

ロジスティック回帰モデル 追加レポート

参考サイト：

- [python で等高線を描くなら meshgrid して contour \(http://ailaby.com/contour/\)](http://ailaby.com/contour/)
- [matplotlibでの描写 等高線プロットの基礎をわかりやすく説明してみた \(https://qiita.com/kazukiii/items/8f0e9559524c0bb0f5ba\)](https://qiita.com/kazukiii/items/8f0e9559524c0bb0f5ba)

In [1]:

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

等高線の描き方

1. 等間隔の格子点リストを作成する。格子点は、(X, Y)座標と格子点の高さZが必要。
2. 等高線を描画する。

In [2]:

```
x = np.linspace(0, 2, 3) # 等間隔の一次元配列を2つ作成する。今回の例はx, yの2つ。
print(x)
```

```
[0. 1. 2.]
```

In [3]:

```
y = np.linspace(0, 10, 3)
print(y)
```

```
[ 0.  5. 10.]
```

In [4]:

```
X, Y = np.meshgrid(x, y) # 格子点となる二次元配列を作成する。
```

In [5]:

```
print(X)
```

```
[[0. 1. 2.]
 [0. 1. 2.]
 [0. 1. 2.]]
```

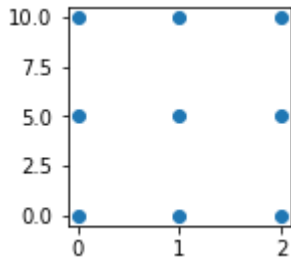
In [6]:

```
print(Y)
```

```
[[ 0.  0.  0.]
 [ 5.  5.  5.]
 [10. 10. 10.]]
```

In [7]:

```
# 散布図により、格子点を描画する。
plt.figure(figsize=(2, 2))
plt.scatter(X, Y)
plt.show()
```



`np.meshgrid` により、上記のような格子点を作成できる。
等高線を描画するには、`plt.contour` を使用する。
等高線に必要な情報として、格子点の高さ Z を決める必要がある。

In [8]:

```
Z = np.sqrt(X**2 + Y**2) # 格子点の高さは、原点からの距離とする。
```

In [14]:

```
print(Z)
```

```
[[ 0.          5.09901951 10.19803903]
 [ 0.          5.09901951 10.19803903]
 [ 0.          5.09901951 10.19803903]]
```

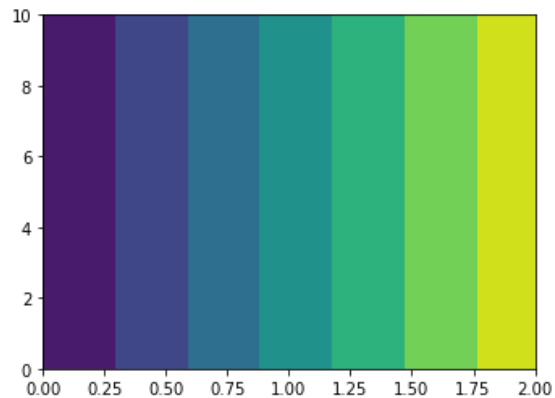
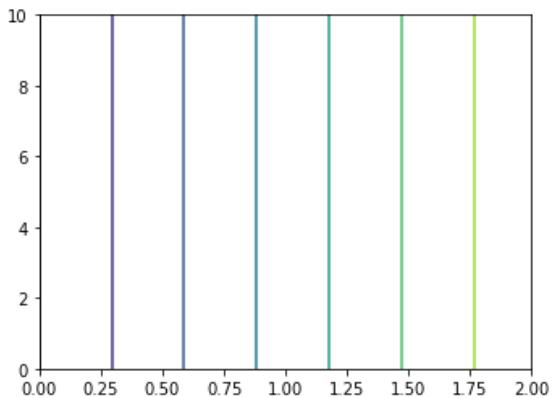
In [25]:

```
fig = plt.figure(figsize=(12, 4))

fig.add_subplot(1,2,1)
plt.contour(X, Y, Z) # 等高線

fig.add_subplot(1,2,2)
plt.contourf(X, Y, Z) # 色付き等高線

plt.show()
```



In [11]:

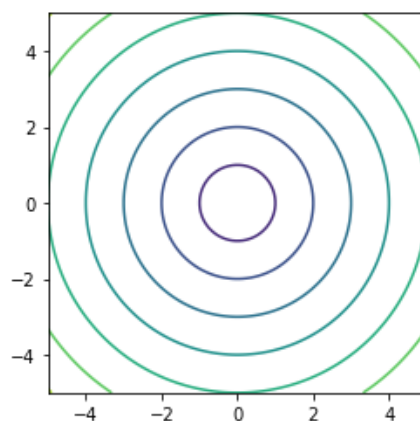
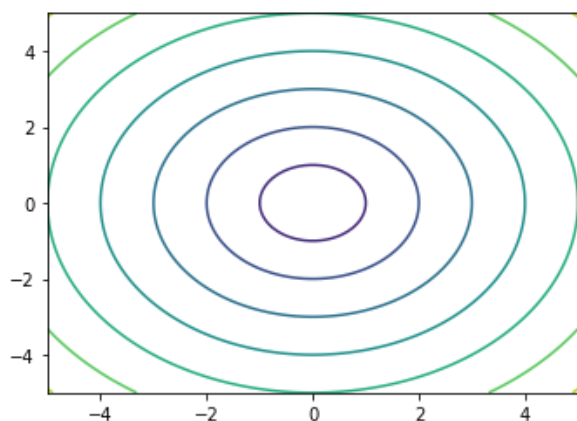
```
x2 = np.linspace(-5, 5, 100) # 格子点が多いほど、等高線は滑らかになる。今回は100を設定。  
y2 = np.linspace(-5, 5, 100)  
X2, Y2 = np.meshgrid(x2, y2) # 格子点の作成（描画しようとしても格子点が多くただの四角に見える）
```

In [12]:

```
Z2 = np.sqrt(X2**2 + Y2**2) # 格子点の高さは、原点からの距離とする。
```

In [13]:

```
fig = plt.figure(figsize=(12, 4))  
  
fig.add_subplot(1,2,1) # 等高線の描画  
plt.contour(X2, Y2, Z2)  
  
fig.add_subplot(1,2,2)  
plt.contour(X2, Y2, Z2)  
plt.gca().set_aspect('equal') # アスペクト比を1:1にする  
  
plt.show()
```



In []: