

# 程式設計 (二)

Matplotlib 的應用

Ming-Hung Wang 王銘宏

tonymhwang@cs.ccu.edu.tw

Department of Computer Science and Information Engineering  
National Chung Cheng University

Spring Semester, 2022

# 本章目錄

1. Matplotlib 介紹
2. 基本使用
3. 畫圖種類
4. 多圖合併顯示

# 為什麼要用 Matplotlib?

如果某天你發現自己要學習 Matplotlib，很可能是因為：

- 手中有許多數據，可是不知道該怎麼呈現這些數據。
- Matplotlib 是一個非常強大的 Python 畫圖工具。

所以就使用了 Matplotlib，它可以幫你畫出美麗的圖形，  
像是：線圖，散點圖，等高線圖... 等。

## 安裝 Matplotlib

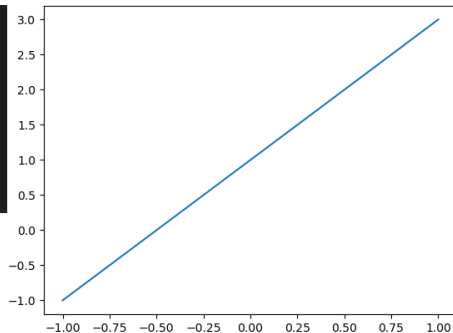
**`pip install matplotlib`**

Anaconda 預設已經安裝了。

## 基礎應用

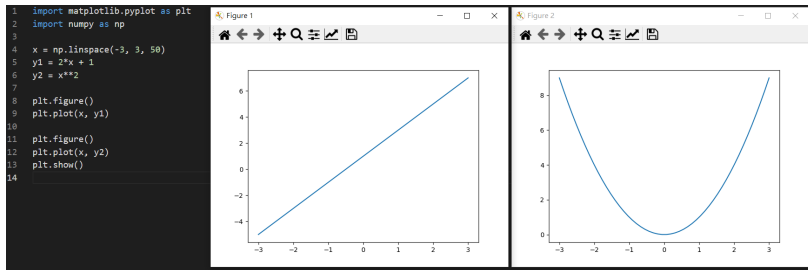
試試看如何畫出一條  $y = 2*x + 1$  的直線。

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 x = np.linspace(-1, 1, 50)
5 y = 2*x + 1
6
7 plt.figure()
8 plt.plot(x, y)
9 plt.show()
```



## Figure 圖像

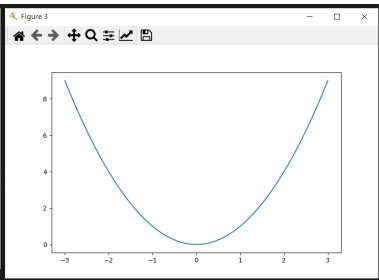
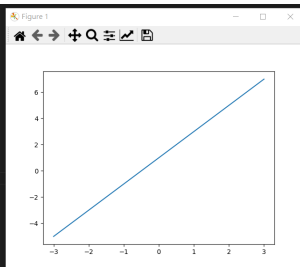
matplotlib 的 figure 就是一個單獨的 figure 小窗口，小窗口裡面還可以有更多的小圖片。



# 基本使用

在 `plt.figure` 裡加入一些設定，可以讓顯示的圖像有些許不一樣。

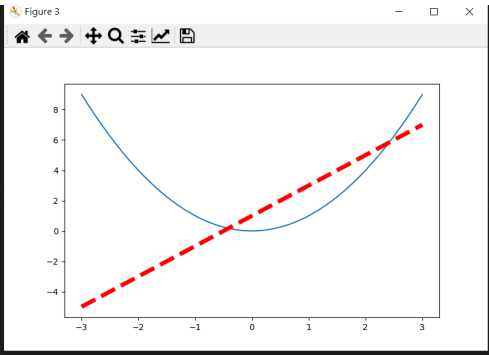
```
1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 x = np.linspace(-3, 3, 50)
5 y1 = 2*x + 1
6 y2 = x**2
7
8 plt.figure()
9 plt.plot(x, y1)
10
11 plt.figure(num = 3, figsize=(8, 5))
12 plt.plot(x, y2)
13 plt.show()
14
```



# 基本使用

使用 `plt.plot` 畫  $(x, y)$  曲線，可以加入一些曲線顏色、寬度、類型的設定。

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 x = np.linspace(-3, 3, 50)
5 y1 = 2*x + 1
6 y2 = x**2
7
8 plt.figure(num=3, figsize=(8, 5))
9 plt.plot(x, y2)
10 plt.plot(x, y1, color='red', linewidth=5.0, linestyle='--')
11 plt.show()
12
```

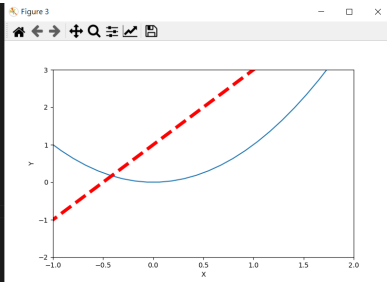




## 設置坐標軸

使用 `plt.xlim` 來設置  $x$  座標軸的範圍，`plt.ylim` 設置  $y$  座標軸的範圍，`plt.xlabel` 設置  $x$  座標軸名稱，`plt.ylabel` 設置  $y$  座標軸的名稱。

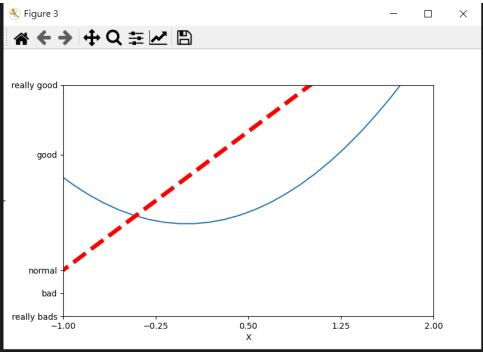
```
1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 x = np.linspace(-3, 3, 50)
5 y1 = 2*x + 1
6 y2 = x**2
7
8 plt.figure(num=3, figsize=(8, 5))
9 plt.plot(x, y2)
10 plt.plot(x, y1, color='red', linewidth=5.0, linestyle='--')
11
12 plt.xlim(-1, 2)
13 plt.ylim(-2, 3)
14 plt.xlabel('X')
15 plt.ylabel('Y')
16 plt.show()
17
```



# 基本使用

使用 `plt.xticks` 來設置  $x$  軸刻度，`plt.yticks` 來設置  $y$  軸的刻度，也能夠設置名稱。

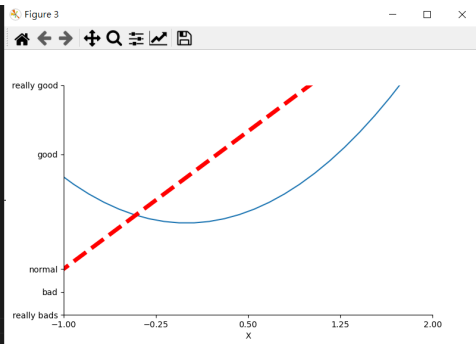
```
1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 x = np.linspace(-3, 3, 50)
5 y1 = 2*x + 1
6 y2 = x**2
7
8 plt.figure(num=3, figsize=(8, 5))
9 plt.plot(x, y2)
10 plt.plot(x, y1, color='red', linewidth=5.0, linestyle='--')
11
12 plt.xlim((-1, 2))
13 plt.ylim((-2, 3))
14 new_ticks = np.linspace(-1, 2, 5)
15 plt.xticks(new_ticks)
16 plt.yticks([-2, -1.5, -1, 1.5, 3], ['really bads',
17                                     'bad', 'normal', 'good', 'really good'])
18 plt.xlabel('X')
19 plt.ylabel('Y')
20
21 plt.show()
```



# 基本使用

使用 `plt.gca` 來獲取當前座標軸信息，`.spines` 設置邊框，  
`.set_color` 設置邊框顏色

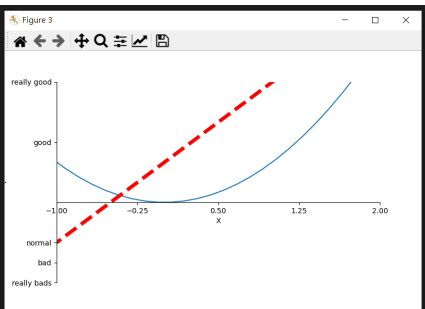
```
1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 x = np.linspace(-3, 3, 50)
5 y1 = 2*x + 1
6 y2 = x**2
7
8 plt.figure(num=3, figsize=(8, 5))
9 plt.plot(x, y2)
10 plt.plot(x, y1, color='red', linewidth=5.0, linestyle='--')
11
12 plt.xlim((-1, 2))
13 plt.ylim((-2, 3))
14 new_ticks = np.linspace(-1, 2, 5)
15 plt.xticks(new_ticks)
16 plt.yticks([-2, -1.5, -1, 1.5, 3], ['really bads',
17                                     'bad', 'normal', 'good', 'really good'])
18 plt.xlabel('X')
19 plt.ylabel('Y')
20
21 ax = plt.gca()
22 ax.spines['right'].set_color('none')
23 ax.spines['top'].set_color('none')
24
25 plt.show()
```



# 基本使用

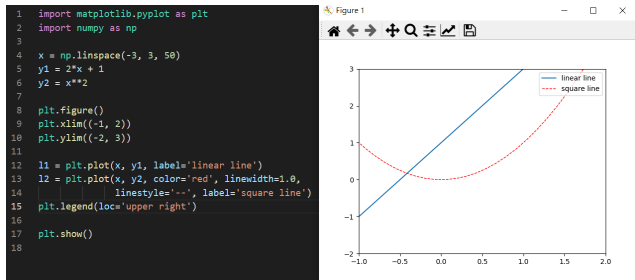
使用 `.axis.set_ticks_position` 來設置  $x$  座標刻度數字和名稱位置，`.axis.set_ticks_position` 則是設置  $y$  座標刻度數字和名稱位置。

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 x = np.linspace(-3, 3, 50)
5 y1 = 2*x + 1
6 y2 = x**2
7
8 plt.figure(num=3, figsize=(8, 5))
9 plt.plot(x, y2)
10 plt.plot(x, y1, color='red', linewidth=5.0, linestyle='--')
11
12 plt.xlim((-1, 2))
13 plt.ylim((-2, 3))
14 new_ticks = np.linspace(-1, 2, 5)
15 plt.xticks(new_ticks)
16 plt.yticks([-2, -1.5, -1, 1.5, 3], ['really bads',
17 |         'bad', 'normal', 'good', 'really good'])
18 plt.xlabel('X')
19 plt.ylabel('Y')
20
21 ax = plt.gca()
22 ax.spines['right'].set_color('none')
23 ax.spines['top'].set_color('none')
24
25 ax.xaxis.set_ticks_position('bottom')
26 ax.spines['bottom'].set_position(('data', 0))
27
28 plt.show()
```



## Legend 圖例

matplotlib 中的 legend 圖例就是為了幫我們展示出每個數據對應的圖像名稱，更好的讓大家知道你的數據結構。



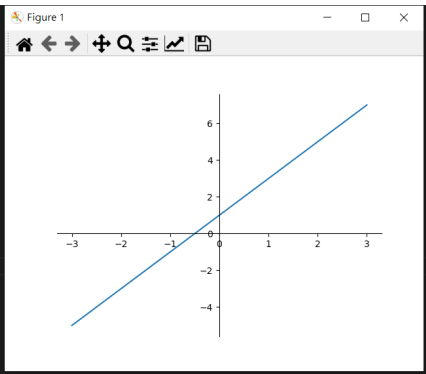
# Annotation 註解

當圖中某些特殊地方需要標示時，可以使用 annotation。matplotlib 中的 annotation 有兩種方法，一種是用 plt 裡面的 annotate，另一種是直接用 plt 裡面的 text 來寫註解。

# 基本使用

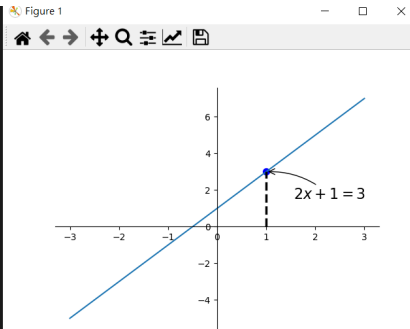
首先繪製一條線，並移動座標軸形成一個新的圖。

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 x = np.linspace(-3, 3, 50)
5 y = 2*x + 1
6
7 plt.plot(x, y)
8 ax = plt.gca()
9 ax.spines['right'].set_color('none')
10 ax.spines['top'].set_color('none')
11 ax.xaxis.set_ticks_position('bottom')
12 ax.spines['bottom'].set_position(('data', 0))
13 ax.yaxis.set_ticks_position('left')
14 ax.spines['left'].set_position(('data', 0))
15
16 plt.show()
17
```



## 添加 annotate

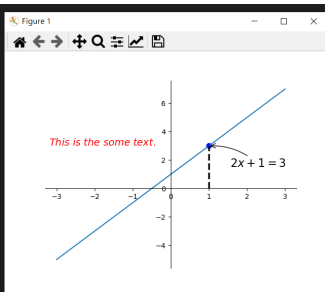
```
1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 x = np.linspace(-3, 3, 50)
5 y = 2*x + 1
6
7 plt.plot(x, y)
8 ax = plt.gca()
9 ax.spines['right'].set_color('none')
10 ax.spines['top'].set_color('none')
11 ax.xaxis.set_ticks_position('bottom')
12 ax.spines['bottom'].set_position(('data', 0))
13 ax.yaxis.set_ticks_position('left')
14 ax.spines['left'].set_position(('data', 0))
15
16 x0 = 1
17 y0 = 2*x0 + 1
18 plt.plot([x0, x0, ], [0, y0, ], 'k--', linewidth=2.5)
19 # set dot styles
20 plt.scatter([x0, ], [y0, ], s=50, color='blue')
21 plt.annotate(r'$2x+1=%s$' % y0, xy=(x0, y0), xycoords='data', xytext=(+30, -30),
22            textcoords='offset points', fontsize=16,
23            arrowprops=dict(arrowstyle='->', connectionstyle="arc3,rad=.2"))
24
25 plt.show()
```





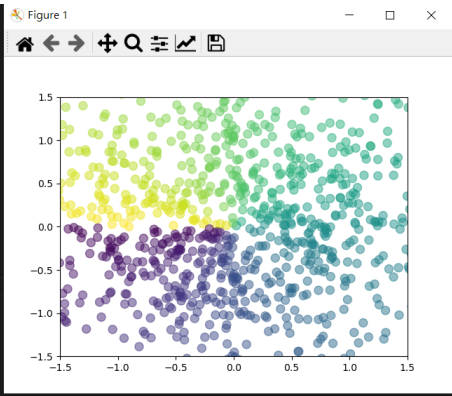
## 添加 text

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 x = np.linspace(-3, 3, 50)
5 y = 2*x + 1
6
7 plt.plot(x, y)
8 ax = plt.gca()
9 ax.spines['right'].set_color('none')
10 ax.spines['top'].set_color('none')
11 ax.xaxis.set_ticks_position('bottom')
12 ax.spines['bottom'].set_position(('data', 0))
13 ax.yaxis.set_ticks_position('left')
14 ax.spines['left'].set_position(('data', 0))
15
16 x0 = 1
17 y0 = 2*x0 + 1
18 plt.plot([x0, x0, ], [y0, y0, ], 'k--', linewidth=2.5)
19 # set dot styles
20 plt.scatter([x0, ], [y0, ], s=50, color='blue')
21 plt.annotate(r'$2x+1=%s$' % y0, xy=(x0, y0), xycoords='data', xytext=(+30, -30),
22            textcoords='offset points', fontsize=16,
23            arrowprops=dict(arrowstyle='->', connectionstyle="arc3,rad=.2"))
24
25 plt.text(-3.2, 3, r'$This is the some\ text.$',
26         fontdict={'size': 14, 'color': 'r'})
27
28 plt.show()
```



## 散佈圖

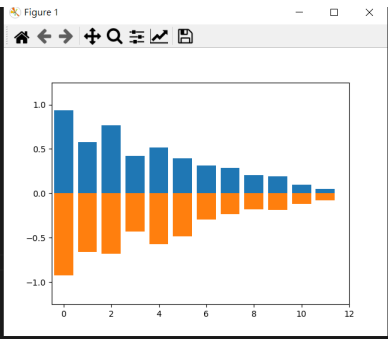
```
1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 n = 1024
5 X = np.random.normal(0, 1, n)
6 Y = np.random.normal(0, 1, n)
7 T = np.arctan2(Y, X)
8
9 plt.scatter(X, Y, s=75, c=T, alpha=.5)
10
11 plt.xlim(-1.5, 1.5)
12 plt.ylim(-1.5, 1.5)
13
14 plt.show()
15
```



## 直方圖

使用 `plt.bar` 生成基本直方圖。

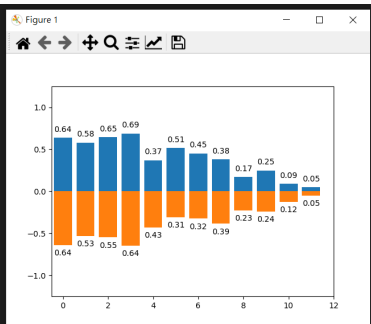
```
1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 n = 12
5 X = np.arange(n)
6 Y1 = (1 - X / float(n)) * np.random.uniform(0.5, 1.0, n)
7 Y2 = (1 - X / float(n)) * np.random.uniform(0.5, 1.0, n)
8
9 plt.bar(X, +Y1)
10 plt.bar(X, -Y2)
11
12 plt.xlim(-.5, n)
13 plt.ylim(-1.25, 1.25)
14
15 plt.show()
16
```



# 畫圖種類

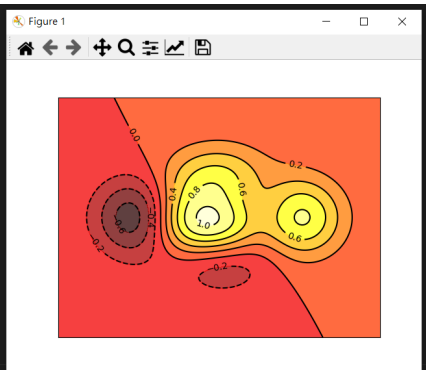
在把上一頁所生成的直方圖加入數據。

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 n = 12
5 X = np.arange(n)
6 Y1 = (1 - X / float(n)) * np.random.uniform(0.5, 1.0, n)
7 Y2 = (1 - X / float(n)) * np.random.uniform(0.5, 1.0, n)
8
9 plt.bar(X, +Y1)
10 plt.bar(X, -Y2)
11
12 plt.xlim(-.5, n)
13 plt.ylim(-1.25, 1.25)
14
15
16 for x, y in zip(X, Y1):
17     plt.text(x, y + 0.05, '%.2f' % y, ha='center', va='bottom')
18
19 for x, y in zip(X, Y2):
20     plt.text(x, -y - 0.05, '%.2f' % y, ha='center', va='top')
21
22 plt.show()
23
```



## 等高線圖

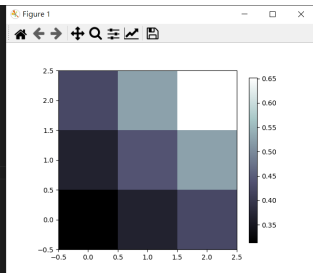
```
1 import matplotlib.pyplot as plt
2 import numpy as np
3
4
5 def f(x, y):
6     return (1 - x / 2 + x**5 + y**3) * np.exp(-x**2 - y**2)
7
8
9 n = 256
10 x = np.linspace(-3, 3, n)
11 y = np.linspace(-3, 3, n)
12 X, Y = np.meshgrid(x, y)
13
14 plt.contourf(X, Y, f(X, Y), 8, alpha=.75, cmap=plt.cm.hot)
15 C = plt.contour(X, Y, f(X, Y), 8, colors='black', linewidth=.5)
16 plt.clabel(C, inline=True, fontsize=10)
17
18 plt.xticks(())
19 plt.yticks(())
20 plt.show()
21
```



## 圖片

這裡我們用  $3 \times 3$  的 2D-array 來表示點的顏色，  
每一個點就是一個 pixel。

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 a = np.array([[0.313660827978, 0.365348418405, 0.423733120134,
5               0.365348418405, 0.439599930621, 0.525083754405,
6               0.423733120134, 0.525083754405, 0.651536351379]).reshape(3, 3)
7
8
9 plt.imshow(a, interpolation='nearest', cmap='bone', origin='lower')
10 plt.colorbar(shrink=.92)
11
12 plt.show()
13
```



## 3D 圖

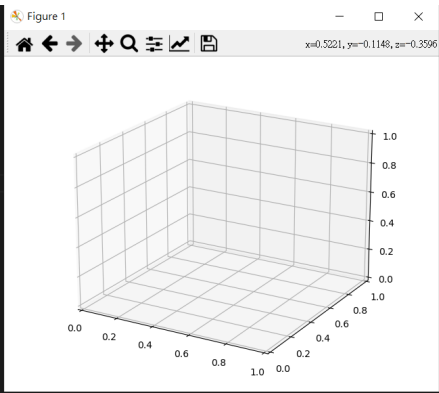
首先在進行 3D 繪圖時除了導入 matplotlib，還要額外添加一個模組，即 Axes3D。

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

# 畫圖種類

接著要先定義一個圖像視窗，在視窗上面添加 3D 座標軸。

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 from mpl_toolkits.mplot3d import Axes3D
4
5 fig = plt.figure()
6 ax = Axes3D(fig, auto_add_to_figure=False)
7 fig.add_axes(ax)
8
9 plt.show()
10
```

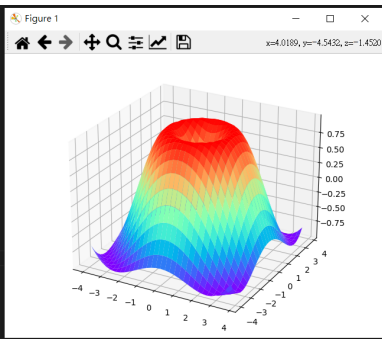




# 畫圖種類

再來我們做出一個三維曲面，並將一個 colormap rainbow 塗上顏色。

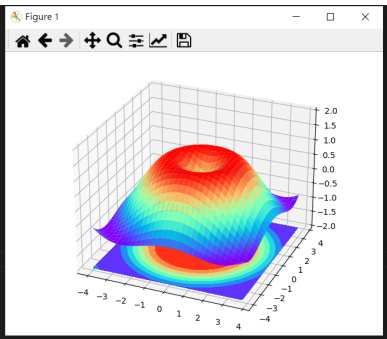
```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 from mpl_toolkits.mplot3d import Axes3D
4
5 fig = plt.figure()
6 ax = Axes3D(fig, auto_add_to_figure=False)
7 fig.add_axes(ax)
8
9 X = np.arange(-4, 4, 0.25)
10 Y = np.arange(-4, 4, 0.25)
11 X, Y = np.meshgrid(X, Y)
12 R = np.sqrt(X ** 2 + Y ** 2)
13 Z = np.sin(R)
14
15 ax.plot_surface(X, Y, Z, rstride=1, cstride=1, cmap=plt.get_cmap('rainbow'))
16
17 plt.show()
18
```



# 畫圖種類

最後再把它投影到  $xy$  平面上做一個等高線圖，就完成了。

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 from mpl_toolkits.mplot3d import Axes3D
4
5 fig = plt.figure()
6 ax = Axes3D(fig, auto_add_to_figure=False)
7 fig.add_axes(ax)
8
9 X = np.arange(-4, 4, 0.25)
10 Y = np.arange(-4, 4, 0.25)
11 X, Y = np.meshgrid(X, Y)
12 R = np.sqrt(X ** 2 + Y ** 2)
13 Z = np.sin(R)
14
15 ax.plot_surface(X, Y, Z, rstride=1, cstride=1, cmap=plt.get_cmap('rainbow'))
16 ax.contourf(X, Y, Z, zdir='z', offset=-2, cmap=plt.get_cmap('rainbow'))
17 ax.set_zlim(-2, 2)
18
19 plt.show()
20
```

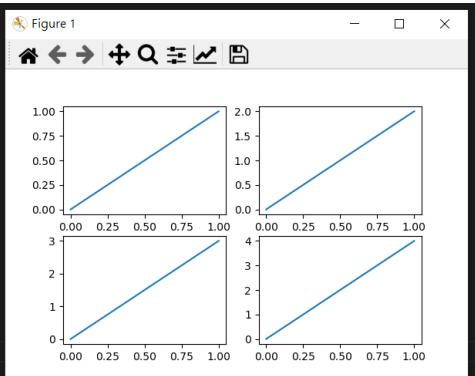


# Subplot 多合一顯示

matplotlib 是可以組合許多的小圖，放在一張大圖裡面顯示的  
使用的方法叫做 subplot。

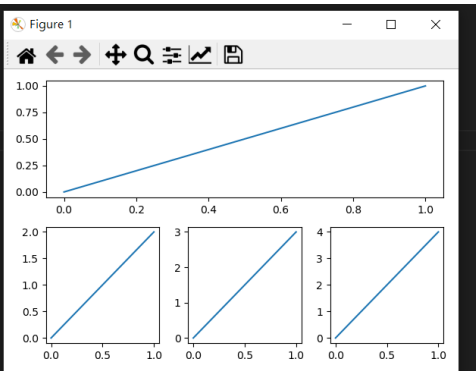
## 均勻的圖中圖

```
1 import matplotlib.pyplot as plt
2
3 plt.figure(figsize=(6, 4))
4
5 plt.subplot(2, 2, 1)
6 plt.plot([0, 1], [0, 1])
7
8 plt.subplot(2, 2, 2)
9 plt.plot([0, 1], [0, 2])
10
11 plt.subplot(2, 2, 3)
12 plt.plot([0, 1], [0, 3])
13
14 plt.subplot(2, 2, 4)
15 plt.plot([0, 1], [0, 4])
16
17 plt.tight_layout()
18
19 plt.show()
```



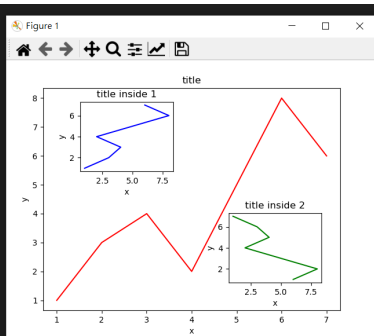
## 不均勻的圖中圖

```
1 import matplotlib.pyplot as plt
2
3 plt.figure(figsize=(6, 4))
4
5 plt.subplot(2, 1, 1)
6 plt.plot([0, 1], [0, 1])
7
8 plt.subplot(2, 3, 4)
9 plt.plot([0, 1], [0, 2])
10
11 plt.subplot(2, 3, 5)
12 plt.plot([0, 1], [0, 3])
13
14 plt.subplot(2, 3, 6)
15 plt.plot([0, 1], [0, 4])
16
17 plt.tight_layout()
18
19 plt.show()
```



## 圖中圖

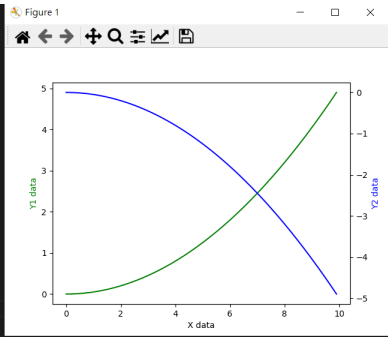
```
1 import matplotlib.pyplot as plt
2
3 fig = plt.figure()
4 x = [1, 2, 3, 4, 5, 6, 7]
5 y = [1, 3, 4, 2, 5, 8, 6]
6
7 left, bottom, width, height = 0.1, 0.1, 0.8, 0.8
8 ax1 = fig.add_axes([left, bottom, width, height])
9 ax1.plot(x, y, 'r')
10 ax1.set_xlabel('x')
11 ax1.set_ylabel('y')
12 ax1.set_title('title')
13
14 ax2 = fig.add_axes([0.2, 0.6, 0.25, 0.25])
15 ax2.plot(y, x, 'b')
16 ax2.set_xlabel('x')
17 ax2.set_ylabel('y')
18 ax2.set_title('title inside 1')
19
20 plt.axes([0.6, 0.2, 0.25, 0.25])
21 plt.plot(y[::-1], x, 'g')
22 plt.xlabel('x')
23 plt.ylabel('y')
24 plt.title('title inside 2')
25
26 plt.show()
27
```



## 副坐標軸

副坐標軸意思就是同一個圖上有第 2 個 y 軸存在。

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 x = np.arange(0, 10, 0.1)
5 y1 = 0.05 * x**2
6 y2 = -1 * y1
7
8 fig, ax1 = plt.subplots()
9
10 ax2 = ax1.twinx() # mirror the ax1
11 ax1.plot(x, y1, 'g-')
12 ax2.plot(x, y2, 'b-')
13
14 ax1.set_xlabel('X data')
15 ax1.set_ylabel('Y1 data', color='g')
16 ax2.set_ylabel('Y2 data', color='b')
17
18 plt.show()
19
```



END