NOTE

- to find relation between categorical and numerical data we use histogram(bar plot)

```
In [1]: import matplotlib.pyplot as plt
        import pandas as pd
        import numpy as np
```

```
In [2]: x = np.linspace(0,10,200)
        x
```

```
Out[2]: array([ 0.        ,  0.05025126,  0.10050251,  0.15075377,  0.20100503,
                0.25125628,  0.30150754,  0.35175879,  0.40201005,  0.45226131,
                0.50251256,  0.55276382,  0.60301508,  0.65326633,  0.70351759,
                0.75376884,  0.8040201 ,  0.85427136,  0.90452261,  0.95477387,
                1.00502513,  1.05527638,  1.10552764,  1.15577889,  1.20603015,
                1.25628141,  1.30653266,  1.35678392,  1.40703518,  1.45728643,
                1.50753769,  1.55778894,  1.6080402 ,  1.65829146,  1.70854271,
                1.75879397,  1.80904523,  1.85929648,  1.90954774,  1.95979899,
                2.01005025,  2.06030151,  2.11055276,  2.16080402,  2.21105528,
                2.26130653,  2.31155779,  2.36180905,  2.4120603 ,  2.46231156,
                2.51256281,  2.56281407,  2.61306533,  2.66331658,  2.71356784,
                2.7638191 ,  2.81407035,  2.86432161,  2.91457286,  2.96482412,
                3.01507538,  3.06532663,  3.11557789,  3.16582915,  3.2160804 ,
                3.26633166,  3.31658291,  3.36683417,  3.41708543,  3.46733668,
                3.51758794,  3.5678392 ,  3.61809045,  3.66834171,  3.71859296,
                3.76884422,  3.81909548,  3.86934673,  3.91959799,  3.96984925,
                4.0201005 ,  4.07035176,  4.12060302,  4.17085427,  4.22110553,
                4.27135678,  4.32160804,  4.3718593 ,  4.42211055,  4.47236181,
                4.52261307,  4.57286432,  4.62311558,  4.67336683,  4.72361809,
                4.77386935,  4.8241206 ,  4.87437186,  4.92462312,  4.97487437,
                5.02512563,  5.07537688,  5.12562814,  5.1758794 ,  5.22613065,
                5.27638191,  5.32663317,  5.37688442,  5.42713568,  5.47738693,
                5.52763819,  5.57788945,  5.6281407 ,  5.67839196,  5.72864322,
                5.77889447,  5.82914573,  5.87939698,  5.92964824,  5.9798995 ,
                6.03015075,  6.08040201,  6.13065327,  6.18090452,  6.23115578,
                6.28140704,  6.33165829,  6.38190955,  6.4321608 ,  6.48241206,
                6.53266332,  6.58291457,  6.63316583,  6.68341709,  6.73366834,
                6.7839196 ,  6.83417085,  6.88442211,  6.93467337,  6.98492462,
                7.03517588,  7.08542714,  7.13567839,  7.18592965,  7.2361809 ,
                7.28643216,  7.33668342,  7.38693467,  7.43718593,  7.48743719,
                7.53768844,  7.5879397 ,  7.63819095,  7.68844221,  7.73869347,
                7.78894472,  7.83919598,  7.88944724,  7.93969849,  7.98994975,
                8.04020101,  8.09045226,  8.14070352,  8.19095477,  8.24120603,
                8.29145729,  8.34170854,  8.3919598 ,  8.44221106,  8.49246231,
                8.54271357,  8.59296482,  8.64321608,  8.69346734,  8.74371859,
                8.79396985,  8.84422111,  8.89447236,  8.94472362,  8.99497487,
                9.04522613,  9.09547739,  9.14572864,  9.1959799 ,  9.24623116,
                9.29648241,  9.34673367,  9.39698492,  9.44723618,  9.49748744,
                9.54773869,  9.59798995,  9.64824121,  9.69849246,  9.74874372,
                9.79899497,  9.84924623,  9.89949749,  9.94974874, 10.        ])
```

```
In [3]: y = np.sin(x)
        y
```

Loading [MathJax]/extensions/Safe.js

Out[3]:

```
array([ 0.        ,  0.05023011,  0.10033341,  0.15018339,  0.19965422,
        0.24862099,  0.29696008,  0.34454944,  0.39126893,  0.43700061,
        0.481629  ,  0.52504145,  0.56712835,  0.60778345,  0.6469041 ,
        0.68439153,  0.72015112,  0.75409257,  0.78613019,  0.8161831 ,
        0.84417544,  0.87003651,  0.89370105,  0.91510929,  0.9342072 ,
        0.95094655,  0.96528509,  0.97718662,  0.98662108,  0.99356467,
        0.99799984,  0.99991541,  0.99930653,  0.99617474,  0.99052796,
        0.98238043,  0.97175273,  0.95867168,  0.94317032,  0.92528777,
        0.90506919,  0.88256563,  0.85783388,  0.8309364 ,  0.80194109,
        0.77092115,  0.7379549 ,  0.70312557,  0.66652108,  0.62823386,
        0.58836056,  0.54700186,  0.50426216,  0.46024937,  0.41507461,
        0.36885193,  0.32169803,  0.27373195,  0.22507478,  0.17584939,
        0.12618003,  0.07619211,  0.02601183, -0.02423412, -0.07441889,
       -0.12441577, -0.17409855, -0.22334179, -0.27202116, -0.32001378,
       -0.36719847, -0.41345611, -0.45866992, -0.50272574, -0.54551235,
       -0.58692173, -0.62684933, -0.66519435, -0.70185999, -0.73675367,
       -0.7697873 , -0.80087747, -0.82994571, -0.85691862, -0.88172811,
       -0.90431153, -0.92461187, -0.94257789, -0.95816422, -0.97133152,
       -0.98204653, -0.99028221, -0.99601778, -0.99923873, -0.99993695,
       -0.99811068, -0.99376451, -0.98690943, -0.97756275, -0.96574805,
       -0.95149517, -0.93484009, -0.91582485, -0.89449748, -0.8709118 ,
       -0.84512737, -0.81720929, -0.78722803, -0.75525929, -0.72138377,
       -0.68568702, -0.64825913, -0.60919462, -0.56859209, -0.52655407,
       -0.48318668, -0.4385994 , -0.39290482, -0.34621828, -0.29865766,
       -0.25034303, -0.20139637, -0.15194126, -0.10210255, -0.05200606,
       -0.00177827,  0.048454  ,  0.09856395,  0.14842506,  0.19791144,
        0.24689816,  0.29526155,  0.34287951,  0.38963181,  0.43540043,
        0.48006981,  0.52352718,  0.56566282,  0.60637036,  0.64554701,
        0.68309389,  0.71891618,  0.75292346,  0.78502987,  0.81515434,
        0.84322083,  0.86915847,  0.89290179,  0.91439084,  0.93357136,
        0.95039493,  0.96481908,  0.9768074 ,  0.98632961,  0.99336168,
        0.99788585,  0.99989069,  0.99937116,  0.99632856,  0.99077057,
        0.98271122,  0.97217086,  0.95917611,  0.94375976,  0.92596075,
        0.905824  ,  0.88340035,  0.85874643,  0.83192446,  0.80300216,
        0.77205257,  0.7391538 ,  0.70438892,  0.66784571,  0.62961641,
        0.58979754,  0.54848964,  0.50579699,  0.46182738,  0.41669181,
        0.37050423,  0.32338126,  0.27544187,  0.22680707,  0.17759967,
        0.12794389,  0.07796509,  0.02778946, -0.02245633, -0.07264543,
       -0.12265112, -0.17234716, -0.22160808, -0.27030952, -0.31832851,
       -0.36554384, -0.4118363 , -0.45708901, -0.50118772, -0.54402111])
```

In [4]:

```
lin = np.linspace(0,10,5)
lin
```

Out[4]:

```
array([  0. ,   2.5,   5. ,   7.5,  10. ])
```

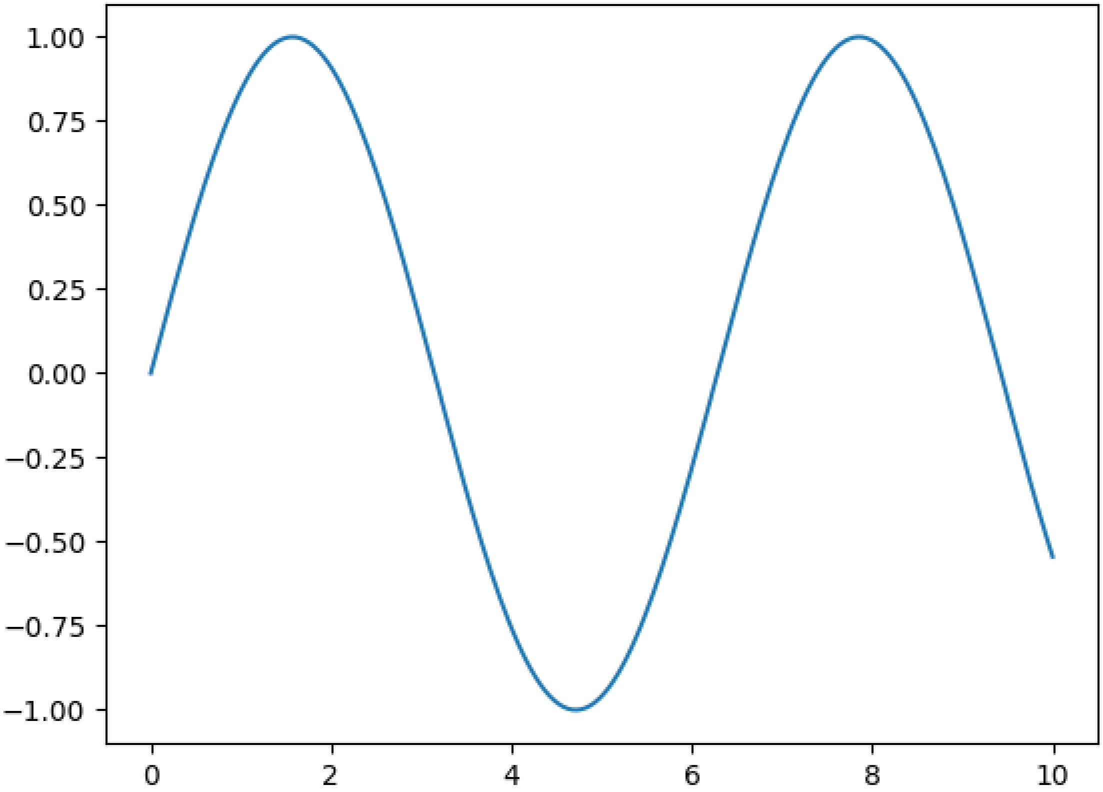
In [5]:

```
plt.plot(x,y)
```

Out[5]:

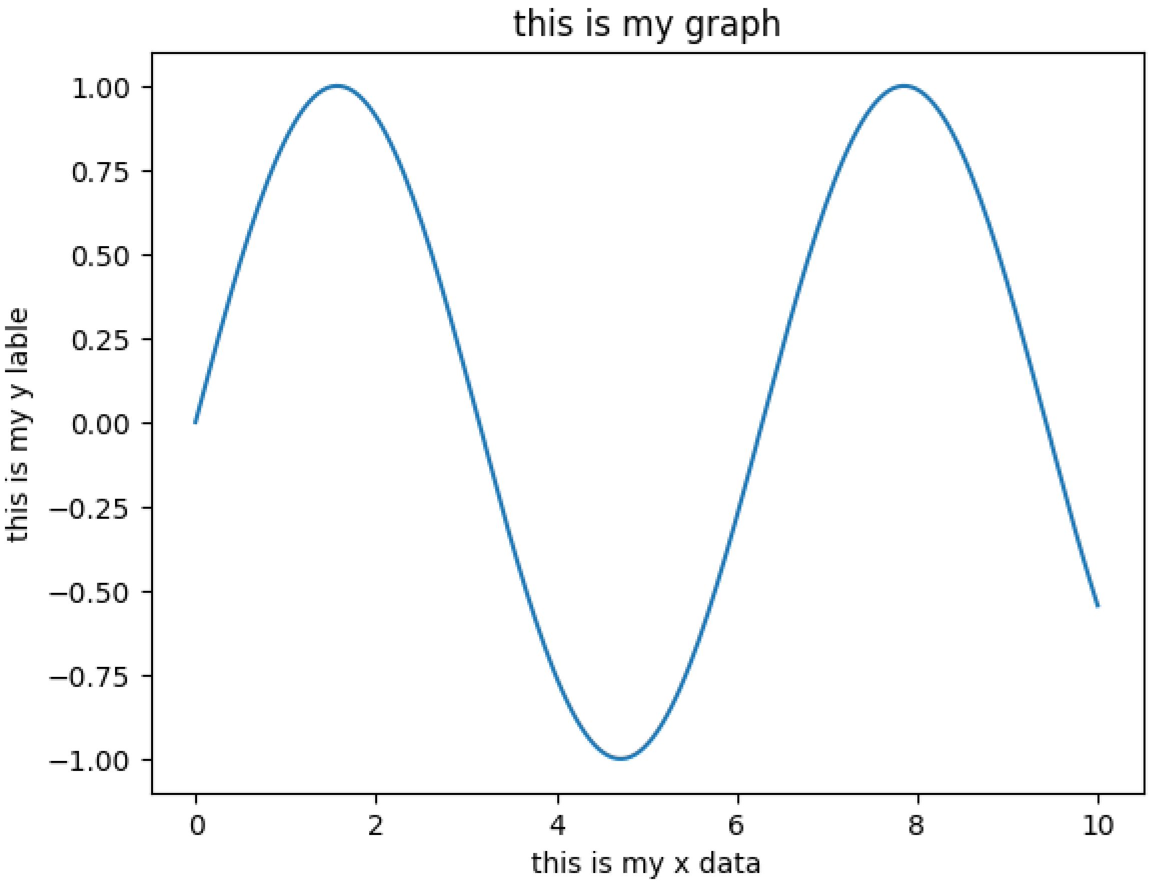
```
[<matplotlib.lines.Line2D at 0x20730760a10>]
```

Loading [MathJax]/extensions/Safe.js



In [6]:
```
plt.plot(x,y)
plt.xlabel('this is my x data')
plt.ylabel('this is my y lable')
plt.title('this is my graph')
```

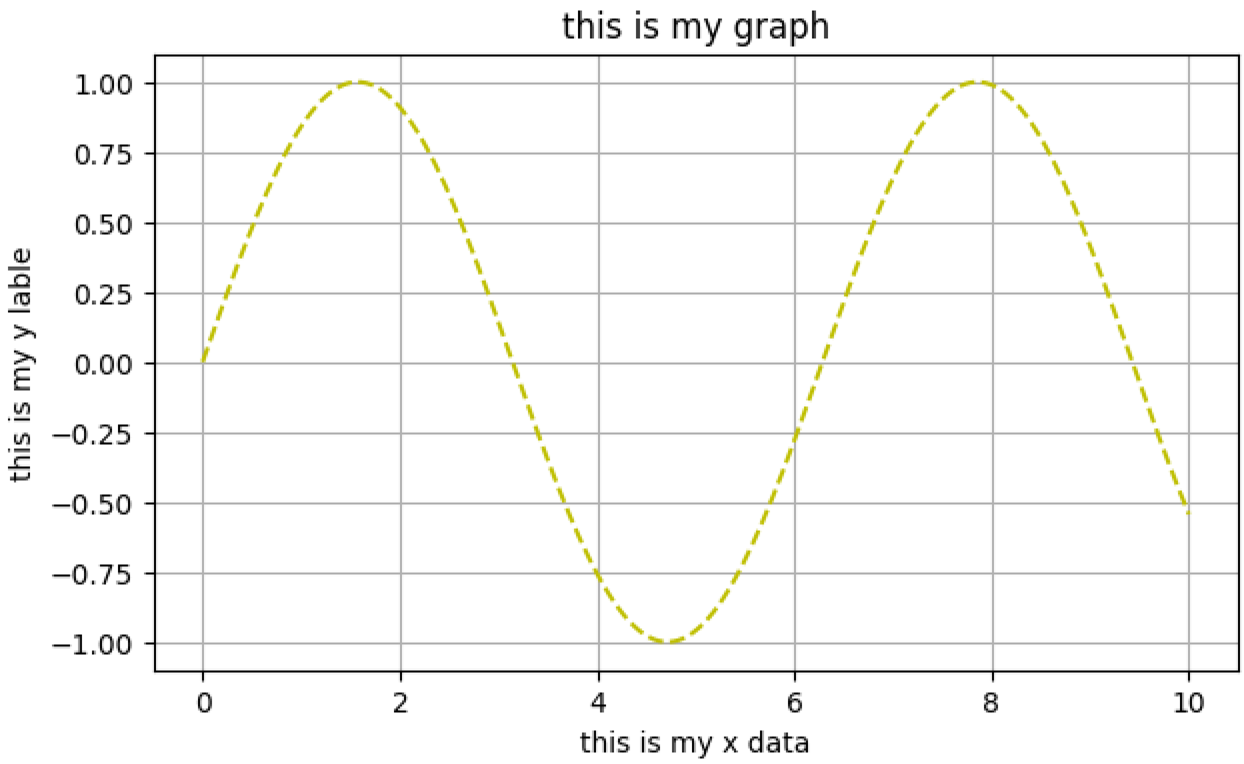
Out[6]:
```
Text(0.5, 1.0, 'this is my graph')
```



In [7]:
```
# to control plot's size
plt.figure(figsize=(7,4))
```

Loading [MathJax]/extensions/Safe.js

```
plt.plot(x,y,'--y')
plt.xlabel('this is my x data')
plt.ylabel('this is my y lable')
plt.title('this is my graph')
plt.grid()
```



In [8]:

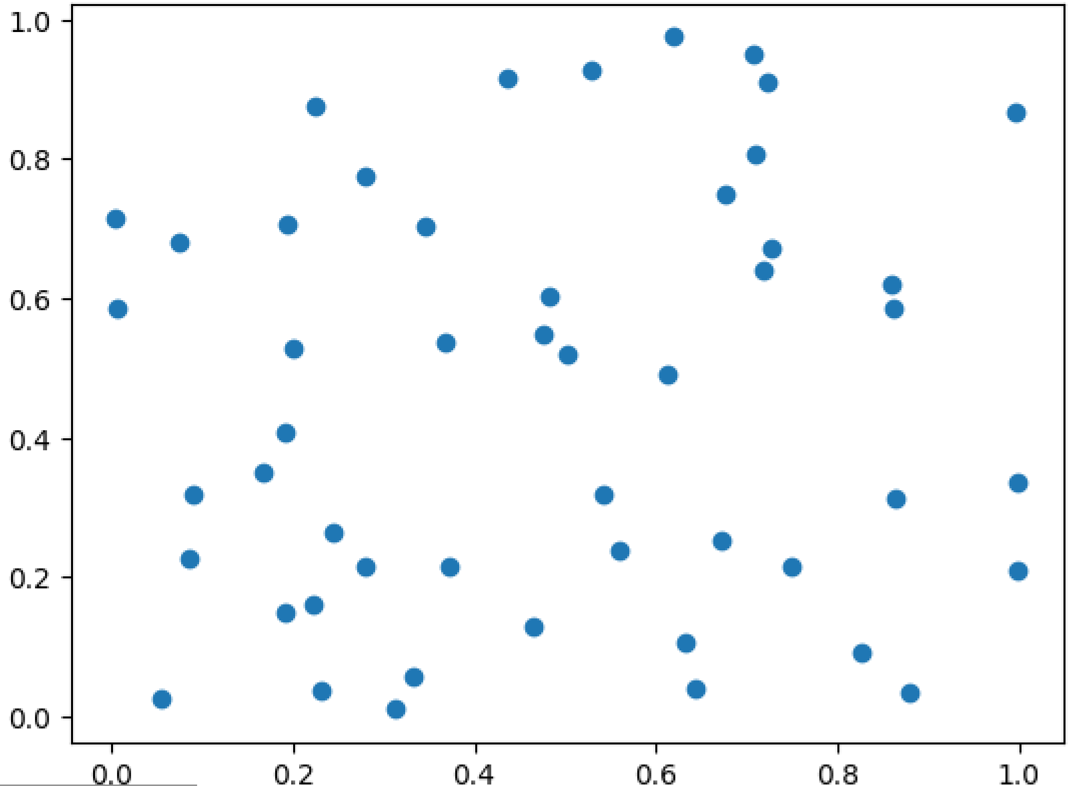
```
x1 = np.random.rand(50)
x1
y1 = np.random.rand(50)
```

In [9]:

```
plt.scatter(x1,y1)
```

Out[9]:

```
<matplotlib.collections.PathCollection at 0x207307b75d0>
```



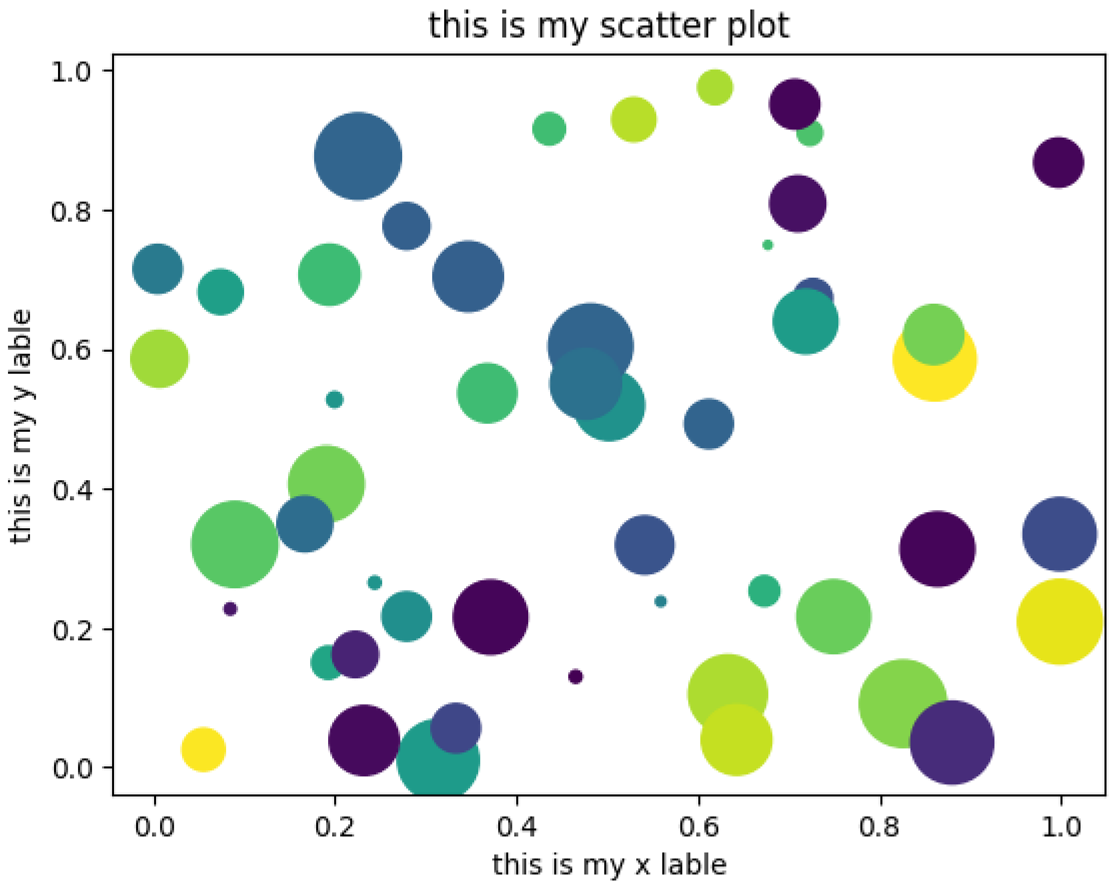
Loading [MathJax]/extensions/Safe.js

In [10]:

```
# to give different colours to each data points
colours = np.random.rand(50)
# to give different sizes to each data point
sizes = 1000*np.random.rand(50)
plt.scatter(x=x1,y=y1,c=colours,s= sizes,alpha=.999)
plt.xlabel('this is my x lable')
plt.ylabel('this is my y lable')
plt.title('this is my scatter plot')
```

Out[10]:

```
Text(0.5, 1.0, 'this is my scatter plot')
```



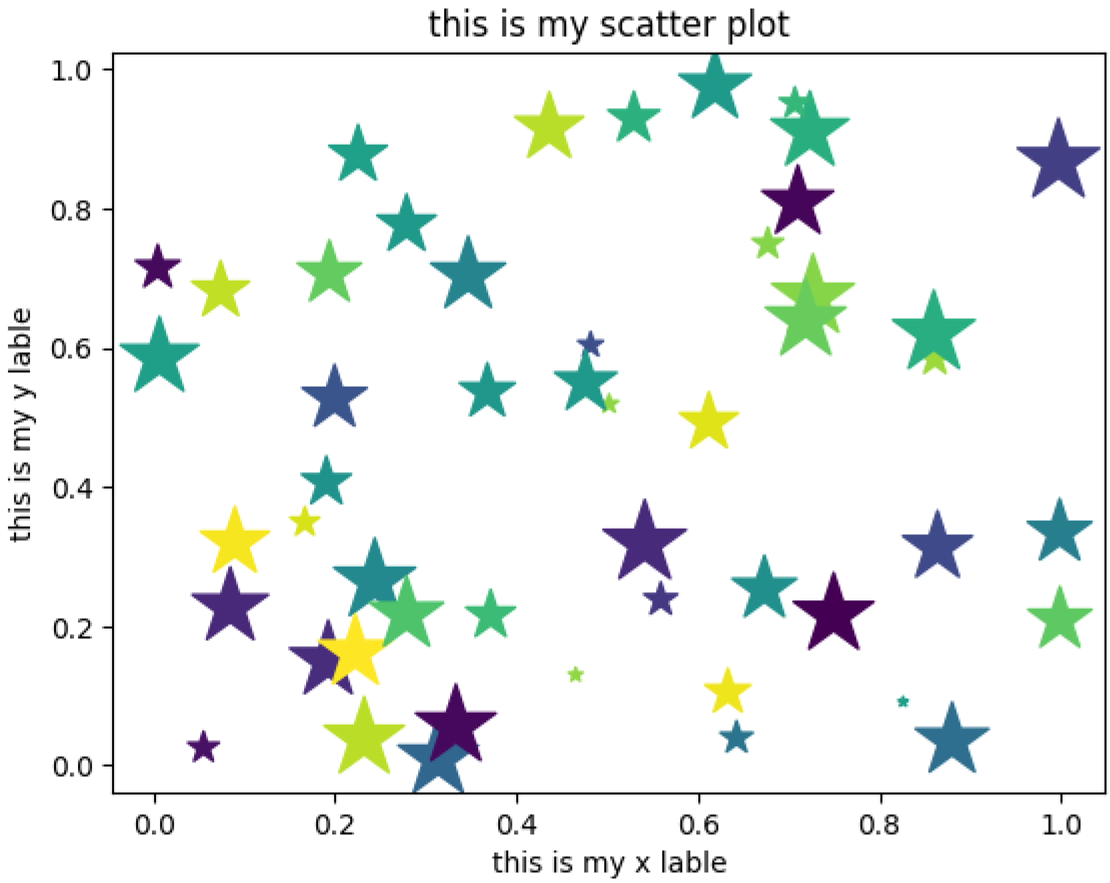
In [11]:

```
# to give different colours to each data points
colours = np.random.rand(50)
# to give different sizes to each data point
sizes = 1000*np.random.rand(50)
plt.scatter(x=x1,y=y1,c=colours,s= sizes,alpha=.999,marker='*')
plt.xlabel('this is my x lable')
plt.ylabel('this is my y lable')
plt.title('this is my scatter plot')
```

Out[11]:

```
Text(0.5, 1.0, 'this is my scatter plot')
```

Loading [MathJax]/extensions/Safe.js



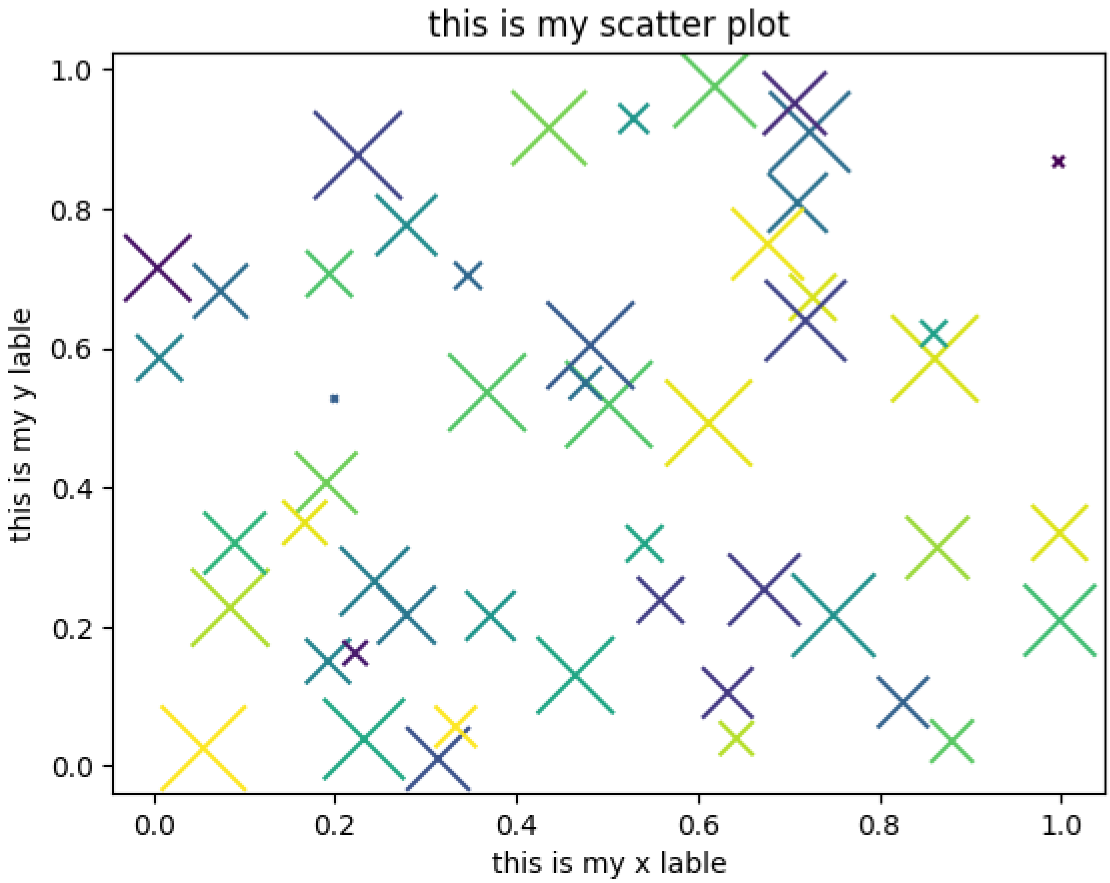
In [12]:

```
# to give different colours to each data points
colours = np.random.rand(50)
# to give different sizes to each data point
sizes = 1000*np.random.rand(50)
plt.scatter(x=x1,y=y1,c=colours,s= sizes,alpha=.999,marker='x')
plt.xlabel('this is my x lable')
plt.ylabel('this is my y lable')
plt.title('this is my scatter plot')
```

Out[12]:

```
Text(0.5, 1.0, 'this is my scatter plot')
```

Loading [MathJax]/extensions/Safe.js



In [13]:

```
x3 = ['a','b','c','d','e']
y3 = np.random.randint(1,200,5)
y3
```

Out[13]:

```
array([190, 155,   7,  73, 180])
```

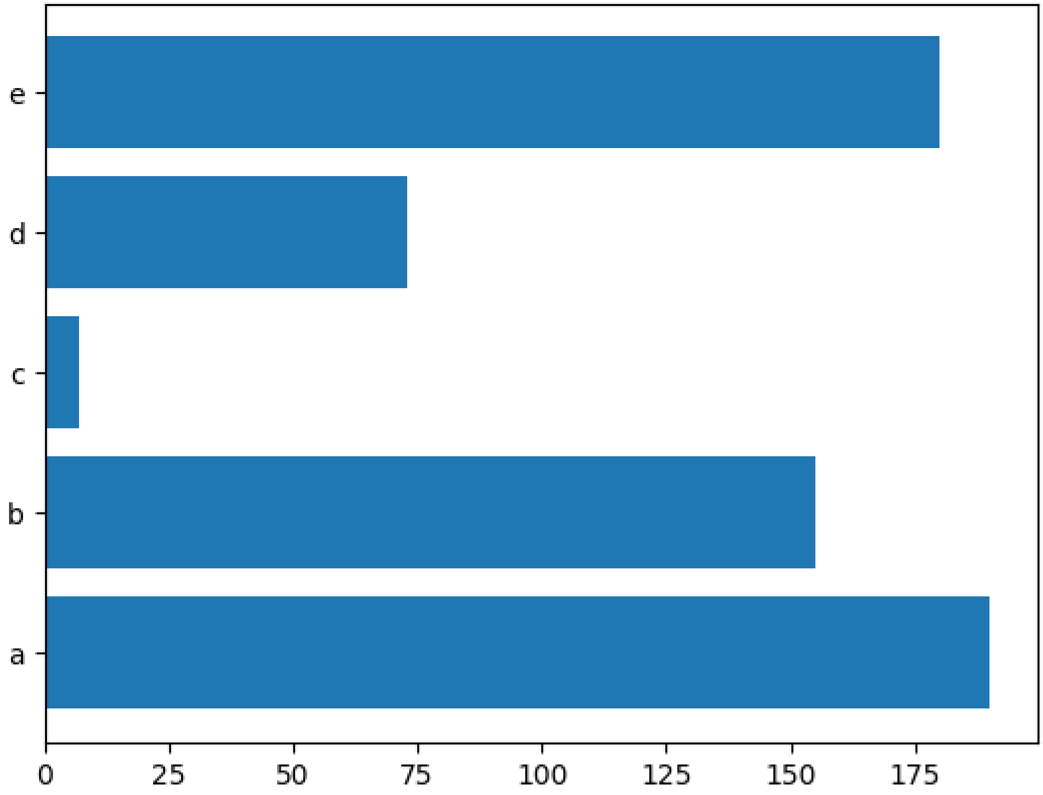
In [14]:

```
# to have barplot horizontally we should append h
plt.barh(x3,y3)
```

Out[14]:

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

Loading [MathJax]/extensions/Safe.js

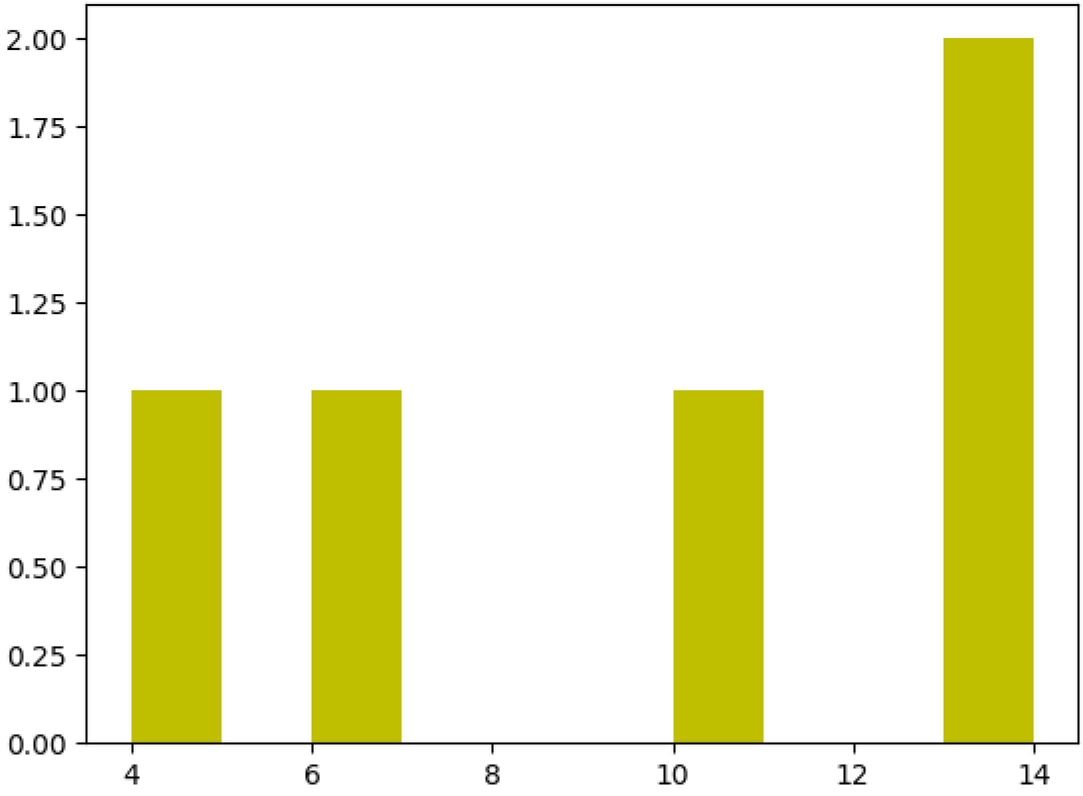


In [15]:

```
data = np.random.randint(1,20,5)
# colour = ['#8C505F','red','blue','black','yello']
plt.hist(data,color='y')
data
```

Out[15]:

```
array([ 6, 13, 10, 14,  4])
```



to plot 3D plot

Loading [MathJax]/extensions/Safe.js

In [16]:

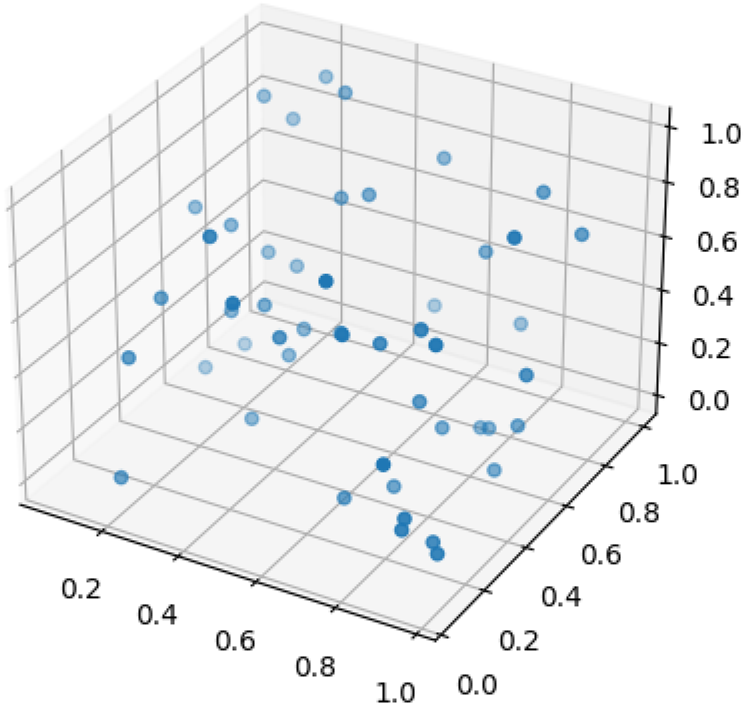
```
# 3D data
# this 3D data didn't worked
data = np.array([[np.random.rand(50)],[np.random.rand(50)],[np.random.rand(50)]])
data

x = np.random.rand(50)
y = np.random.rand(50)
z = np.random.rand(50)
```

In [17]:

```
fig = plt.figure()
ax = fig.add_subplot(projection='3d')
ax.scatter(x,y,z)
```

Out[17]:

```
<mpl_toolkits.mplot3d.art3d.Path3DCollection at 0x20730912690>
```



Loading [MathJax]/extensions/Safe.js