

## 02\_March\_Ass

April 11, 2023

```
[ ]: Q1: What is Matplotlib? Why is it used? Name five plots that can be plotted
      ↳ using the Pyplot module of
      Matplotlib.
```

```
[ ]: ANS -
```

```
[ ]: Matplotlib is a comprehensive library for creating static, animated , and
      ↳ interactive visualization in python
      Matplotlib makes easy thing and hard things possible.
      - Create publication quality plots.
      - Make interactive figures that an zoom , pan , updates.
      - Customize visual style and layout.
      - Embed in jupyterlab and graphical user interfaces.
```

```
[ ]: .Plot(x,y)
      .Scatter(x,y)
      .bar(x,y)
      .fill_between(x,y,y1)
      .step(x,y)
      .stackplot(x,y)
```

```
[ ]:
```

```
[ ]:
```

```
[ ]: Q2: What is a scatter plot? Use the following code to generate data for x and y.
      ↳ Using this generated data
      plot a scatter plot.
```

```
[ ]: import numpy as np
      np.random.seed(3)
      x = 3 + np.random.normal(0, 2, 50)
      y = 3 + np.random.normal(0, 2, len(x))
      Note: Also add title, xlabel, and ylabel to the plot.
```

```
[ ]: ANS -
```

```
[2]: import matplotlib.pyplot as plt
```

```
[3]: import numpy as np
```

```
[4]: np.random.seed(3)
```

```
[6]: x = 3 + np.random.normal(0, 2, 50)  
y = 3 + np.random.normal(0, 2, len(x))
```

```
[8]: x
```

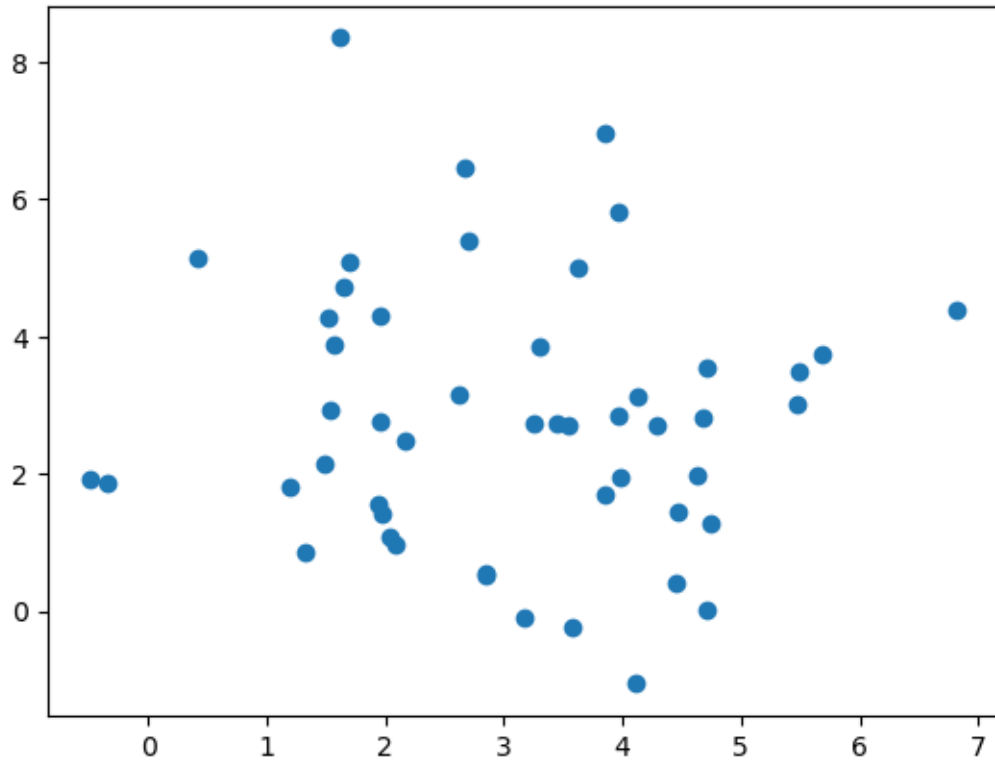
```
[8]: array([ 4.74857145,  0.41292674,  2.84051812,  4.12897104,  5.46694209,  
          3.29797279,  1.93883571,  1.53894671,  4.29012397,  3.62612075,  
          1.96670415,  2.62185667,  2.16760397,  4.44931532,  1.62007865,  
          3.97282895,  4.7030379 ,  3.97249865,  1.3315203 ,  5.68998491,  
          1.64357464,  3.85287015,  1.49333041, -0.4882205 ,  3.45150053,  
          3.57407033,  2.84511808,  3.55213699,  1.70317822,  1.52507033,  
          2.6638198 ,  6.81855362,  4.62962908,  1.96001649,  4.11742641,  
          2.04327068,  2.08547843,  4.71856802,  1.94947071, -0.35126927,  
          1.1870106 ,  3.17683041,  3.25601564,  5.48323304,  1.5679484 ,  
          4.46293147,  3.8519335 ,  2.70197246,  4.6716878 ,  3.98423781])
```

```
[9]: y
```

```
[9]: array([ 1.27538303,  5.14336786,  0.55819616,  3.11923087,  3.00488832,  
          3.84927144,  1.54913304,  2.93011323,  2.71875995,  4.99417673,  
          1.40817061,  3.14549106,  2.47751903,  0.40390672,  8.35222494,  
          2.85756195,  0.02668386,  5.81725392,  0.858829 ,  3.74173994,  
          4.72566419,  1.70313595,  2.13821989,  1.91945947,  2.74127798,  
         -0.24492234,  0.52872675,  2.71842712,  5.07790423,  4.26348835,  
          6.45883486,  4.38810454,  1.97774201,  2.75431319, -1.06078711,  
          1.07844978,  0.95928144,  3.54118685,  4.29565959,  1.87925316,  
          1.82299676, -0.09311165,  2.74447588,  3.49633605,  3.89156192,  
          1.43458191,  6.97697936,  5.39011667,  2.8095248 ,  1.94562444])
```

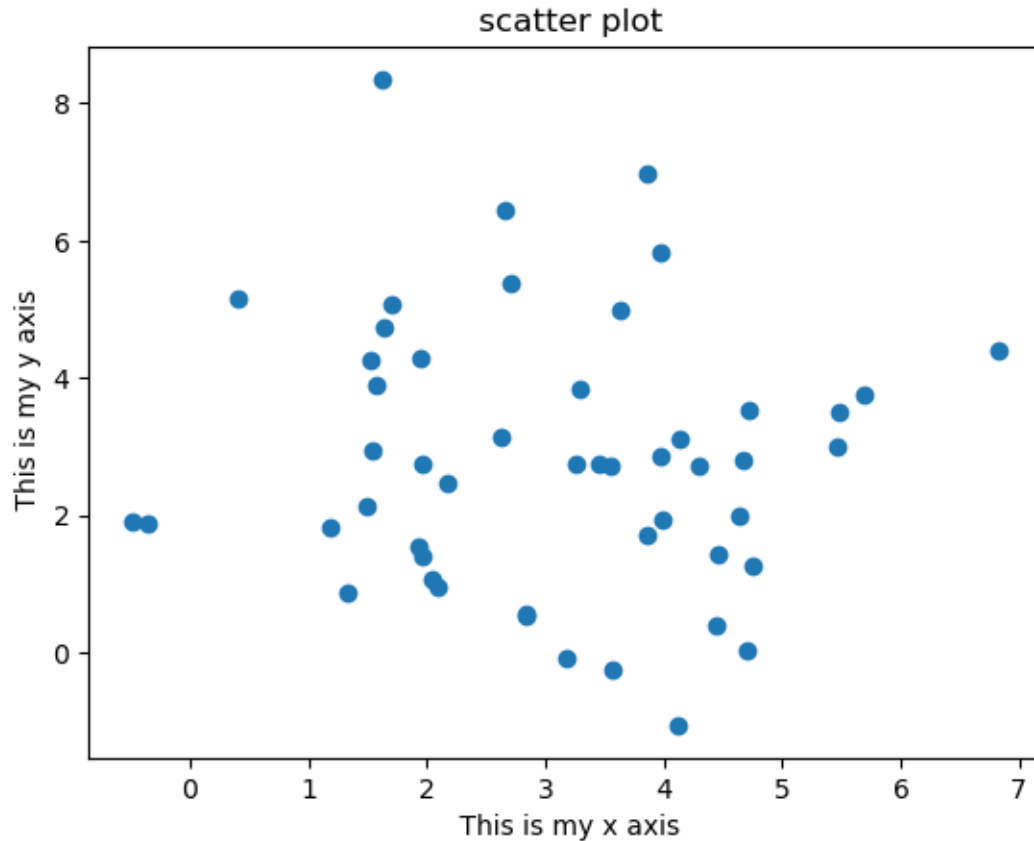
```
[10]: plt.scatter(x,y)
```

```
[10]: <matplotlib.collections.PathCollection at 0x7fe628e841f0>
```



```
[11]: plt.scatter(x,y)
plt.xlabel(" This is my x axis")
plt.ylabel(" This is my y axis")
plt.title(" scatter plot")
```

```
[11]: Text(0.5, 1.0, ' scatter plot')
```



[ ]:

[ ]:

[ ]: Q3: Why **is** the subplot() function used? Draw four line plots using the subplot() function.  
Use the following data:

[ ]: line 1: x = np.array([0, 1, 2, 3, 4, 5]) and y = np.array([0, 100, 200, 300, 400, 500])  
line 2: x = np.array([0, 1, 2, 3, 4, 5]) and y = np.array([50, 20, 40, 20, 60, 70])  
line 3: x = np.array([0, 1, 2, 3, 4, 5]) and y = np.array([10, 20, 30, 40, 50, 60])  
line 4: x = np.array([0, 1, 2, 3, 4, 5]) and y = np.array([200, 350, 250, 550, 450, 150])

[ ]: ANS -

```
[ ]: Matplotlib spyplot API has a convenience function called subplots() which acts
    ↪as a utility wrapper and
    help in creating common layouts of subplot, including the enclosing figure
    ↪object , in a single call.
    The two integer arguments to this function specify the number of rows and
    ↪columns of the subplot grid.
```

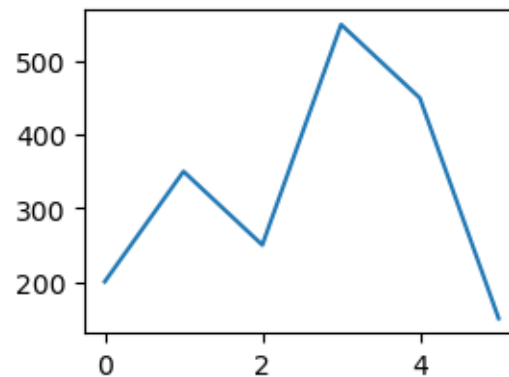
```
[12]: import matplotlib.pyplot as plt
```

```
[13]: import numpy as np
```

```
[15]: x = np.array([0, 1, 2, 3, 4, 5])
      y = np.array([0, 100, 200, 300, 400, 500])
```

```
[37]: plt.subplot(2,2,1)
      plt.plot(x,y)
```

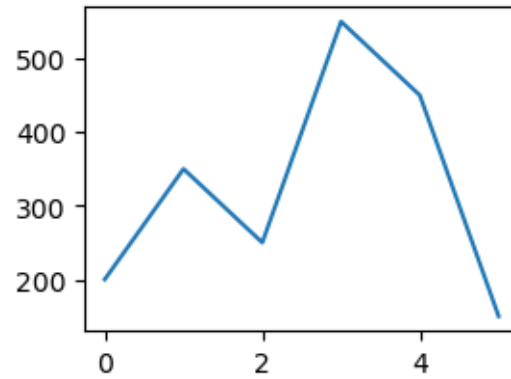
```
[37]: [<matplotlib.lines.Line2D at 0x7fe6207b1bd0>]
```



```
[17]: x = np.array([0, 1, 2, 3, 4, 5])
      y = np.array([50, 20, 40, 20, 60, 70])
```

```
[38]: plt.subplot(2,2,2)
      plt.plot(x,y)
```

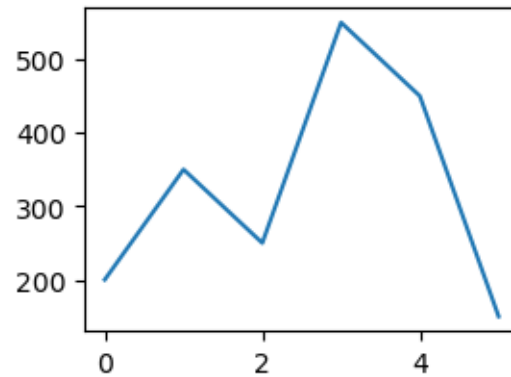
```
[38]: [<matplotlib.lines.Line2D at 0x7fe620863700>]
```



```
[20]: x = np.array([0, 1, 2, 3, 4, 5])
      y = np.array([10, 20, 30, 40, 50, 60])
```

```
[39]: plt.subplot(2,2,3)
      plt.plot(x,y)
```

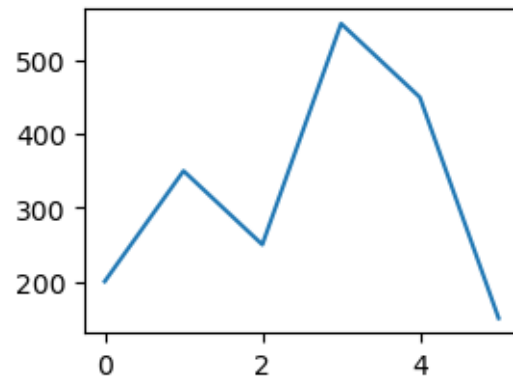
```
[39]: [<matplotlib.lines.Line2D at 0x7fe620915c00>]
```



```
[22]: x = np.array([0, 1, 2, 3, 4, 5])
      y = np.array([200, 350, 250, 550, 450, 150])
```

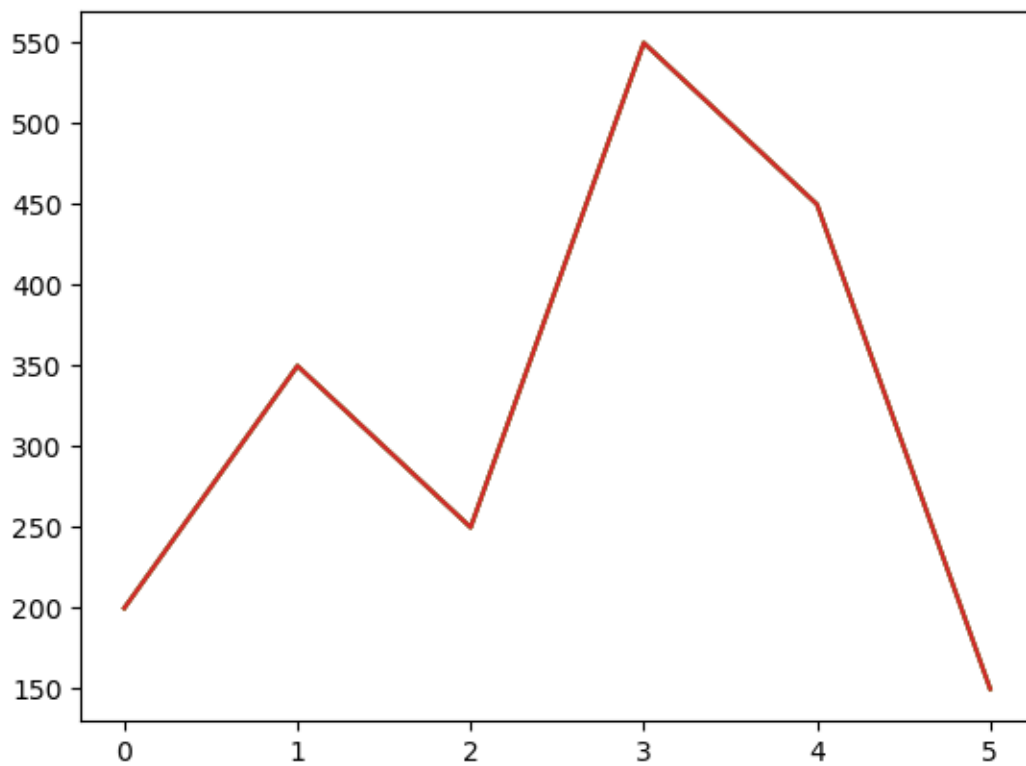
```
[40]: plt.subplot(2,2,4)
      plt.plot(x,y)
```

```
[40]: [<matplotlib.lines.Line2D at 0x7fe62087c280>]
```



```
[43]: plt.show()
```

```
[45]: plt.plot(x , y , label = "line 1")  
plt.plot(x , y , label = "line 2")  
plt.plot(x , y , label = "line 3")  
plt.plot(x , y , label = "line 4")  
plt.show()
```



[ ]:

[ ]:

[ ]: Q4: What **is** a bar plot? Why **is** it used? Using the following data plot a bar plot and a horizontal bar plot.

```
[ ]: company = np.array(["Apple", "Microsoft", "Google", "AMD"])
profit = np.array([3000, 8000, 1000, 10000])
```

[ ]: ANS -

[ ]: A bar plot **is** used when you want to show a distribution of data points **or** perform a comparison of metric values across different subgroups of your data, **from a** bar chart , we can see which group are highest **or** most common , and how other groups compare against the others.

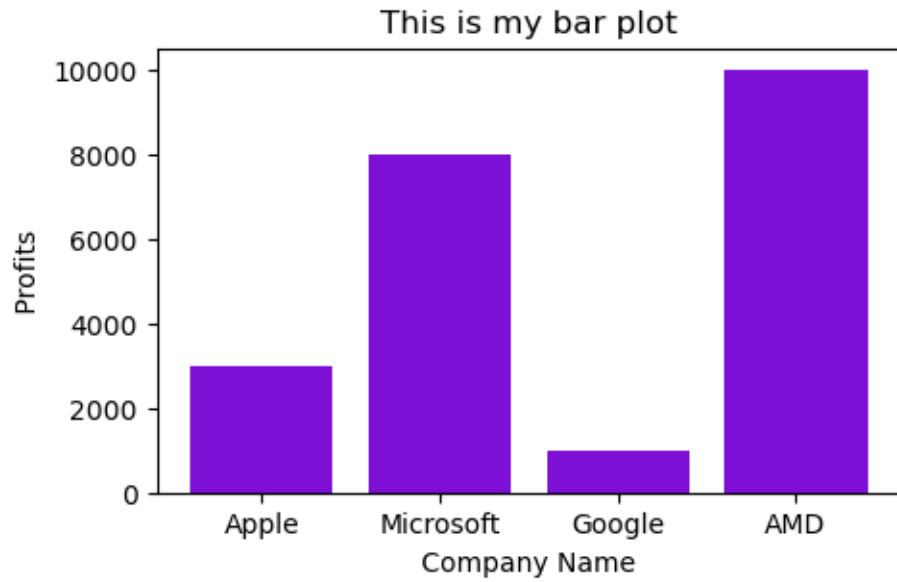
```
[46]: import matplotlib.pyplot as plt
```

```
[47]: x = np.array(["Apple", "Microsoft", "Google", "AMD"])
y = np.array([3000, 8000, 1000, 10000])
```

```
[50]: plt.figure(figsize=(5,3))
plt.bar(x,y,color = '#7E10D5')
plt.xlabel(" Company Name ")
plt.ylabel(" Profits ")
plt.title("This is my bar plot")
```

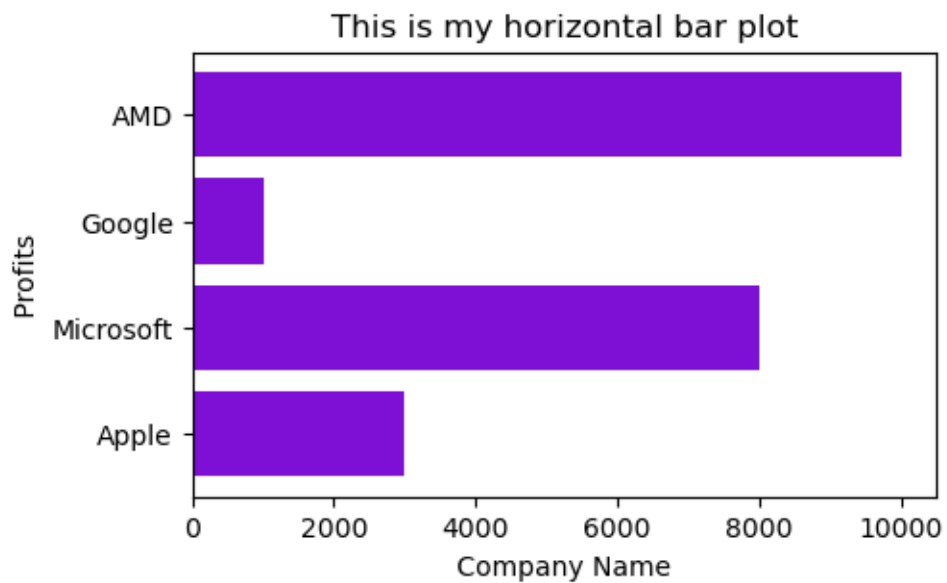
```
[50]: Text(0.5, 1.0, 'This is my bar plot')
```





```
[52]: plt.figure(figsize=(5,3))
plt.barh(x,y,color = '#7E10D5')
plt.xlabel(" Company Name ")
plt.ylabel(" Profits ")
plt.title("This is my horizontal bar plot")
```

```
[52]: Text(0.5, 1.0, 'This is my horizontal bar plot')
```



```
[ ]:
```

```
[ ]:
```

```
[ ]: Q5: What is a box plot? Why is it used? Using the following data plot a box
    plot.
```

```
[ ]: box1 = np.random.normal(100, 10, 200)
    box2 = np.random.normal(90, 20, 200)
```

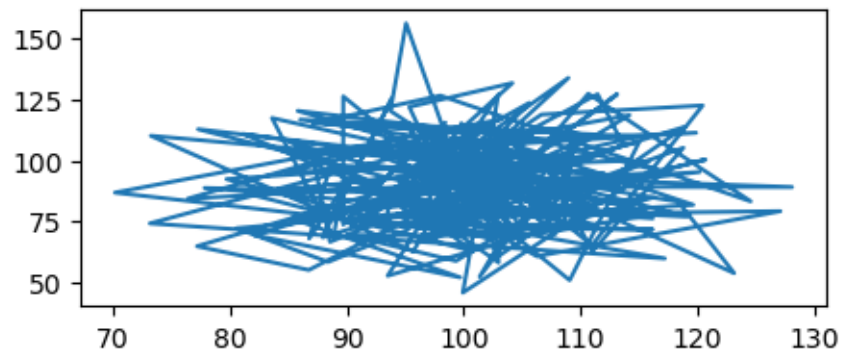
```
[ ]: ANS -
```

```
[ ]: Box plot are used to show distributions of numeric data values, especially when
    you want to compare
    them between multiple groups. They are built to provide high-level information
    at a glance, offering general
    information about a group of datas symmetry , skew , variance and outliers.
```

```
[53]: x = np.random.normal(100, 10, 200)
    y = np.random.normal(90, 20, 200)
```

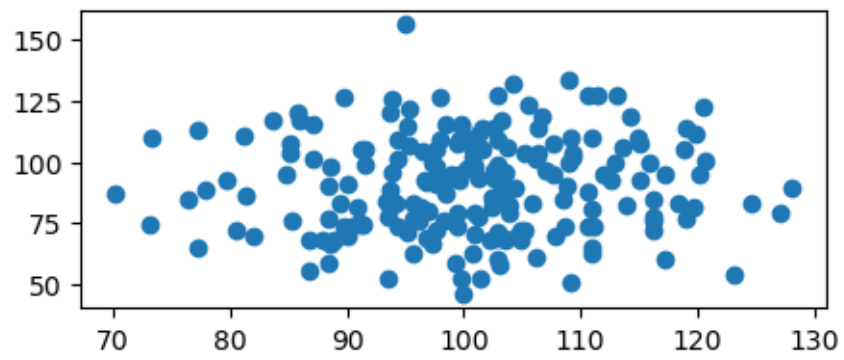
```
[56]: plt.figure(figsize=(5,2))
    plt.plot(x,y)
```

```
[56]: [<matplotlib.lines.Line2D at 0x7fe61fefafb0>]
```



```
[57]: plt.figure(figsize=(5,2))
    plt.scatter(x,y)
```

```
[57]: <matplotlib.collections.PathCollection at 0x7fe620077f10>
```



[ ]: