matplotlib

■グラフ作成の基本

matplotlib の最も重要な働きは、グラフを作成することです。この機能は「pyplot」というモジュールを使います。

```
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()
```

※プログラム内で numpy を使っているのはデータを作成するため

◆sin カーブを描く

-π からπまでの間を 0.1 刻みで描く

```
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()
```

◆複数のグラフを同時に描く

```
x = np.arange(-2*np.pi, 2*np.pi, 0.1)
y0 = np.sin(x)
y1 = np.cos(x)

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

plot を 2 回呼び出し実行している。

※以下、import の表示をしていない

◆凡例表示

```
x = np.arange(-2*np.pi, 2*np.pi, 0.1)
y0 = np.sin(x)
y1 = np.cos(x)

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

◆タイトル、XY 軸のラベル表示[1]

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

[1] 日本語は NG

◆グリッド表示

```
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, 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.show()
```

plt.grid の引数

color：グリッドの色

alpha：透明度

which：メジャーな線（major）、マイナーな線（minor）、両方（both）

axis：描画する方向

linestyle：「－」直線、「：」点線など

linewidth：ラインの太さ

◆表示エリア

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

・X 方向の表示エリアを設定

```
plt.xlim([xmin, xmax])
```

・Y 方向の表示エリアを設定

```
plt.ylim([ymin, ymax])
```

◆課題

上記「表示エリア」を使って、各グラフの「頂点」、「交差点」を拡大表記する

◆2 つのグラフに分割する

```
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()
```

◆矢印の追加

```
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.arrow(2.,0., -1.5, 0., width=0.05,
                head_width=0.2, head_length=0.5, color='y')
plt.show()
```

arrow（X 値、Y 値、X 幅、Y 幅、width、矢印の幅、矢印の長さ、色）

◆テキストの追加

```
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.text(-6, 0.8, 'This is Sample!', fontsize=16, color='r')

plt.show()
```

上の「矢印」に説明を付けるために使われる

◆矢印を一定幅での塗りつぶし

```
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()
```

◆指定領域の塗りつぶし

```
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()
```

■いろいろなグラフ

■棒グラフ

```
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()
```

◆棒グラフを積み上げる

```
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()
```

■円グラフを描く

```
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()
```

円グラフの引数

labels：ラベル

shadow：ture で影付き

explode：特定の部分を少し円から話す

startangle：スタート位置。デフォルトは 3 時方向から左回り

autopct：パーセント値を表示

radius：半径

■ヒストグラム

```
(sigma, mu) = (10, 50)
value = np.random.randn(1000)*sigma+mu
plt.hist(value, 25)
plt.show()
```

◆ヒストグラムに確率密度曲線の表示

```
(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()
```

※from scipy.stats import norm が必要

◆散布図

```
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()
```

◆3D グラフ

```
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()
```

※gca の代わりに、add_subplot