```
## Hands-on Activity 9.2 Customized Visualizations using Seaborn

**Supplementary**

**Name: Delloson, Angelo Dan Renz G.**

**Section: CPE22S3**

**Performed on:**

**Submitted on:**

**Submitted to: Engr. Roman M. Richard**
```

Hands-on Activity 9.2 Customized Visualizations using Seaborn

Supplementary

Name: Delloson, Angelo Dan Renz G.

Section: CPE22S3

Performed on:

Submitted on:

Submitted to: Engr. Roman M. Richard

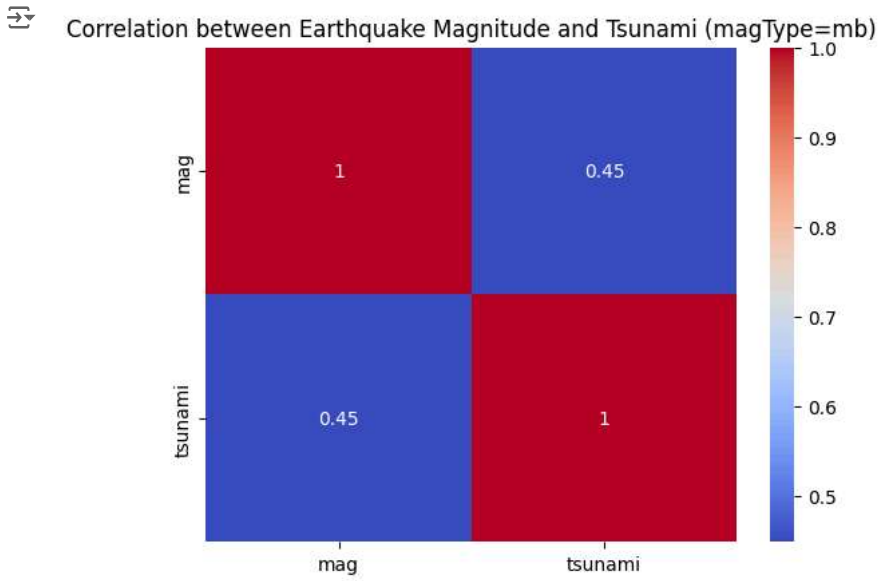
```
%matplotlib inline
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns

fb = pd.read_csv(
  '/content/fb_stock_prices_2018 (1).csv', index_col='date', parse_dates=True
)
eq = pd.read_csv('/content/earthquakes-1 (1).csv')

```

```
eq = pd.read_csv('/content/earthquakes-1 (1).csv')
eq
```

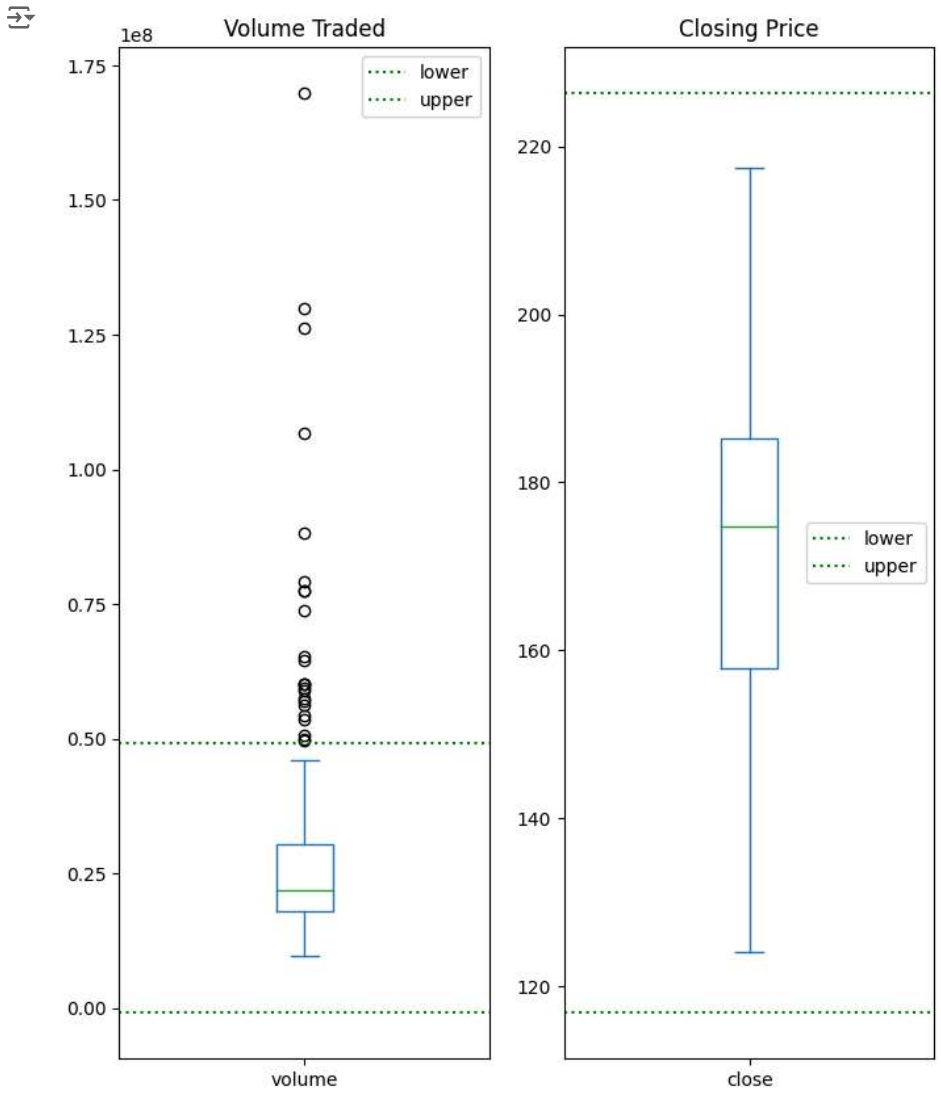
	mag	magType	time	place	tsunami	parsed_place
0	1.35	ml	1539475168010	9km NE of Aguanga, CA	0	California
1	1.29	ml	1539475129610	9km NE of Aguanga, CA	0	California
2	3.42	ml	1539475062610	8km NE of Aguanga, CA	0	California
3	0.44	ml	1539474978070	9km NE of Aguanga, CA	0	California
4	2.16	md	1539474716050	10km NW of Avenal, CA	0	California
...	...	...	...	...	...	...
9327	0.62	md	1537230228060	9km ENE of Mammoth Lakes, CA	0	California
9328	1.00	ml	1537230135130	3km W of Julian, CA	0	California
9329	2.40	md	1537229908180	35km NNE of Hatillo, Puerto Rico	0	Puerto Rico
9330	1.10	ml	1537229545350	9km NE of Aguanga, CA	0	California
9331	0.66	ml	1537228864470	9km NE of Aguanga, CA	0	California

Next steps: Generate code with eq View recommended plots

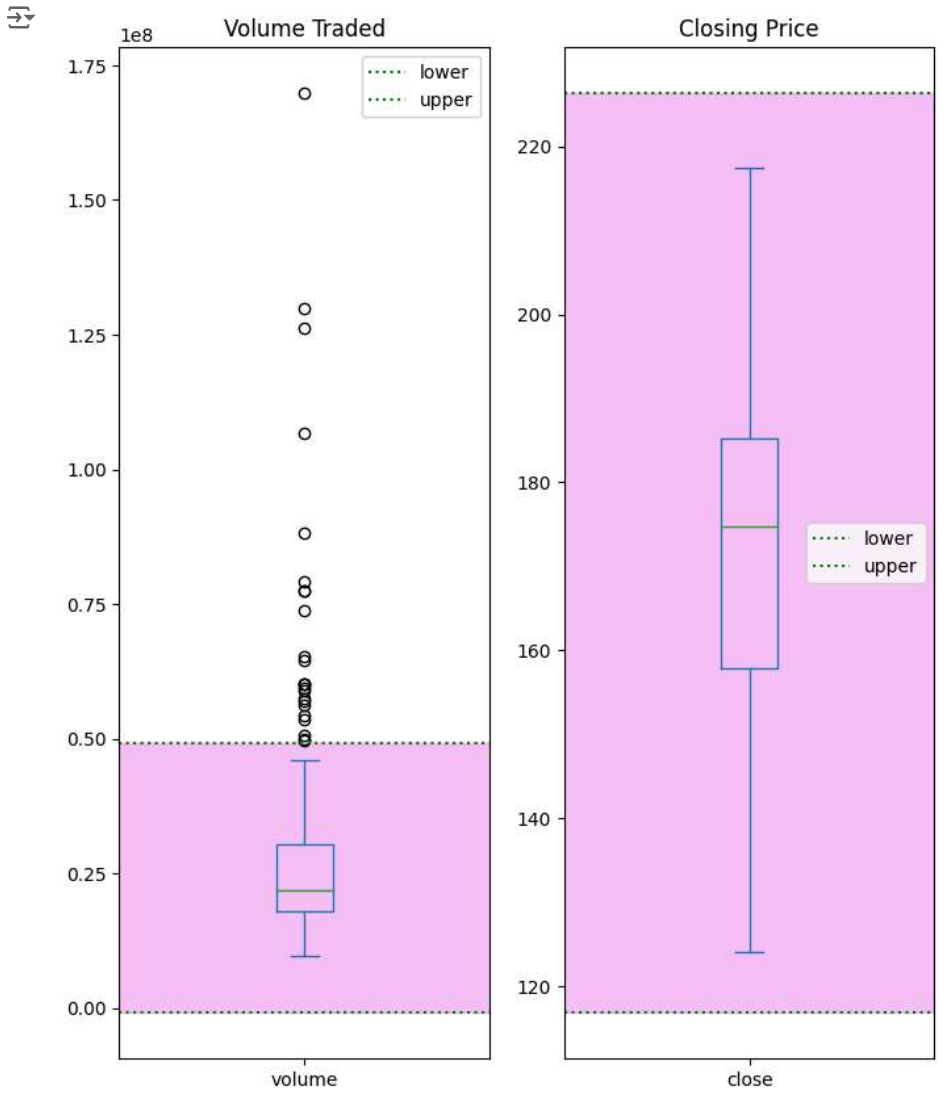
```
eq_mb = eq[(eq['magType'] == 'mb')]

correlation_matrix = eq_mb[['mag', 'tsunami']].corr()

sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation between Earthquake Magnitude and Tsunami (magType=mb)')
plt.show()
```




```
Vol = ['volume', 'close']
gra = fb[Vol]
qtl = gra.quantile([0.25, 0.75])
qtl.loc['iqr',:] = qtl.loc[0.75,:] - qtl.loc[0.25,:]

A = gra.plot(
    kind='box',
    subplots=True,
    figsize=(8, 10),
    title=['Volume Traded', 'Closing Price'])

for ax, col in zip(A, Vol):
  stats = qtl[col]
  lower = stats.loc[0.25] - 1.5 * stats['iqr']
  upper = stats.loc[0.75] + 1.5 * stats['iqr']
  for bound, name in zip([lower, upper], ['lower', 'upper']):
    ax.axhline( bound, color='green', linestyle='dotted', label=name )
    ax.legend()

```



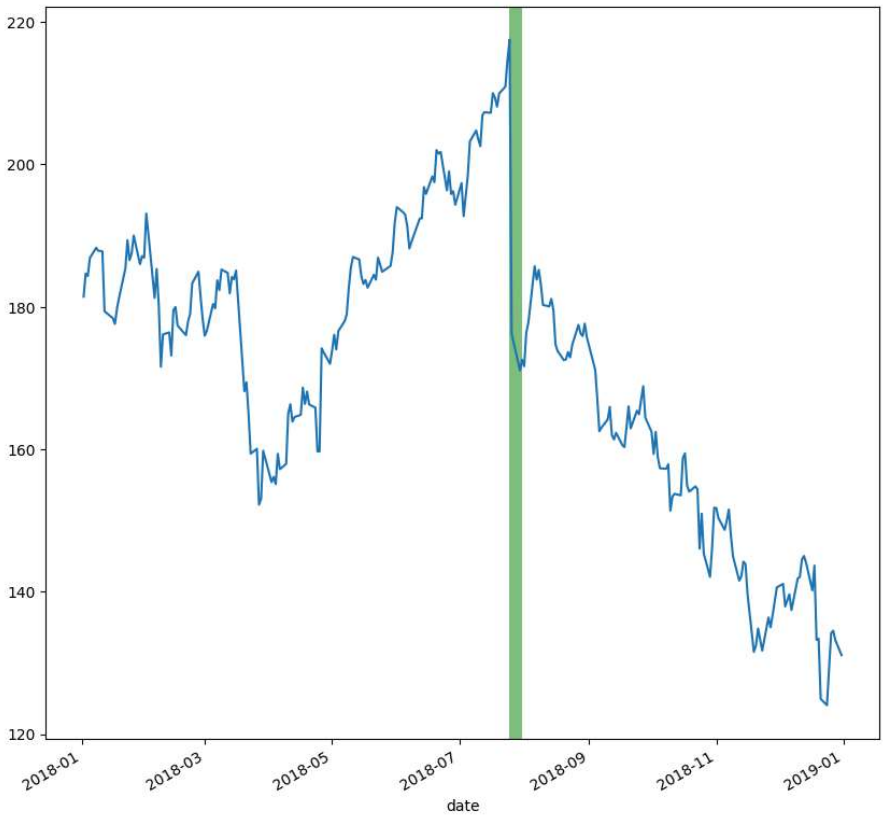

```
volum = ['volume', 'close']
graph = fb[volum]
qtl = graph.quantile([0.25, 0.75])
qtl.loc['iqr',:] = qtl.loc[0.75,:] - qtl.loc[0.25,:]

L = graph.plot(
    kind='box',
    subplots=True,
    figsize=(8, 10),
    title=['Volume Traded', 'Closing Price'])

for ax, col in zip(L, volum):
  stats = qtl[col]
  lower = stats.loc[0.25] - 1.5 * stats['iqr']
  upper = stats.loc[0.75] + 1.5 * stats['iqr']
  for bound, name in zip([lower, upper], ['lower', 'upper']):
    ax.axhline( bound, color='green', linestyle='dotted', label=name )
    ax.legend()
  ax.axhspan(lower, upper, color='violet', alpha=0.5)
```




```
1 fb = pd.read_csv('/content/fb_stock_prices_2018 (1).csv')
2 fb['date'] = pd.to_datetime(fb['date'])
3 fb.set_index('date', inplace = True)
4
```

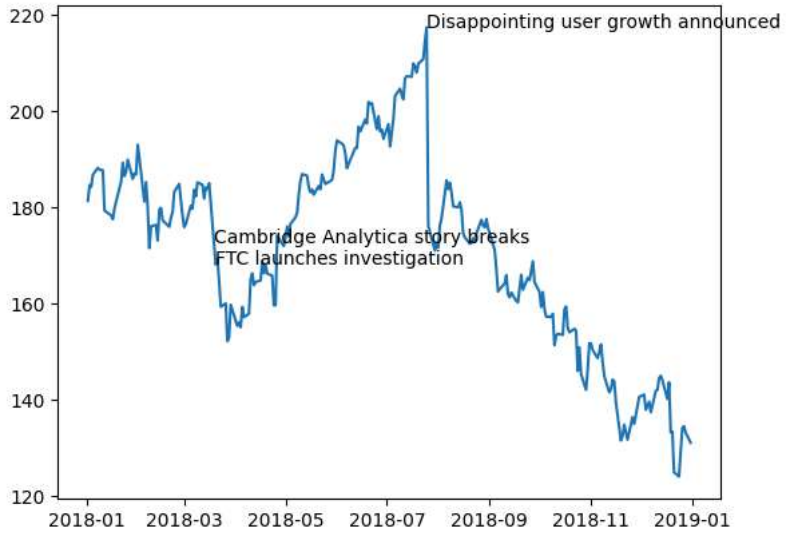
```
1 start = '2018-07-25'
2 end = '2018-07-31'
3
4 fb.close.plot(figsize = (10,10))
5 plt.axvspan(start, end, facecolor = 'green', alpha = 0.5)
6
```

```
<matplotlib.patches.Polygon at 0x7d69499f3a60>
```




```
1 import matplotlib.dates as plt_date
2
3 plt.plot(fb.close)
4 plt.annotate('Disappointing user growth announced',
5              xy=(plt_date.date2num(pd.to_datetime('2018-07-25')), fb.loc['2018-07-25', 'close']))
6 plt.annotate('Cambridge Analytica story breaks',
7              xy=(plt_date.date2num(pd.to_datetime('2018-03-19')), fb.loc['2018-03-19', 'close']))
8 plt.annotate('FTC launches investigation',
9              xy=(plt_date.date2num(pd.to_datetime('2018-03-20')), fb.loc['2018-03-20', 'close']))
```

```
Text(17610.0, 168.15, 'FTC launches investigation')
```



```
import itertools

def reg_resid_plots(data):

    graphs = data.shape[1]
    permutation_count = graphs * (graphs - 1)
    fig, ax = plt.subplots(permutation_count, 2, figsize=(15, 8))
    chplo = plt.cm.get_cmap('viridis')
    colors = [chplo(i) for i in np.linspace(0, 1, permutation_count)]
    for (x, y), axes, color in zip(
            itertools.permutations(data.columns, 2),
            ax,
            colors):
        for subplot, func in zip(axes, (sns.regplot, sns.residplot)):
            func(x=x, y=y, data=data, ax=subplot, color='blue')
    plt.close()
    return fig
```