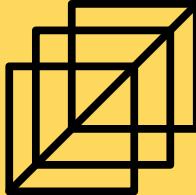


Matplotlib & OpenCV

資管碩—612630060 張瑋珉



ADD COMPANY NAME



Agenda

Matplotlib

安裝並載入matplotlib

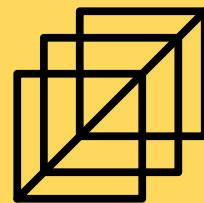
介紹 pyplot 的使用方式

範例₁：人口成長圖表

範例₂：座標圖

範例₃：其他類型圖

OpenCV



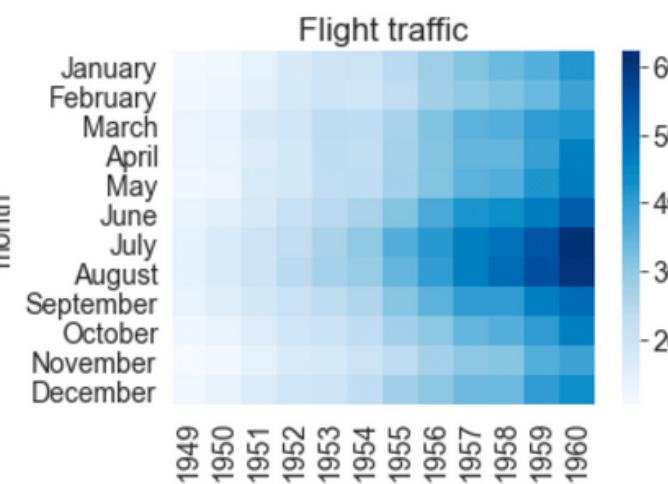
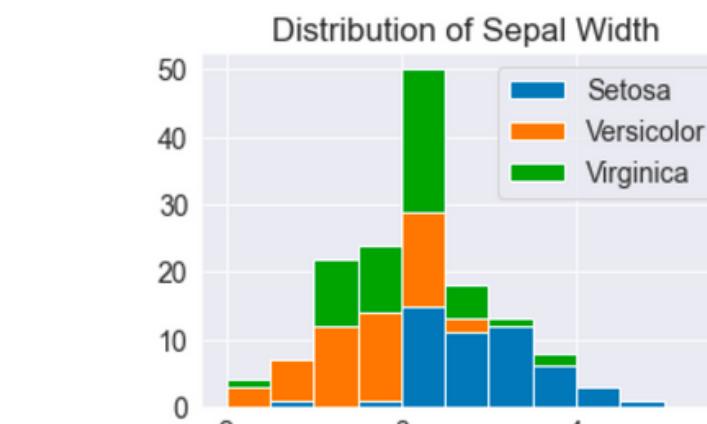
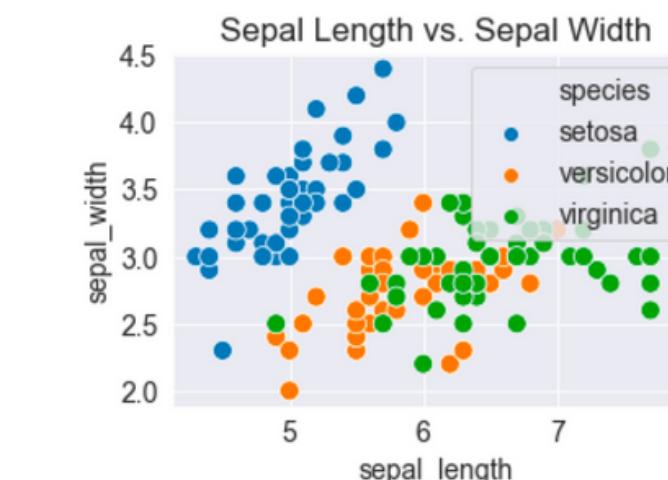
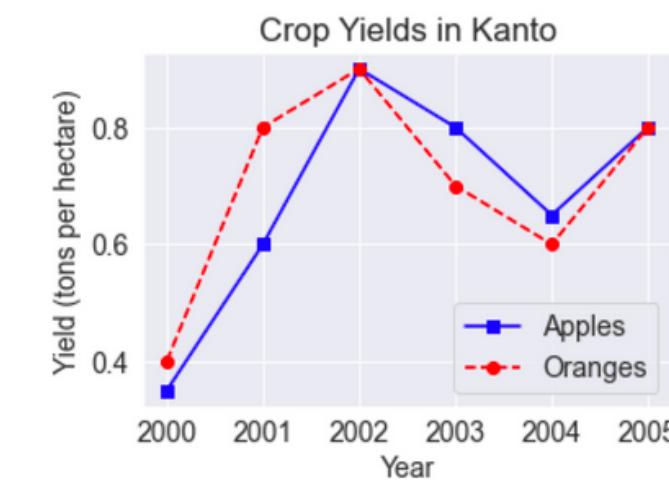
ADD COMPANY NAME

Matplotlib



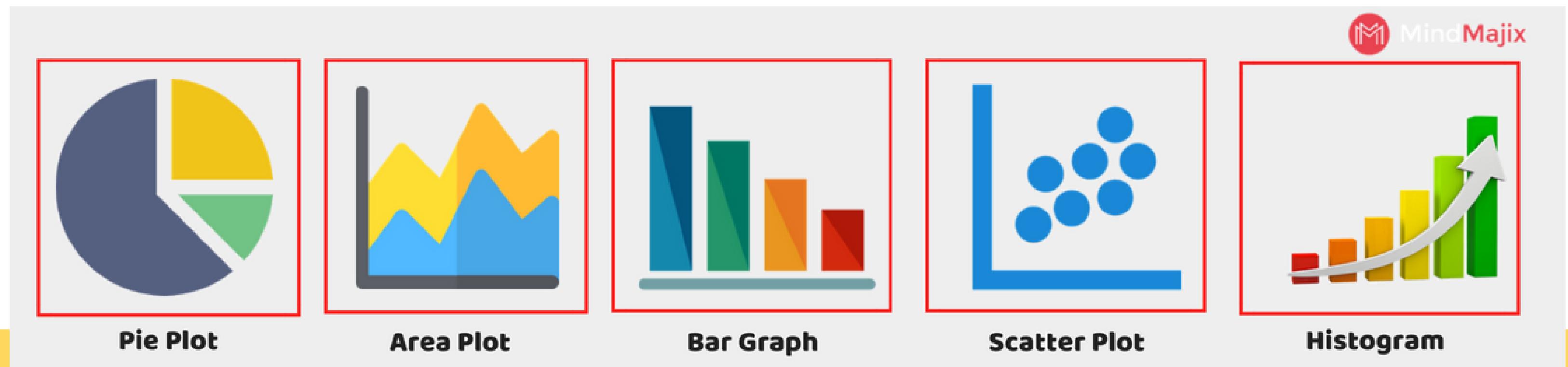
Matplotlib

- Matplotlib 是一個 Python 2D 繪圖庫，同時是 Python 當中用來繪圖、圖表呈現及數據表示非常重要的一個綜合函式庫 (Library)。
- Matplotlib 擁有一個功能強大的開源 (Open Source) 工具包，用於表示及可視覺化數據。
- 創建者 John Hunter 深受 MATLAB 程式環境的啟發，所以蘊藏了一些其元素想法。



Matplotlib

- Matplotlib 其中包含兩個最重要的模組 — pylab 和 pyplot，我們可以利用 Matplotlib 創建各種圖形，包括簡單的折線圖，直方圖，散佈圖...。
- Python 數據計算中經常使用它來完成視覺化的工作。



安裝並載入matplotlib

畫圖前,我們需要安裝並載入matplotlib

`pip install matplotlib`

介紹 pyplot 的使用方式

Python 的 Matplotlib 可繪製簡單的折線圖，只要使用 plt.plot ，給 X, Y 值就可以畫出圖表！

範例：假設有資料 (X 與 Y) 為座標,要繪製成折線圖

```
x = [1, 3, 5 ,7 ,9]
```

```
y = [2, 4, 6, 8, 10]
```

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

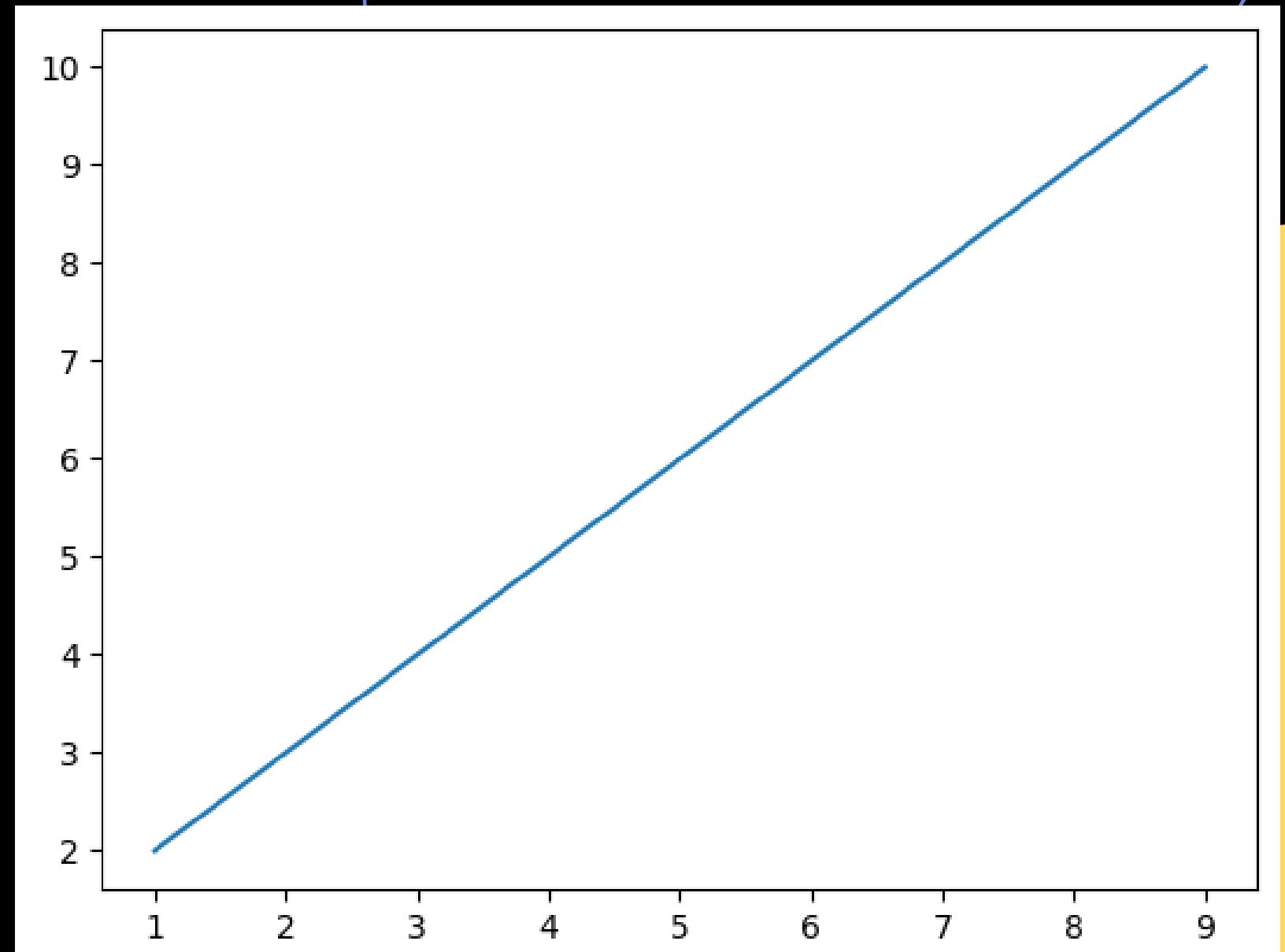
```
x = [1, 3, 5, 7, 9]
```

```
y = [2, 4, 6, 8, 10]
```

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

```
plt.show()
```

打印出來的折線圖就會是下面這樣：



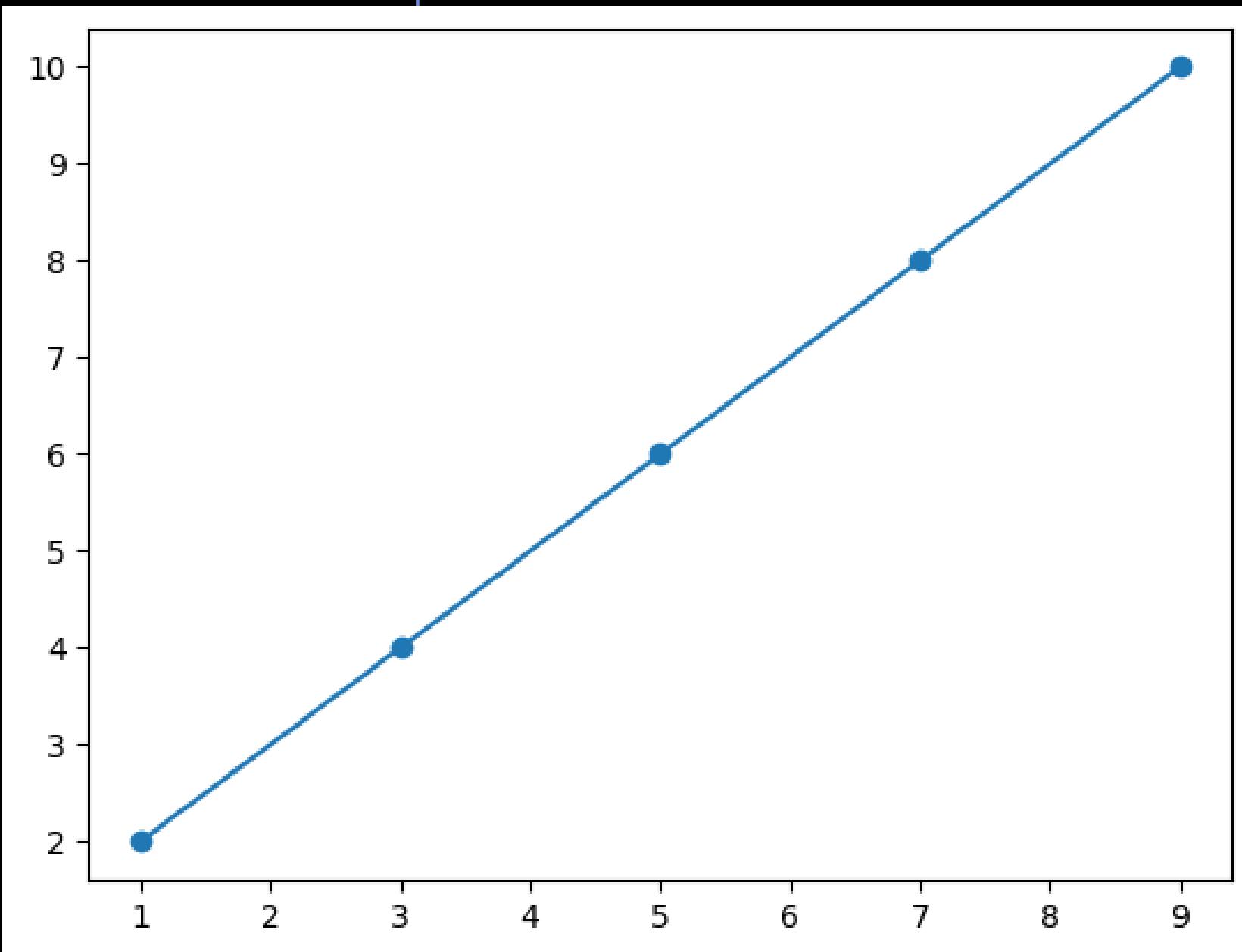
plot 點&線的樣式

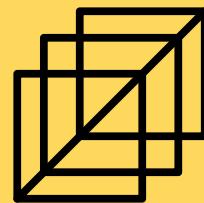
plt.plot 有不同的參數可以做出不同的線或是點：

- plt.plot(x, y, 'o') # 只有點
- plt.plot(x, y) # 只有線(預設)
- plt.plot(x, y, '-o') # 點和線
- plt.plot(x, y, '--o') # 點和虛線

```
x = [1, 3, 5, 7, 9]
y = [2, 4, 6, 8, 10]
plt.plot(x, y, "-o")
plt.show()
```

打印出來的折線圖就會是下面這樣：





ADD COMPANY NAME

範例：人口成長圖表

Matplotlib 讓資料視覺化！

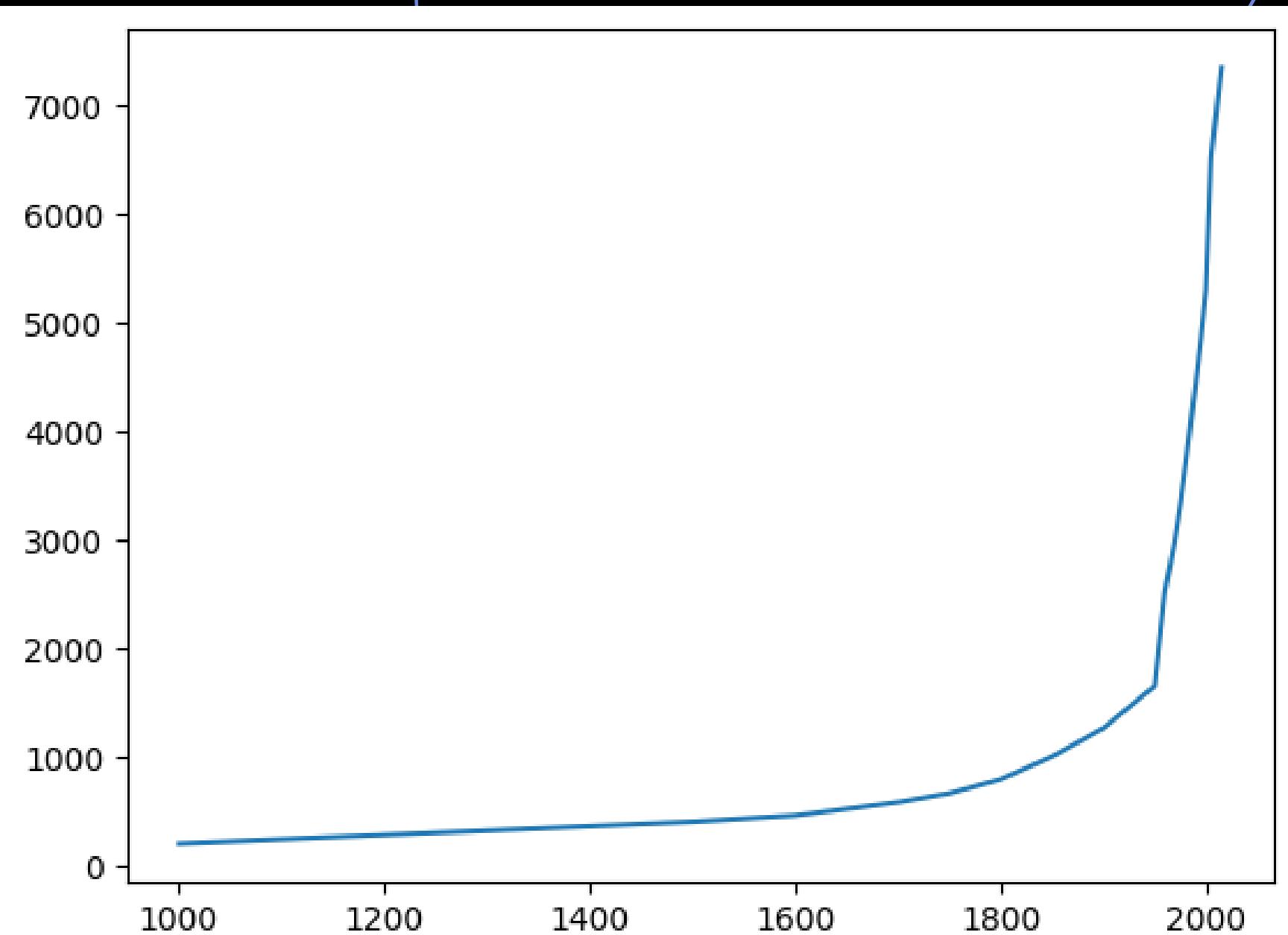


資料的走向和結果

使用 `plt.plot()`，將`years,pops`資料內容放到函數內：

```
import matplotlib.pyplot as plt  
years = [1000,1500,1600,1700,1750,1800,1850,  
    1900,1950,1960,1965,1970,1975,1980,  
    1985,1990,1995,2000,2005,  
    2010,2015]  
  
pops = [200,400,458,580,662,791,1000,  
    1262,1650,2525,2758,3018,3322,3682,  
    4061,4440,4853,5310,6520,  
    6930,7349]  
plt.plot(years,pops)  
plt.show()
```

打印出來的折線圖就會是下面這樣：



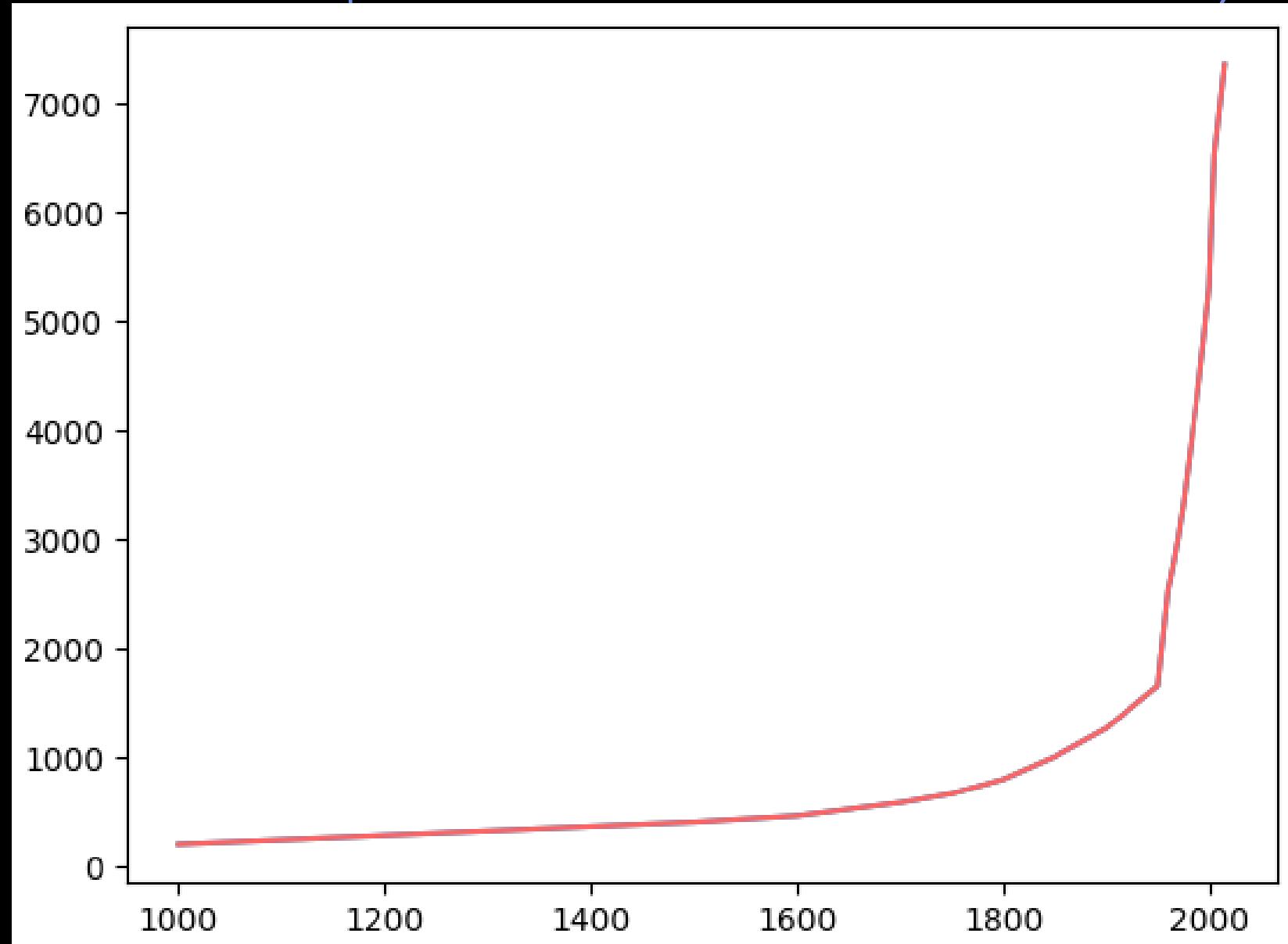
Change color

要怎麼改變線條的顏色呢？

只需要在 `plt.plot()` 內加入一個color參數：

```
import matplotlib.pyplot as plt  
years = [1000,1500,1600,1700,1750,1800,1850,  
        1900,1950,1960,1965,1970,1975,1980,  
        1985,1990,1995,2000,2005,  
        2010,2015]  
  
pops = [200,400,458,580,662,791,1000,  
        1262,1650,2525,2758,3018,3322,3682,  
        4061,4440,4853,5310,6520,  
        6930,7349]  
plt.plot(years,pops)  
plt.plot(years,pops,color(255/255,100/255,100/255))  
plt.show()
```

打印出來的折線圖就會是下面這樣：

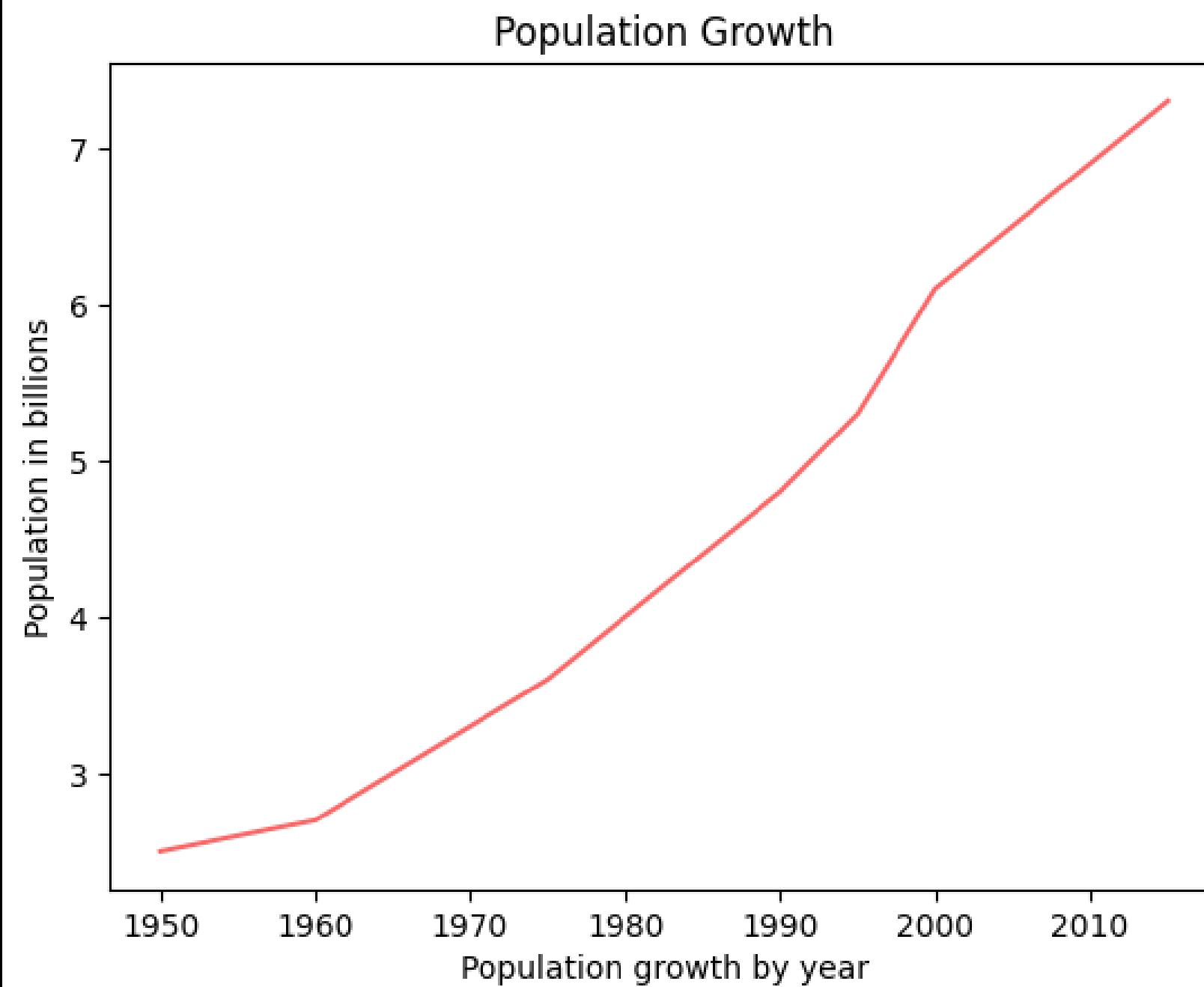


label

將圖型顯示出來之後，可以新增label來寫個人口成長的圖表

```
years = [1950,1960,1965,1970,1975,1980,  
        1985,1990,1995,2000,2005,  
        2010,2015]  
pops = [2.5,2.7,3,3.3,3.6,4.0,  
        4.4,4.8,5.3,6.1,6.5,6.9,7.3]  
  
plt.plot(years,pops,color=(255/255,100/255,100/255))  
#設定label  
plt.title("Population Growth") # title  
plt.ylabel("Population in billions") # y label  
plt.xlabel("Population growth by year") # x label  
plt.show()
```

打印出來的折線圖就會是下面這樣：



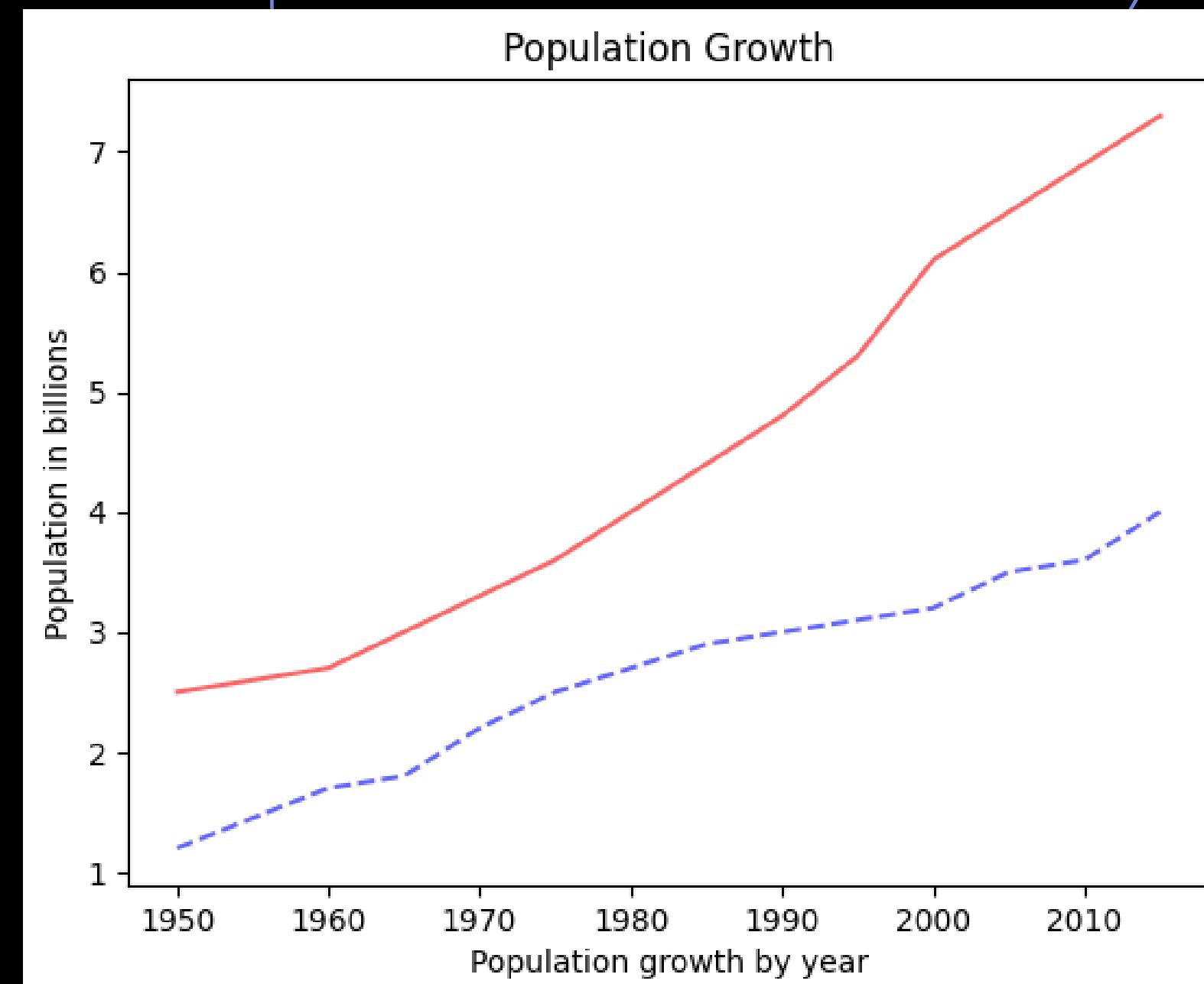
Multilines

上面我們顯示了人口成長的圖表，假設我們想再加上死亡率：

新增一筆資料：

```
years = [1950,1960,1965,1970,1975,1980,  
        1985,1990,1995,2000,2005,  
        2010,2015]  
  
pops = [2.5,2.7,3,3.3,3.6,4.0,  
        4.4,4.8,5.3,6.1,6.5,6.9,7.3]  
  
deaths = [1.2,1.7,1.8,2.2,2.5,2.7,2.9,3,3.1,3.2,3.5,3.6,4]  
# 修改plt.plot()的部分：  
plt.plot(years,pops, color=(255/255,100/255,100/255))  
# 看到 '--' 部分，這個是顯示顯條的樣子  
plt.plot(years,deaths, '--', color=(100/255,100/255,255/255))  
  
plt.title("Population Growth") # title  
plt.ylabel("Population in billions") # y label  
plt.xlabel("Population growth by year") # x label  
plt.show()
```

打印出來的折線圖就會是下面這樣：



Configure graph

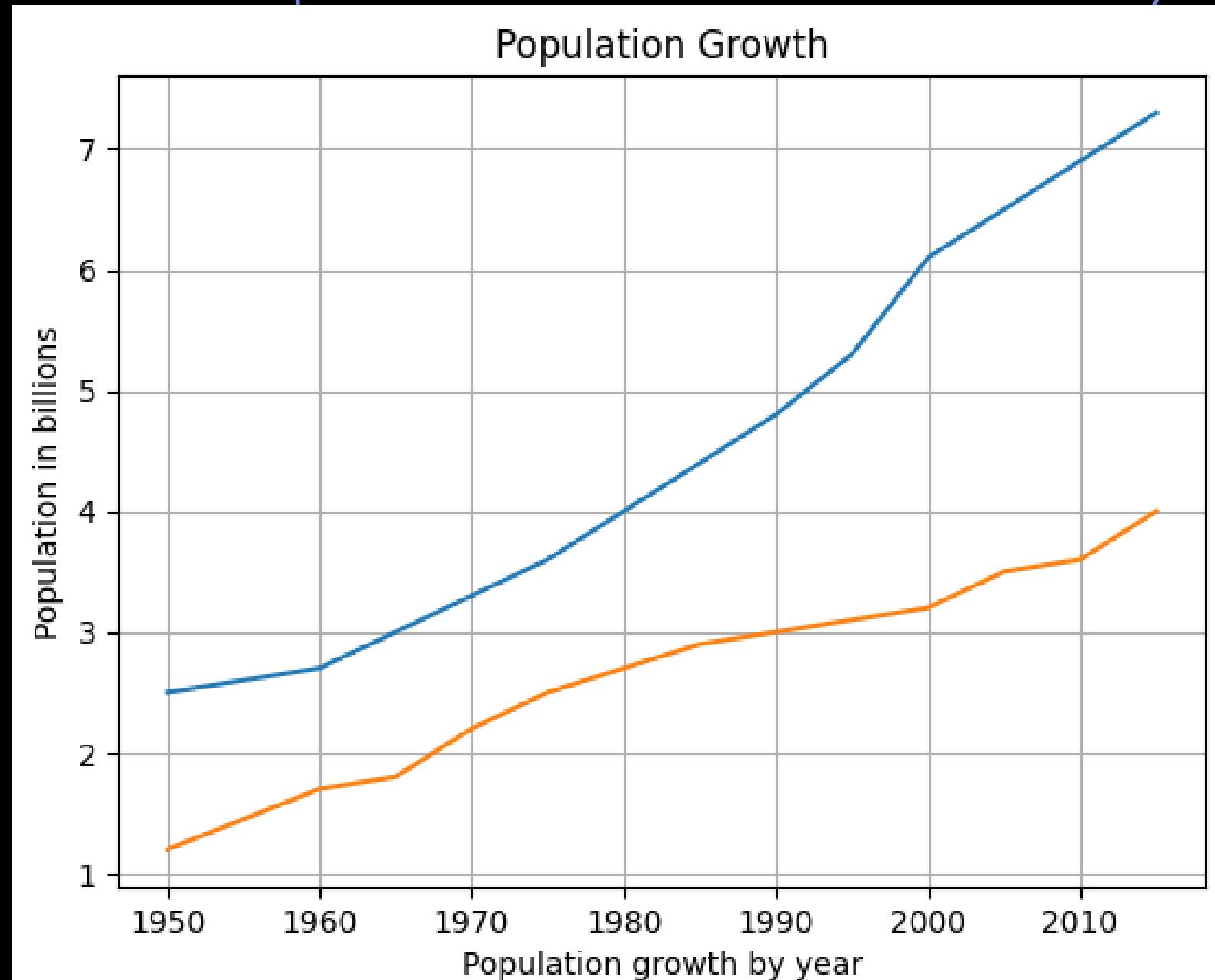
圖型可以有哪些設置：

`plt.grid()`

`grid()` 用來顯示網格：

```
years = [1950,1960,1965,1970,1975,1980,  
        1985,1990,1995,2000,2005,  
        2010,2015]  
pops = [2.5,2.7,3,3.3,3.6,4.0,  
        4.4,4.8,5.3,6.1,6.5,6.9,7.3]  
deaths = [1.2,1.7,1.8,2.2,2.5,2.7,2.9,3,3.1,3.2,3.5,3.6,4]  
  
plt.title("Population Growth") # title  
plt.ylabel("Population in billions") # y label  
plt.xlabel("Population growth by year") # x label  
  
lines = plt.plot(years,pops,years,deaths)  
plt.grid(True)  
plt.show()
```

打印出來的折線圖就會是下面這樣：

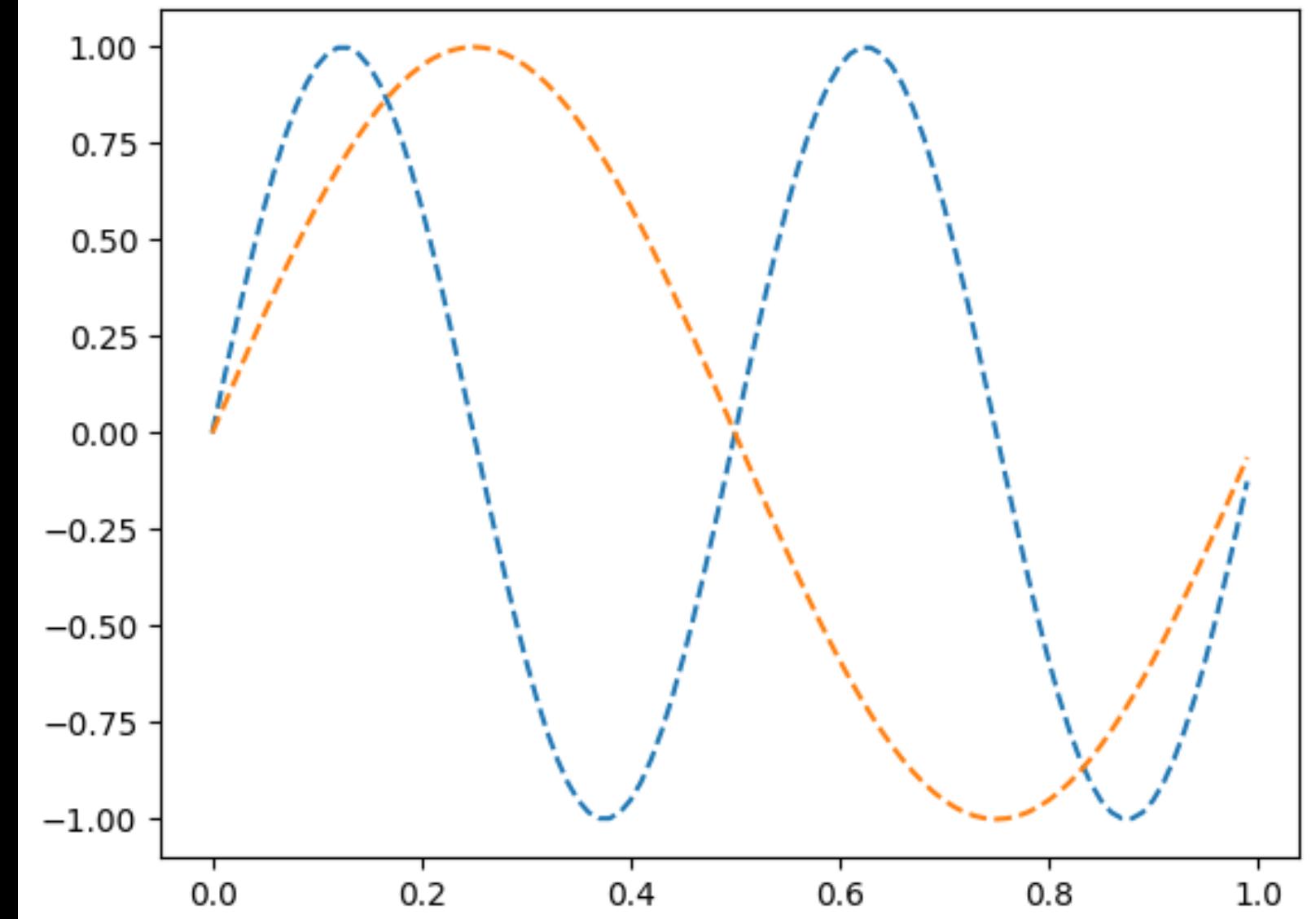


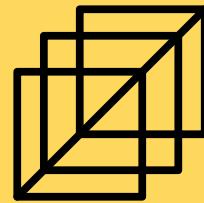
With Numpy

結合Numpy來繪出兩個不同速率的sin波形

```
import matplotlib.pyplot as plt import numpy as np  
x = np.arange(0, 1.0, 0.01)  
y1 = np.sin(4*np.pi*x)  
y2 = np.sin(2*np.pi*x) lines = plt.plot(x, y1, x, y2)  
l1, l2 = lines  
plt.setp(lines, linestyle='--')  
plt.show()
```

打印出來的折線圖就會是下面這樣：





ADD COMPANY NAME

範例2：座標

Matplotlib 讓資料視覺化！

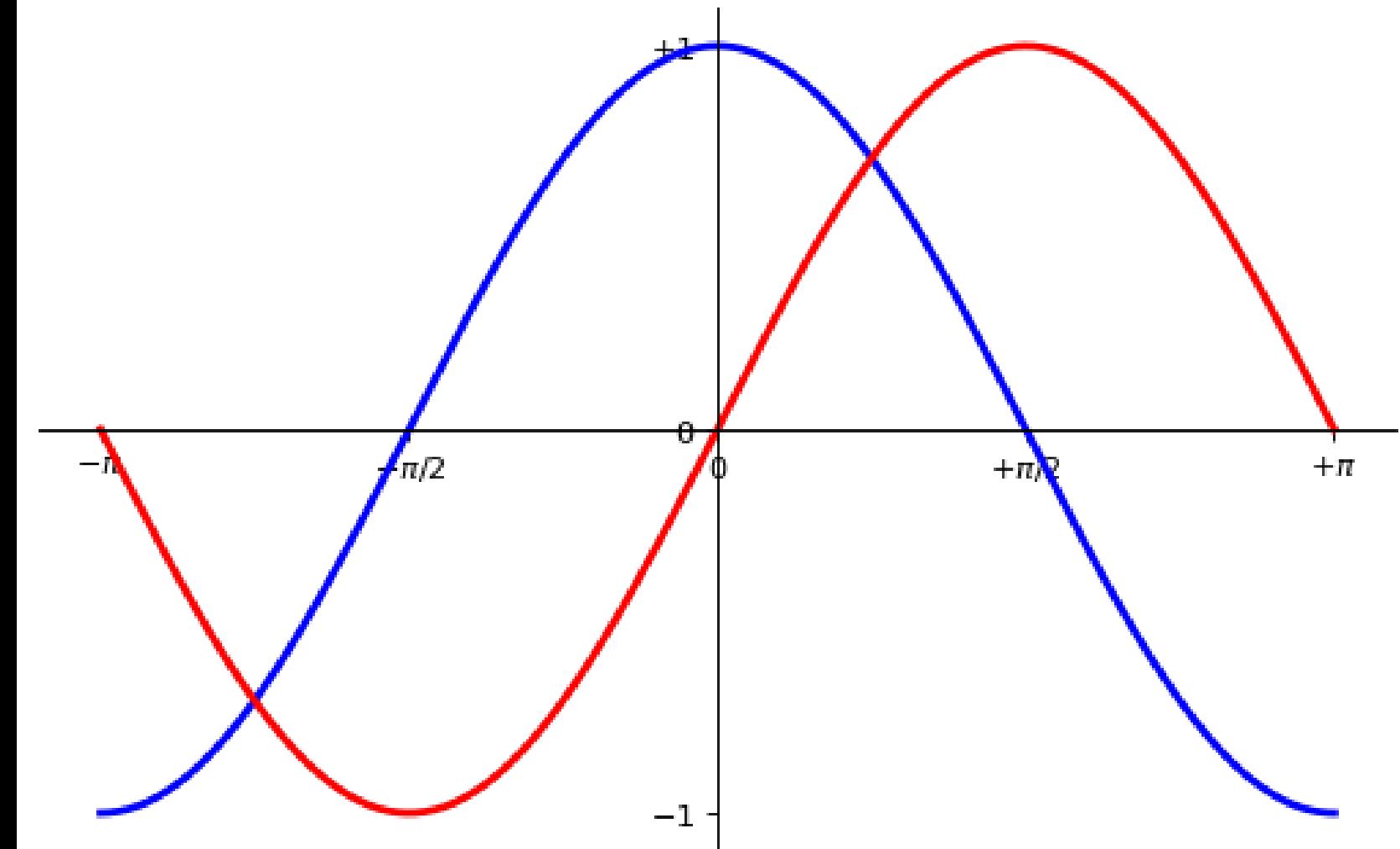


坐標軸線

(Spines) 放在圖的中間，我們必須將其中的兩條（上和右）設置為無色，然後調整剩下的兩條到合適的位置

```
...  
ax = gca()  
ax.spines['right'].set_color('none')  
ax.spines['top'].set_color('none')  
ax.xaxis.set_ticks_position('bottom')  
ax.spines['bottom'].set_position((‘data’,0))  
ax.yaxis.set_ticks_position(‘left’)  
ax.spines[‘left’].set_position((‘data’,0))  
...
```

打印出來的折線圖就會是下面這樣：

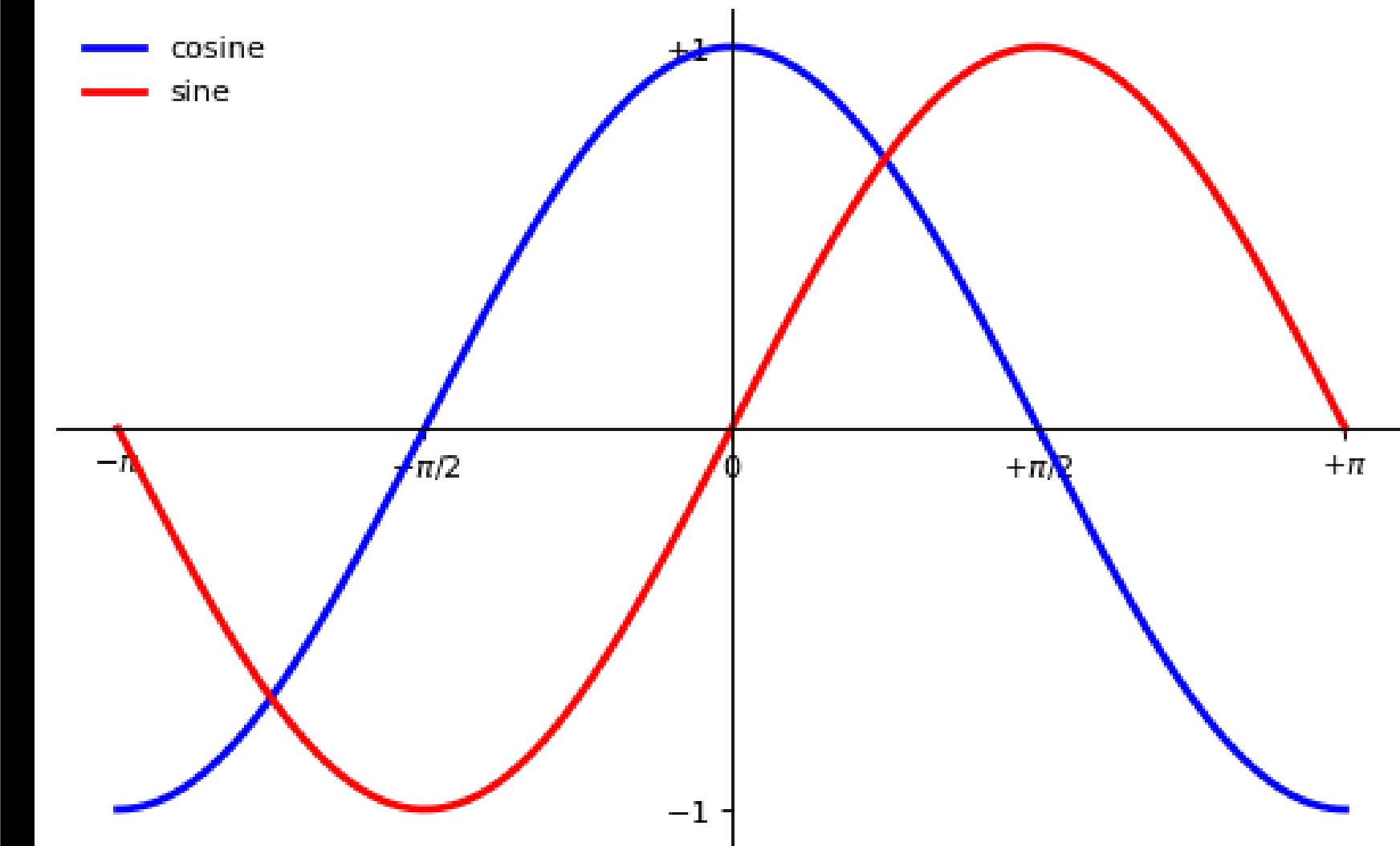


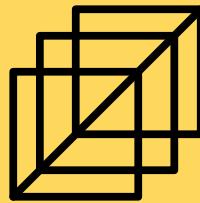
添加圖例

在圖的左上角添加一個圖例。為此，我們只需要在 `plot` 函數里以「鍵 - 值」的形式增加一個參數。

```
....  
plot(X, C, color="blue", linewidth=2.5,  
linestyle="-", label="cosine")  
plot(X, S, color="red", linewidth=2.5,  
linestyle="-", label="sine")  
  
legend(loc='upper left')  
...
```

打印出來的折線圖就會是下面這樣：





ADD COMPANY NAME

範例3：其他類型圖

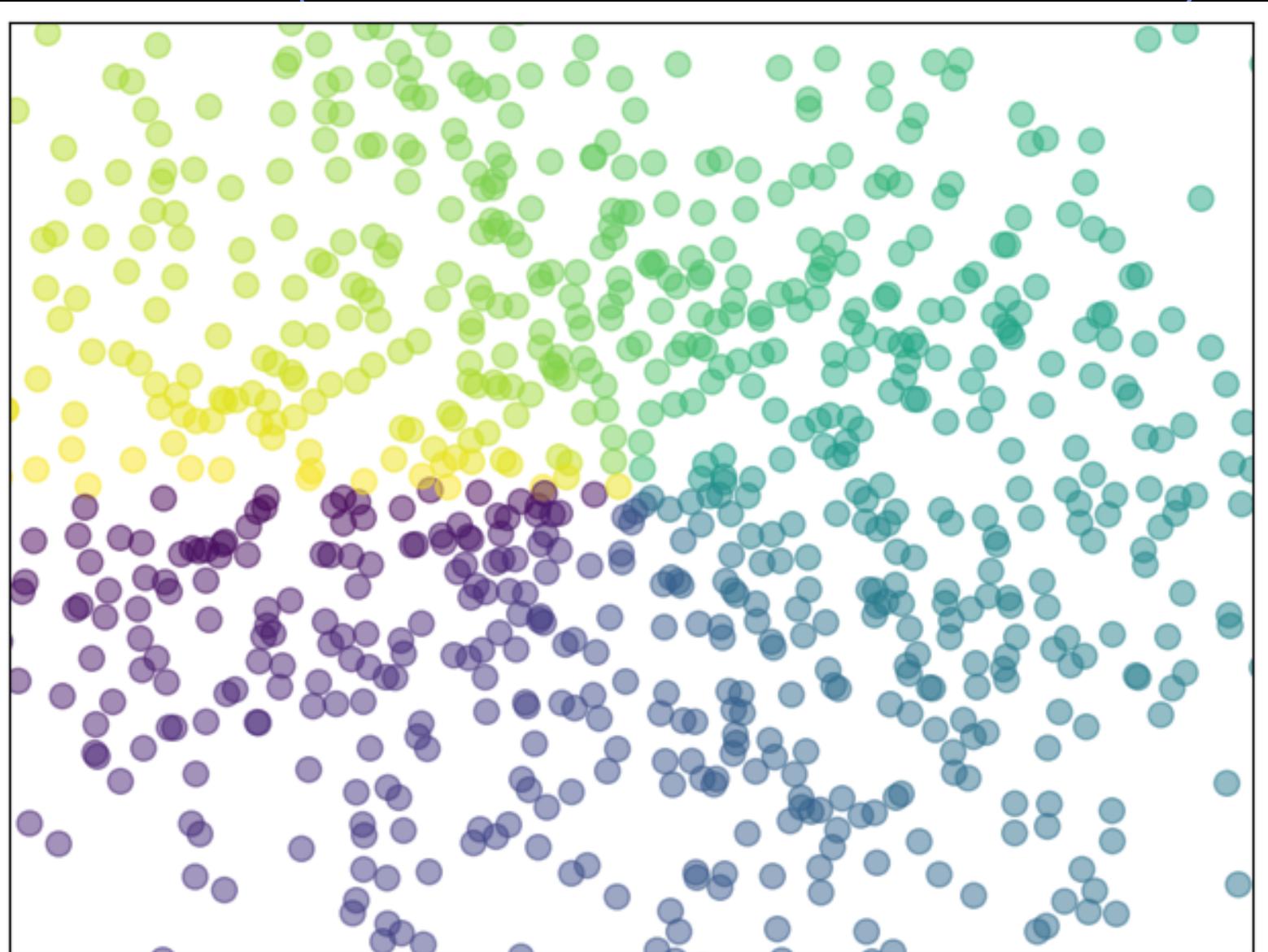
Matplotlib 讓資料視覺化！



散點圖

```
import numpy as np  
import matplotlib.pyplot as plt  
  
n = 1024  
X = np.random.normal(0,1,n)  
Y = np.random.normal(0,1,n)  
T = np.arctan2(Y,X)  
  
plt.axes([0.025,0.025,0.95,0.95])  
plt.scatter(X,Y, s=75, c=T, alpha=.5)  
  
plt.xlim(-1.5,1.5), plt.xticks([])  
plt.ylim(-1.5,1.5), plt.yticks([])  
  
plt.show()
```

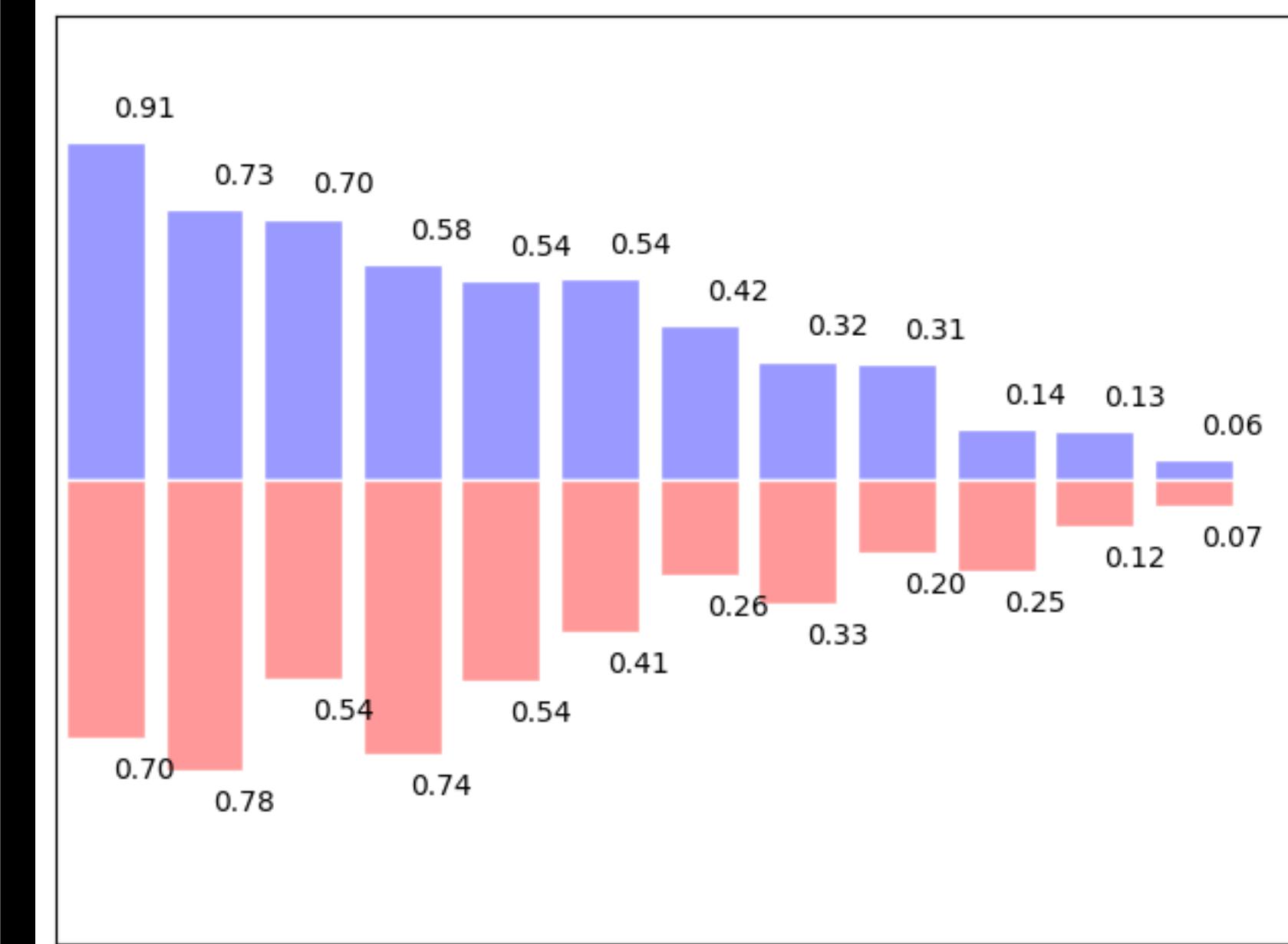
打印出來的折線圖就會是下面這樣：



條形圖

```
n = 12  
X = np.arange(n)  
Y1 = (1-X/float(n)) * np.random.uniform(0.5,1.0,n)  
Y2 = (1-X/float(n)) * np.random.uniform(0.5,1.0,n)  
  
plt.axes([0.025,0.025,0.95,0.95])  
plt.bar(X, +Y1, facecolor='#9999ff', edgecolor='white')  
plt.bar(X, -Y2, facecolor='#ff9999', edgecolor='white')  
for x,y in zip(X,Y1):  
    plt.text(x+0.4, y+0.05, '%.2f' % y, ha='center', va=  
        'bottom')  
  
for x,y in zip(X,Y2):  
    plt.text(x+0.4, -y-0.05, '%.2f' % y, ha='center', va= 'top')  
  
plt.xlim(-.5,n), plt.xticks([])  
plt.ylim(-1.25,+1.25), plt.yticks([])  
  
plt.show()
```

打印出來的折線圖就會是下面這樣：



等高線圖

```
def f(x,y):
    return (1-x/2+x**5+y**3)*np.exp(-x**2-y**2)

n = 256
x = np.linspace(-3,3,n)
y = np.linspace(-3,3,n)
X,Y = np.meshgrid(x,y)

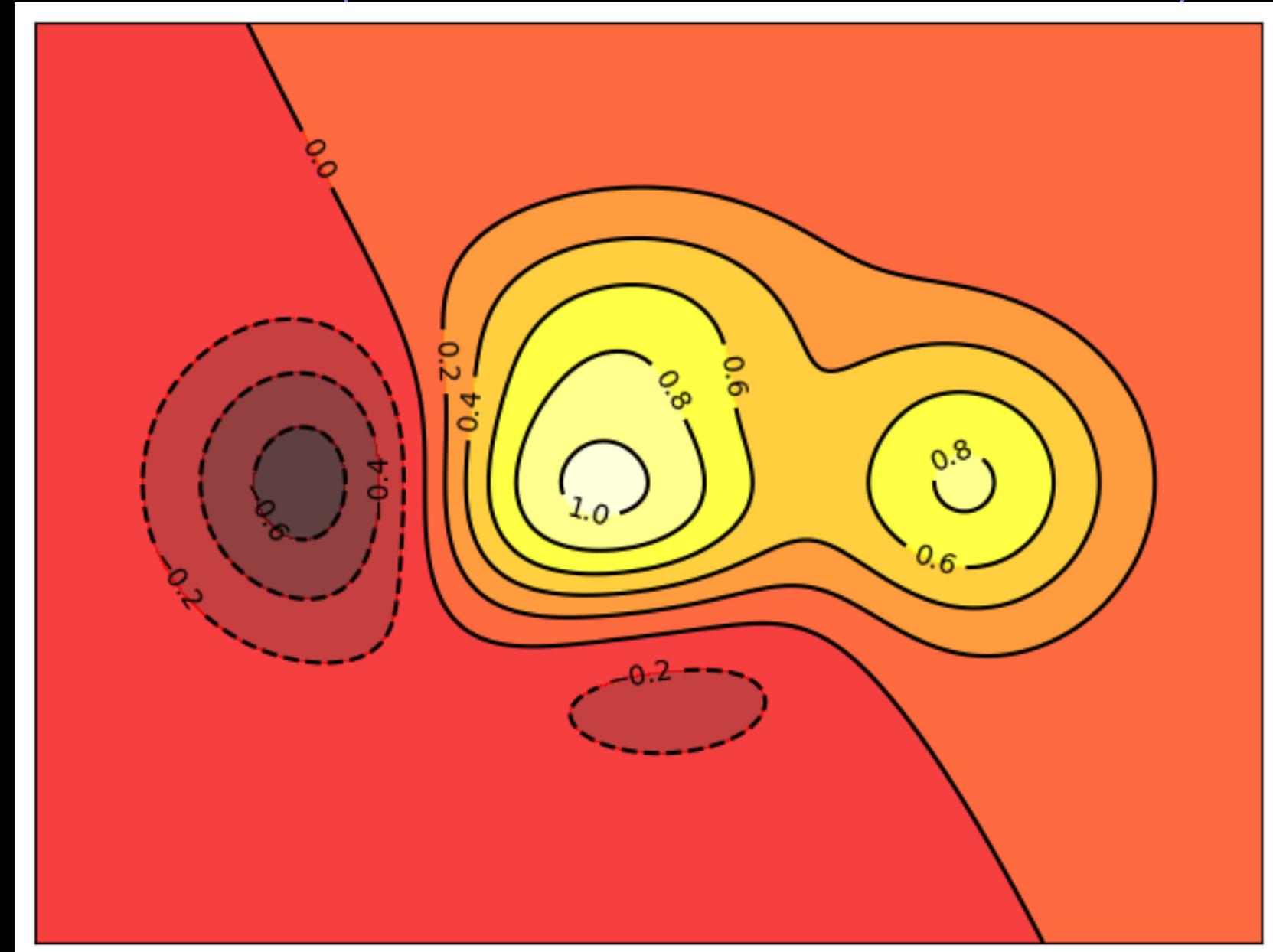
plt.axes([0.025,0.025,0.95,0.95])

plt.contourf(X, Y, f(X,Y), 8, alpha=.75, cmap=plt.cm.hot)
C = plt.contour(X, Y, f(X,Y), 8, colors='black',
linewidth=.5)
plt.clabel(C, inline=1, fontsize=10)

plt.xticks([]), plt.yticks([])

plt.show()
```

打印出來的折線圖就會是下面這樣：



多重網格圖

```
1.  
2. import numpy as np  
3. import matplotlib.pyplot as plt  
4. from pylab import *  
5. import pandas as pd  
6.  
7. fig = plt.figure()  
8. ax = fig.add_subplot(111,projection='3d')  
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.cm.hot)  
16. ax.contourf(X, Y, Z, zdir='z', offset=-2,  
   cmap=plt.cm.hot)  
17. ax.set_zlim(-2,2)  
18. show()
```

打印出來的折線圖就會是下面這樣：



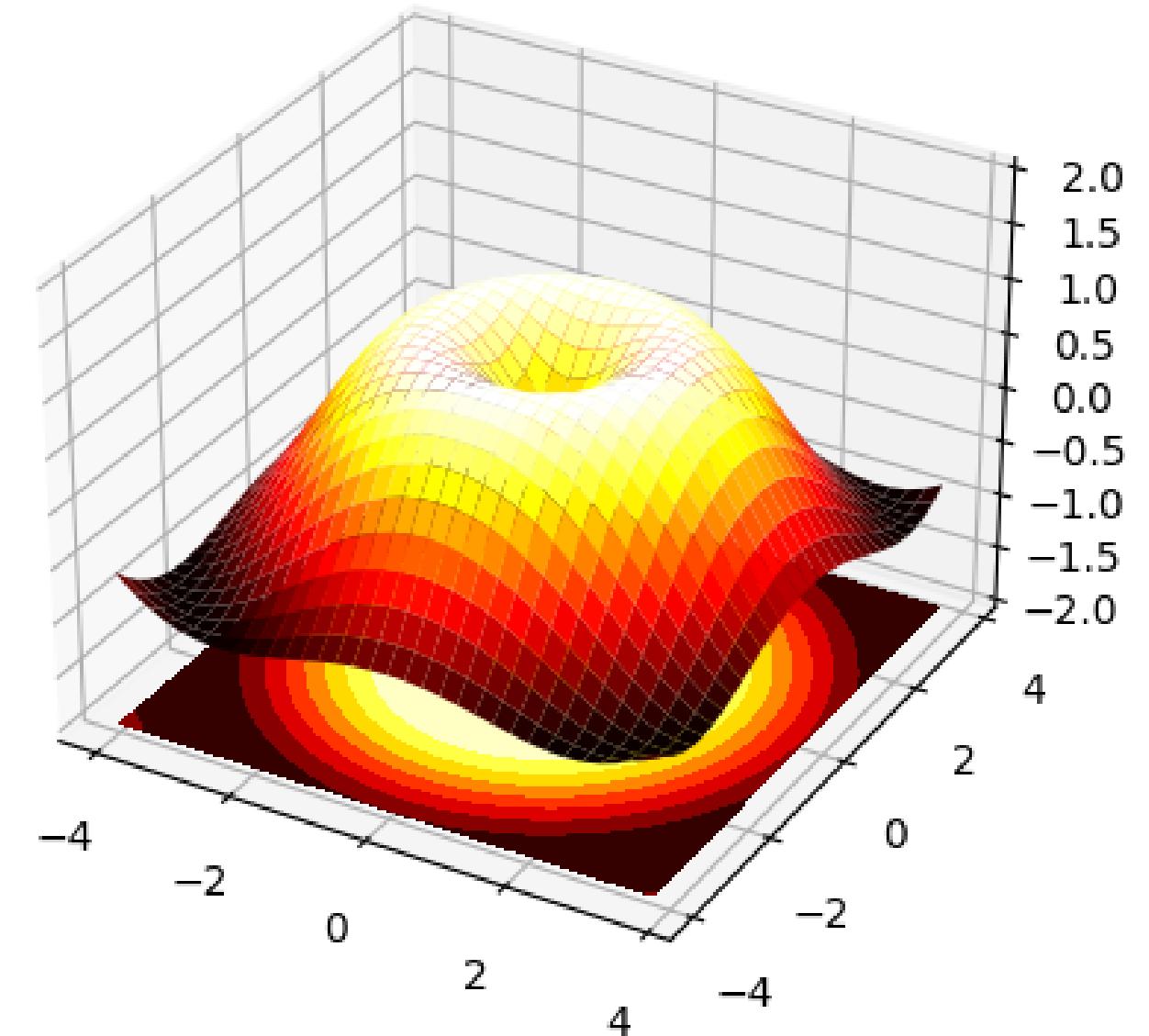
3D 立體圖

```
import numpy as np
import matplotlib.pyplot as plt
from pylab import *
import pandas as pd

fig = plt.figure()
ax = fig.add_subplot(111,projection='3d') #生成3D圖
X = np.arange(-4, 4, 0.25)
Y = np.arange(-4, 4, 0.25)
X, Y = np.meshgrid(X, Y)
R = np.sqrt(X**2 + Y**2)
Z = np.sin(R)

ax.plot_surface(X, Y, Z, rstride=1, cstride=1,
cmap=plt.cm.hot)
ax.contourf(X, Y, Z, zdir='z', offset=-2, cmap=plt.cm.hot)
ax.set_zlim(-2,2)
show()
```

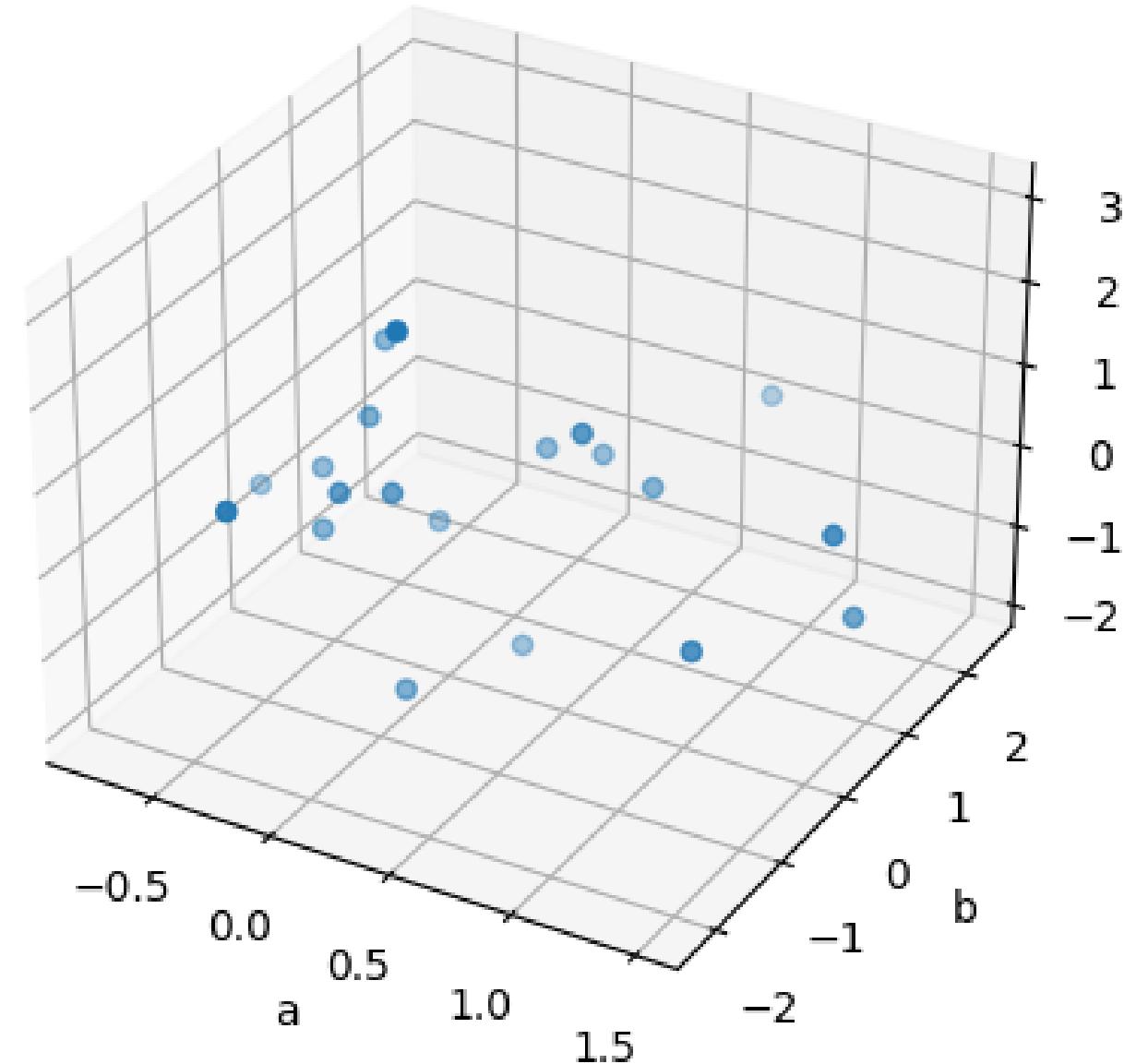
打印出來的折線圖就會是下面這樣：

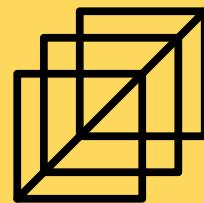


多維度空間圖

```
import numpy as np  
import matplotlib.pyplot as plt  
from pylab import *  
import pandas as pd  
  
fig = plt.figure()  
ax = fig.add_subplot(111,projection='3d') #生成3D圖  
X = np.arange(-4, 4, 0.25)  
Y = np.arange(-4, 4, 0.25)  
X, Y = np.meshgrid(X, Y)  
R = np.sqrt(X**2 + Y**2)  
Z = np.sin(R)  
  
ax.plot_surface(X, Y, Z, rstride=1, cstride=1,  
cmap=plt.cm.hot)  
ax.contourf(X, Y, Z, zdir='z', offset=-2, cmap=plt.cm.hot)  
ax.set_zlim(-2,2)  
show()
```

打印出來的折線圖就會是下面這樣：





ADD COMPANY NAME

OpenCV

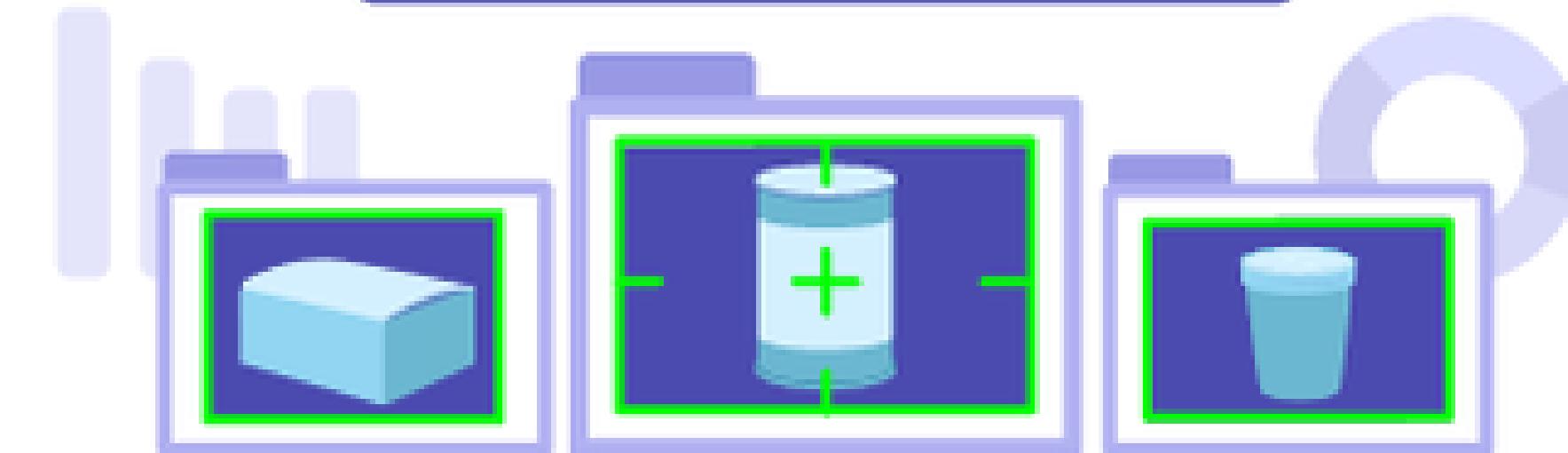


OpenCV

- OpenCV是個開源電腦視覺函示庫，由Intel公司發起並參與開發，目的為針對電腦視覺技術，建構開源的程式庫，以提供學術界或產業界使用。
- 在第三方開發者的協助下，目前OpenCV以內建許多好用的功能模組。

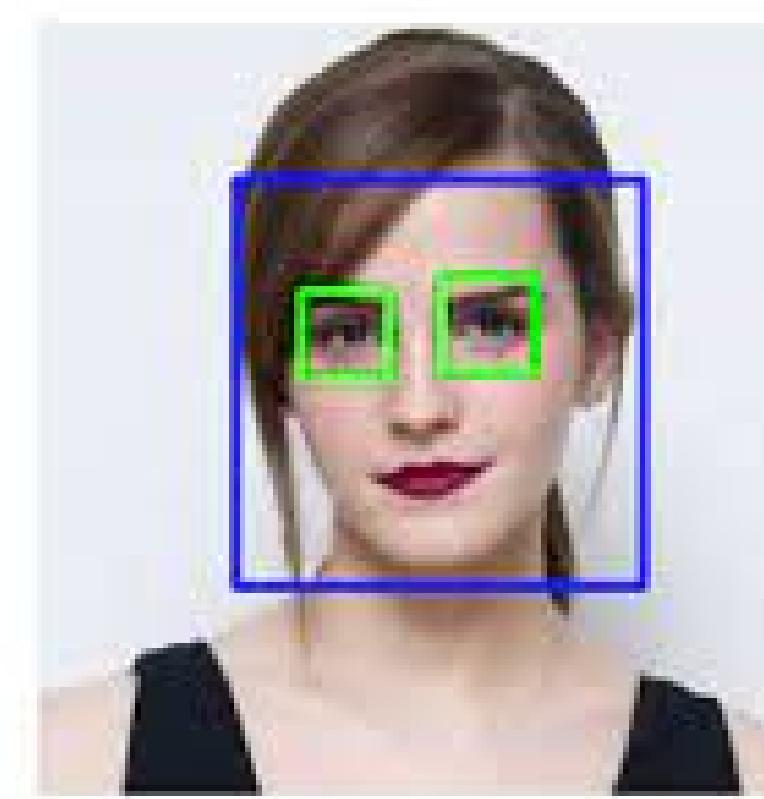


商品影像AI辨識實務



OpenCV

- OpenCV在影像處理方面應用廣泛，可以讀取儲存圖片、視訊、矩陣運算、統計、影像處理等，可用在物體追蹤、人臉辨識、紋理分析、動態視訊的影像處理等。
- OpenCV提供簡單的GUI介面，例如將影像顯示在螢幕上，在視窗上加上滑動桿和偵測滑鼠和鍵盤輸入，方便我們驗證或呈現結果。但因為OpenCV當初設計的時候著重在演算法的處理，所以關於系統硬體的支援，和介面元件的完整度都不高，所以假使想要開發完整的C/C++應用程式，還是需要像Qt、wxWidgets之類的應用程式框架。



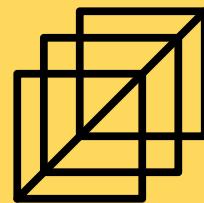
OpenCV

- 讀取與顯示數位影像: 既然要對影像進行處理，最基本的要先進行影像的輸出與輸入，OpenCV支援的影像檔案格式很多，包含:BMP、JPEG、GIF、PNG、TIFF等等。
- 其中BMP檔案格式通常沒有採用壓縮技術，在讀寫過程中不會造成影像失真，因此本次將以BMP檔案格式實作。程式碼如下:

```
import numpy as np
import cv2
import pathlib

img_file='0000982117.jfif'
path=pathlib.Path(f'C:/Users/TKU-STAFF/Pictures/Saved Pictures/{img_file}')
if not path.exists():
    print('檔案不存在')
    exit()

img = cv2.imread(str(path))
cv2.imshow("Example", img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```



THE END

