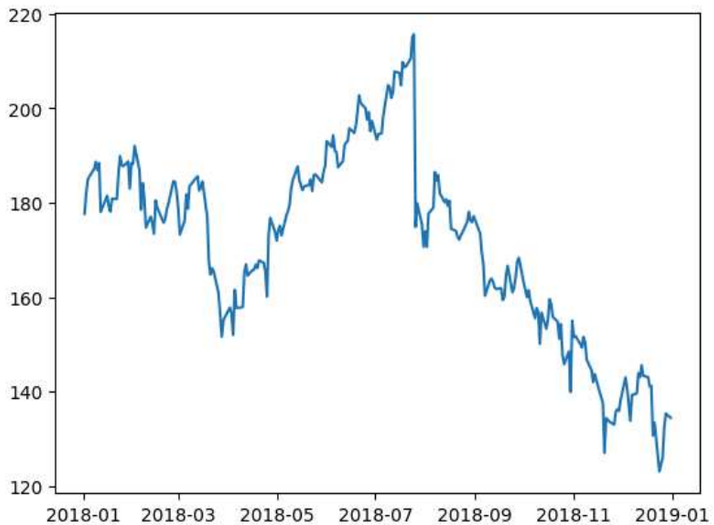
Submitted by: Angelo Luis C. Cu

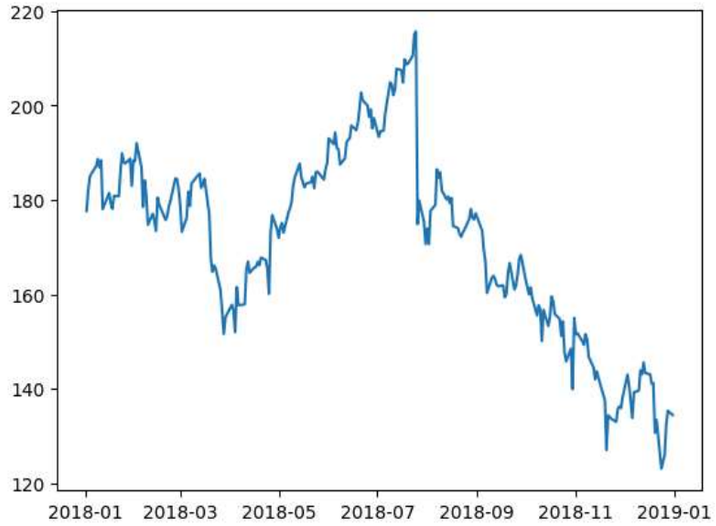
```
import matplotlib.pyplot as plt
import pandas as pd


fb = pd.read_csv(
  'data/fb_stock_prices_2018.csv', index_col='date', parse_dates=True
)
plt.plot(fb.index, fb.open) # parameters are (x, y)
plt.show() # shows a line graph
```



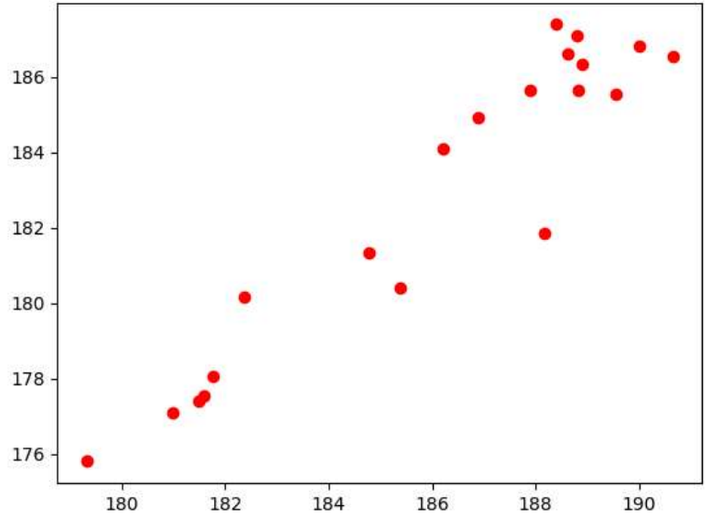

```
%matplotlib inline # removes the need to do plt.show()
import matplotlib.pyplot as plt
import pandas as pd
fb = pd.read_csv(
  'data/fb_stock_prices_2018.csv', index_col='date', parse_dates=True
)
plt.plot(fb.index, fb.open)
```

```
[<matplotlib.lines.Line2D at 0x7a0fbf973970>]
```



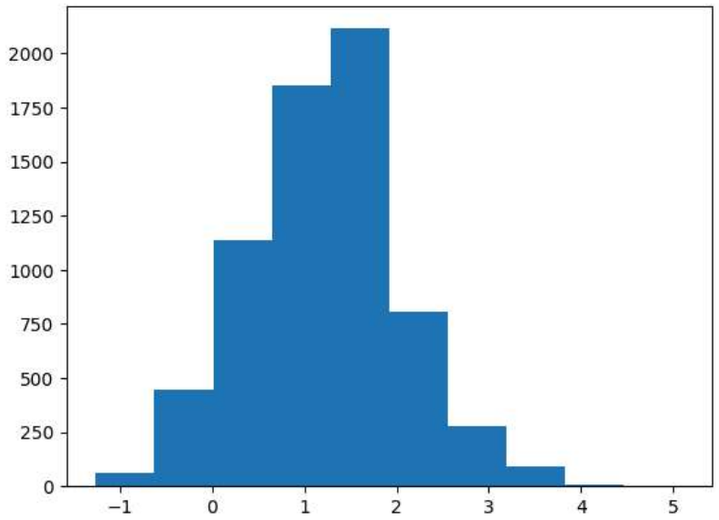

```
plt.plot('high', 'low', 'ro', data=fb.head(20)) # for the first 20 datapoints
# x = fb.high
# y = fb.low
# style = ro (red scatter plot)
```

```
[<matplotlib.lines.Line2D at 0x7a0fbf8ac760>]
```



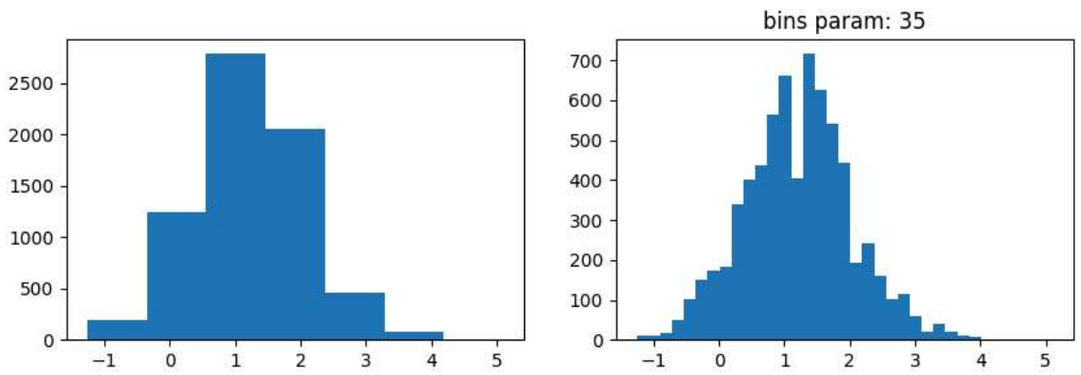
```
quakes = pd.read_csv('data/earthquakes.csv') # gets earthquakes data
plt.hist(quakes.query('magType == "ml"').mag) # gets magnitude data with magType 'ml'
# .hist() creates a histogram
```

```
(array([6.400e+01, 4.450e+02, 1.137e+03, 1.853e+03, 2.114e+03, 8.070e+02,
        2.800e+02, 9.200e+01, 9.000e+00, 2.000e+00]),
 array([-1.26 , -0.624,  0.012,  0.648,  1.284,  1.92 ,  2.556,  3.192,
         3.828,  4.464,  5.1  ]),
 <BarContainer object of 10 artists>)
```




```
x = quakes.query('magType == "ml"').mag
fig, axes = plt.subplots(1, 2, figsize=(10, 3)) # creates 2 subgraphs with size 10 and 3
for ax, bins in zip(axes, [7, 35]): # for each subgraph,
  ax.hist(x, bins=bins) # creates a histogram with the specified number of bins
ax.set_title(f'bins param: {bins}')
```

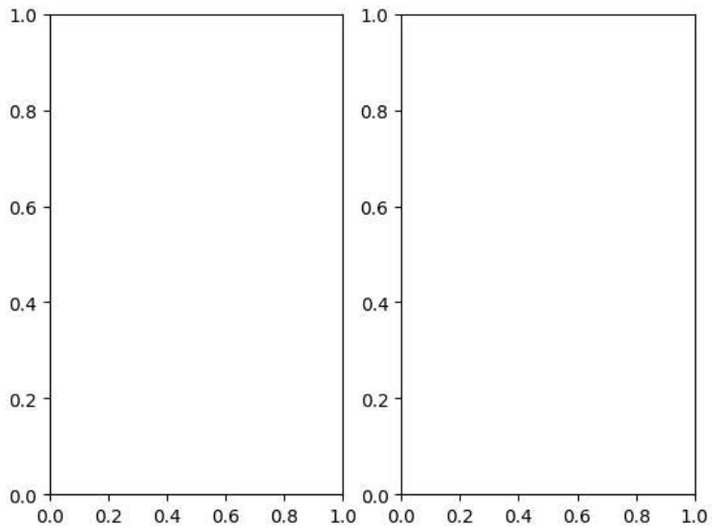
```
Text(0.5, 1.0, 'bins param: 35')
```



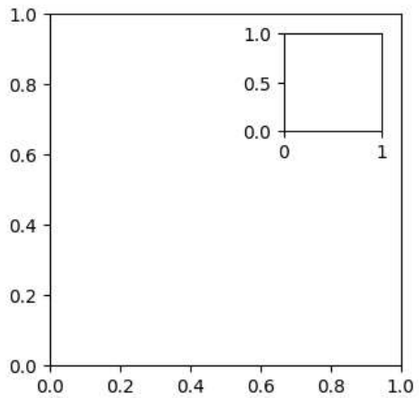
```
fig = plt.figure() # contains the graph itself
```

```
<Figure size 640x480 with 0 Axes>
```

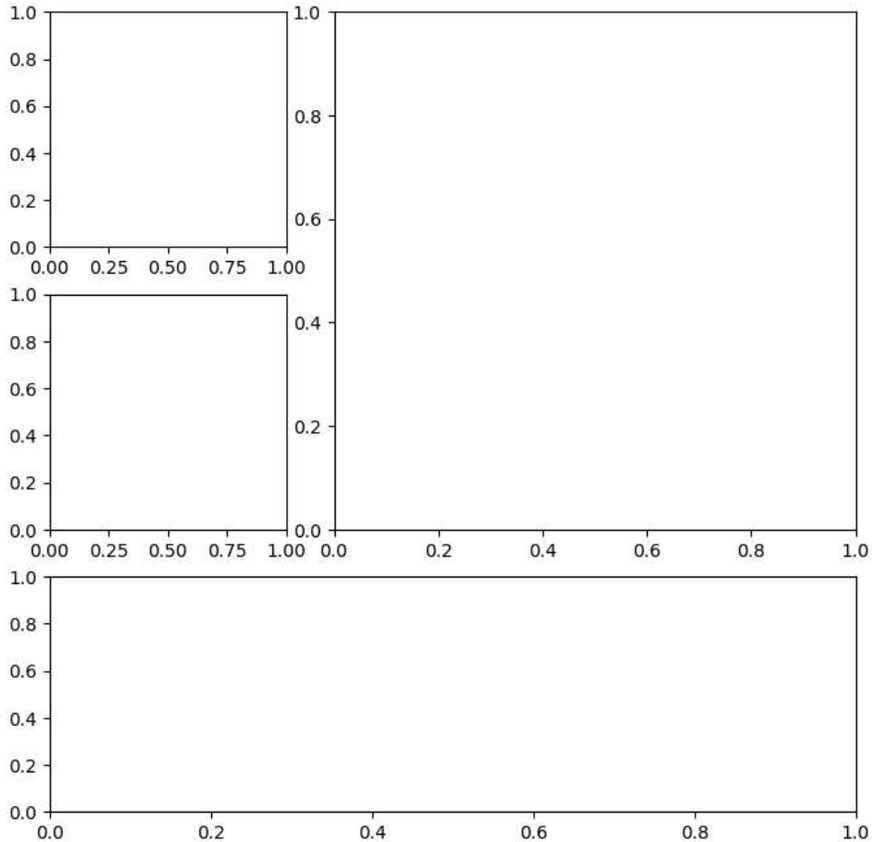
```
fig, axes = plt.subplots(1, 2) # creates 2 subgraphs with .subplots() method
# parameters are (row, col)
```




```
fig = plt.figure(figsize=(3, 3))
# creates a graph with width and height of 3 inches
outside = fig.add_axes([0.1, 0.1, 0.9, 0.9])
inside = fig.add_axes([0.7, 0.7, 0.25, 0.25])
```



```
fig = plt.figure(figsize=(8, 8))
gs = fig.add_gridspec(3, 3) # adds a 3x3 gridspace
top_left = fig.add_subplot(gs[0, 0]) # gets the [0][0]
mid_left = fig.add_subplot(gs[1, 0]) # gets the [1][0]
top_right = fig.add_subplot(gs[:2, 1:]) # gets the 4 grids
bottom = fig.add_subplot(gs[2,:]) # gets the bottom 3 grids
```



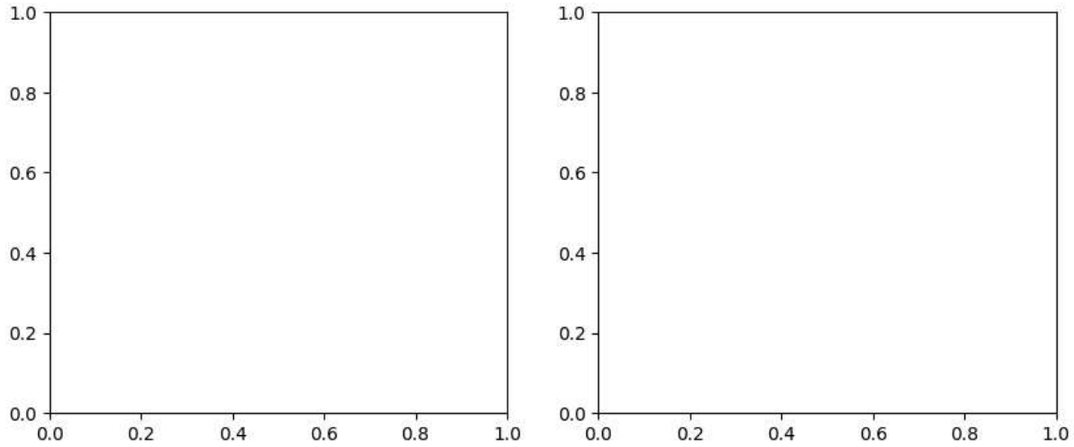

```
fig.savefig('empty.png') # saves the figure above to a png
```

```
plt.close('all')
# it is important to close figures, like how we use close in file manipulation
```

```
fig = plt.figure(figsize=(10, 4)) # size of 10x4 inches
```

```
<Figure size 1000x400 with 0 Axes>
```

```
fig, axes = plt.subplots(1, 2, figsize=(10, 4)) # creates 2 subgraphs
# figsize is for the entire figure with the subgraphs
```




```
import random
import matplotlib as mpl

rcparams_list = list(mpl.rcParams.keys()) # list all the parameters of matplotlib
random.seed(20) # gets 20 random params
random.shuffle(rcparams_list) # shuffles the params
sorted(rcparams_list[:20]) # outputs them
```

```
['animation.convert_args',
 'axes.edgecolor',
 'axes.formatter.use_locale',
 'axes.spines.right',
 'boxplot.meanprops.markersize',
 'boxplot.showfliers',
 'keymap.home',
 'lines.markerfacecolor',
 'lines.scale_dashes',
 'mathtext.rm',
 'patch.force_edgecolor',
 'savefig.facecolor',
 'svg.fonttype',
 'text.hinting_factor',
 'xtick.alignment',
 'xtick.minor.top',
 'xtick.minor.width',
 'ytick.left',
 'ytick.major.left',
 'ytick.minor.width']
```

```
mpl.rcParams['figure.figsize'] # checks the current default figsize
```

```
[6.4, 4.8]
```

```
mpl.rcParams['figure.figsize'] = (300, 10) # changes the default to 300x10
mpl.rcParams['figure.figsize']
```

```
[300.0, 10.0]
```

```
mpl.rcdefaults() # restores default settings
mpl.rcParams['figure.figsize']
```

```
[6.4, 4.8]
```

```
plt.rc('figure', figsize=(20, 20)) # another way to change settings using pyplot
plt.rcdefaults() # restore to default
```