Engineering Analytics (ESE1007) Seminar 5

Data Visualization

Histogram

- It is an estimate of the probability distribution of a continuous variable (quantitative variable) and was first introduced by Karl Pearson.
- It differs from a bar graph, in the sense that a bar graph relates two variables, but a histogram relates only one.
- To construct a histogram, the first step is to "bin" the range of values—that is, divide the entire range of values into a series of intervals—and then count how many values fall into each interval.
- The bins are usually specified as consecutive, non-overlapping intervals of a variable. The bins (intervals) must be adjacent, and are often (but are not required to be) of equal size.

Histogram

https://www.youtube.com/watch?v=0Ul8SOlOu8c

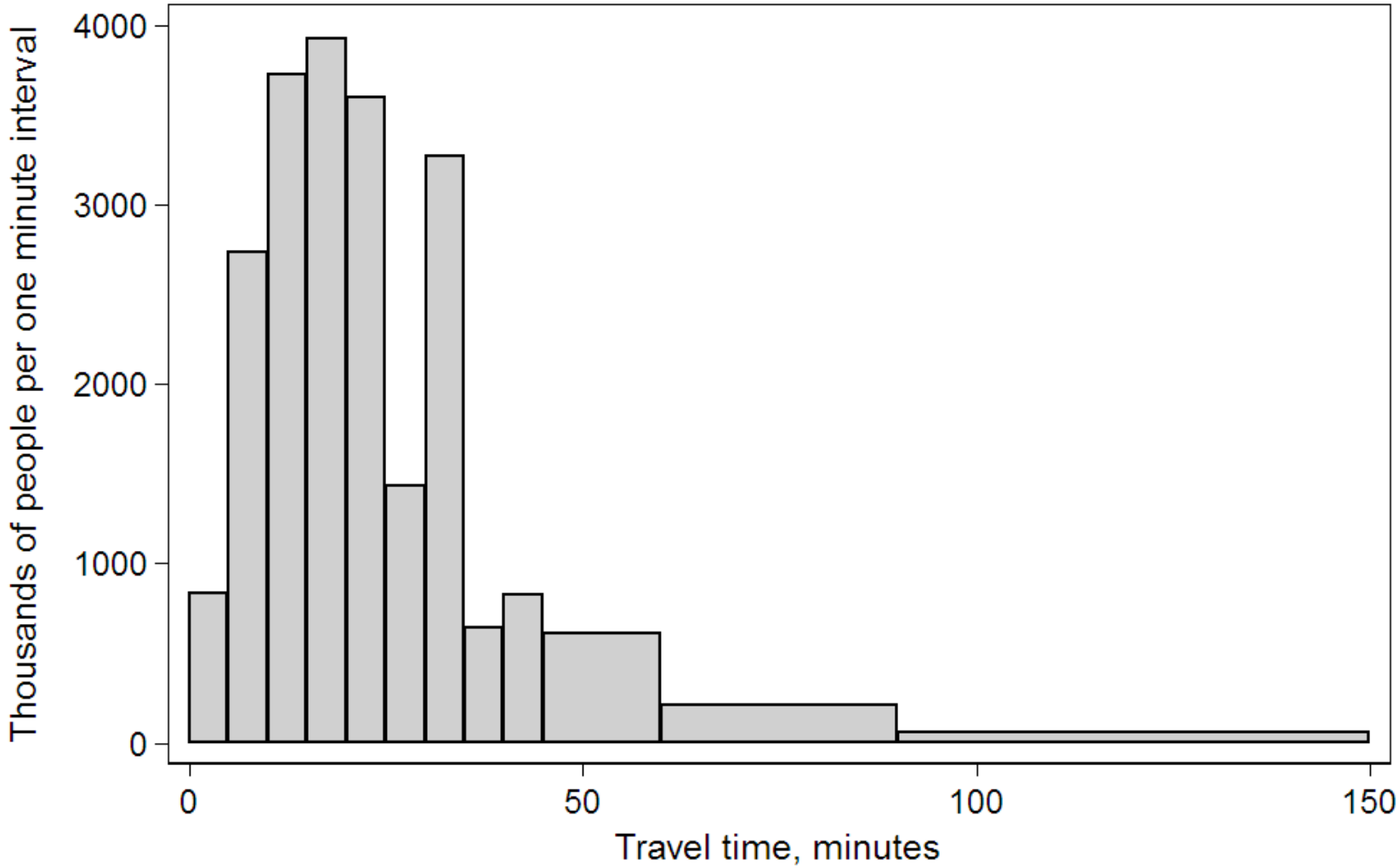
Example

- The U.S. Census Bureau found that there were 124 million people who worked outside of their homes.
- Using their data on the time occupied by travel to work, the table below shows the absolute number of people who responded with travel times "at least 30 but less than 35 minutes" is higher than the numbers for the categories above and below it.
- This is likely due to people rounding their reported journey time.
- The problem of reporting values as somewhat arbitrarily rounded numbers is a common phenomenon when collecting data from people.

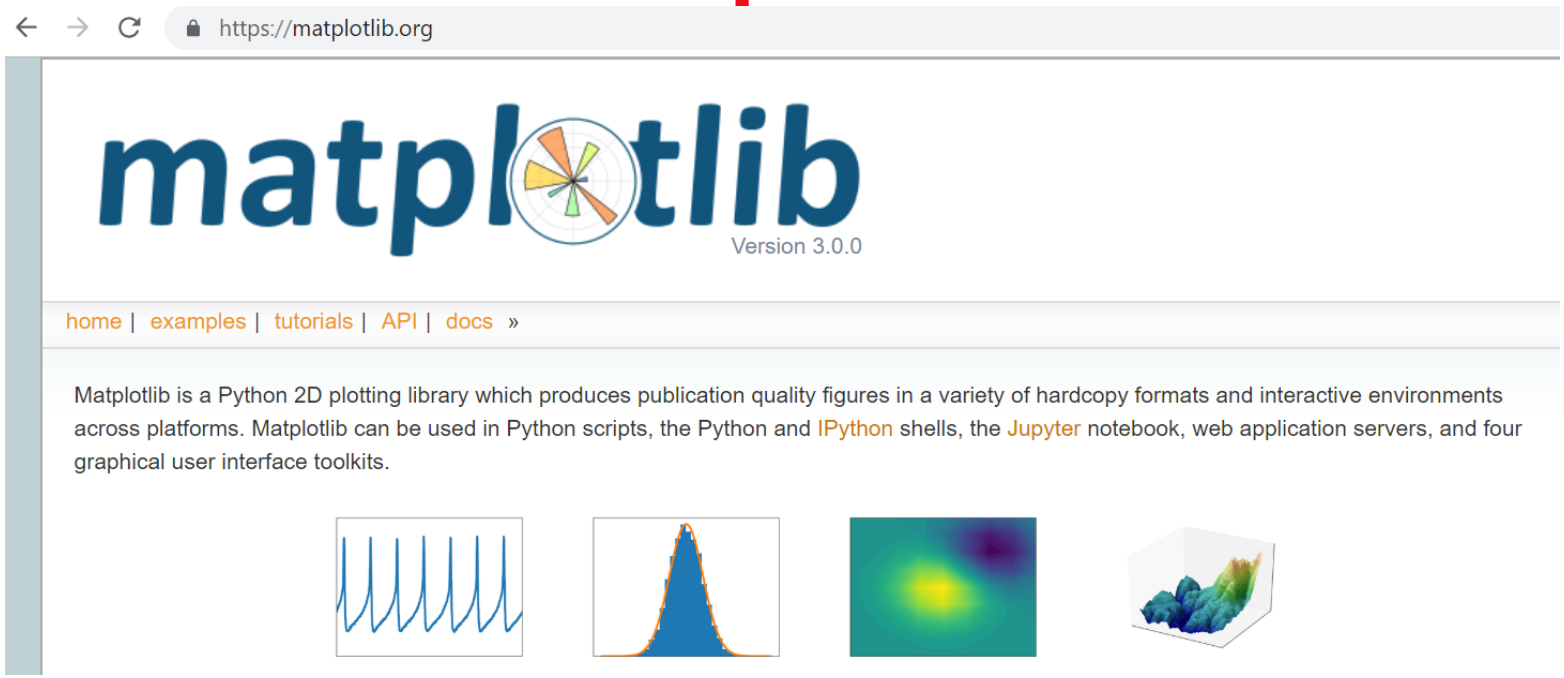
Example - Data

Data by absolute numbers			
Interval	Width	Quantity	Quantity/width
0	5	4180	836
5	5	13687	2737
10	5	18618	3723
15	5	19634	3926
20	5	17981	3596
25	5	7190	1438
30	5	16369	3273
35	5	3212	642
40	5	4122	824
45	15	9200	613
60	30	6461	215
90	60	3435	57

Example - Histogram



Matplotlib



- Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms.

https://matplotlib.org/

Matplotlib: Simple line Graph

Inline command only for Jupyter Notebook to display the graph

```
1  %matplotlib inline
2  #import library required
3  import matplotlib.pyplot as plt
4  plt.style.use('seaborn-whitegrid') #set t he stype to use
5  import numpy as np
6
7  fig=plt.figure()
8  ax=plt.axes()
9  x=np.linspace(0.0, 10.0, num=20)
10 print(x)
11 y=2*x+5
12 ax.plot(x,y)
13 plt.show()
```

Import matplotlib for used

Set the style of the graph

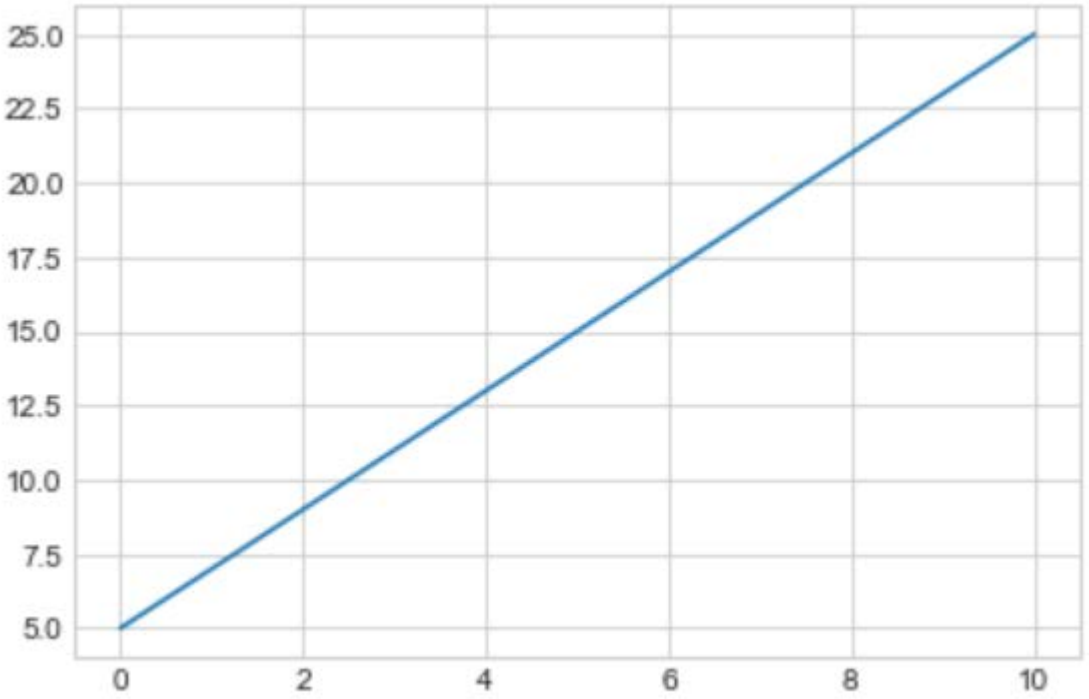
Start figure instance

Create a ax

Plot at axes ax with x and y

Show the graph

Matplotlib: Simple Line Graph



- A simple line graph displayed

Matplotlib: Color

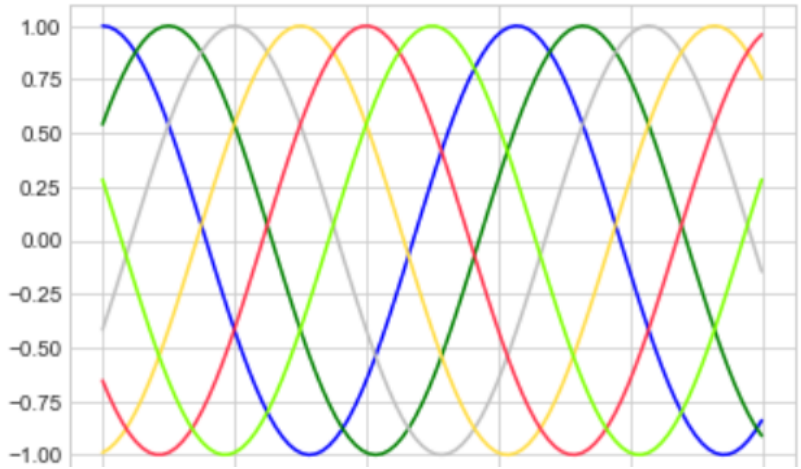
- In almost all places in matplotlib where a color can be specified by the user it can be provided as:
  - an RGB or RGBA tuple of float values in [0, 1] (e.g., (0.1, 0.2, 0.5) or (0.1, 0.2, 0.5, 0.3))
  - a hex RGB or RGBA string (e.g., '#0F0F0F' or '#0F0F0F0F')
  - a string representation of a float value in [0, 1] inclusive for gray level (e.g., '0.5')
  - one of {'b', 'g', 'r', 'c', 'm', 'y', 'k', 'w'}
  - a X11/CSS4 color name
  - a name from the xkcd color survey prefixed with 'xkcd:' (e.g., 'xkcd:sky blue')
  - one of {'C0', 'C1', 'C2', 'C3', 'C4', 'C5', 'C6', 'C7', 'C8', 'C9'}
  - one of {'tab:blue', 'tab:orange', 'tab:green', 'tab:red', 'tab:purple', 'tab:brown', 'tab:pink', 'tab:gray', 'tab:olive', 'tab:cyan'} which are the Tableau Colors from the ‘T10’ categorical palette (which is the default color cycle).

Matplotlib: Color

```
#we probably want to specify the color of a particular plot rather than leave it to the system.
fig=plt.figure()
ax=plt.axes()
x=np.linspace(0.0, 10.0, num=100)

ax.plot(x,np.cos(x),color='blue')   #blue
ax.plot(x,np.cos(x-1),color='g')    #short  color code "rgbcmyk
ax.plot(x,np.cos(x-2),color='0.75') #grey scale from 0 to 1
ax.plot(x,np.cos(x-3),color='#FFDD44')  #Hex code(RRGGBB from OO to FF)
ax.plot(x,np.cos(x-4),color=(1.0,0.2,0.3)) #RGB tuple, values 0 to 1
ax.plot(x,np.cos(x-5),color='chartreuse')  #all HTML color names are supported
```

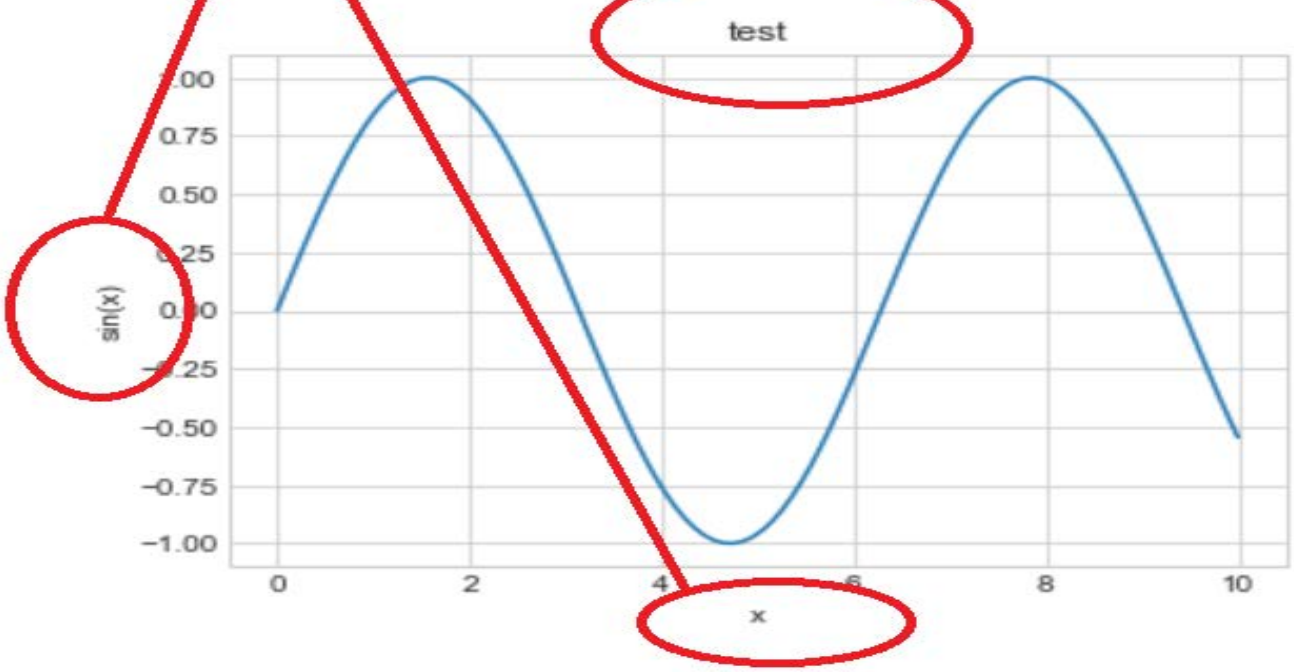
```
[<matplotlib.lines.Line2D at 0x23f4ca9e358>]
```



Matplotlib: Customize axes

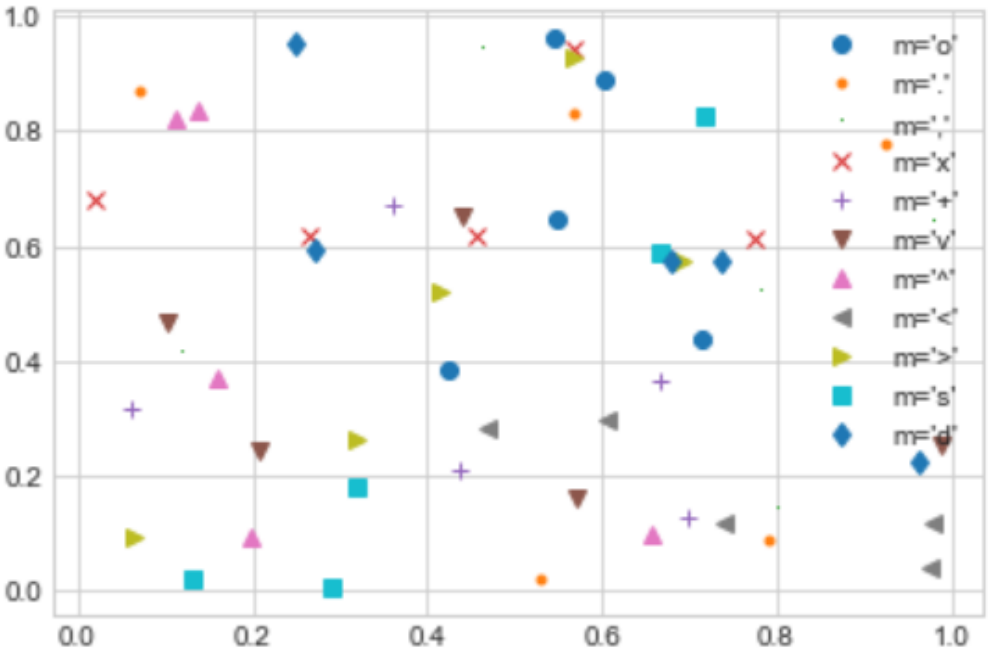
```
2]:  1  fig=plt.figure()
     2  ax=plt.axes()
     3  ax.plot(x, np.sin(x))
     4  ax.set_title('test')
     5  ax.set_xlabel('x')
     6  ax.set_ylabel('sin(x)')
```

```
2]: Text(0,0.5,'sin(x)')
```



Matplotlib: Scatter Plots

```
:  1  fig=plt.figure()
   2  rng=np.random.RandomState(0)
   3  ax=plt.axes()
   4  for m in ['o','.',',','x','+','v','^','<','>','s','d']:
   5      ax.plot(rng.rand(5),rng.rand(5),m,
   6      label="m='{0}'".format(m))
   7      ax.legend(numpoints=1)
```



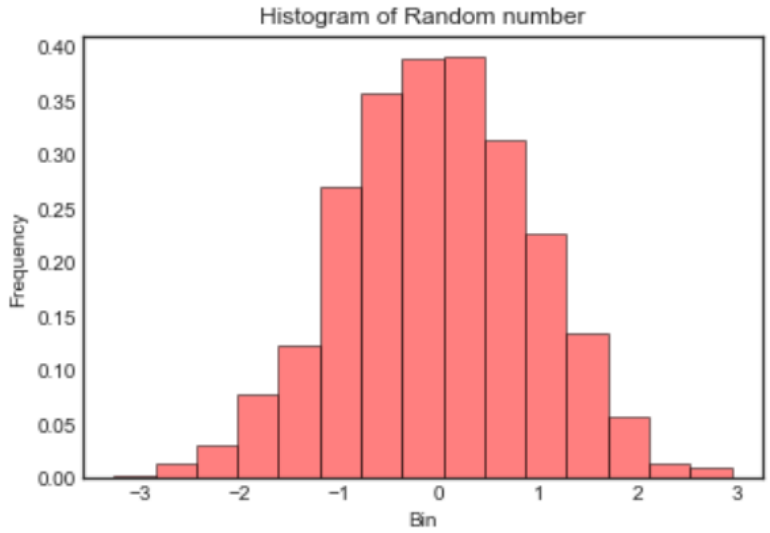
Matplotlib: Histogram

```
#more control over the look and feel of the histogram
#20 bin with and density =true is to have probability density=1

fig=plt.figure()
ax=plt.axes()
ax.hist(data,bins=15,density=True,alpha=0.5,histtype='bar',edgecolor='black',color='r')
ax.set_title('Histogram of Random number')
ax.set_xlabel('Bin')
ax.set_ylabel('Frequency')

```

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

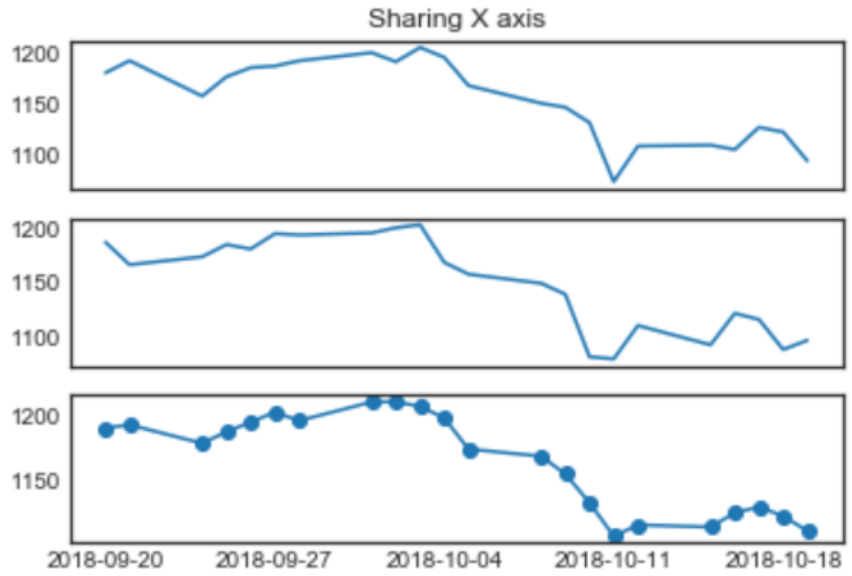


Matplotlib: Subplot

```
#Creates two subplots and unpacks the output array immediately with sharing same X axis
f, (ax1, ax2,ax3) = plt.subplots(3, 1, sharex=True)   #3 rows 1 column
ax1.plot(gdata.index,gdata['Open'])
ax2.plot(gdata.index,gdata['Close'])
ax1.set_title('Sharing X axis')
ax3.plot(gdata.index,gdata['High'],'-o')

```

```
[<matplotlib.lines.Line2D at 0x23f4c911ef0>]
```



Matplotlib: Subplot

```
:
 1  import matplotlib.pyplot as plt
 2
 3  # Make a fake dataset:
 4  product1 = [3, 12, 5, 18, 45]
 5  bars = ('A', 'B', 'C', 'D', 'E')
 6  x_pos = np.arange(len(bars))
 7  fig, ax = plt.subplots()
 8  # Create bars
 9  ax.bar(x_pos, product1,label='Product1')
10
11  # Create names on the x-axis
12  ax.set_xticks(x_pos)
13  ax.set_xticklabels(('A', 'B', 'C','D','E'))
14  ax.set_title('Bar plot of Product  1 vs different parameters')
15  ax.set_xlabel('Parameters')
16  ax.set_ylabel('value')
17
18  legend = ax.legend(loc='upper center', fontsize='x-large')
19
20  # Put a nicer background color on the legend.
21  legend.get_frame().set_facecolor('C0')
```

Matplotlib: Subplot