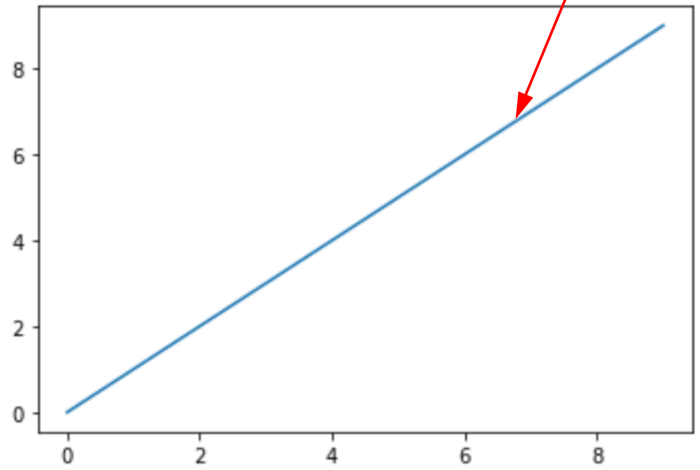
matplotlib

①WindowsPowershellあるいはコマンドプロンプトを
用いて, py −m pip install matplotlibでインストール

```
import numpy as np
import matplotlib.pyplot as plt

x=np.array(range(0,10))
print(x)
y=x

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

plt.plot([Xデータ],[Yデータ])

```
[0 1 2 3 4 5 6 7 8 9]
```



②plotでグラフを作成してshowする

グラフ表示
plt.show()

Sin曲線

```python
import numpy as np
import matplotlib.pyplot as plt
x=np.arange(-np.pi,np.pi,0.1)
y=np.sin(x)
print(x[:10])
print(y[:10])

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

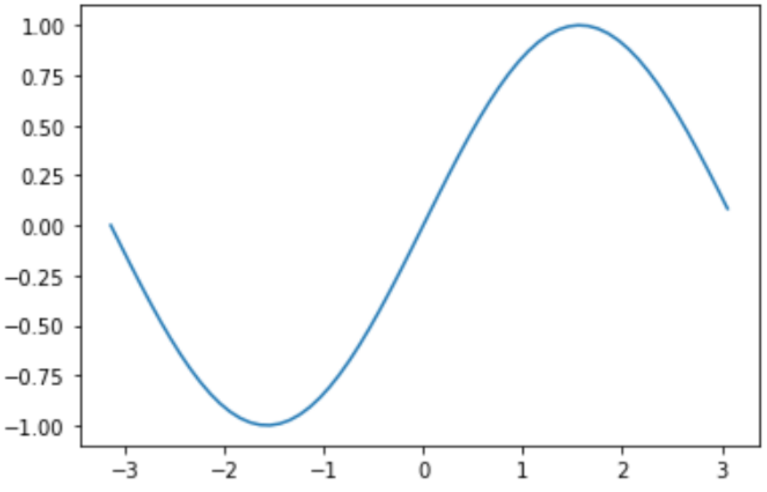
ーπ～πまで0.1刻みの数列作成　np.pi

```
[-3.14159265 -3.04159265 -2.94159265 -2.84159265 -2.74159265 -2.64159265
 -2.54159265 -2.44159265 -2.34159265 -2.24159265]
[-1.22464680e-16 -9.98334166e-02 -1.98669331e-01 -2.95520207e-01
 -3.89418342e-01 -4.79425539e-01 -5.64642473e-01 -6.44217687e-01
 -7.17356091e-01 -7.83326910e-01]
```

←y=np.sin(x)

```
<function matplotlib.pyplot.show(close=None, block=None)>
```

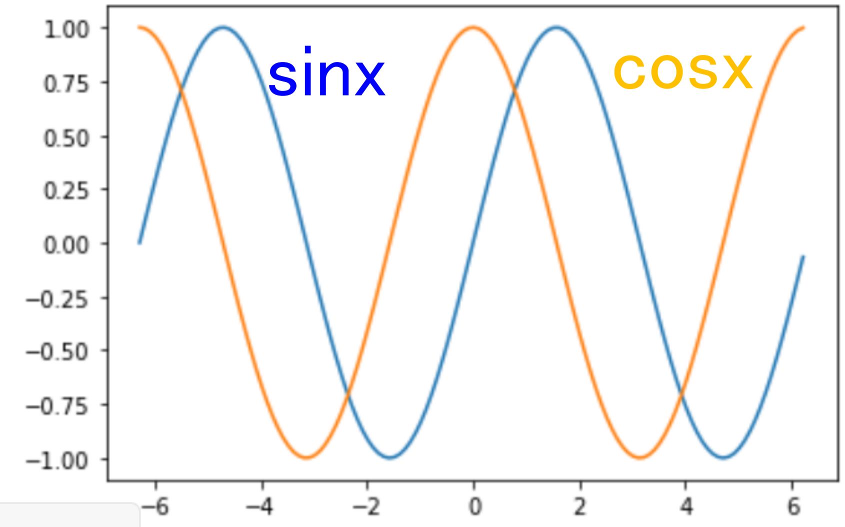
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



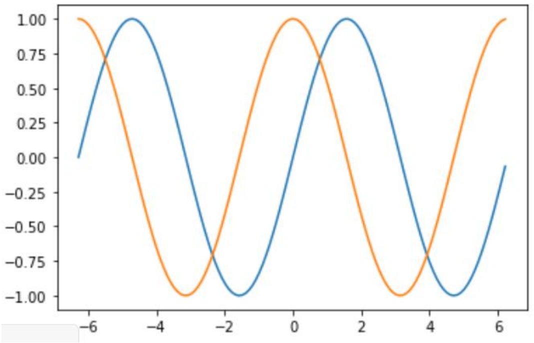
複数の曲線

```python
import numpy as np
import matplotlib.pyplot as plt
x=np.arange(-2*np.pi,2*np.pi,0.1)
y0=np.sin(x)
y1=np.cos(x)
print(x[:10])
print(y[:10])

plt.plot(x,y0)
plt.plot(x,y1)
plt.show
```

```
[-6.28318531 -6.18318531 -6.08318531 -5.98318531 -5.88318531 -5.78318531
 -5.68318531 -5.58318531 -5.48318531 -5.38318531]
[-1.22464680e-16 -9.98334166e-02 -1.98669331e-01 -2.95520207e-01
 -3.89418342e-01 -4.79425539e-01 -5.64642473e-01 -6.44217687e-01
 -7.17356091e-01 -7.83326910e-01]

<function matplotlib.pyplot.show(close=None, block=None)>
```

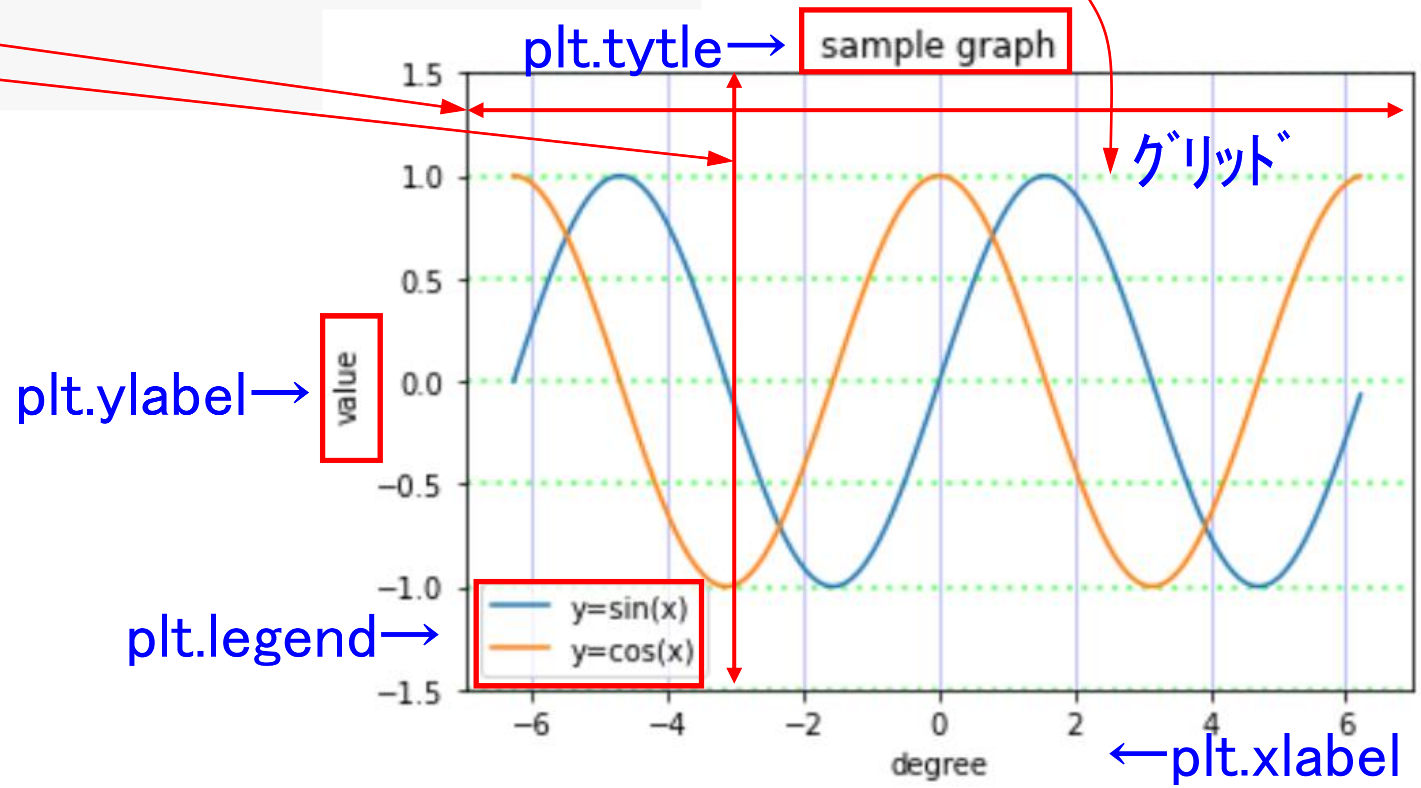


```python
import numpy as np
import matplotlib.pyplot as plt
x=np.arange(-2*np.pi,2*np.pi,0.1)
y0=np.sin(x)
y1=np.cos(x)
print(x[:10])
print(y[:10])

plt.plot(x,y0)
plt.plot(x,y1)
plt.show
```



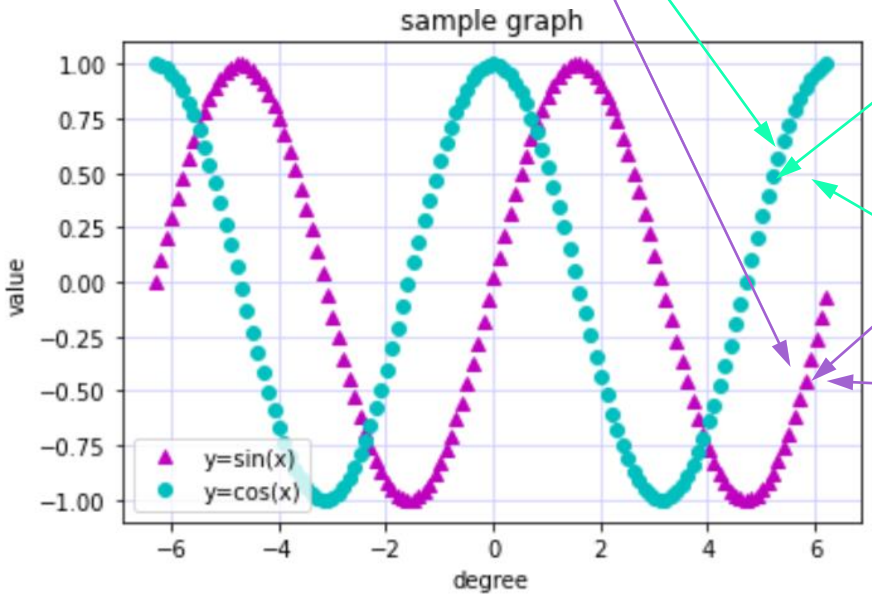
```python
import numpy as np
import matplotlib.pyplot as plt
x=np.arange(-2*np.pi,2*np.pi,0.1)
y0=np.sin(x)
y1=np.cos(x)
print(x[:10])
print(y[:10])

plt.plot(x,y0,label='y=sin(x)')
plt.plot(x,y1,label='y=cos(x)')
plt.legend()

plt.title('sample graph')
plt.xlabel('degree')
plt.ylabel('value')

plt.grid(which='both',axis='x',color='#0000ff',alpha=0.25,linestyle='-',linewidth=1)
plt.grid(which='major',axis='y',color='#00ff00',alpha=0.5,linestyle=':',linewidth=2)

plt.xlim([-7,7])
plt.ylim([-1.5,1.5])
plt.show()
```

color　ｸﾞﾘｯﾄﾞの色、6桁の16進数
alpha　透過度、0～1の実数
which　major、minorとboth
axis　描画方向
linestyle　‘-’は直線　‘:’は点線
linewidth　ラインの太さ



```python
import numpy as np
import matplotlib.pyplot as plt
x=np.arange(-2*np.pi,2*np.pi,0.1)
y0=np.sin(x)
y1=np.cos(x)

plt.plot(x,y0,'m^',label='y=sin(x)')
plt.plot(x,y1,'co',label='y=cos(x)')
plt.legend()

plt.title('sample graph')
plt.xlabel('degree')
plt.ylabel('value')

plt.grid(which='both',color='#ccccff')

plt.show()
```

タイトル
xラベル
yラベル



r	赤
b	青
g	緑
c	シアン
k	黒
m	マジェンダ
y	黄色

‘-’	実線
--’	破線
‘:’	点線
‘-.’	一点鎖線
‘s’	四角形
‘^’	三角形
‘o’	円

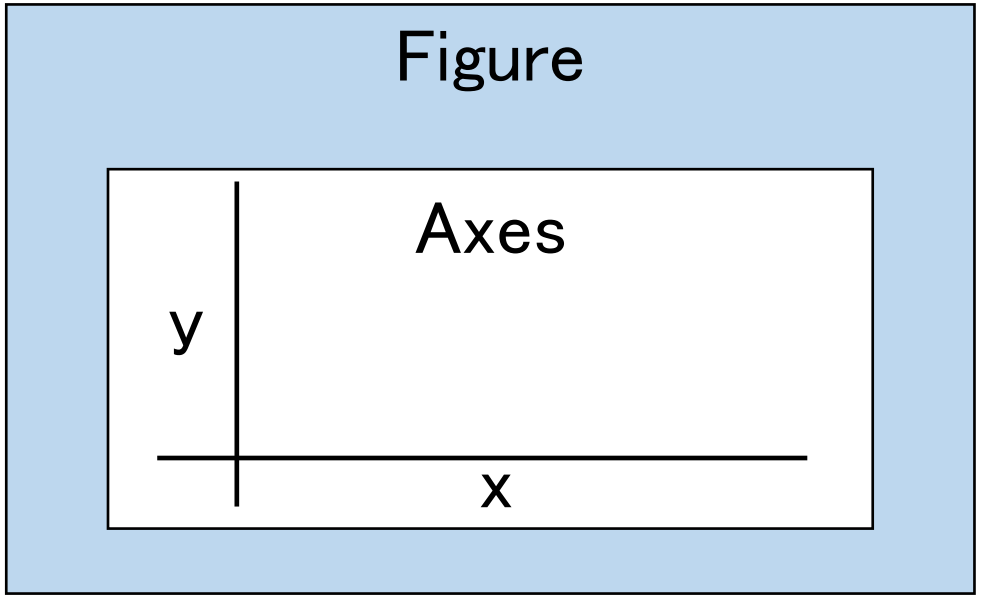


Figure
Axes
y
x

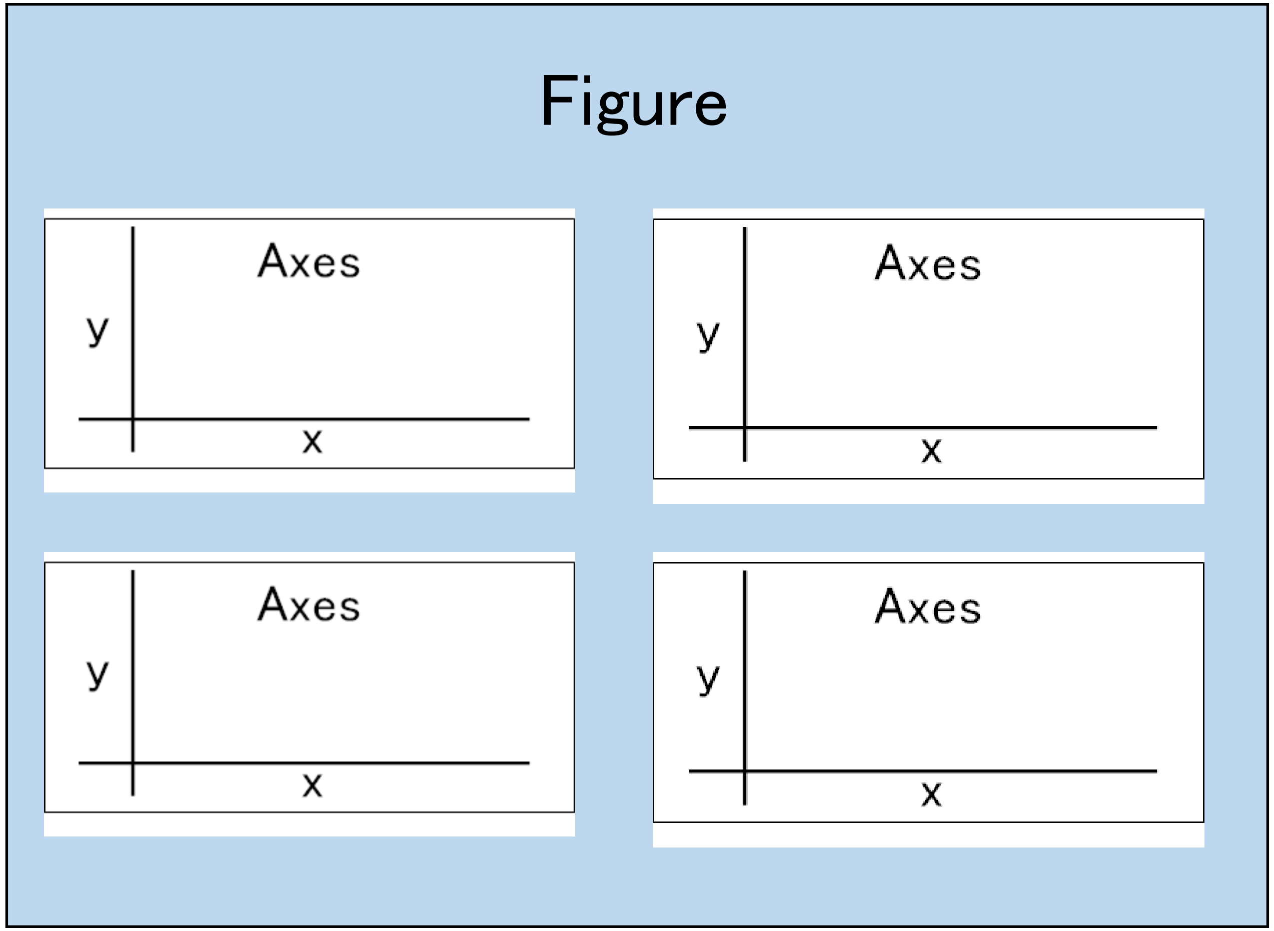


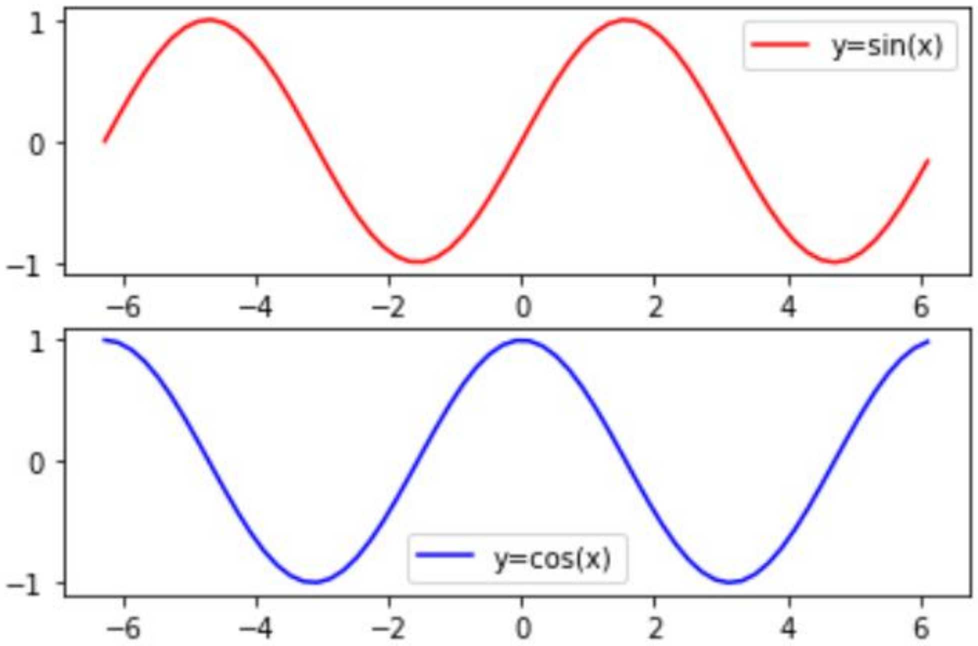
Figure
Axes
y
x
Axes
y
x
Axes
y
x
Axes
y
x

変数、(変数)＝plt. subplots（整数、整数）

```
import numpy as np
import matplotlib.pyplot as plt
x=np.arange(-2*np.pi,2*np.pi,0.2)
y0=np.sin(x)
y1=np.cos(x)

fig,(p_a1,p_a2)=plt.subplots(2,1)
p_a1.plot(x,y0,'r',label='y=sin(x)')
p_a1.legend()
p_a2.plot(x,y1,'b',label='y=cos(x)')
p_a2.legend()
plt.show()
```

x方向に2つ

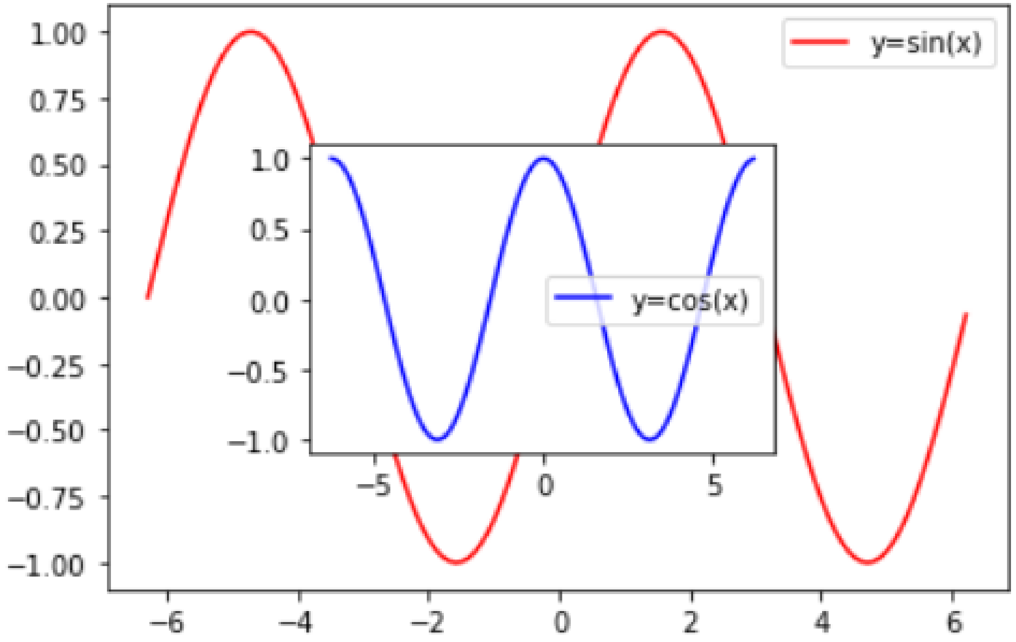
x方向に2つ



Figureに追加

変数＝plt. axes（[横位置、縦位置、幅、高さ]）

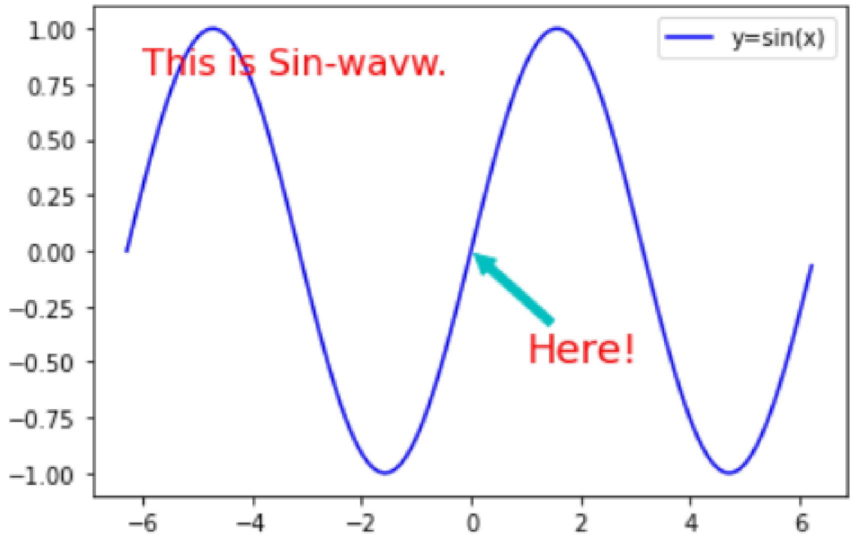
```
import numpy as np
import matplotlib.pyplot as plt
x=np.arange(-2*np.pi,2*np.pi,0.1)
y0=np.sin(x)
y1=np.cos(x)

plt.plot(x,y0,'r',label='y=sin(x)')
plt.legend()
ax2=plt.axes([0.3,0.3,0.4,0.4])
ax2.plot(x,y1,'b',label='y=cos(x)')
plt.legend()

plt.show()
```



テキストに追加

変数＝plt. text（X値、Y値、テキスト）

```
import numpy as np
import matplotlib.pyplot as plt
x=np.arange(-2*np.pi,2*np.pi,0.1)
y=np.sin(x)

plt.plot(x,y,'b',label='y=sin(x)')
plt.legend()
plt.text(-6,0.8,'This is Sin-wavw.',fontsize=16,color='r')

plt.annotate('Here!',xy=(0,0),xytext=(1,-0.5),
             arrowprops=dict(arrowstyle='simple',color='c'),
             fontsize=18,color='r')
plt.show()
```



矢印追加

xy＝plt. arrow（X値、Y値、 X値、Y値）

注釈追加

変数＝plt. annotate（テキスト）

矢印の先端位置

変数＝（X値、Y値）

テキスト位置

xytext＝（X値、Y値）

フォントサイズ

fontsize＝ 数値

テキストの色

color＝ 色の指定

矢印の細かな設定

arrowprps＝ dict（属性＝値、・・・）

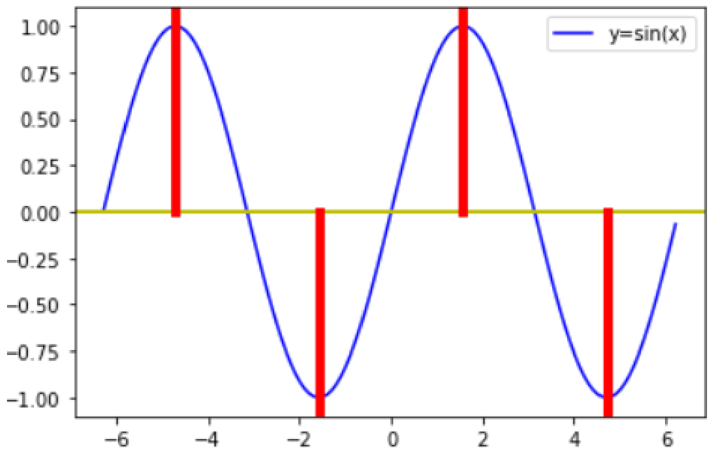
線の描画

plt. axhline（y＝値）　plt. axvline（x＝値）

alpha　透明度0～1の実数

xmin,xmax　axhlineの最小値と最大値
ymin,ymax　axvlineの最小値と最大値

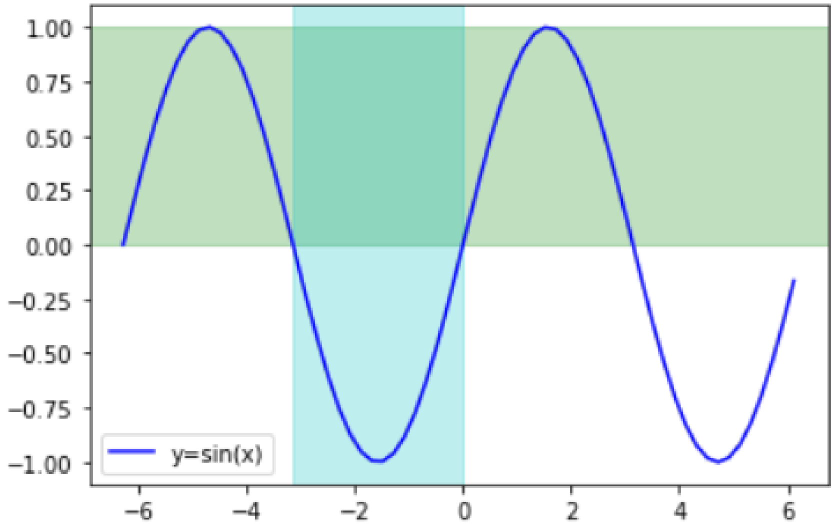
```
import numpy as np
import matplotlib.pyplot as plt
x=np.arange(-2*np.pi,2*np.pi,0.1)
y=np.sin(x)

plt.plot(x,y,'b',label='y=sin(x)')
plt.legend()

plt.axhline(y=0.,linewidth=2,color='y')
plt.axvline(x=-np.pi*1.5,ymin=0.5,ymax=1.,linewidth=5,color='r')
plt.axvline(x=-np.pi*0.5,ymin=0.,ymax=0.5,linewidth=5,color='r')
plt.axvline(x=np.pi*0.5,ymin=0.5,ymax=1.,linewidth=5,color='r')
plt.axvline(x=np.pi*1.5,ymin=0.,ymax=0.5,linewidth=5,color='r')
plt.show()
```



塗りつぶし

plt. axhspan（Y最小値、Y最大値）
plt. axvspan（X最小値、X最大値）

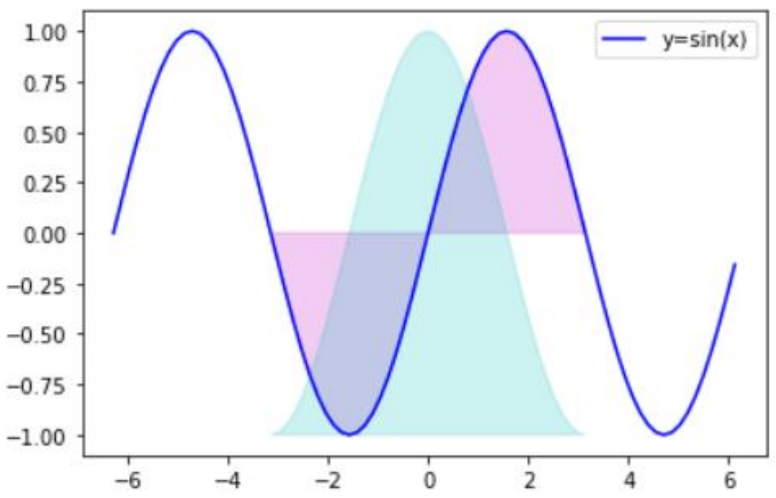
```
import numpy as np
import matplotlib.pyplot as plt
x=np.arange(-2*np.pi,2*np.pi,0.2)
y=np.sin(x)

plt.plot(x,y,'b',label='y=sin(x)')
plt.legend()

plt.axhspan(0.,1.,color='g',alpha=0.25)
plt.axvspan(-np.pi,0.,color='c',alpha=0.25)
plt.show()
```



指定領域塗りつぶし

plt. fill([Xリスト]、[Yリスト])

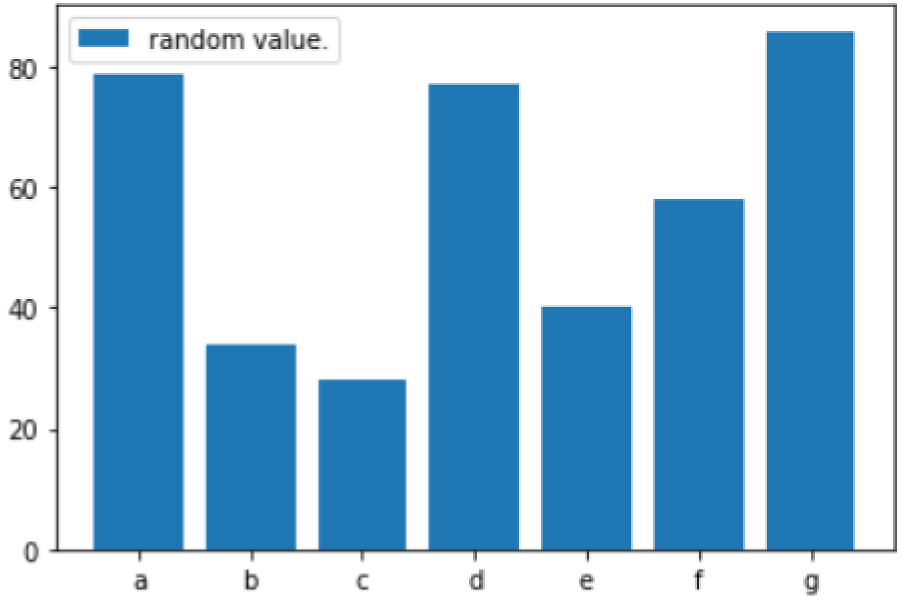
```
import numpy as np
import matplotlib.pyplot as plt
x=np.arange(-2*np.pi,2*np.pi,np.pi/20)
y=np.sin(x)

s_x=np.arange(-np.pi,np.pi+0.001,np.pi/20)
s_y=np.sin(s_x)
c_x=np.arange(-np.pi,np.pi+0.001,np.pi/20)
c_y=np.cos(c_x)
plt.plot(x,y,'b',label='y=sin(x)')
plt.legend()
plt.fill(s_x,s_y,color='m',alpha=0.2)
plt.fill(c_x,c_y,color='c',alpha=0.2)
plt.show()
```



棒グラフ

plt. bar([Xデータ]、[Yデータ])

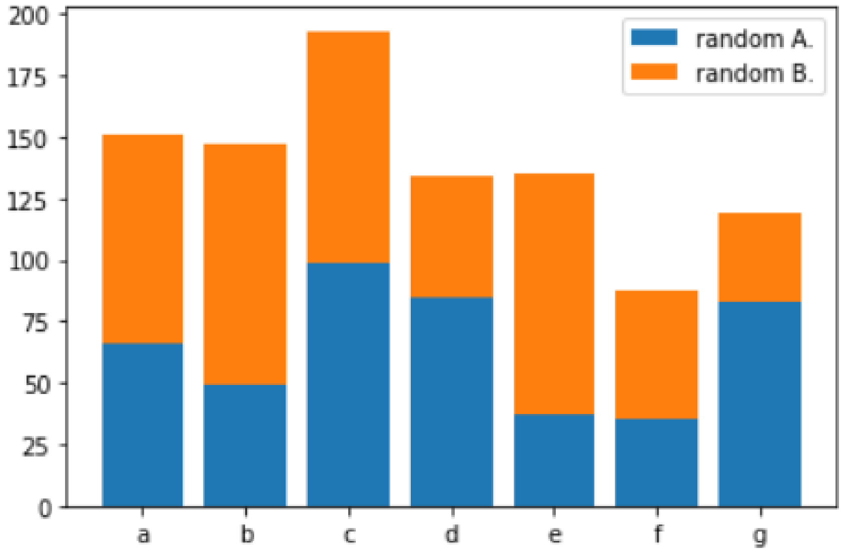
```
import numpy as np
import matplotlib.pyplot as plt

x=list('abcdefg')
y=np.array([np.random.randint(75)+25 for i in range(7)])

plt.bar(x,y,label='random value.')
plt.legend()
plt.show()
```



棒グラフを重ねる

bottomで棒の下端を指定

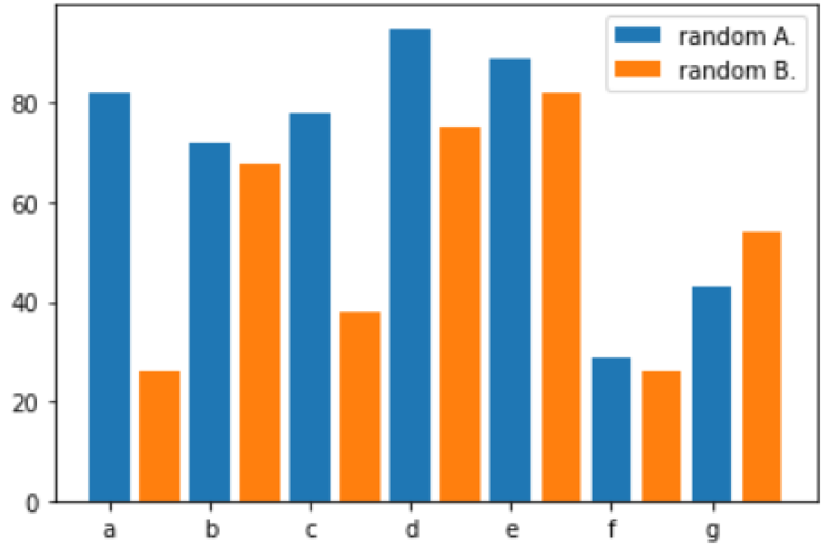
```
import numpy as np
import matplotlib.pyplot as plt

x=list('abcdefg')
y0=np.array([np.random.randint(75)+25 for i in range(7)])
y1=np.array([np.random.randint(75)+25 for i in range(7)])

plt.bar(x,y0,label='random A.')
plt.bar(x,y1,bottom=y0,label='random B.')
plt.legend()
plt.show()
```



棒グラフを並べる

tick_label　でX軸のラベル設定

```
import numpy as np
import matplotlib.pyplot as plt

x0=np.arange(0,13,2)
x1=np.arange(1,14,2)
y0=np.array([np.random.randint(75)+25 for i in range(7)])
y1=np.array([np.random.randint(75)+25 for i in range(7)])
lb=list('abcdefg')
plt.bar(x0,y0,tick_label=lb,label='random A.')
plt.bar(x1,y1,label='random B.')
plt.legend()
plt.show()
```

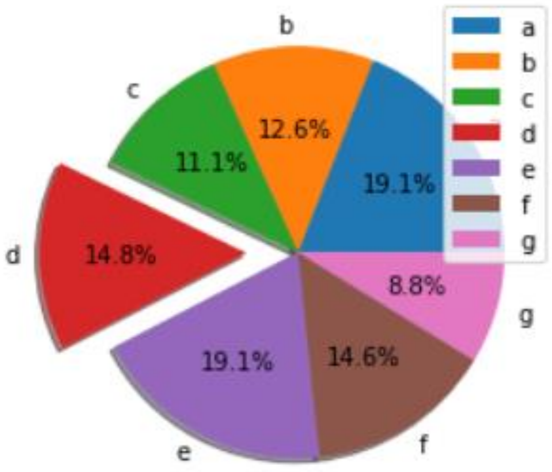


円グラフを描く

plt. pie（データ）

shadow　Trueで影を付ける
explode　離して描く
startangle　ﾃﾞﾌｫﾙﾄでは3時の方向が最初
autopct　パーセント表示　‘%1.nf%%’

```
import numpy as np
import matplotlib.pyplot as plt

x=np.array([np.random.randint(75)+25 for i in range(7)])
y=list('abcdefg')
ex=[0,0,0,0.25,0,0,0]

plt.pie(x,labels=y,shadow=True,autopct='%1.1f%%',explode=ex)
plt.legend()
plt.show()
```



散布図

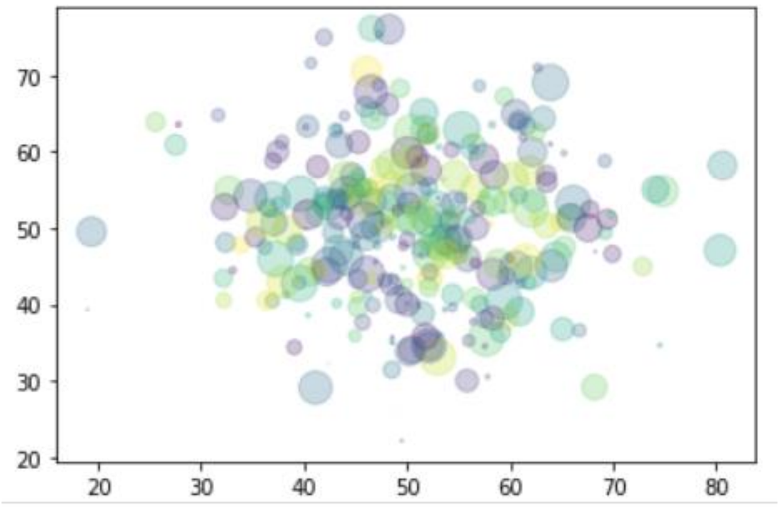
plt. scatter（Xデータ、Yデータ）

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import norm

(n,sigma,mu)=(300,10,50)
x=np.random.randn(n)*sigma+mu
y=np.random.randn(n)*sigma+mu
c=np.random.rand(n)
s=np.random.rand(n)**2 * np.pi * 100
plt.scatter(x,y,s=s,c=c,alpha=0.25)
plt.show()
```

s　ｻｲｽﾞﾃﾞｰﾀ
c　色ﾃﾞｰﾀ
marker　図形の形状

alphaは透明性

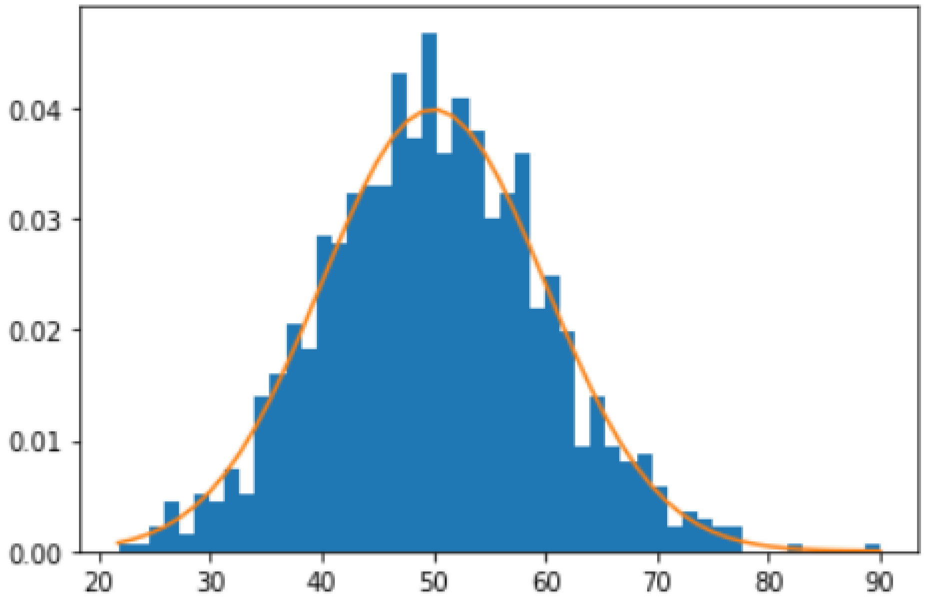
plt. pcolormesh（ Xデータ、Yデータ、色ﾃﾞｰﾀ）



ヒストグラムを描く

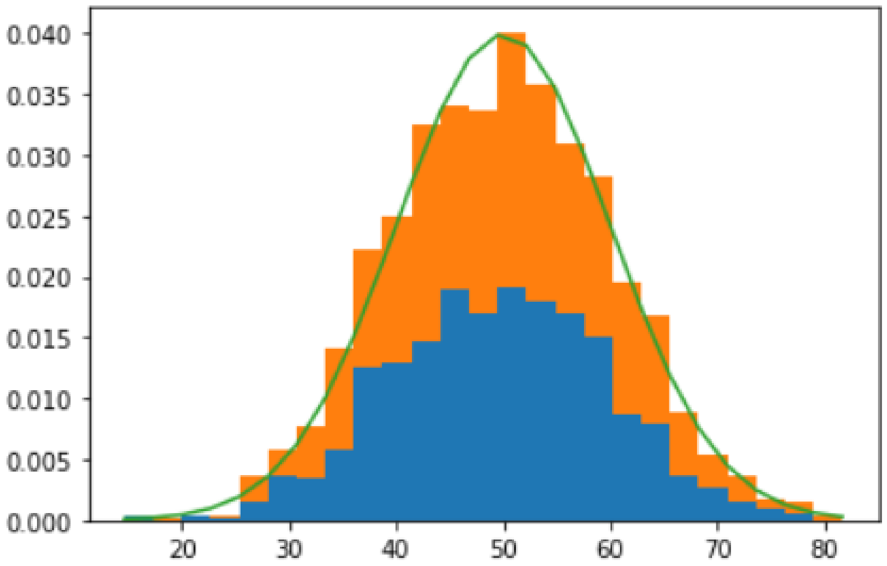
（n,bin,patches）＝plt.hist（データ、分割数）

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import norm
(sigma,mu)=(10,50)
value=np.random.randn(1000)*sigma+mu
(n,bins,patches)=plt.hist(value,50,density=True)
plt.plot(bins,norm.pdf(bins,loc=mu,scale=sigma))
plt.show()
```



range　グラフの範囲
orientation　グラフの方向
stacked　データの積み重ね
density　確率密度関数の近似値
histtype　バーのスタイル

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import norm
(sigma,mu)=(10,50)
value0=np.random.randn(1000)*sigma+mu
value1=np.random.randn(1000)*sigma+mu
(n,bins,patches)=plt.hist([value0,value1],25,stacked=True,density=True)
plt.plot(bins,norm.pdf(bins,loc=mu,scale=sigma))
plt.show()
```



戻り値
n　各ﾊﾞｰのデータ数のﾘｽﾄ
bins　ﾊﾞｰの境界値のﾘｽﾄ
分割数10なら11個
patches　書くﾊﾞｰのｵﾌﾞｼﾞｪｸﾄ

density＝True
scipy.starsのnorm.pdf関数

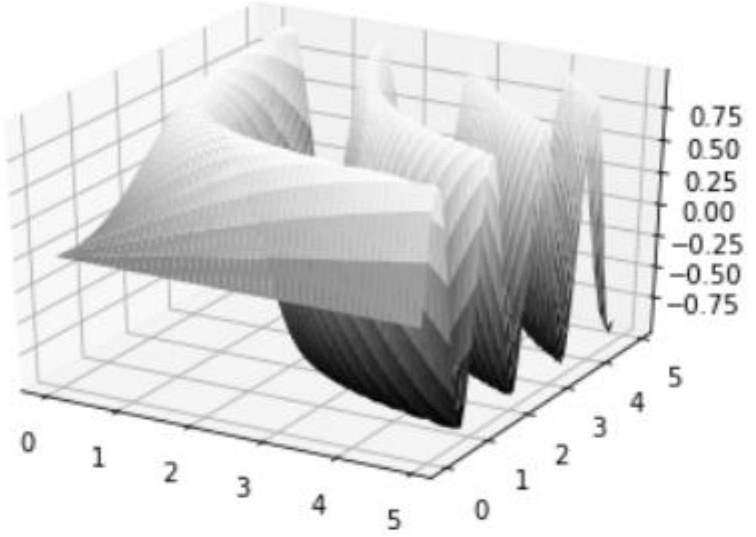
norm.pdf関数は因数にbins指定で確率密度を計算
曲線の位置をlocとscaleで調整

3Dグラフを描く

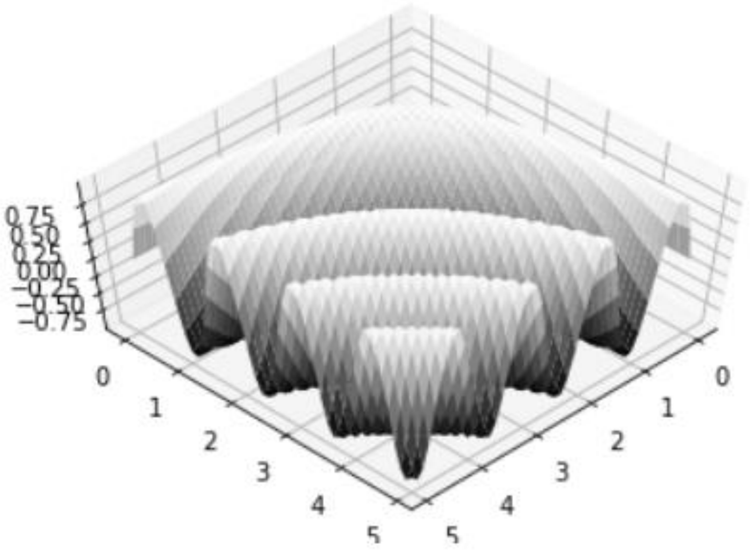
サーフェースで描く　ax.plot_surface(Xﾃﾞｰﾀ、Yﾃﾞｰﾀ、Zﾃﾞｰﾀ)
ワイヤーフレームで描く　ax.plot_wireframe(Xﾃﾞｰﾀ、Yﾃﾞｰﾀ、Zﾃﾞｰﾀ)

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

ax=plt.figure().gca(projection='3d')

x0=np.arange(0,5,0.1)
y0=np.arange(0,5,0.1)
(x,y)=np.meshgrid(x0,y0)
z=np.sin(x*y)

surf=ax.plot_surface(x,y,z,cmap='gray',antialiased=True)
plt.show()
```

antialiased=True なめらかに描く




```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

ax=plt.figure().gca(projection='3d')

x0=np.arange(0,5,0.1)
y0=np.arange(0,5,0.1)
(x,y)=np.meshgrid(x0,y0)
z=np.sin(x*y)

surf=ax.plot_surface(x,y,z,cmap='gray',antialiased=True)
ax.view_init(60,45)
plt.show()
```

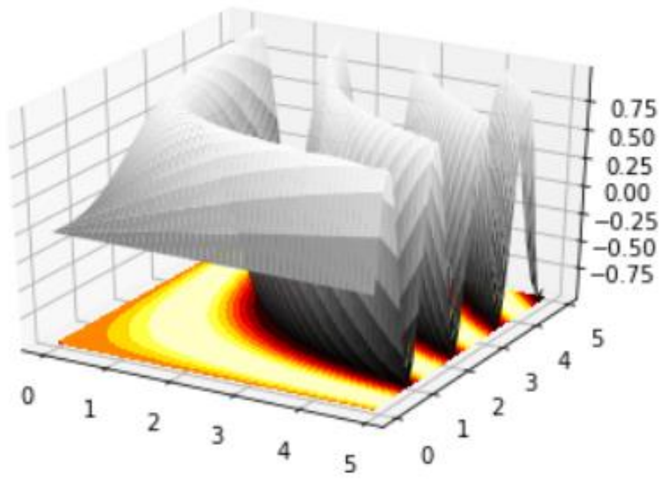


ax.view_init(縦角度、回転角度)

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

ax=plt.figure().gca(projection='3d')

x0=np.arange(0,5,0.1)
y0=np.arange(0,5,0.1)
(x,y)=np.meshgrid(x0,y0)
z=np.sin(x*y)

surf=ax.plot_surface(x,y,z,cmap='gray',antialiased=True)
ax.contourf(x,y,z,zdir='z',offset=-1.,cmap='hot')
plt.show()
```

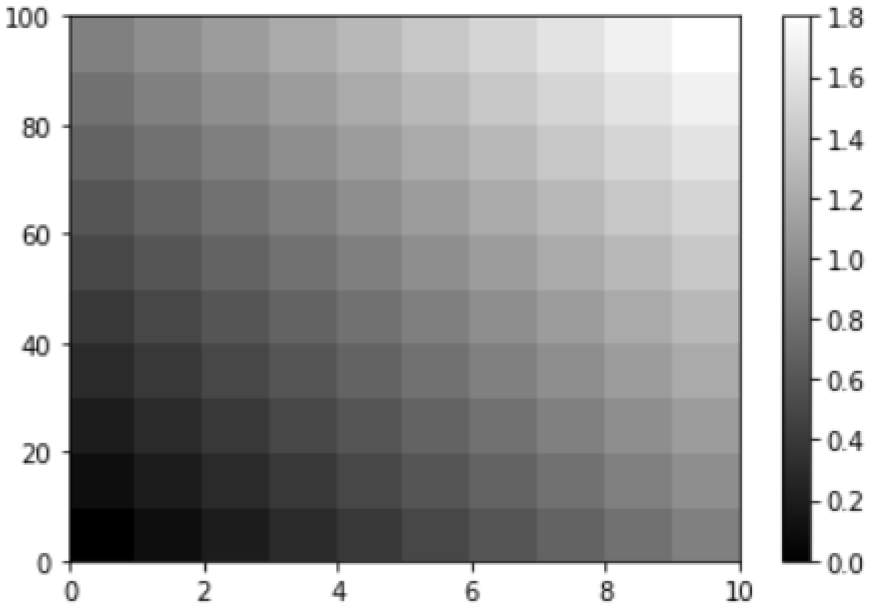
ax.contourf(Xﾃﾞｰﾀ、Yﾃﾞｰﾀ、Zﾃﾞｰﾀ)

zdir　投影する面（x、y、z）を指定
offset　面からの離れ度合



グラデーション作成

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import norm

x=np.arange(0,11,1)
y=np.arange(0,110,10)
c=[np.arange(n/10,1.+n/10,0.1)for n in range(0,10)]
plt.pcolormesh(x,y,c,cmap='gray',vmin=0,vmax=1.8)
plt.colorbar()
plt.show()
```



cmap

カラーマップ名
autumn,bone,copper,flag,gray,hot,jet,pink,prism,spring,
summer,winter