Exp No.3 Study of Scatter plot, Bar chart, Line chart for the given data in Python using matplotlib, seaborn and pandas

Roll no :- 22354 Name : Malhar Choure

Performed Date: 16/02/22

Submission Date: 19/03/22

Importing the libraries

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
sns.set()
```

1. Creating an Array

a.) array()

```
arr = np.array([1, 2, 3, 4, 5]).astype(int)
arr
```

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

b.) np.linspace()

```
arr = np.linspace(0, 10, 11).astype(int)
arr
```

```
array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10])
```

```
arr = np.linspace(0, 10).astype(float)
arr
```

```
array([ 0.        ,  0.20408163,  0.40816327,  0.6122449 ,
0.81632653,
        1.02040816,  1.2244898 ,  1.42857143,  1.63265306,
1.83673469,
        2.04081633,  2.24489796,  2.44897959,  2.65306122,
2.85714286,
        3.06122449,  3.26530612,  3.46938776,  3.67346939,
3.87755102,
        4.08163265,  4.28571429,  4.48979592,  4.69387755,
4.89795918,
        5.10204082,  5.30612245,  5.51020408,  5.71428571,
5.91836735,
        6.12244898,  6.32653061,  6.53061224,  6.73469388,
6.93877551,
        7.14285714,  7.34693878,  7.55102041,  7.75510204,
7.95918367,
        8.16326531,  8.36734694,  8.57142857,  8.7755102 ,
8.97959184,
        9.18367347,  9.3877551 ,  9.59183673,  9.79591837,
10.        ])
```

c.) np.logspace()

```
arr = np.logspace(0, 10, 11).astype(int)
arr
```

```
array([          1,          10,         100,        1000,
10000,
            100000,     1000000,    10000000,   100000000,
1000000000,
       10000000000])
```

```
arr = np.logspace(0, 10).astype(float)
arr
```

```
array([1.00000000e+00, 1.59985872e+00, 2.55954792e+00, 4.09491506e+00,
       6.55128557e+00, 1.04811313e+01, 1.67683294e+01, 2.68269580e+01,
       4.29193426e+01, 6.86648845e+01, 1.09854114e+02, 1.75751062e+02,
       2.81176870e+02, 4.49843267e+02, 7.19685673e+02, 1.15139540e+03,
       1.84206997e+03, 2.94705170e+03, 4.71486636e+03, 7.54312006e+03,
       1.20679264e+04, 1.93069773e+04, 3.08884360e+04, 4.94171336e+04,
       7.90604321e+04, 1.26485522e+05, 2.02358965e+05, 3.23745754e+05,
       5.17947468e+05, 8.28642773e+05, 1.32571137e+06, 2.12095089e+06,
       3.39322177e+06, 5.42867544e+06, 8.68511374e+06, 1.38949549e+07,
       2.22299648e+07, 3.55648031e+07, 5.68986603e+07, 9.10298178e+07,
       1.45634848e+08, 2.32995181e+08, 3.72759372e+08, 5.96362332e+08,
       9.54095476e+08, 1.52641797e+09, 2.44205309e+09, 3.90693994e+09,
       6.25055193e+09, 1.00000000e+10])
```

d.) np.arange()

```
arr = np.arange(0, 10, 1).astype(int)
arr
```

```
array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

e.) np.zeros()

```
np.zeros((5, 5))
```

```
array([[0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0.]])
```

f.) np.ones()

```
np.ones((5, 5))
```

```
array([[1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1.]])
```

g.) np.eye()

```
np.eye(5)
```

```
array([[1., 0., 0., 0., 0.],
       [0., 1., 0., 0., 0.],
       [0., 0., 1., 0., 0.],
       [0., 0., 0., 1., 0.],
       [0., 0., 0., 0., 1.]])
```

2. 2D Arrays

```
arr2D = np.array([[1, 2, 3],
                  [4, 5, 6]])
arr2D
```

```
array([[1, 2, 3],
       [4, 5, 6]])
```

```
np.matrix('1, 2; 3 4')
```

```
matrix([[1, 2],
        [3, 4]])
```

3. Array Operations

```
arr1 = np.array([[1, 2, 3],
                  [4, 5, 6]]).astype(int)
arr2 = np.array([[7, 8, 9],
                  [10, 11, 12]]).astype(int)
```

```
arr1 + arr2
```

```
array([[ 8, 10, 12],
       [14, 16, 18]])
```

```
arr2 - arr1
```

```
array([[6, 6, 6],
       [6, 6, 6]])
```

```
arr1 * arr2
```

```
array([[ 7, 16, 27],
       [40, 55, 72]])
```

```
arr2 / arr1
```

```
array([[7. , 4. , 3. ],
       [2.5, 2.2, 2. ]])
```

```
arr1.astype(complex)
```

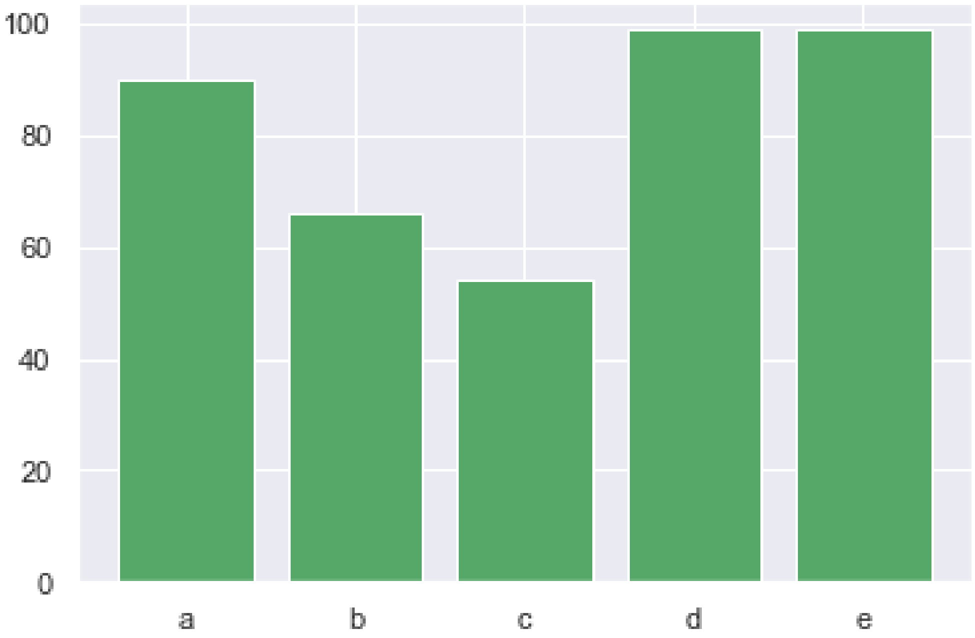
```
array([[1.+0.j, 2.+0.j, 3.+0.j],
       [4.+0.j, 5.+0.j, 6.+0.j]])
```

PLOTTING

BAR PLOT

```
name = np.array(['a', 'b', 'c', 'd', 'e'])
marks = np.array([90, 66, 54, 99, 99])

plt.bar(x=name, height=marks, color='g')
```

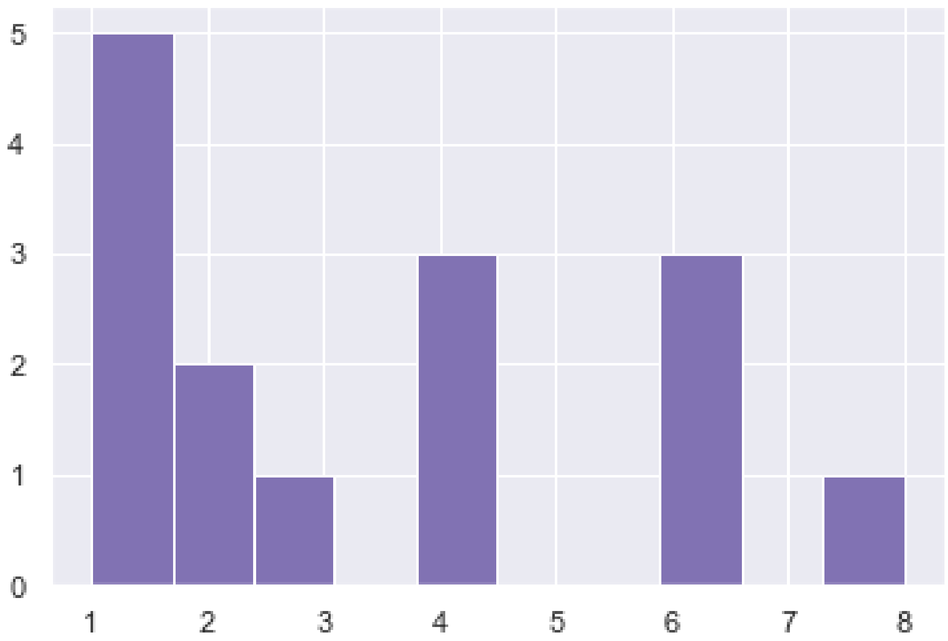
```
<BarContainer object of 5 artists>
```



HISTOGRAM PLOT

```
numRange2 = np.array([1, 1, 1, 1, 3, 4, 2, 2, 4, 4, 6 , 6, 6, 8, 1])

plt.hist(numRange2, color='m')
```

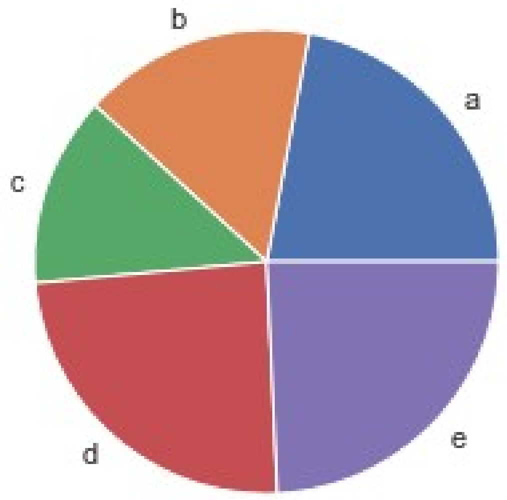
```
(array([5., 2., 1., 0., 3., 0., 0., 3., 0., 1.]),
 array([1. , 1.7, 2.4, 3.1, 3.8, 4.5, 5.2, 5.9, 6.6, 7.3, 8. ]),
 <BarContainer object of 10 artists>)
```



PIE PLOT

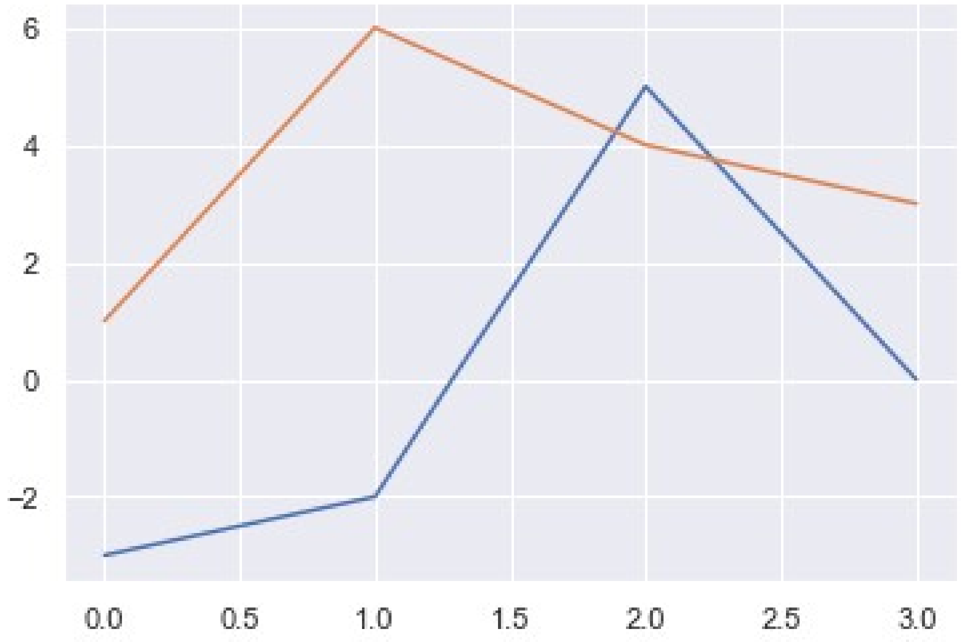
```
plt.pie(marks, labels=name, counterclock=True)
```

```
([<matplotlib.patches.Wedge at 0x7fa973581ee0>,
  <matplotlib.patches.Wedge at 0x7fa9a1a10400>,
  <matplotlib.patches.Wedge at 0x7fa9a1a10880>,
  <matplotlib.patches.Wedge at 0x7fa9a1a10d00>,
  <matplotlib.patches.Wedge at 0x7fa9a1a1d1c0>],
 [Text(0.8462673642109039, 0.7027314908779381, 'a'),
  Text(-0.349570590260835, 1.0429767027238392, 'b'),
  Text(-1.0429767354529953, 0.3495704926103071, 'c'),
  Text(-0.7220909402926393, -0.8298100228047937, 'd'),
  Text(0.7955759040521009, -0.7596439829891912, 'e')])
```



LINE PLOT

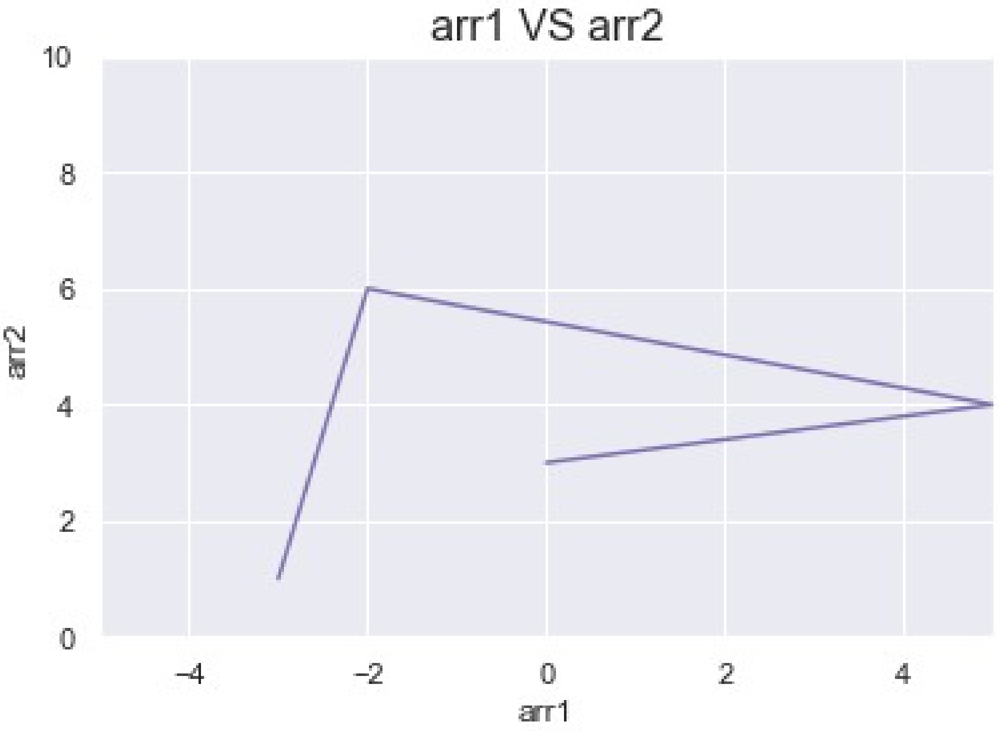
```
arr1 = np.array([-3, -2, 5, 0])
arr2 = np.array([1, 6, 4, 3])

plt.plot(arr1);
plt.plot(arr2);
```



Plot the arrays

```
plt.plot(arr1, arr2, c='m')
plt.xlim(-5, 5); plt.ylim(0, 10);
plt.xlabel('arr1', fontsize=12)
plt.ylabel('arr2', fontsize=12)
plt.title('arr1 VS arr2', fontsize=16)
```

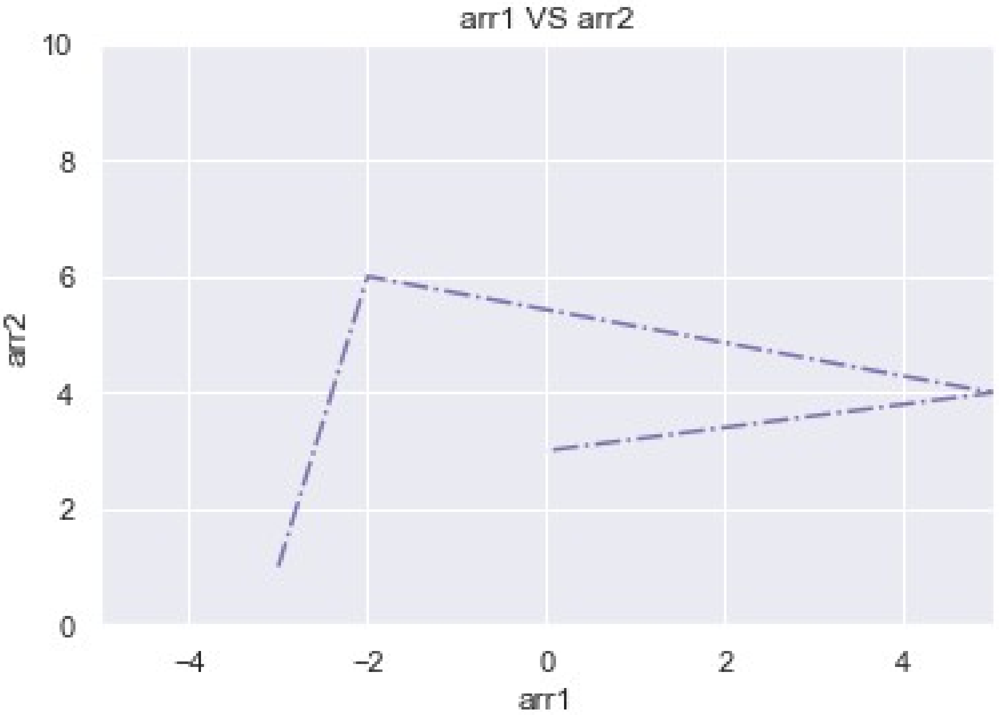
```
Text(0.5, 1.0, 'arr1 VS arr2')
```



Linestyle

```
plt.plot(arr1, arr2, c='m', linestyle='-.')
plt.xlim(-5, 5)
plt.ylim(0, 10)

plt.xlabel('arr1')
plt.ylabel('arr2')
plt.title('arr1 VS arr2')
```

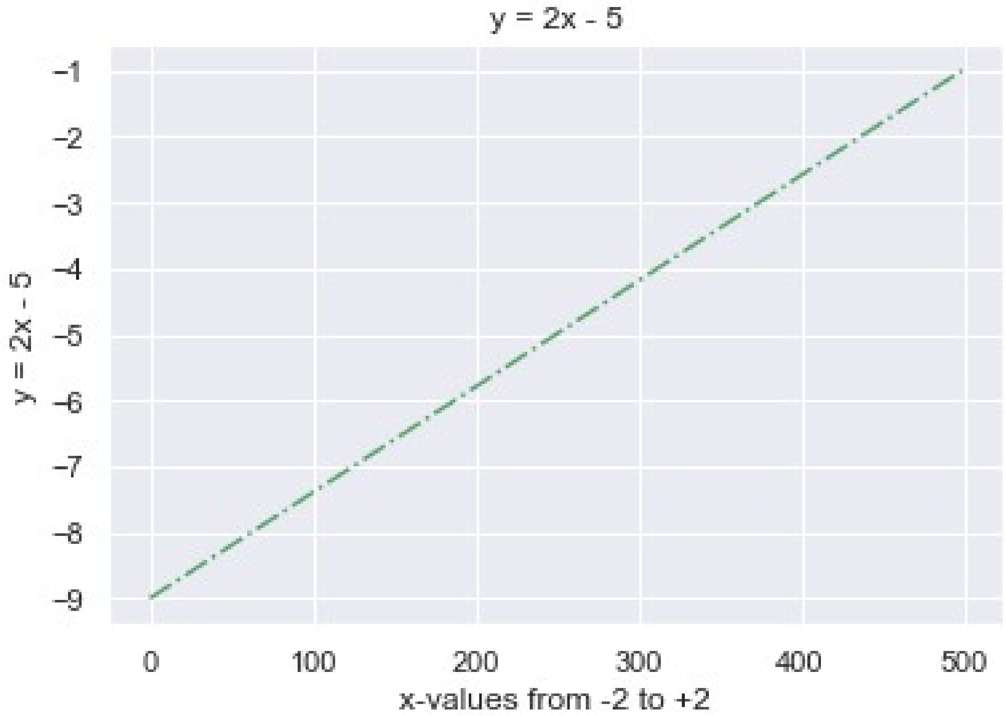
```
Text(0.5, 1.0, 'arr1 VS arr2')
```



Problem Statement:- Using numpy linspace function create an array, 'x' containing 500 floats ranging from -2 to +2. Create an 2nd Array, 'y' as y=2x-5

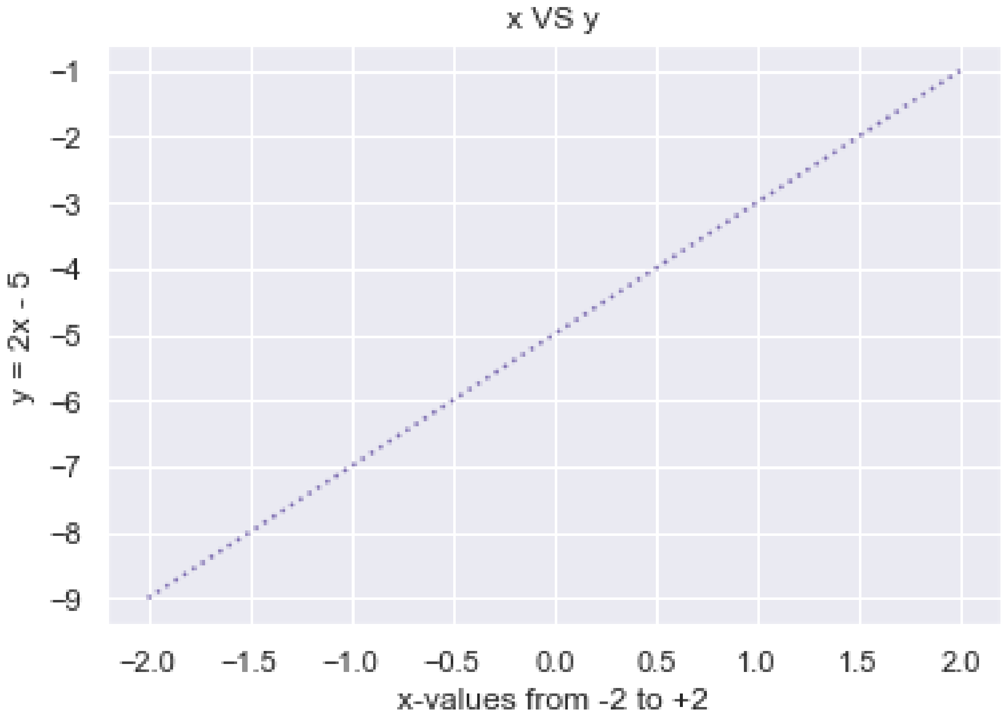
```
x = np.linspace(-2, 2, 500).astype(float)
y = (2 * x) - 5

plt.plot(y, c='g', linestyle='-.')
plt.xlabel('x-values from -2 to +2')
plt.ylabel('y = 2x - 5')
plt.title('y = 2x - 5')
```

```
Text(0.5, 1.0, 'y = 2x - 5')
```



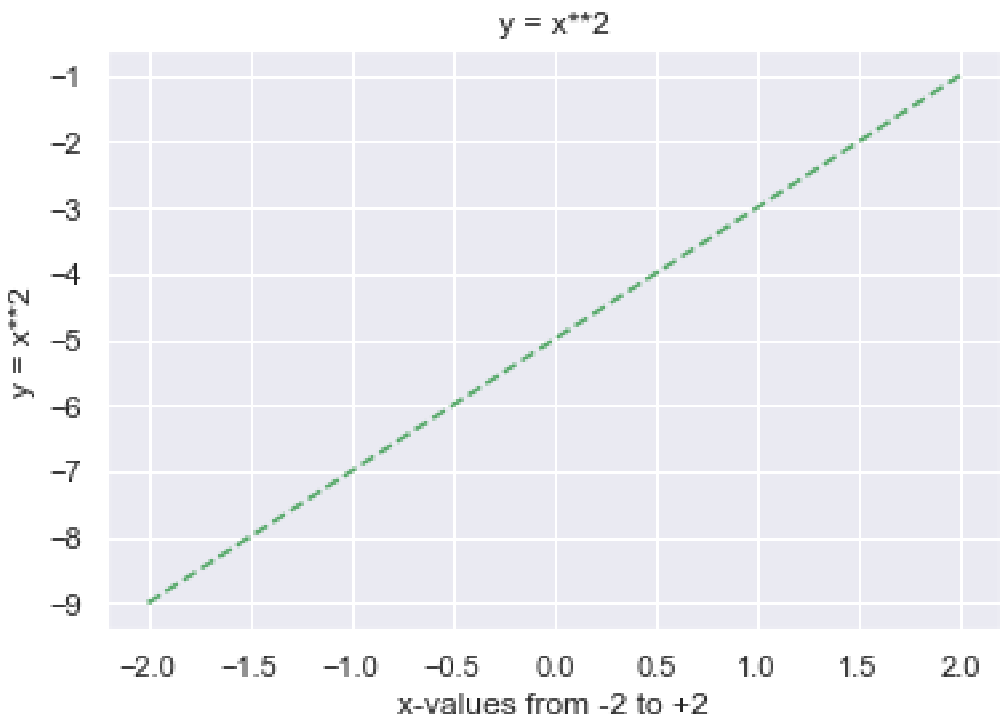
```
plt.plot(x, y, c='m', linestyle='dotted')
plt.xlabel('x-values from -2 to +2')
plt.ylabel('y = 2x - 5')
plt.title('x VS y')
```

```
Text(0.5, 1.0, 'x VS y')
```



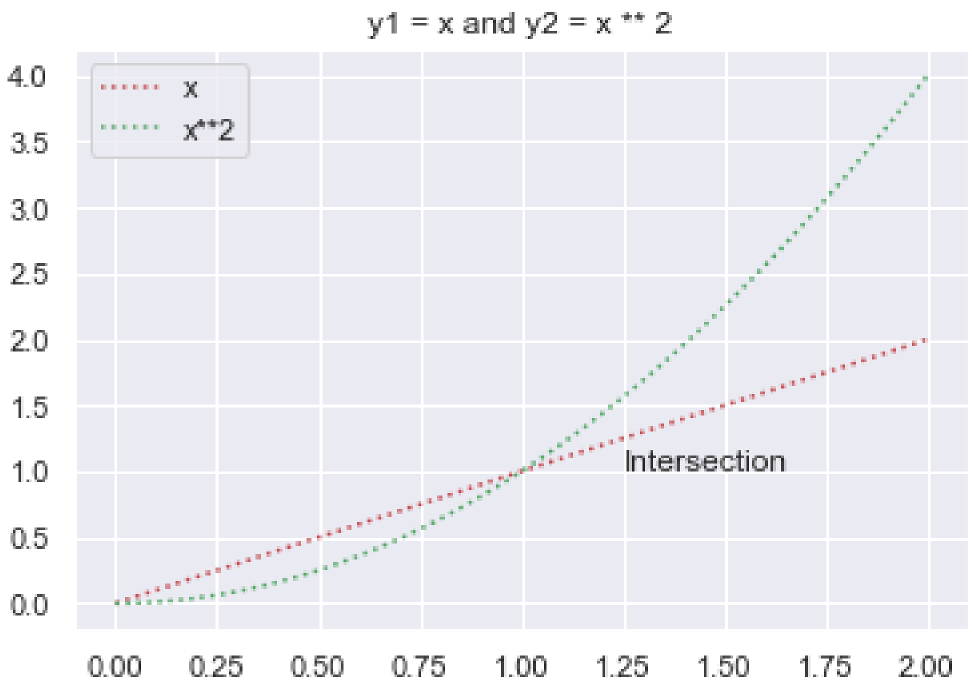
```
z = x ** 2
plt.plot(x, y, c='g', linestyle='--')
plt.xlabel('x-values from -2 to +2')
plt.ylabel('y = x**2')
plt.title('y = x**2')
```

```
Text(0.5, 1.0, 'y = x**2')
```

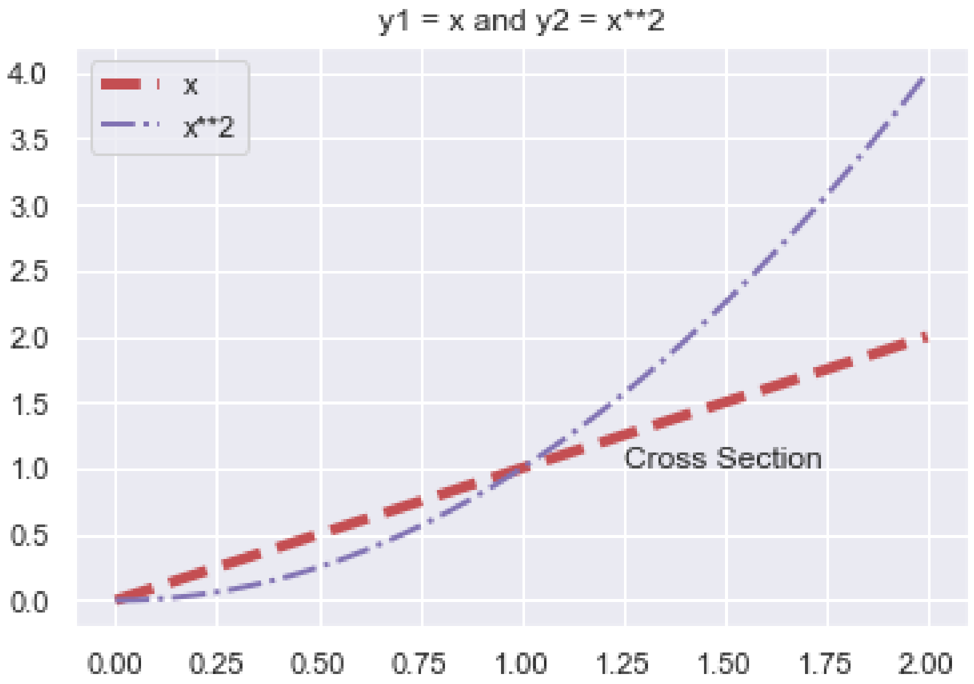


```
x = np.linspace(0, 2, 500).astype(float)

y1 = x ** 1
y2 = x ** 2

plt.plot(x, y1, 'r', x, y2, 'g', linestyle='dotted')
plt.title('y1 = x and y2 = x ** 2')
plt.legend(['x', 'x**2'])

plt.annotate(text='Intersection', xy=(1.25, 1))
```

```
Text(1.25, 1, 'Intersection')
```



```
x = np.linspace(0, 2, 500).astype(float)
y1 = x**1
y2 = x**2

# To change individual linewidth, linestyle
lin1, lin2 = plt.plot(x, y1, 'r', x, y2, 'm')
lin1.set_linewidth(4.0), lin2.set_linewidth(2.0),
lin1.set_linestyle('--'), lin2.set_linestyle('-.')

# title and legend
plt.title('y1 = x and y2 = x**2')
plt.legend(["x", "x**2"])

# to show the intersection of 2 lines
plt.annotate(text='Cross Section', xy=(1.25, 1))

# Save the plot in .png .jpg .jpeg .pdf
plt.savefig("y1=x and y2=x2.png")
```

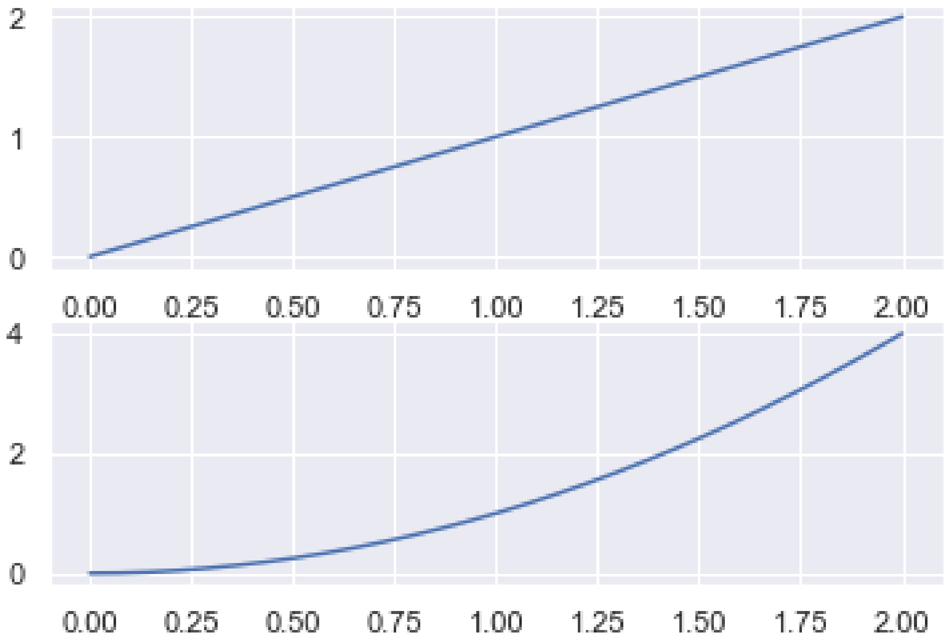


SubPlots

Single Column, Multiple Rows

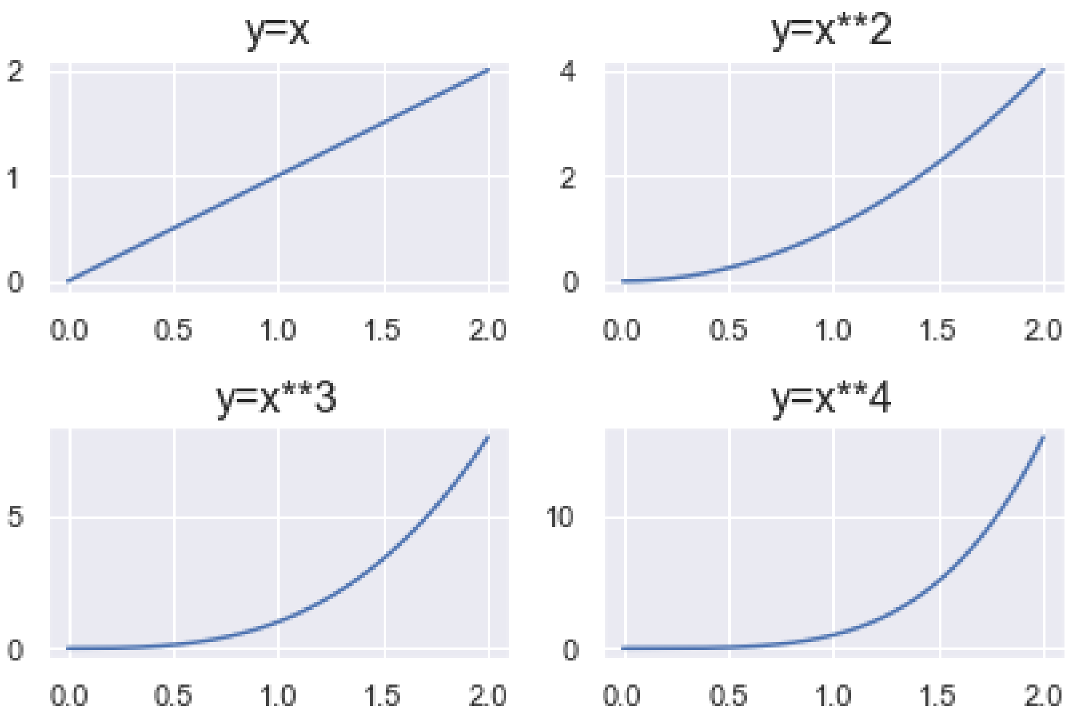
```
x = np.linspace(0, 2, 500)
y1, y2 = x, x**2

plt.subplot(2, 1, 1)
plt.plot(x, y1)

plt.subplot(2, 1, 2)
plt.plot(x, y2)
```

```
[<matplotlib.lines.Line2D at 0x7fa9738281f0>]
```

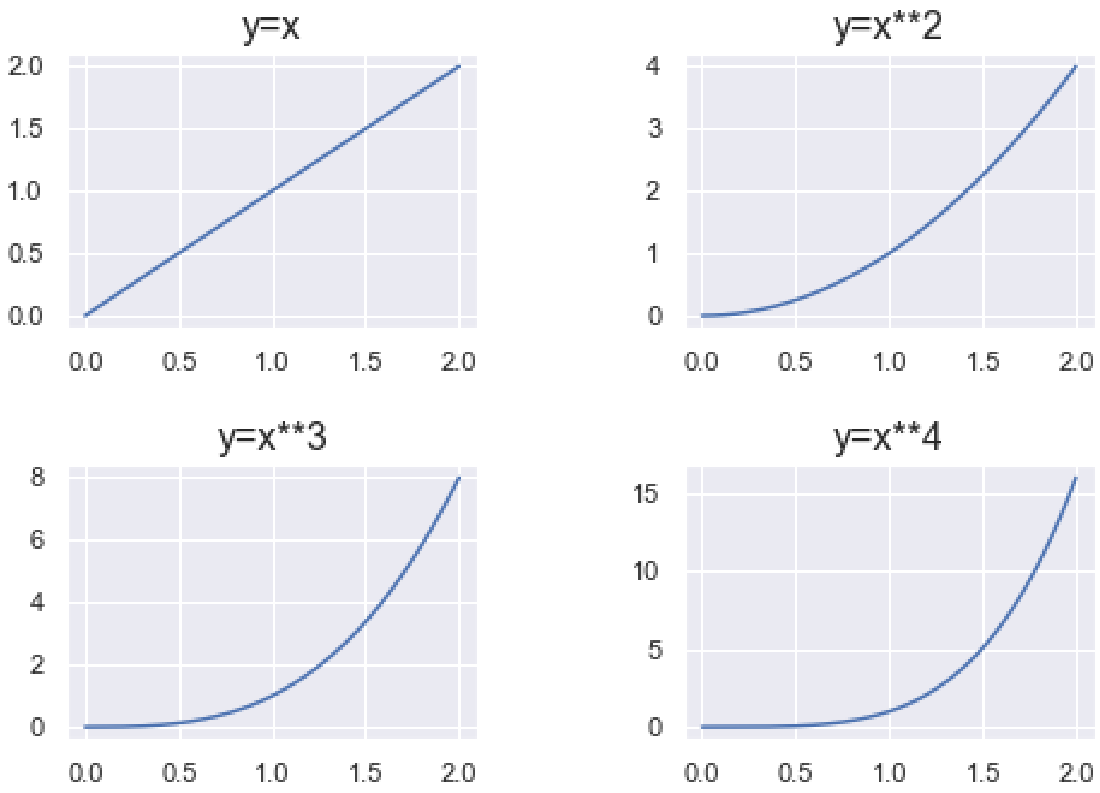


Multiple Columns, Multiple Rows

```
x = np.linspace(0, 2, 500).astype(float)
y1, y2, y3, y4 = x, x**2, x**3, x**4

plt.subplot(2, 2, 1)
plt.plot(x, y1)
plt.title('y=x', fontsize=16)

plt.subplot(2, 2, 2)
plt.plot(x, y2)
plt.title('y=x**2', fontsize=16)

plt.subplot(2, 2, 3)
plt.plot(x, y3)
plt.title('y=x**3', fontsize=16)

plt.subplot(2, 2 ,4)
plt.plot(x, y4)
plt.title('y=x**4', fontsize=16)

# To avoid overlapping of the axis
plt.tight_layout()
```



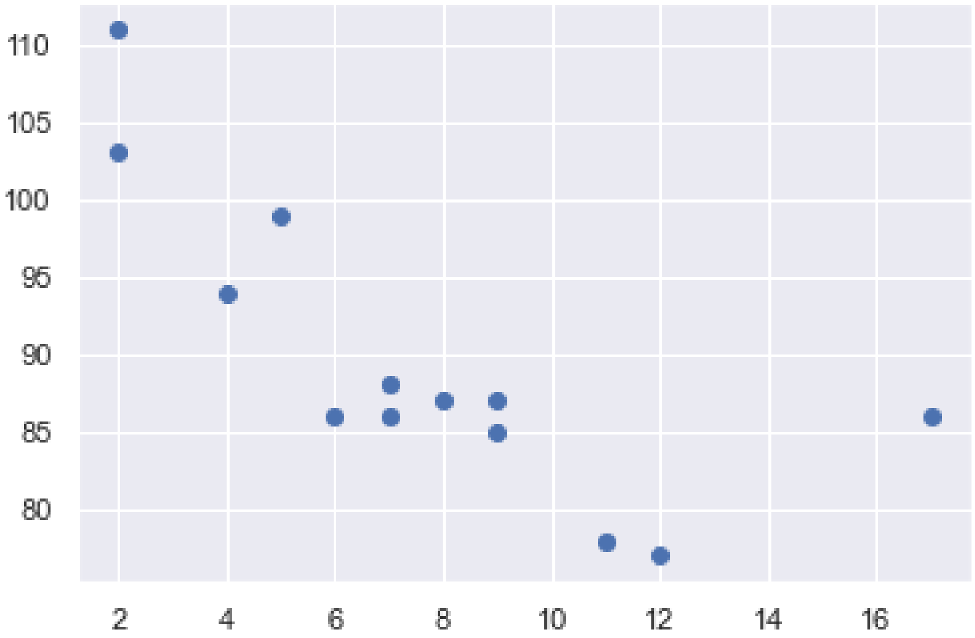
```
x = np.linspace(0, 2, 500).astype(float)
y1, y2, y3, y4 = x, x**2, x**3, x**4

plt.subplot(2, 2, 1)
plt.plot(x, y1)
plt.title('y=x', fontsize=16)

plt.subplot(2, 2, 2)
plt.plot(x, y2)
plt.title('y=x**2', fontsize=16)

plt.subplot(2, 2, 3)
plt.plot(x, y3)
plt.title('y=x**3', fontsize=16)

plt.subplot(2, 2 ,4)
plt.plot(x, y4)
plt.title('y=x**4', fontsize=16)

# To avoid overlapping of the axis
# plt.subplots_adjust(left, bottom, right, top, wspace, hspace)
plt.subplots_adjust(left=0.01, right=1, top=1, bottom=0.01,
                    wspace=0.5, hspace=0.5)
```



SCATTER PLOT

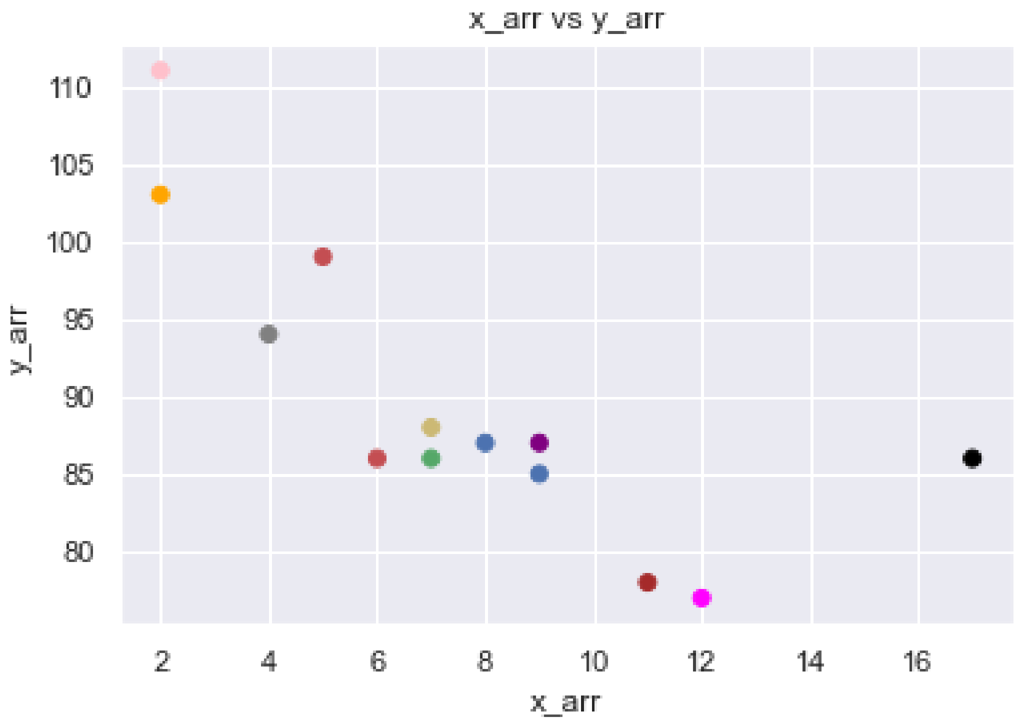
```
x_arr = np.array([5, 7, 8, 7, 2, 17, 2, 9, 4, 11, 12, 9, 6])
y_arr = np.array([99, 86, 87, 88, 111, 86, 103, 87, 94, 78, 77, 85,
86])

plt.scatter(x=x_arr, y=y_arr)
```

```
<matplotlib.collections.PathCollection at 0x7fa9a25a7550>
```



```
colors = np.array(['r', 'g', 'b', 'y', 'pink', 'black', 'orange',
'purple', 'grey', 'brown', 'magenta', 'b', 'r'])
plt.scatter(x=x_arr, y=y_arr, c=colors)
plt.xlabel('x_arr')
plt.ylabel('y_arr')
plt.title('x_arr vs y_arr')
```

```
Text(0.5, 1.0, 'x_arr vs y_arr')
```


x_arr vs y_arr
110
105
100
95
90
85
80
y_arr
2
4
6
8
10
12
14
16
x_arr