plotting scatter plot

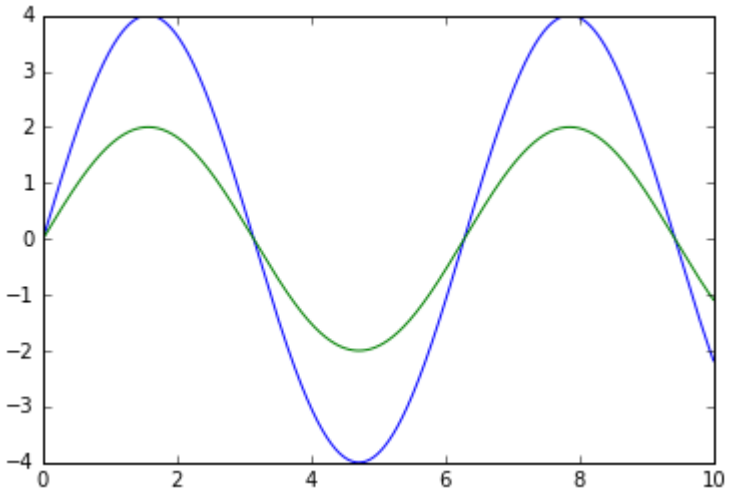
2D Plotting

In [1]:

```
import numpy as np
import matplotlib.pyplot as plt
plt.style.use('classic')
```

In [2]:

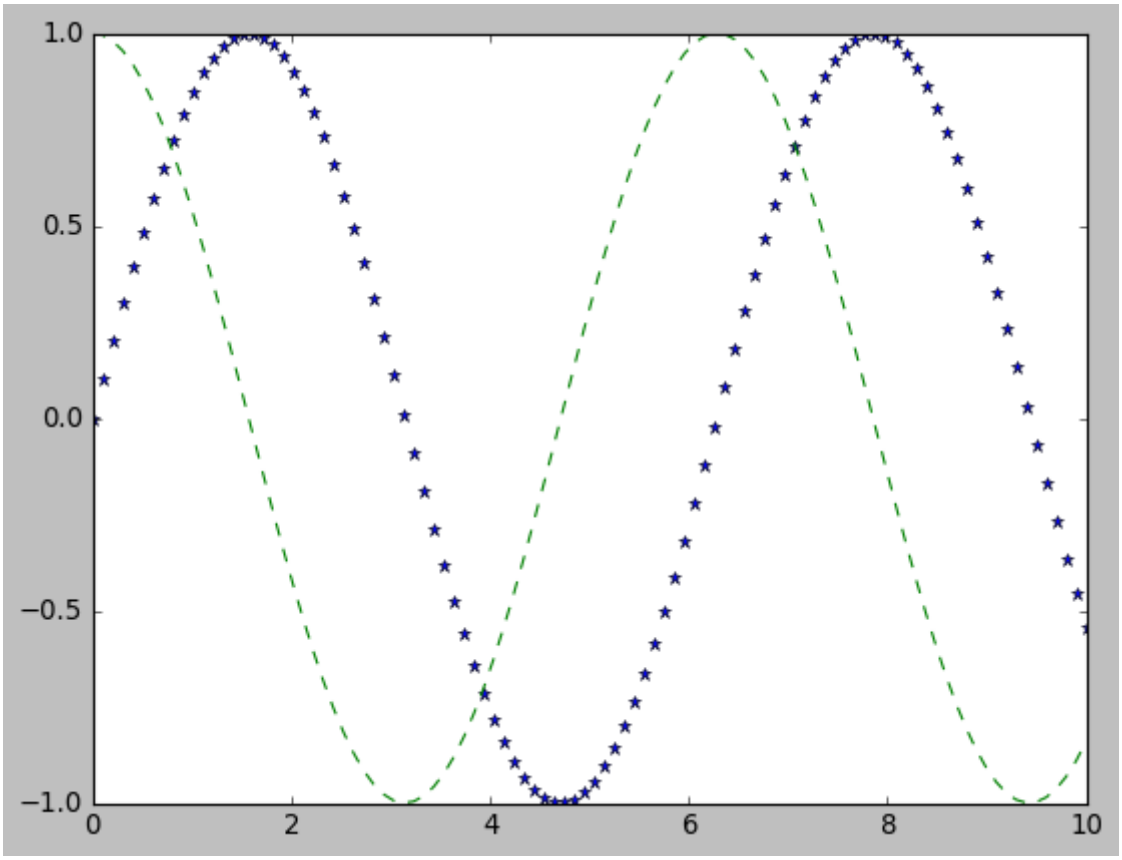
```
import numpy as np
q=np.linspace(0,10,100000)
plt.plot(q,np.sin(q)*4,)
plt.plot(q,np.sin(q)*2)
fig=plt.figure()
fig.savefig(r'C:\Users\Lenovo\Desktop\Jupyter note book\saved_figure\sin_and_cos_graph
5.png')
```




```
<Figure size 432x288 with 0 Axes>
```

In [3]:

```
plt.style.use('classic')
import numpy as np
x = np.linspace(0,10,100)
fig=plt.figure()
plt.plot(x, np.sin(x),'*')
plt.plot(x, np.cos(x),'--')
plt.show()
```



In [4]:

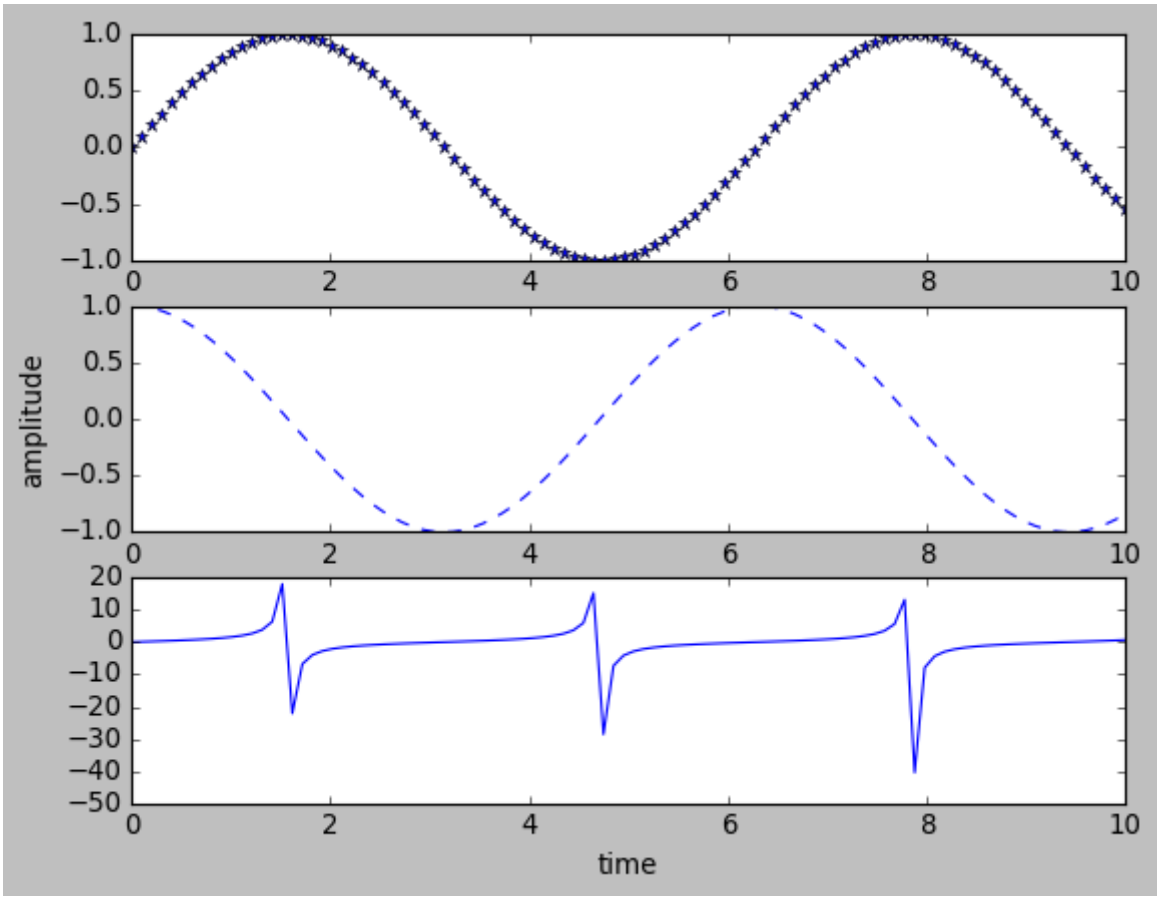
```
plt.style.use('classic')
fig=plt.plot()
plt.subplot(3,1,1) # rows, column, panel number
plt.plot(x, np.sin(x),'*')

plt.subplot(3,1,2)
plt.plot(x,np.cos(x),'--')
plt.ylabel('amplitude')

plt.subplot(3,1,3)
plt.plot(x, np.tan(x),'-')
plt.xlabel('time')
```

Out[4]:

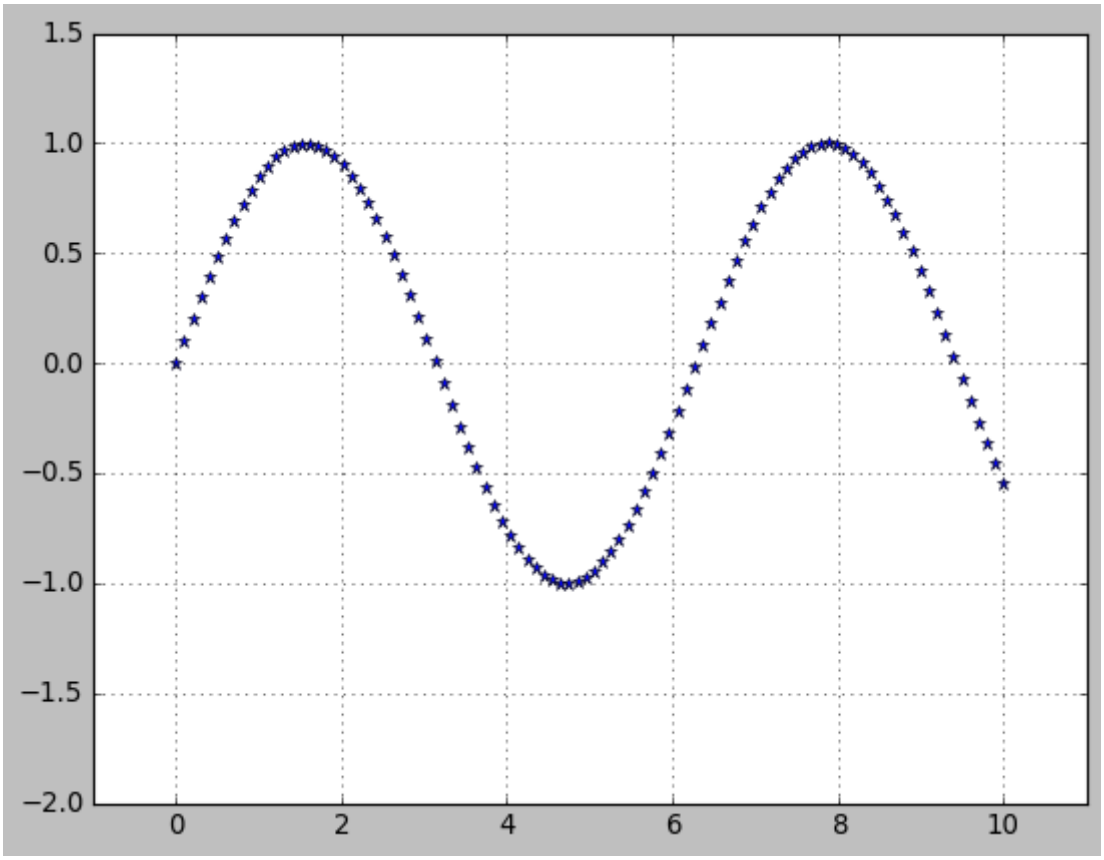
```
Text(0.5, 0, 'time')
```



In [5]:

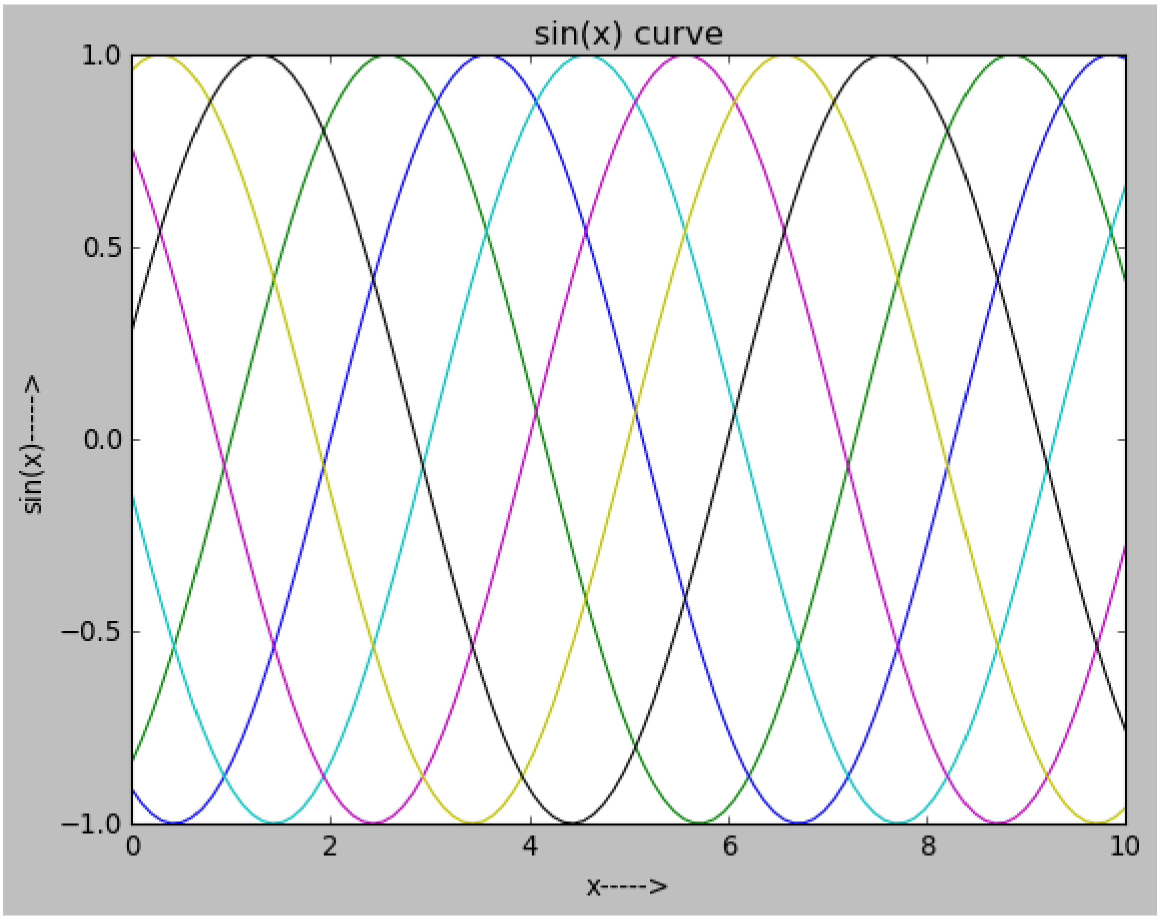
```
plt.style.use('classic')
fig = plt.figure()
ax = plt.axes()

x = np.linspace(0,10,100)
ax.plot(x, np.sin(x),'*')
plt.xlim(-1,11)# or plt.axis([-1,11,-2,1.5])
plt.ylim(-2,1.5)# or plt.axis([-1,11,-2,1.5])
plt.grid()
fig.savefig(r'C:\Users\Lenovo\Desktop\Jupyter note book\saved_figure\sin_graph2.jpg')
```



In [6]:

```
import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt
import pandas as pd
fig=plt.figure()
plt.plot(x, np.sin(x-1),'-',color='g')
plt.plot(x, np.sin(x-2),'-',color='b')
plt.plot(x, np.sin(x-3),'-',color='c')
plt.plot(x, np.sin(x-4),'-',color='m')
plt.plot(x, np.sin(x-5),'-',color='y')
plt.plot(x, np.sin(x-6),'-',color='k')
plt.title('sin(x) curve')
plt.xlabel("x----->")
plt.ylabel("sin(x)----->")
fig.savefig(r'C:\Users\Lenovo\Desktop\Jupyter note book\saved_figure\sin_and_cos_graph
1.jpg')
```

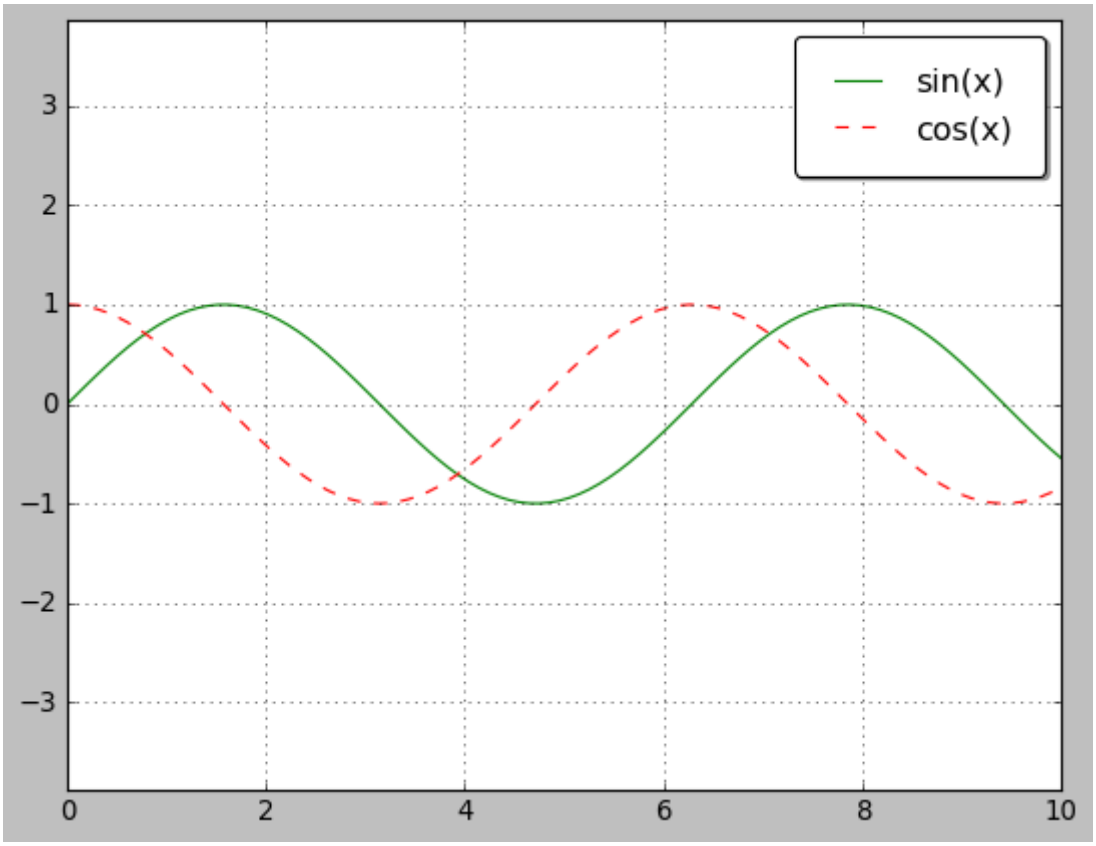


In [7]:

```
fig=plt.figure()
plt.plot(x, np.sin(x),'-g',label='sin(x)')
plt.plot(x, np.cos(x),'--r',label='cos(x)')
plt.axis('equal')
plt.grid()
plt.legend(loc='lower center',frameon='true',ncol=2,framealpha=1);
fig.savefig(r'C:\Users\Lenovo\Desktop\Jupyter note book\saved_figure\sin_and_cos_(with
 legend function)graph2.jpg')
plt.legend(fancybox=True, framealpha=1, shadow=True, borderpad=1)
```

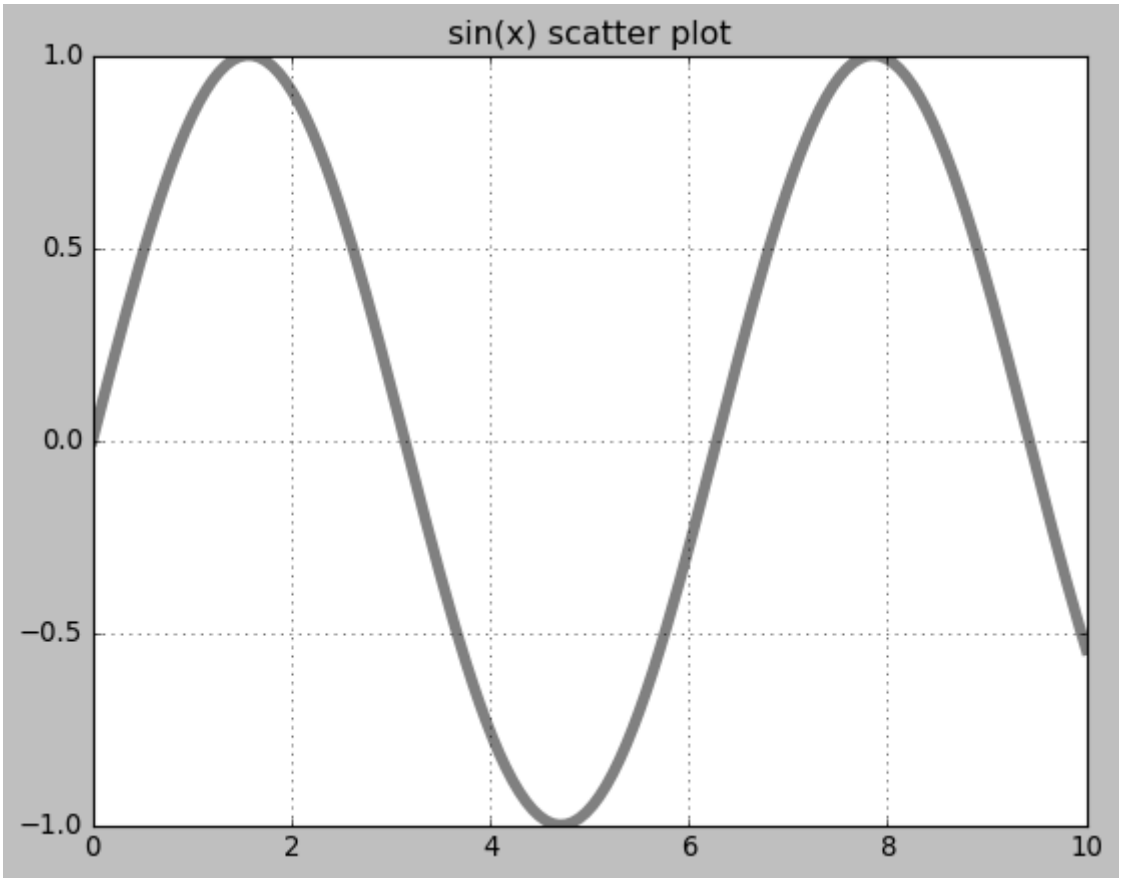
Out[7]:

```
<matplotlib.legend.Legend at 0x191c44c8630>
```



In [207]:

```
x = np.linspace(0,10,150)
fig=plt.figure()
plt.plot(x,np.sin(x),'-',color='gray',linewidth=5)
plt.title('sin(x) scatter plot')
plt.grid()
fig.savefig(r'C:\Users\Lenovo\Desktop\Jupyter note book\saved_figure\scatter plot(with
 cirle).png')
```

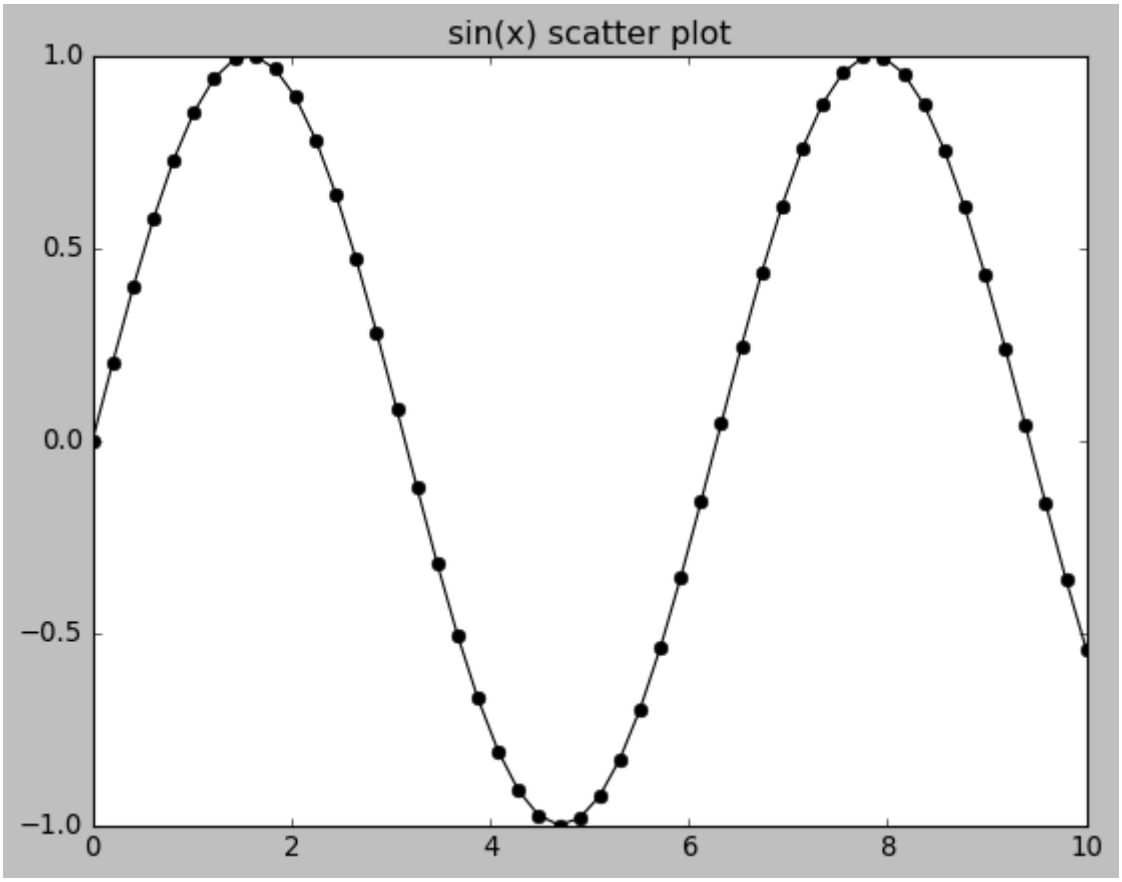


In [40]:

```
x = np.linspace(0,10,50)
fig=plt.figure()
plt.plot(x,np.sin(x),'-ok') ## K= black
plt.title('sin(x) scatter plot')
```

Out[40]:

```
Text(0.5, 1.0, 'sin(x) scatter plot')
```

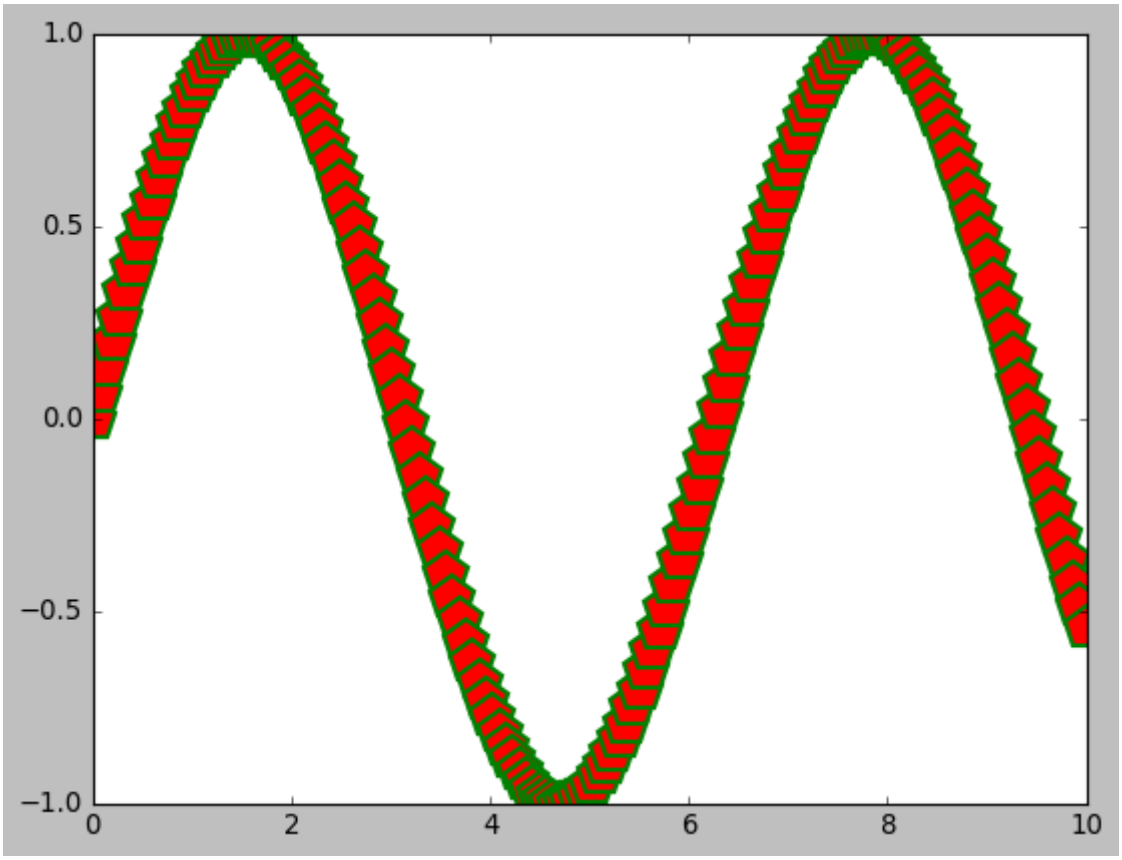


In [225]:

```
plt.plot(x,np.sin(x),'-p',color='gray',markersize=20,linewidth=5,markerfacecolor='r',ma
rkeredgecolor='g',markeredgewidth=2)
# marker size = size of the ploygon
# line width = size of the line
# markerfacecolor = polygon inside color
# markeredgecolor= polygon edge color
# markeredgewidth =size of the polygon width
```

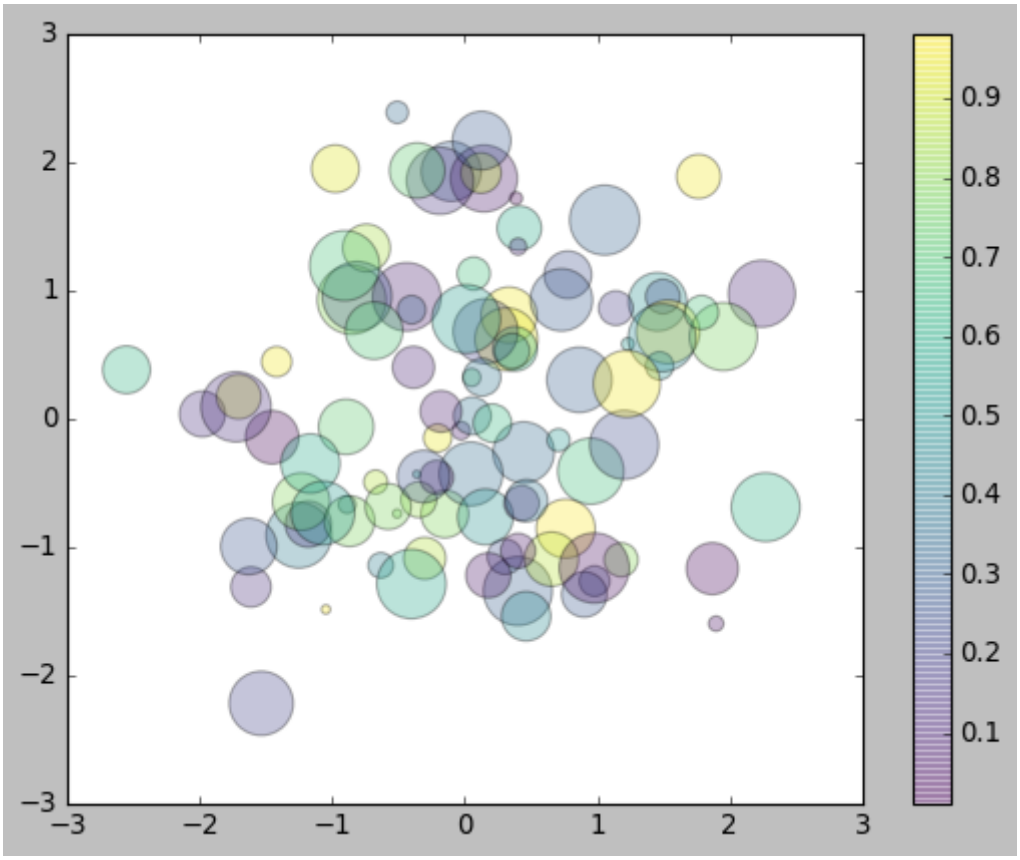
Out[225]:

```
[<matplotlib.lines.Line2D at 0x233c9e58e48>]
```



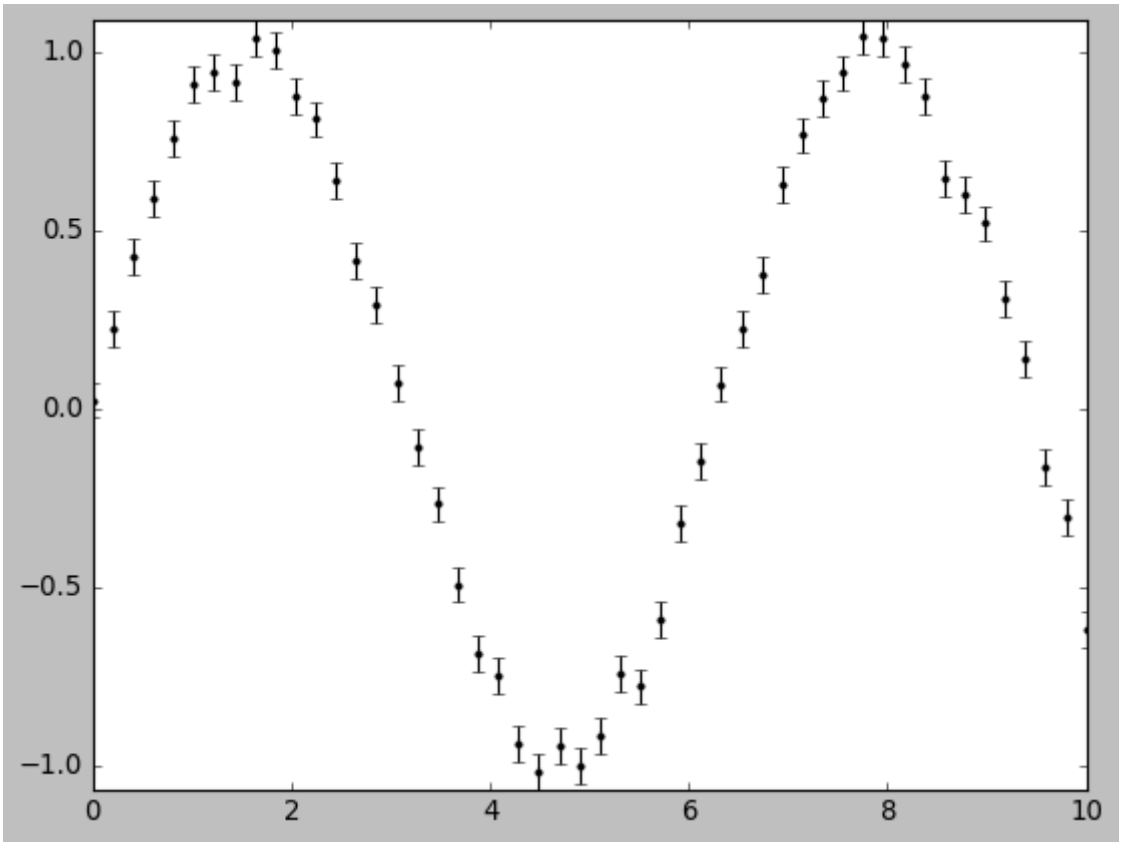
In [75]:

```
qwer=np.random.RandomState(0)
x=qwer.randn(100)
y=qwer.randn(100)
color=qwer.rand(100)
size=1000*qwer.rand(100)
plt.scatter(x, y, c=color, s=size, alpha=0.3,cmap='viridis')
plt.colorbar();
```



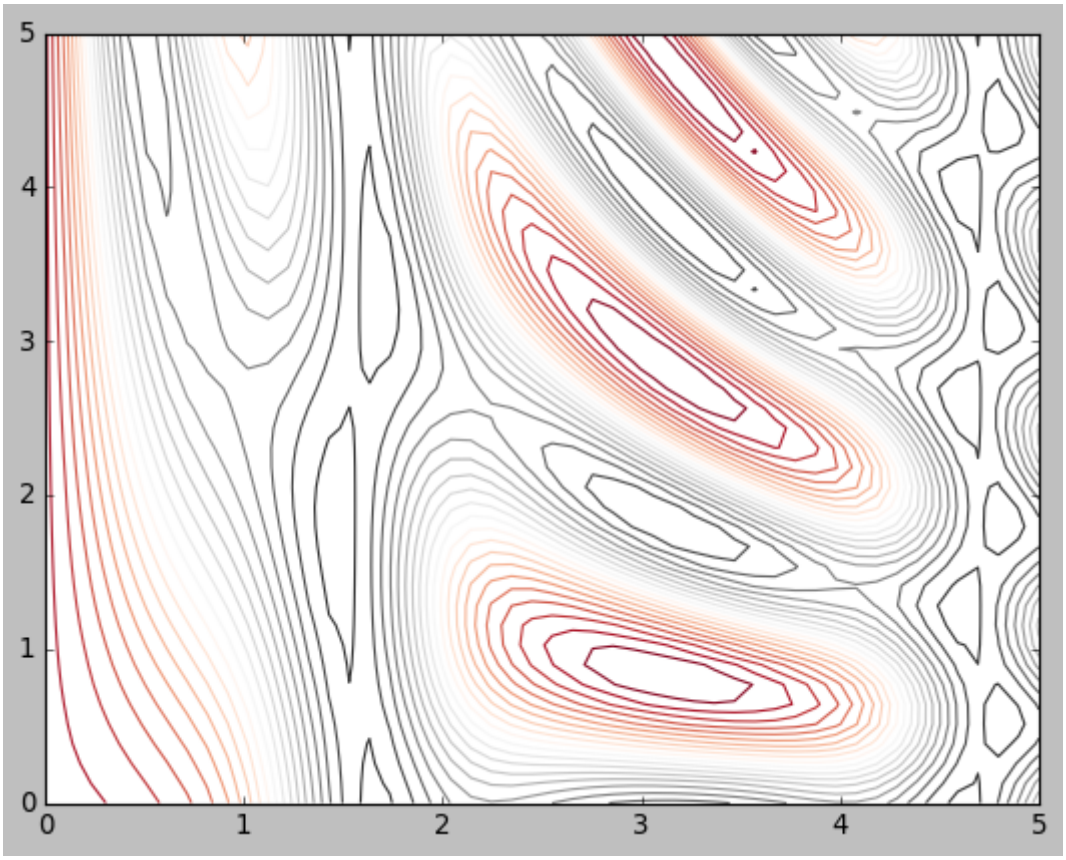
In [104]:

```
fig=plt.figure()
x = np.linspace(0,10,50)
dy=0.05
y=np.sin(x) + dy * np.random.randn(50)
plt.errorbar(x,y,yerr=dy,fmt='.k');
plt.axis('tight')
plt.savefig(r'C:\Users\Lenovo\Desktop\Jupyter note book\saved_figure\hubble_law')
```



In [118]:

```
def f(x,y):
    return np.sin(x)**10+np.cos(10+y*x)*np.cos(x)
x=np.linspace(0,5,50)
y=np.linspace(0,5,40)
x,y=np.meshgrid(x,y)
z=f(x,y)
plt.contour(x,y,z,20,cmap='RdGy');
```



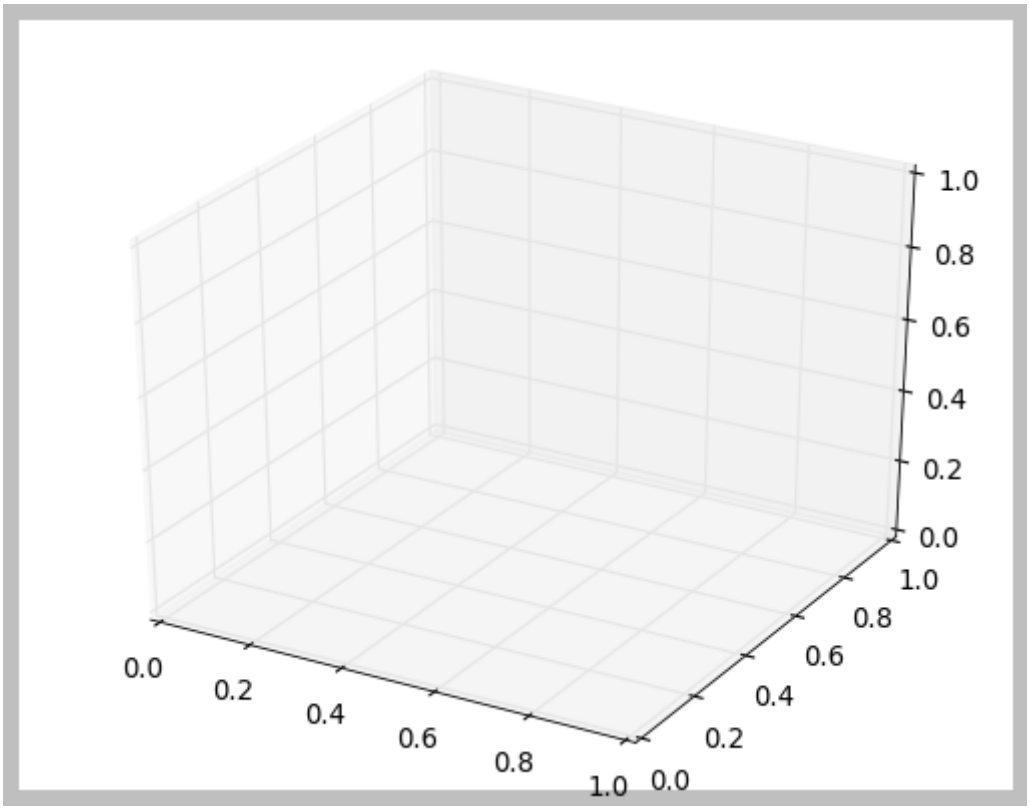
3D plotting

In [121]:

```
from mpl_toolkits import mplot3d
```

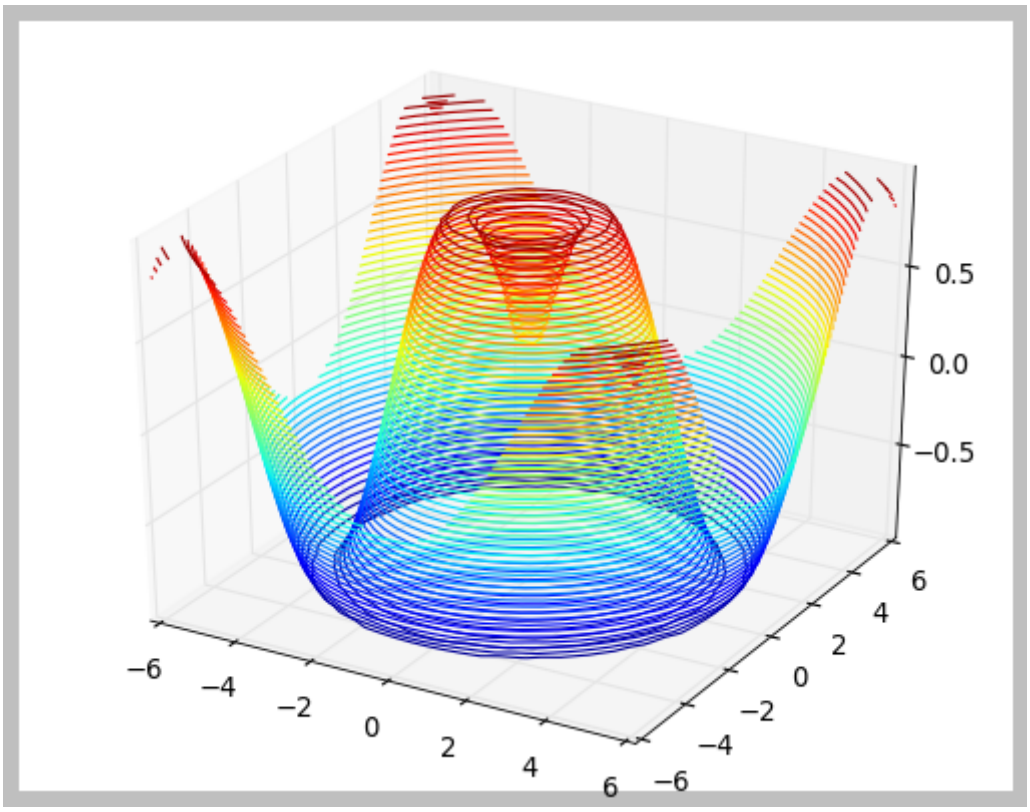
In [126]:

```
fig=plt.figure()
ax=plt.axes(projection='3d')
```



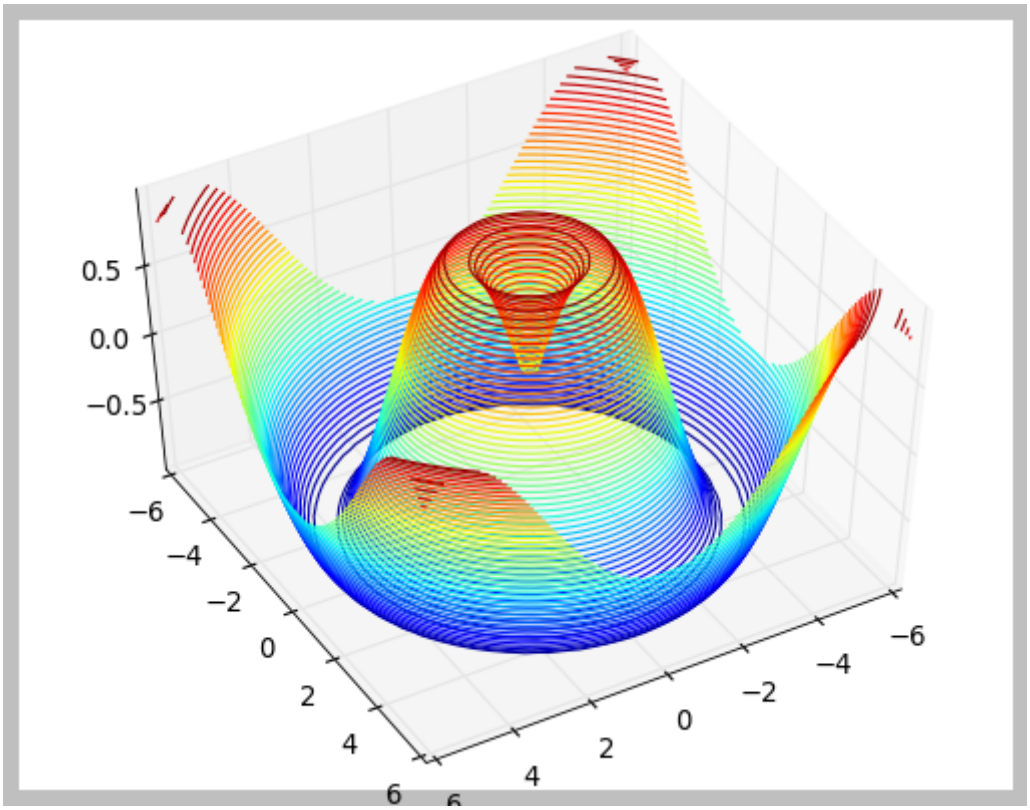
In [192]:

```
def f(x,y):
    return np.sin(np.sqrt(x**2 + y**2))
x= np.linspace(-6,6,30)
y= np.linspace(-6,6,30)
x,y=np.meshgrid(x,y)
z=f(x,y)
fig=plt.figure()
ax=plt.axes(projection='3d')
ax.contour3D(x,y,z,50)
fig.savefig(r'C:\Users\Lenovo\Desktop\Jupyter note book\saved_figure\sin(sqrt(x^2+y^
2))')
```



In [249]:

```
def f(x,y):
    return np.sin(np.sqrt(x**2+y**2))
x = np.linspace(-6,6,50)
y = np.linspace(-6,6,50)
x,y=np.meshgrid(x,y)
z=f(x,y)
fig=plt.figure()
ax=plt.axes(projection='3d')
ax.contour3D(x,y,z,50)
ax.view_init(50,60)
```



In [ ]: