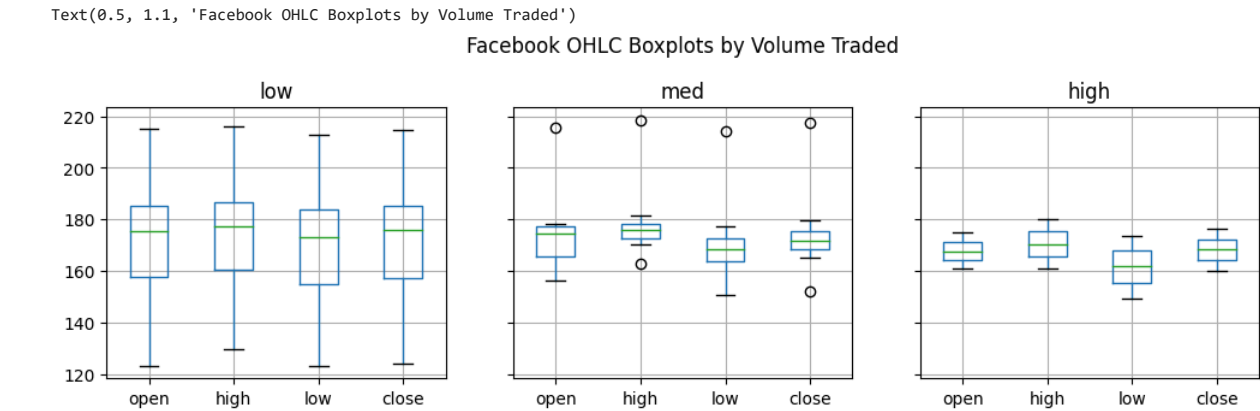
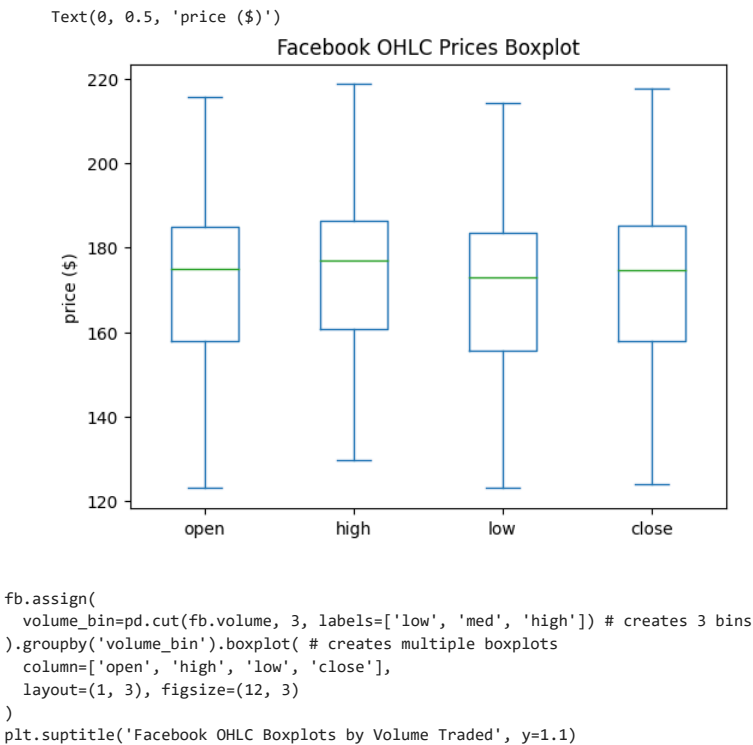
```
from statsmodels.distributions.empirical_distribution import ECDF

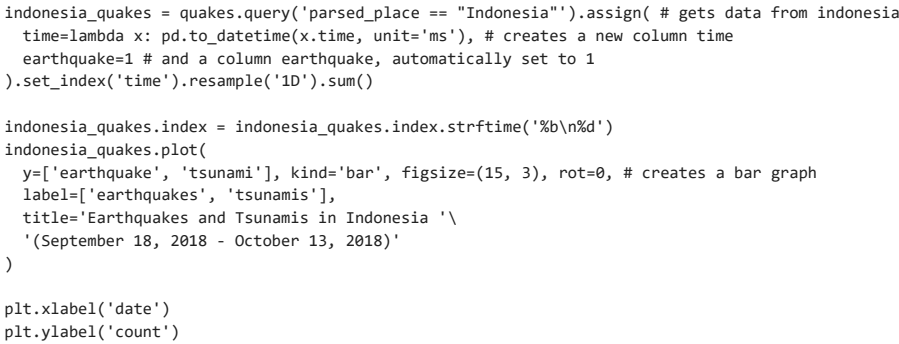
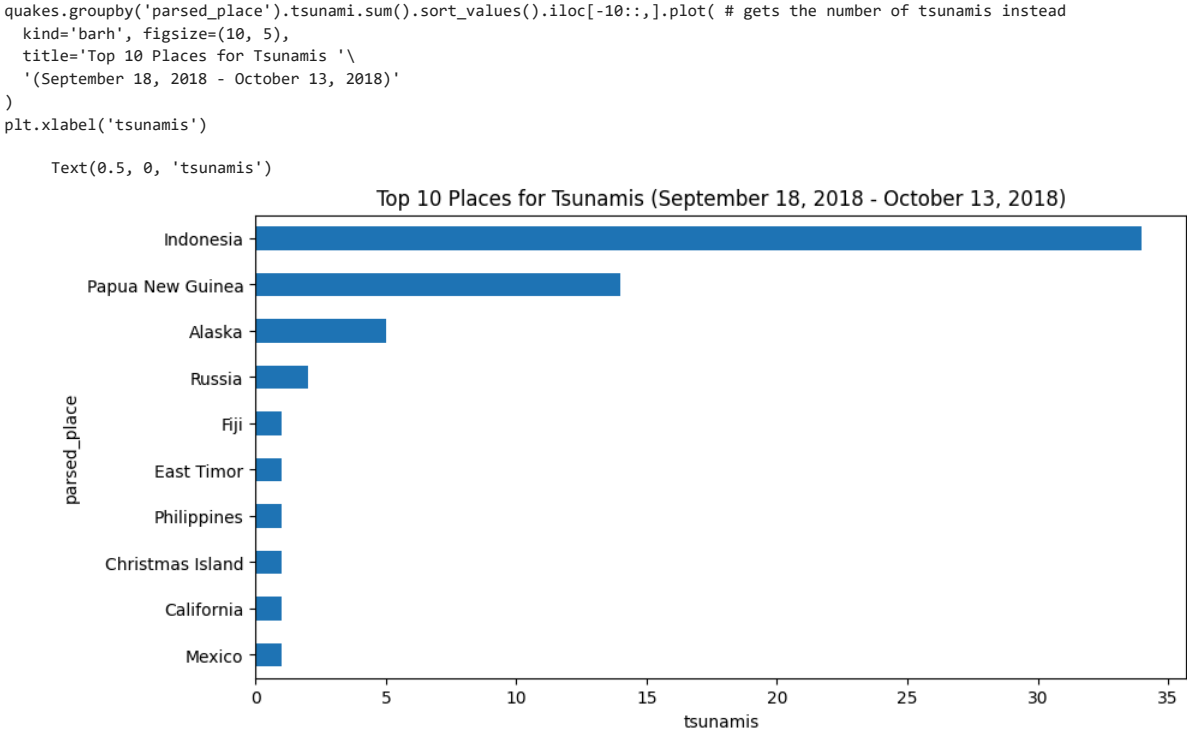
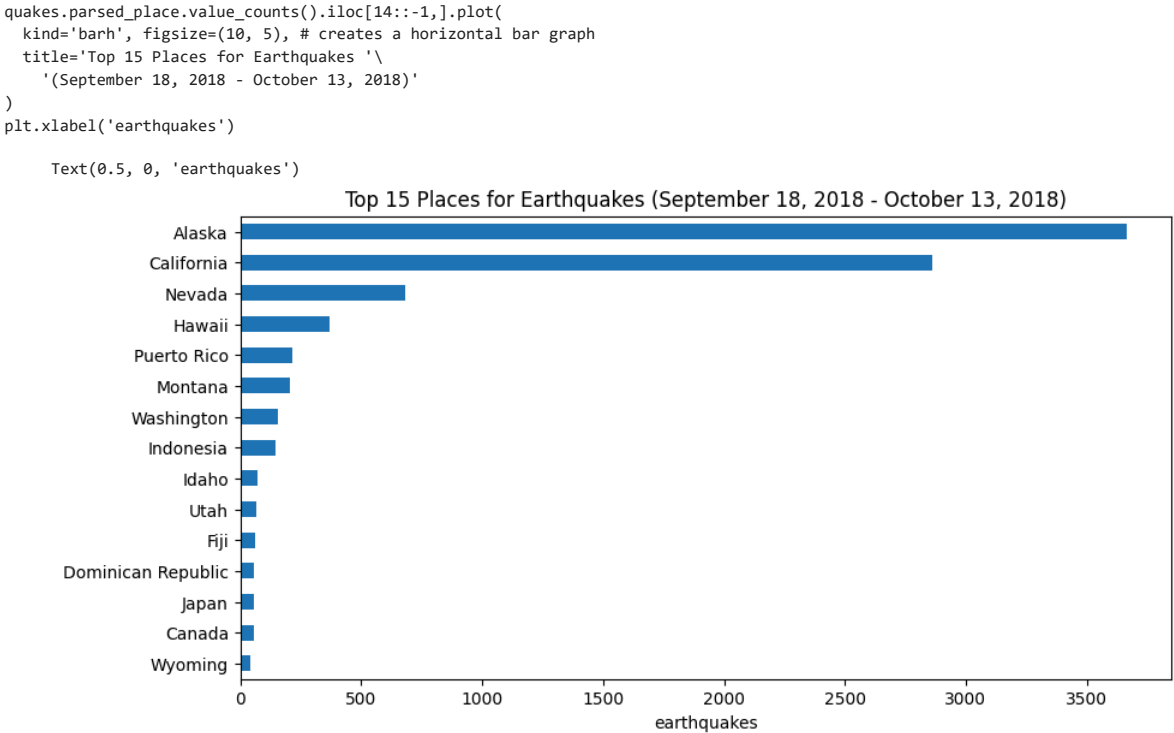
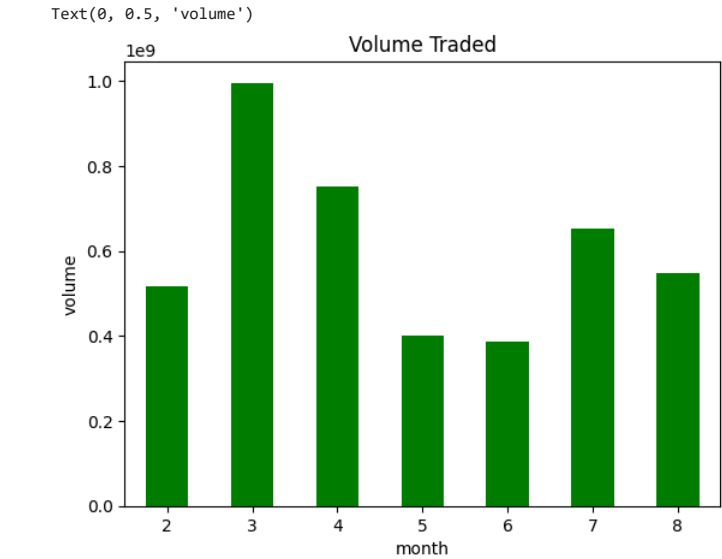
ecdf = ECDF(quakes.query('magType == "ml").mag)
plt.plot(ecdf.x, ecdf.y)
plt.xlabel('mag')
plt.ylabel('cumulative probability')

plt.plot( # creates a line to visualize the magnitudes less than 3
[3, 3], [0, .98], 'k--',
[-1.5, 3], [0.98, 0.98], 'k--', alpha=0.4
)
plt.ylim(0, None)
plt.xlim(-1.25, None)
plt.title('P(mag <= 3) = 98%')
```



```
fb.iloc[:,4].plot(kind='box', title='Facebook OHLC Prices Boxplot') # creates a boxplot
plt.ylabel('price ($)')
```

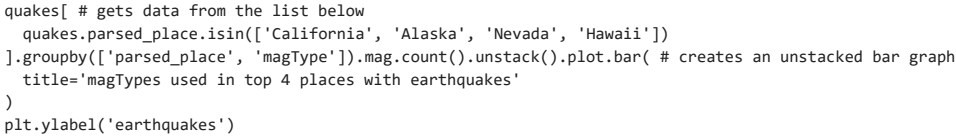




Bar chart titled "Earthquakes and Tsunamis in Indonesia (September 18, 2018 - October 13, 2018)". The x-axis represents the date, ranging from Sep 18 to Oct 13. The y-axis represents the count, ranging from 0 to 40. The chart displays two data series: earthquakes (blue bars) and tsunamis (orange bars). A significant peak in earthquakes is observed on Sep 28, with a count of 40. Tsunamis are also present on Sep 28, with a count of 18. Other notable earthquake counts include 14 on Oct 01 and Oct 02, and 8 on Sep 29. Tsunami counts are generally low, with a maximum of 4 on Sep 26 and Oct 08.

date	earthquakes	tsunamis
Sep 18	5	0
Sep 19	4	0
Sep 20	4	0
Sep 21	2	0
Sep 22	5	0
Sep 23	4	0
Sep 24	4	0
Sep 25	4	0
Sep 26	4	2
Sep 27	4	1
Sep 28	40	18
Sep 29	9	2
Sep 30	4	0
Oct 01	14	5
Oct 02	14	3
Oct 03	6	2
Oct 04	7	0
Oct 05	0	0
Oct 06	3	0
Oct 07	4	1
Oct 08	4	2
Oct 09	1	0
Oct 10	4	0
Oct 11	2	0
Oct 12	1	0
Oct 13	4	0

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



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

