

matplotlib

September 24, 2023

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

matplotlib-matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in python and used to create 2D plots of array. It allows users to create numerous and diverse plot types.

plots that can be plotted using pyplot module of matplotlib:

- 1.scatterplot
- 2.Barplot
- 3.Histogramplot
- 4.lineplot
- 5.Areaplot
- 6.pieplot

```
[ ]: 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.
```

scatterplot-A scatter plot is a diagram where each value in the data set is represented by a dot and it is a type of data visualization that shows the relationship between different variables.

```
[1]: 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))
```

```
[2]: x
```

```
[2]: array([ 6.57725695,  3.8730197 ,  3.19299494, -0.72698541,  2.44522359,
            2.29048204,  2.83451704,  1.74599865,  2.91236366,  2.04556394,
            0.37227049,  4.76924476,  4.76263608,  6.41914613,  3.10006728,
            2.19064517,  1.9092801 , -0.09295463,  4.96473487,  0.79786474,
            0.62990695,  2.5887002 ,  5.97229671,  3.47343253,  0.95242972,
            1.5740136 ,  4.25048993,  2.67897327,  1.4623273 ,  2.53993856,
```

```

4.49011253, 6.95222157, 0.51175334, 1.74716618, 1.39246781,
-1.83816635, 1.15241596, 0.95224848, 5.24795592, 2.73617153,
-0.24657089, 4.2933509 , 2.28745848, -0.48628207, 1.80670072,
1.82281124, 1.2522354 , 3.05942763, -1.49651554, 2.46447627])

```

```
[3]: y
```

```

[3]: array([ 5.02636688,  4.70559568,  5.216375 ,  5.23878131,  5.97508626,
  0.76339863,  4.69166681, -0.72177906,  1.79422979, -0.82894409,
  5.09629502,  5.66747564,  2.60517064,  6.54929006,  1.65054498,
  3.30123373,  3.30589141,  0.87160945,  3.87589322,  6.87795692,
  0.95013825,  4.79867689,  2.6909863 ,  6.53925461,  3.9675767 ,
  4.3524328 ,  4.28632656,  3.49817341,  0.20847299,  5.78332582,
  0.25866197,  3.47712638,  4.22815418,  1.32417545,  3.29012643,
  5.33576457,  2.95179106,  1.22268516, -2.8314755 ,  1.05631899,
  1.81784252,  1.96716526,  1.08000764,  3.75459047,  1.85058316,
  2.78109133,  4.3581432 ,  1.28912566,  2.39958785,  7.31629868])

```

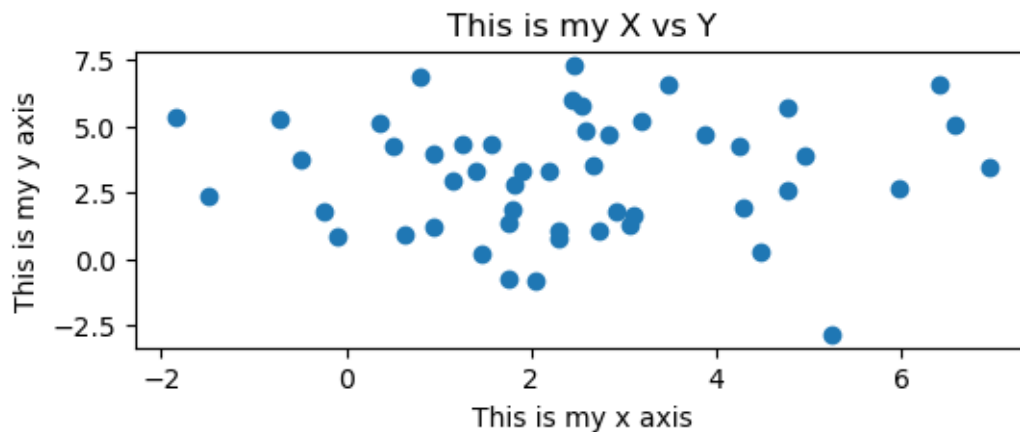
```
[6]: import matplotlib.pyplot as plt
```

```

[9]: plt.figure(figsize = (6,2))
plt.xlabel("This is my x axis")
plt.ylabel("This is my y axis")
plt.title("This is my X vs Y")
plt.scatter(x,y)

```

```
[9]: <matplotlib.collections.PathCollection at 0x7fee24b20ca0>
```



```

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

```

subplots-subplots method provides a way to plot multiple plots on a single `figure`.

```
[27]: import numpy as np
x1 = np.array([0, 1, 2, 3, 4, 5])
y1 = np.array([0, 100, 200, 300, 400, 500])
x2 = np.array([0, 1, 2, 3, 4, 5])
y2 = np.array([50, 20, 40, 20, 60, 70])
x3 = np.array([0, 1, 2, 3, 4, 5])
y3 = np.array([10, 20, 30, 40, 50, 60])
x4 = np.array([0, 1, 2, 3, 4, 5])
y4 = np.array([200, 350, 250, 550, 450, 150])
```

```
[28]: x1
```

```
[28]: array([0, 1, 2, 3, 4, 5])
```

```
[29]: y1
```

```
[29]: array([ 0, 100, 200, 300, 400, 500])
```

```
[30]: x2
```

```
[30]: array([0, 1, 2, 3, 4, 5])
```

```
[31]: y2
```

```
[31]: array([50, 20, 40, 20, 60, 70])
```

```
[32]: x3
```

```
[32]: array([0, 1, 2, 3, 4, 5])
```

```
[33]: y3
```

```
[33]: array([10, 20, 30, 40, 50, 60])
```

```
[34]: x4
```

```
[34]: array([0, 1, 2, 3, 4, 5])
```

```
[35]: y4
```

```
[35]: array([200, 350, 250, 550, 450, 150])
```

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

```
[82]: fig = plt.figure()
```

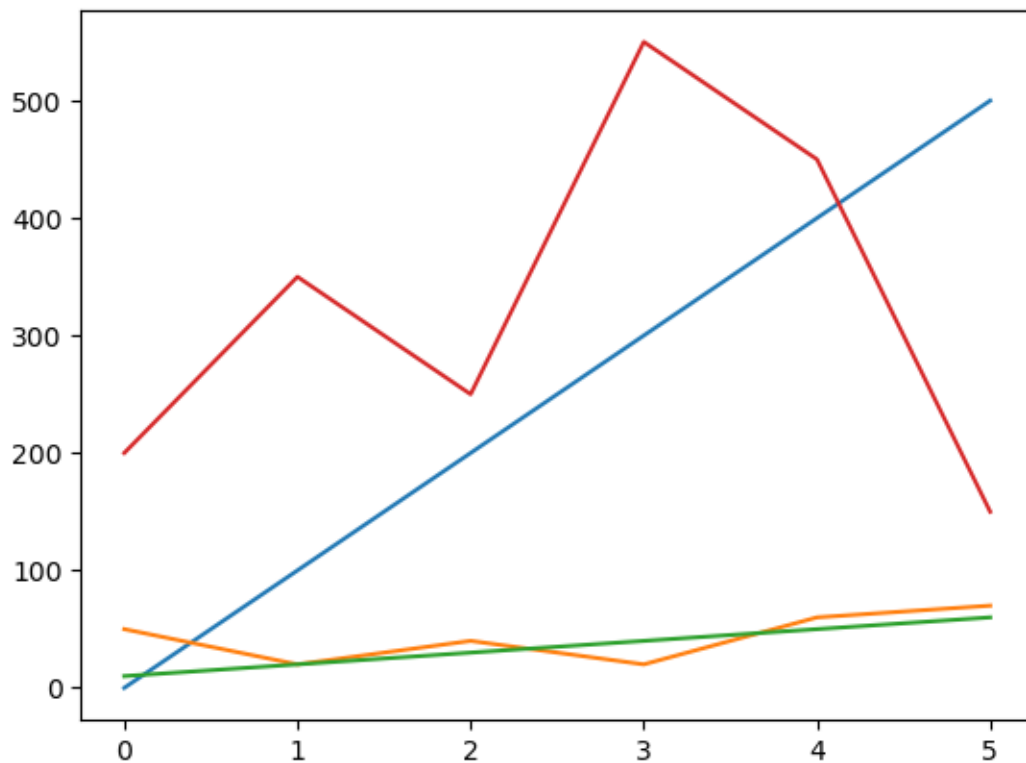
<Figure size 640x480 with 0 Axes>

```
[88]: fig, (ax) == fig.add_subplot(122)
```

```
[88]: (<Figure size 640x480 with 3 Axes>, False)
```

```
[93]: plt.plot(x1,y1)
plt.plot(x2,y2)
plt.plot(x3,y3)
plt.plot(x4,y4)
```

```
[93]: [<matplotlib.lines.Line2D at 0x7fee21a763b0>]
```



```
[ ]:
```

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

bar plot - It shows the relationship between a numeric **and** a categorical variable.
Bar plots are a **type** of data visualization used to represent data **in** the form of rectangular bars.

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

```
[41]: company
```

```
[41]: array(['Apple', 'Microsoft', 'Google', 'AMD'], dtype='<U9')
```

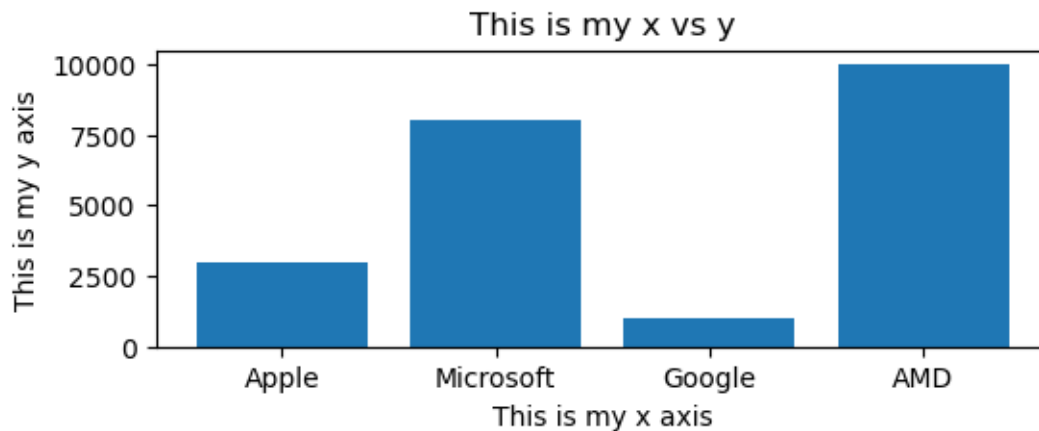
```
[42]: profit
```

```
[42]: array([ 3000,  8000,  1000, 10000])
```

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

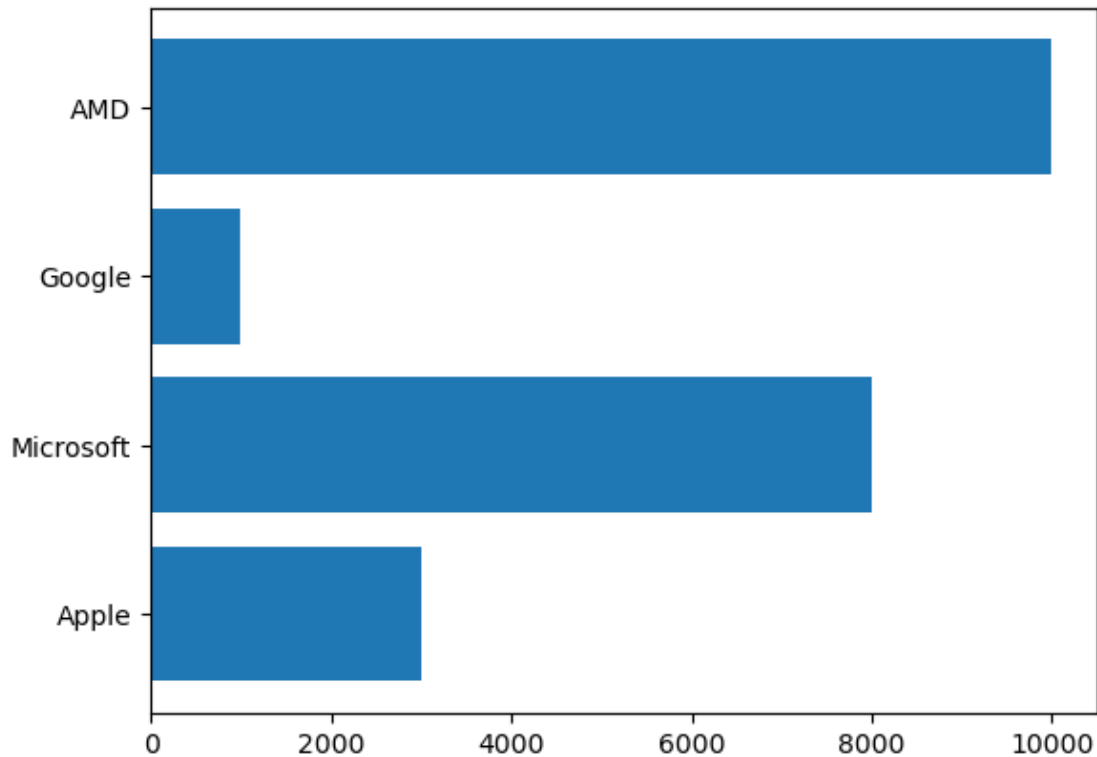
```
[45]: plt.figure(figsize = (6,2))
plt.xlabel("This is my x axis")
plt.ylabel("This is my y axis")
plt.title("This is my x vs y")
plt.bar(company,profit)
```

```
[45]: <BarContainer object of 4 artists>
```



```
[46]: plt.barh(company,profit)
```

```
[46]: <BarContainer object of 4 artists>
```



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

box plot- Box plots provide a quick visual summary of the variability of values in a dataset. They show the median, upper and lower quartiles, minimum and maximum values, and any outliers in the dataset.

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

```
[48]: box1
```

```
[48]: array([108.74285723,  87.06463368,  99.20259062, 105.64485518,
          112.33471044, 101.48986395,  94.69417856,  92.69473356,
          106.45061985, 103.13060374,  94.83352075,  98.10928334,
          95.83801985, 107.24657658,  93.10039323, 104.86414475,
          108.5151895 , 104.86249326,  91.65760149, 113.44992457,
          93.21787321, 104.26435074,  92.46665206,  82.55889749,
          102.25750266, 102.87035165,  99.22559039, 102.76068497,
          93.51589112,  92.62535163,  98.31909901, 119.09276809,
          108.14814541,  94.80008246, 105.58713205,  95.2163534 ,
          95.42739213, 108.59284008,  94.74735355,  83.24365366,
```

```

90.93505299, 100.88415206, 101.28007821, 112.41616518,
92.83974202, 107.31465736, 104.2596675 , 98.50986228,
108.35843902, 104.92118903, 91.37691513, 110.7168393 ,
87.7909808 , 100.59615433, 100.02444162, 104.24635721,
92.7456652 , 99.65056614, 98.59379973, 109.97088367,
92.04085307, 100.72745529, 97.38759515, 87.01953362,
126.76112468, 99.28780973, 85.13341929, 114.08626959,
89.29414501, 103.70869972, 108.62832095, 93.51567977,
95.69109945, 94.59729736, 98.7063899 , 83.7753883 ,
87.64363376, 98.59213558, 110.38952117, 106.31744177,
117.29417429, 106.94052272, 94.88871007, 98.77156594,
79.69606446, 90.3922489 , 89.79640721, 102.70593425,
106.47829797, 94.39626581, 94.1149838 , 84.53444177,
98.72237942, 102.48168027, 104.45780959, 92.17290957,
119.8848968 , 111.95058336, 99.04762401, 94.72812221,
96.78415307, 101.51130372, 99.81372284, 104.83528787,
107.68965158, 113.66242844, 111.47264793, 98.89770845,
103.88250414, 96.12872819, 94.12779688, 119.10826851,
95.4015385 , 119.90737812, 96.50964607, 102.52825091,
110.89409548, 100.23922022, 103.93125281, 97.586152 ,
95.24475142, 98.34222977, 93.50282579, 116.31382954,
98.32301397, 117.22669198, 73.14891322, 100.18420795,
105.61951672, 97.06178762, 110.94653079, 106.39692355,
97.2539988 , 104.35009258, 128.11878375, 102.51995127,
102.99502331, 95.60008689, 101.33497041, 87.10738806,
98.01709743, 124.57587626, 110.67215551, 106.41420656,
111.03921665, 118.81754991, 105.93588115, 120.70878526,
110.69798358, 101.6651951 , 117.19476555, 76.40785262,
94.28651038, 102.65787047, 90.87909087, 98.43941597,
93.61209104, 93.45584788, 127.11926335, 106.27473889,
99.46052151, 113.15167224, 97.62662524, 108.85339675,
103.50815841, 116.26573605, 85.80128694, 107.65721072,
101.22249745, 88.42616624, 110.65420182, 91.27624339,
116.19238485, 105.13093008, 106.95483983, 100.80457388,
109.04528487, 81.34348828, 100.74727794, 93.71749193,
102.82660307, 99.52843093, 106.16577883, 91.62368553,
118.39151887, 123.15858597, 97.91716582, 99.85026814,
102.8755765 , 112.6408575 , 118.96900956, 87.94197209,
93.84891436, 89.37843877, 88.8721885 , 83.60703753,
103.62803486, 88.40963535, 115.03261953, 109.08318716])

```

[49]: box2

```

[49]: array([ 69.40580949,  69.39582765,  77.75231165, 117.99930992,
  73.00785281,  60.12900664,  89.01152229,  97.46779888,
  76.8548503 , 122.38861613,  94.80913561,  99.06075574,
  72.88934558,  89.20540323,  86.86962372,  44.55511771,

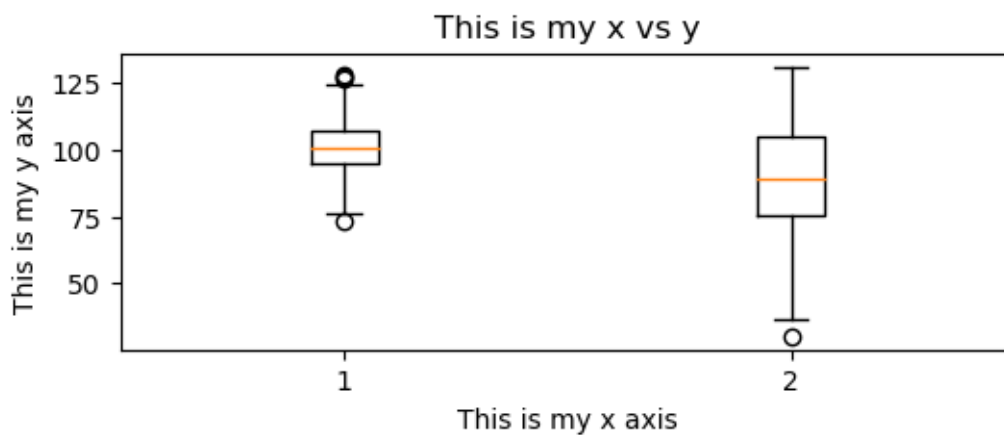
```

95.75314457, 54.10385929, 89.42428709, 60.52114346,
 130.39301284, 96.52524546, 107.22715228, 108.38016262,
 63.51022605, 44.3641537 , 83.40843694, 107.94299494,
 91.82069023, 105.70864494, 108.73265146, 60.15205798,
 95.75367447, 129.32949256, 78.5893541 , 49.41855749,
 85.36190894, 80.70662558, 98.36762045, 72.15187331,
 91.81433546, 45.65164778, 107.07926743, 121.73733086,
 115.95755228, 59.69568683, 96.38392352, 30.3205959 ,
 95.66371844, 88.71279656, 70.08411511, 96.87589037,
 92.7602931 , 108.79005708, 92.54617048, 94.70054026,
 51.09855476, 66.80554098, 80.48201226, 95.93689062,
 89.87402442, 120.01780714, 72.5968162 , 85.20735853,
 95.1016139 , 85.37005181, 99.9116079 , 78.58874352,
 118.4068461 , 83.60819718, 112.31890369, 89.39393266,
 119.83044693, 62.03767978, 100.33595108, 81.34847248,
 94.62801522, 113.85610331, 67.21806402, 63.55940531,
 70.03266243, 95.08775149, 52.26326149, 91.93187469,
 64.27541345, 67.12556347, 82.6164451 , 97.61195169,
 77.47170993, 80.15912214, 89.16311544, 84.54528928,
 36.4695727 , 81.39798786, 91.69928116, 111.95558975,
 130.92665553, 103.33975699, 91.58184385, 70.70473065,
 91.78106742, 105.57793805, 115.29289823, 72.38977332,
 94.72811191, 106.31208932, 127.21623338, 95.11180971,
 79.16992568, 76.20800686, 82.85118543, 76.96159592,
 106.53071696, 111.38611445, 104.49713645, 113.84372486,
 80.92462923, 97.60670117, 82.30673649, 90.87317377,
 114.49971485, 89.40529376, 52.70388405, 84.94368025,
 75.74300409, 59.8216576 , 74.19268622, 109.21249607,
 123.61821298, 80.21987916, 110.05071674, 113.56444203,
 66.80415464, 89.21274591, 89.11079658, 93.44771358,
 58.12498379, 83.01715522, 111.15642414, 115.24406398,
 126.62724162, 83.24981909, 127.39015113, 103.31810223,
 62.81597665, 105.23218564, 82.95439931, 100.38152528,
 87.94952115, 114.16477282, 95.13120319, 84.34989964,
 109.29931537, 95.12443553, 81.74087198, 115.54548724,
 81.83309515, 77.25730274, 79.20850776, 60.69055824,
 78.93585661, 127.21755384, 71.8343213 , 90.16837807,
 67.83633301, 77.77293696, 120.37384318, 107.92711472,
 77.794993 , 90.12440232, 73.47992076, 74.31589295,
 71.70355604, 72.05720446, 96.51898546, 101.93586334,
 99.76484985, 86.61126286, 62.83909599, 88.65779057,
 71.51412516, 107.62602265, 101.12885889, 104.93783061,
 83.03263367, 53.77961653, 109.15792787, 114.5264267 ,
 60.33366783, 92.75832088, 113.86505158, 68.48552901,
 125.35365552, 83.01935087, 68.49219779, 117.43316082,
 95.89064273, 76.93723677, 107.59575906, 50.88465743])


```
[104]: box = [box1,box2]
```

```
[105]: import matplotlib.pyplot as plt
plt.figure(figsize = (6,2))
plt.xlabel("This is my x axis")
plt.ylabel("This is my y axis")
plt.title("This is my x vs y")
plt.boxplot(box)
```

```
[105]: {'whiskers': [<matplotlib.lines.Line2D at 0x7fee218b2a10>,
<matplotlib.lines.Line2D at 0x7fee218b2cb0>,
<matplotlib.lines.Line2D at 0x7fee218b3c70>,
<matplotlib.lines.Line2D at 0x7fee218b3f10>],
'caps': [<matplotlib.lines.Line2D at 0x7fee218b2f50>,
<matplotlib.lines.Line2D at 0x7fee218b31f0>,
<matplotlib.lines.Line2D at 0x7fee216c01f0>,
<matplotlib.lines.Line2D at 0x7fee216c0490>],
'boxes': [<matplotlib.lines.Line2D at 0x7fee218b2770>,
<matplotlib.lines.Line2D at 0x7fee218b39d0>],
'medians': [<matplotlib.lines.Line2D at 0x7fee218b3490>,
<matplotlib.lines.Line2D at 0x7fee216c0730>],
'fliers': [<matplotlib.lines.Line2D at 0x7fee218b3730>,
<matplotlib.lines.Line2D at 0x7fee216c09d0>],
'means': []}
```



```
[ ]:
```