

自行車竊案分析

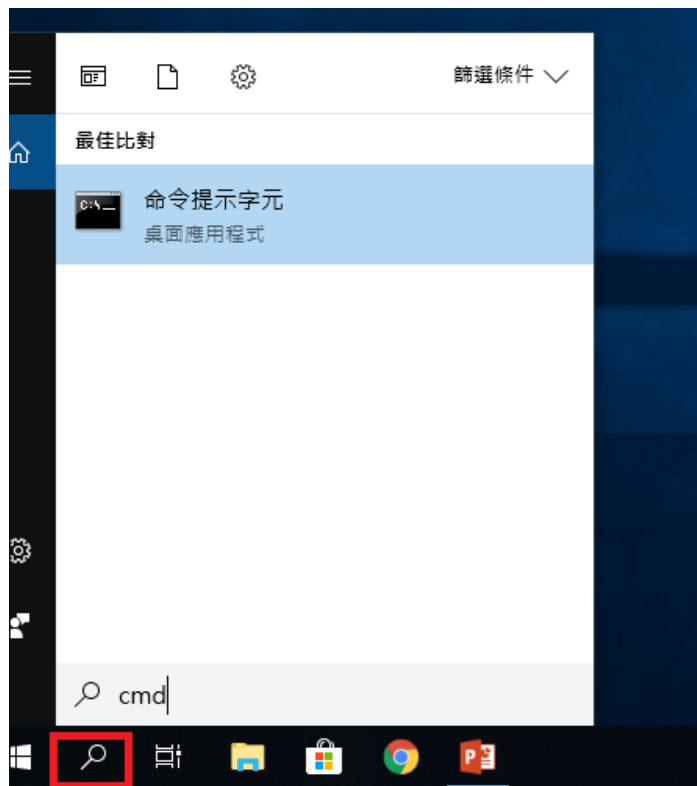
圖表繪製

matplotlib

- ▶ 今天我們要練習使用 matplotlib 這個 Python 的繪圖套件。
- ▶ 只要具備函式與模組的基本知識便可使用。
- ▶ 師父領進門，修行在個人。
 - matplotlib 的繪圖功能很多，課堂上先帶大家瀏覽基本功能。
 - 若有進階需求，詢問 Google 大神通常也都會有答案。

準備工作：安裝 matplotlib 套件

Step 1-1. 點擊左下角搜尋，輸入 **cmd** 後按 Enter 鍵。



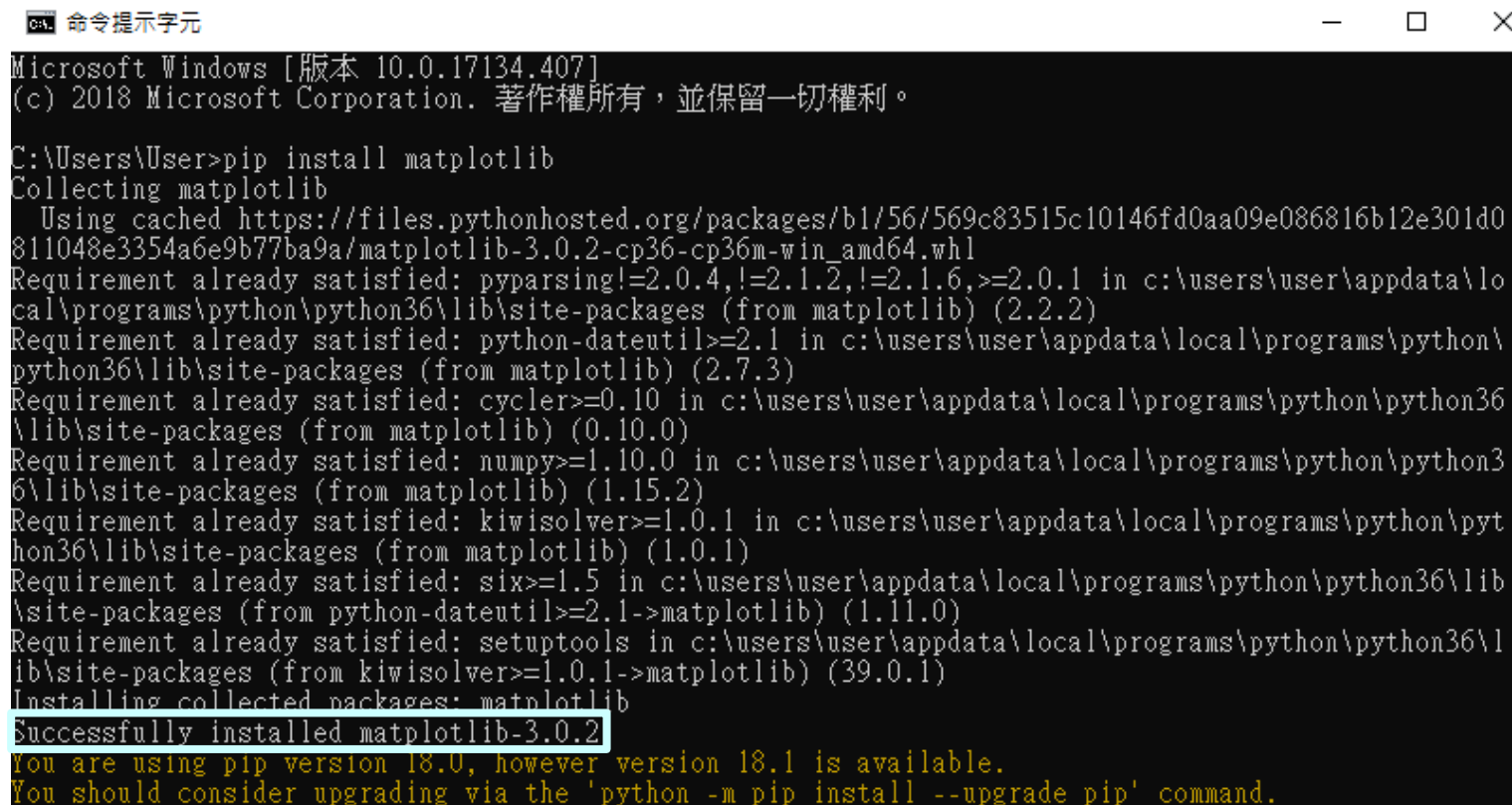
Step 1-2. 在出現的命令提示字元視窗中，輸入 **pip install matplotlib** 後按 Enter 鍵。



p.s. win7 的搜尋在開始裡面 (windows旗幟)

準備工作：安裝 matplotlib 套件

若出現以下畫面表示成功安裝。



```
命令提示字元
Microsoft Windows [版本 10.0.17134.407]
(c) 2018 Microsoft Corporation. 著作權所有，並保留一切權利。

C:\Users\User>pip install matplotlib
Collecting matplotlib
  Using cached https://files.pythonhosted.org/packages/b1/56/569c83515c10146fd0aa09e086816b12e301d0811048e3354a6e9b77ba9a/matplotlib-3.0.2-cp36-cp36m-win_amd64.whl
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in c:\users\user\appdata\local\programs\python\python36\lib\site-packages (from matplotlib) (2.2.2)
Requirement already satisfied: python-dateutil>=2.1 in c:\users\user\appdata\local\programs\python\python36\lib\site-packages (from matplotlib) (2.7.3)
Requirement already satisfied: cycler>=0.10 in c:\users\user\appdata\local\programs\python\python36\lib\site-packages (from matplotlib) (0.10.0)
Requirement already satisfied: numpy>=1.10.0 in c:\users\user\appdata\local\programs\python\python36\lib\site-packages (from matplotlib) (1.15.2)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\user\appdata\local\programs\python\python36\lib\site-packages (from matplotlib) (1.0.1)
Requirement already satisfied: six>=1.5 in c:\users\user\appdata\local\programs\python\python36\lib\site-packages (from python-dateutil>=2.1->matplotlib) (1.11.0)
Requirement already satisfied: setuptools in c:\users\user\appdata\local\programs\python\python36\lib\site-packages (from kiwisolver>=1.0.1->matplotlib) (39.0.1)
Installing collected packages: matplotlib
Successfully installed matplotlib-3.0.2
You are using pip version 18.0, however version 18.1 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
```

出現紅色文字就是有錯誤發生。

準備工作：設定 matplotlib 中文字型

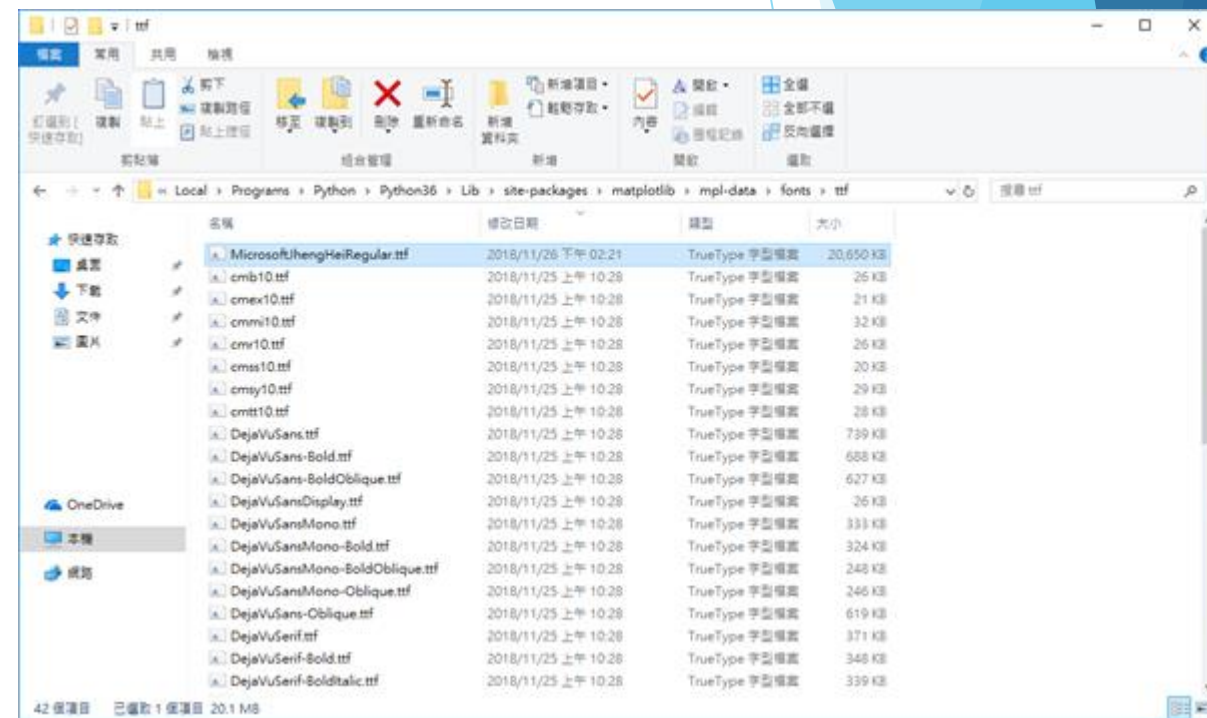
matplotlib 預設的字型不支援中文顯示，需要自己設定。

Step 2-1. 下載微軟正黑體的 ttf 檔案。

Step 2-2. 把 ttf 檔放入到 matplotlib 字型的資料夾中。

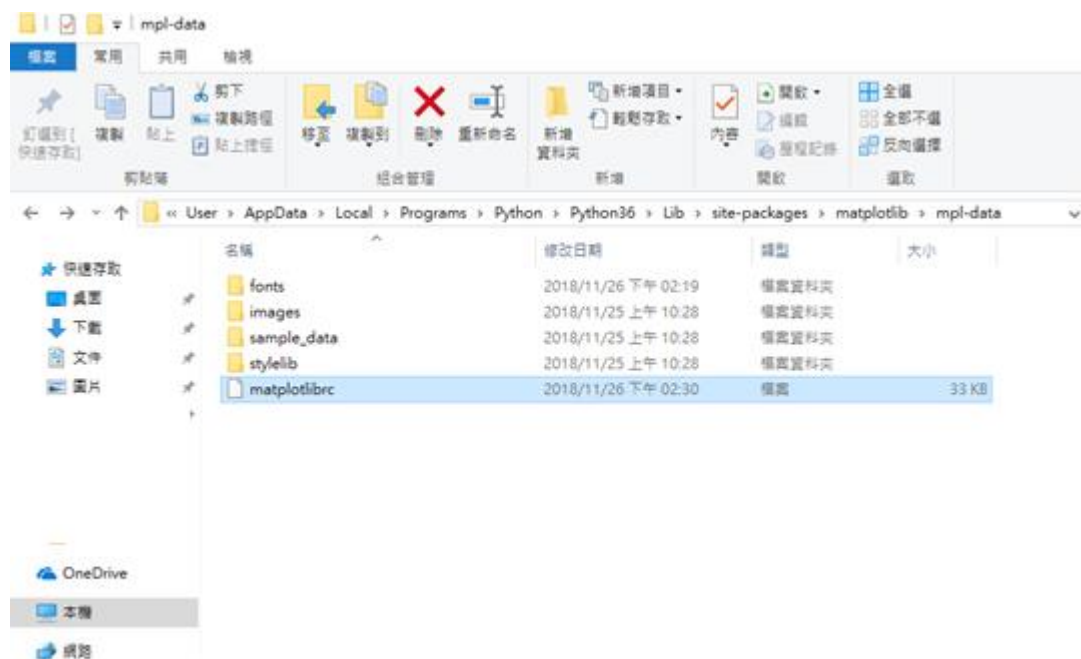
位置是：

安裝 Python 的位置 `Lib\site-packages\matplotlib\mpl-data\fonts\ttf`



準備工作：設定 matplotlib 中文字型

Step 2-3. 回到上兩層的 *mpl-data* 資料夾中，找到 *matplotlibrc* 檔案並開啟。



p.s. 點兩下後選記事本開啟

Step 2-4-1 把下列兩行的 # 去掉

```
#font.family
#font.sans-serif
```

Step 2-4-2 在 `font.sans-serif` 這行的冒號後加上 Microsoft JhengHei, °

```
font.family      : sans-serif
#font.style      : normal
#font.variant    : normal
#font.weight     : normal
#font.stretch    : normal
## note that font.size controls default text sizes. To configure
## special text sizes tick labels, axes, labels, title, etc, see the rc
## settings for axes and ticks. Special text sizes can be defined
## relative to font.size, using the following values: xx-small, x-small,
## small, medium, large, x-large, xx-large, larger, or smaller
font.size        : 10.0
#font.serif      : DejaVu Serif, Bitstream Vera Serif, Computer Modern
font.sans-serif  : Microsoft JhengHei, DejaVu Sans, Bitstream Vera Sans
#font.cursive    : Apple Chancery, Textile, Zapf Chancery, Sand, Scrip
#font.fantasy    : Comic Sans MS, Chicago, Charcoal, ImpactWestern, Hu
#font.monospace  : DejaVu Sans Mono, Bitstream Vera Sans Mono, Compute
```

p.s. 可以用記事本上方編輯內的尋找

準備工作：設定 matplotlib 中文字型

Step 2-5. 找到

```
#axes.unicode_minus : True ,
```

把 # 刪掉並改成 False 後存檔。

```
#axes.spines.left : True ## display axis spines
#axes.spines.bottom : True
#axes.spines.top : True
#axes.spines.right : True
axes.unicode_minus : False ## use unicode for the minus symbol
## rather than hyphen. See
## http://en.wikipedia.org/wiki/Plus_minus
#axes.prop_cycle : cycler('color', ['1f77b4', 'ff7f0e', '2ca02c'])
## color cycle for plot lines as list of strings
## colorspecs: single letter, long name, or wavelength
## Note the use of string
## as opposed to the rest of this file.
#axes.autolimit_mode : data ## How to scale axes limits to the data
## Use "data" to use data limits, plus
## Use "round_number" move to the nearest
#axes.xmargin : .05 ## x margin. See `axes.Axes.margins`
#axes.ymargin : .05 ## y margin See `axes.Axes.margins`
#polaraxes.grid : True ## display grid on polar axes
#axes3d.grid : True ## display grid on 3d axes
```

Step 2-6. 找到 *matplotlib* 的快取資料夾，把它刪掉。

.idlerc	2018/10/1 下午 0...	檔案資料夾
.ipython	2018/10/4 下午 0...	檔案資料夾
.jupyter	2018/10/4 下午 0...	檔案資料夾
.matplotlib	2018/11/26 下午 ...	檔案資料夾
3D 物件	2018/11/15 下午 ...	檔案資料夾
AppData	2018/9/21 上午 1...	檔案資料夾
Intel	2018/9/27 上午 0...	檔案資料夾
MicrosoftEdgeBackups	2018/9/21 上午 1...	檔案資料夾
OneDrive	2018/11/16 上午 ...	檔案資料夾
wekafiles	2018/10/4 下午 0...	檔案資料夾
下載	2018/11/26 下午 ...	檔案資料夾
文件	2018/11/15 下午 ...	檔案資料夾
我的最愛	2018/11/15 下午 ...	檔案資料夾
音樂	2018/11/15 下午 ...	檔案資料夾

準備工作：設定 matplotlib 中文字型

Step 2-7. 若覺得步驟太麻煩，下載 *one_click_to_move_font.py* 並執行，讓他來處理。

```
import matplotlib
import shutil

# matplotlib.__file__ will return path to __init__.py.
dest_name = matplotlib.__file__[:-11] + 'mpl-data\\fonts\\ttf\\MicrosoftJhengHeiRegular.ttf'
# move the ttf file to target folder.
shutil.move('MicrosoftJhengHeiRegular.ttf', dest_name)

# find matplotlibrc and modify the value
brc_loc = matplotlib.__file__[:-11] + 'mpl-data\\matplotlibrc'
with open(brc_loc, "r+") as f:
    old = f.read()
    old = old.replace('#font.family', 'font.family')
    old = old.replace('#font.sans-serif      : ', 'font.sans-serif      : Microsoft JhengHei, ')
    old = old.replace('#axes.unicode_minus : True', 'axes.unicode_minus : False')
    f.seek(0) # sets the file's current position at the offset to 0.
    f.truncate() # truncates the file's size. (Avoiding the new content is shorter than original.)
    f.write(old)

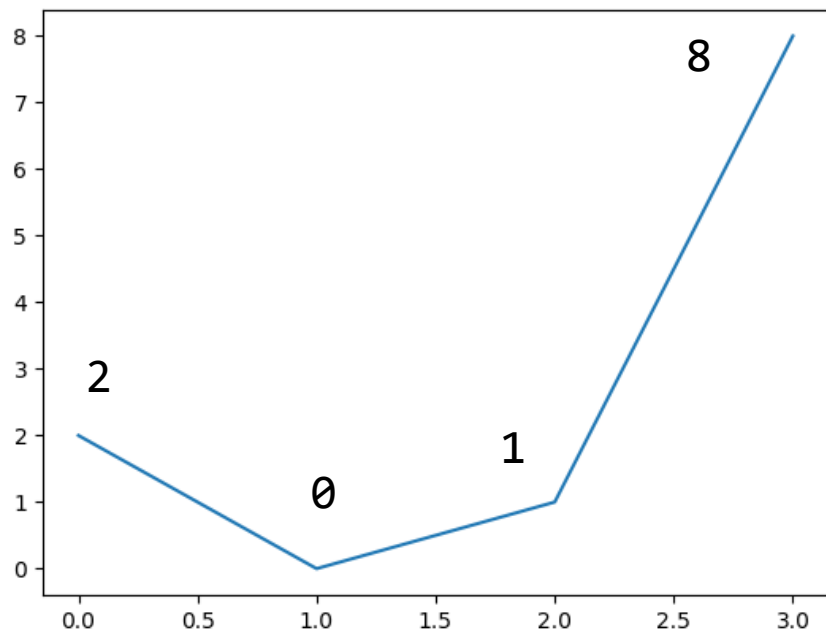
# delete the cache folder
shutil.rmtree(matplotlib.get_cachedir(), ignore_errors=True)
```

p.s. 下載下來的檔案要與剛剛載的字型檔案處於同一個資料夾。

立馬體驗：plot() 與 show()

```
import matplotlib.pyplot as plt
```

```
plt.plot([2, 0, 1, 8]) #繪製 2, 0, 1, 8 四個資料  
plt.show()            #沒有呼叫 show() 就不會秀出來喔
```



最簡單的傳法，把資料點 (就是 y) 傳入 plot()。

x 座標點預設是 0, 1, 2, ...。

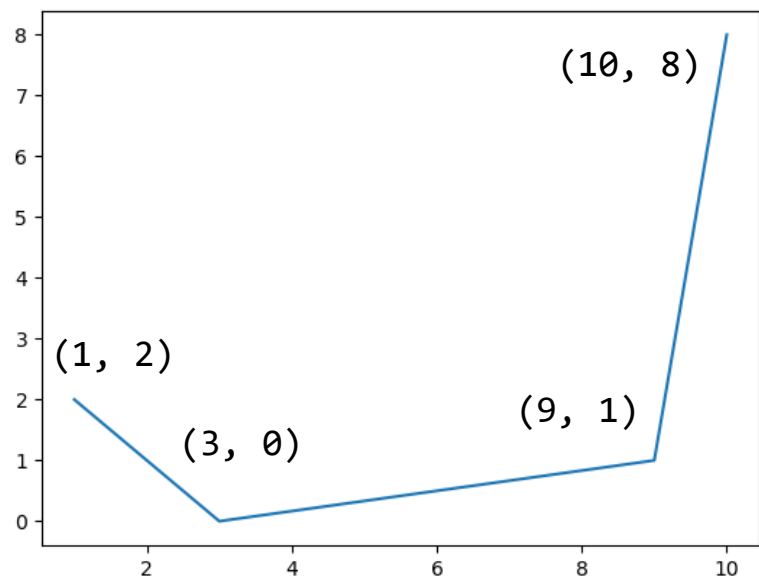
如果只想快速看一個趨勢，咻一下就有得看了。

立馬體驗：plot() 與 show()

▶ 指定 x 座標值。

```
import matplotlib.pyplot as plt

x = [1, 3, 9, 10]
y = [2, 0, 1, 8]
plt.plot(x, y)    #若 x 和 y 長度不同會發生執行錯誤
plt.show()
```



💡 想想看，如果資料是這樣：
data = [1, 2, 3, 0, 9, 1, 10, 8]
該怎麼拿出 x 和 y 資料值？

如果是這樣呢？

```
data = [[1, 2], [3, 0], [9, 1], [10, 8]]
```

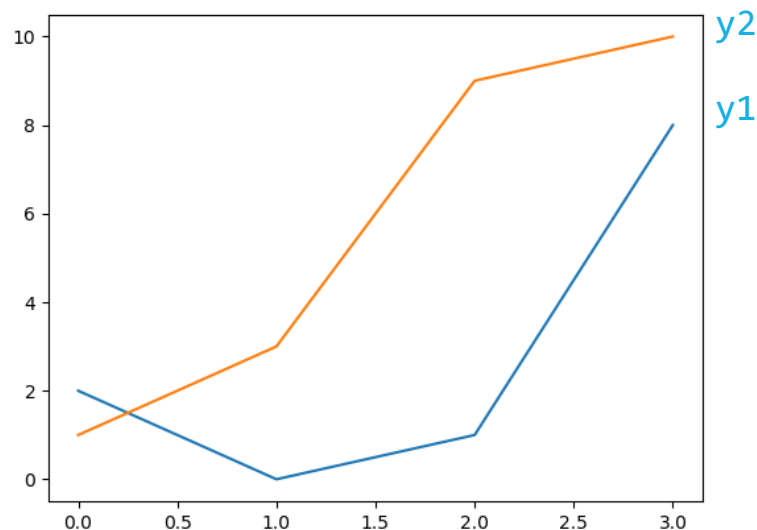
參考答案1 x = data[:,2]

參考答案2 x = [e[0] for e in data]

立馬體驗：plot() 與 show()

▶ 多組資料也沒問題。

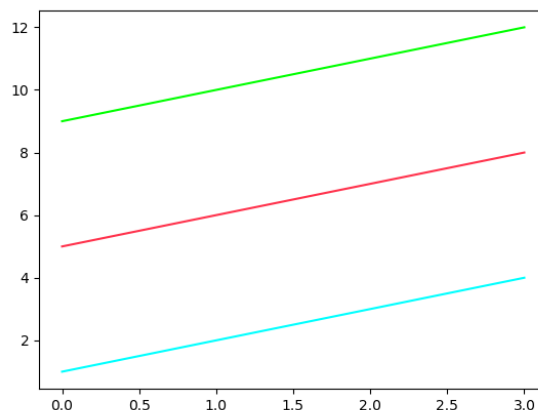
```
import matplotlib.pyplot as plt  
  
y1 = [2, 0, 1, 8]  
y2 = [1, 3, 9, 10]  
plt.plot(y1)  
plt.plot(y2)  
plt.show()
```



折線圖：plot()

▶ 線條參數

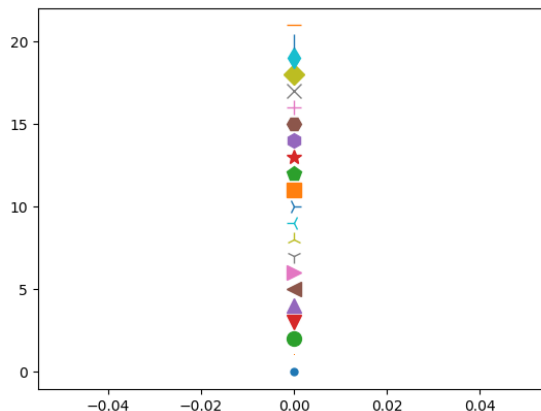
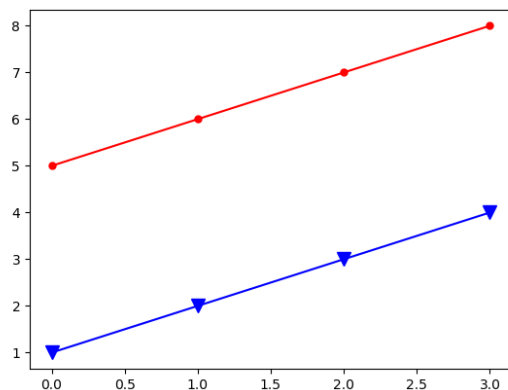
參數名稱	可用值與範例
color	字串：blue, green, red, cyan, magenta, yellow, black, white 三原色浮點數：(1.0, 0.2, 0.3) 三原色十六進位：#00FF00 <pre>plt.plot([9, 10, 11, 12], color='#00FF00') plt.plot([5, 6, 7, 8], color=(1.0, 0.2, 0.3)) plt.plot([1, 2, 3, 4], color='cyan')</pre>



折線圖：plot()

▶ 線條參數

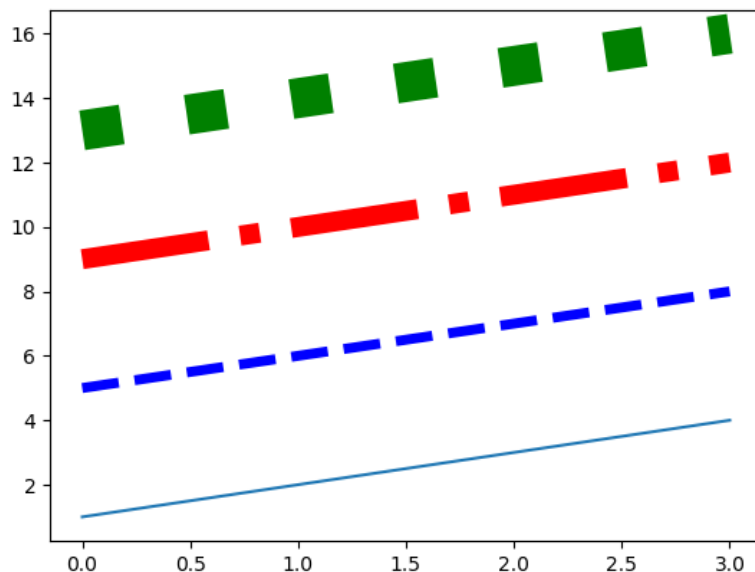
參數名稱	可用值與範例
marker	字串：. , o v ^ < > 1 2 3 4 s p * h H + x D d _
markersize	<pre>plt.plot([5, 6, 7, 8], color='red', marker='v', markersize=5) plt.plot([1, 2, 3, 4], color='blue', marker='o', markersize=10) plt.show()</pre> <pre>mstr = '.,ov^<>1234sp*hH+xDd _' for i in range(len(mstr)): plt.plot([i], marker=mstr[i], markersize=10) plt.show()</pre>



折線圖：plot()

▶ 線條參數

參數名稱	可用值與範例
linestyle	字串：- -- -. :
linewidth	<pre>plt.plot([13, 14, 15, 16], color='green', linestyle=':', linewidth=20) plt.plot([9, 10, 11, 12], color='red', linestyle='-.', linewidth=10) plt.plot([5, 6, 7, 8], color='blue', linestyle='--', linewidth=5) plt.plot([1, 2, 3, 4])</pre>

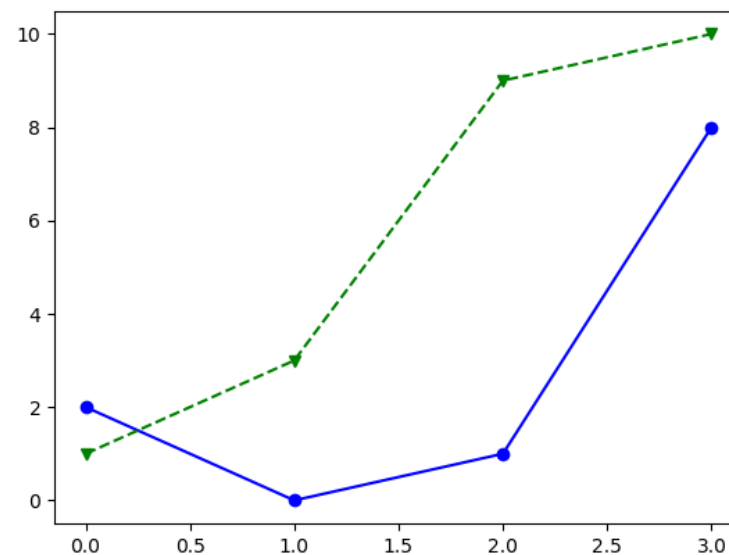


折線圖：plot()

▶ 簡易版字串傳法

- 顏色 (color)、符號 (marker) 和線條樣式 (linestyle) 三位一體！

```
import matplotlib.pyplot as plt  
  
y1 = [2, 0, 1, 8]  
y2 = [1, 3, 9, 10]  
plt.plot(y1, 'bo--')  
plt.plot(y2, 'gv--')  
plt.show()
```

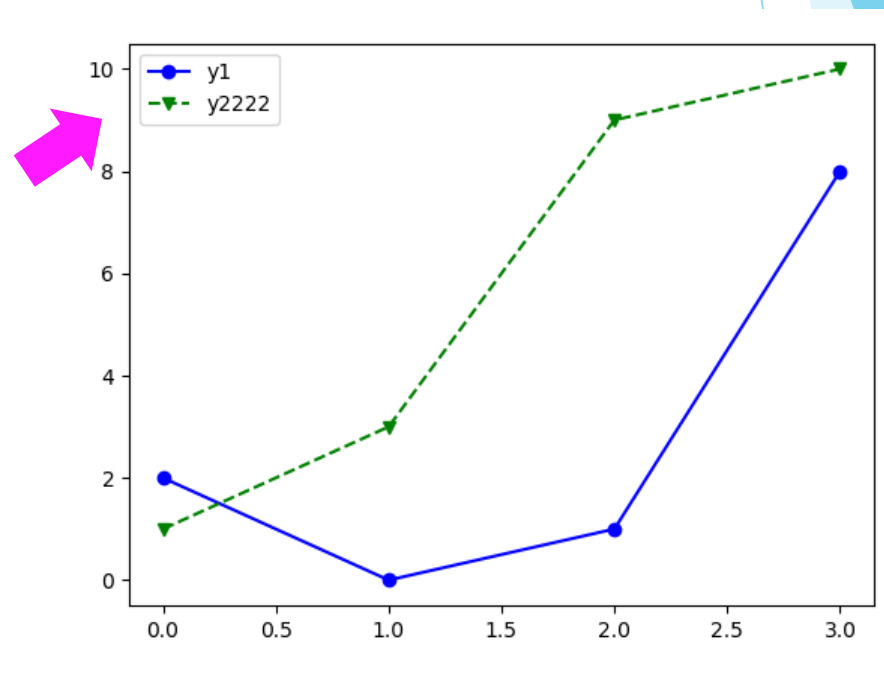


通用：資料標籤

- ▶ 使用 `plot()` 的參數 `label`。
- ▶ 呼叫 `plt.legend()`。

```
import matplotlib.pyplot as plt

y1 = [2, 0, 1, 8]
y2 = [1, 3, 9, 10]
plt.plot(y1, 'bo-', label='y1')
plt.plot(y2, 'gv--', label='y2222')
plt.legend()
plt.show()
```



通用：標籤

▶ 呼叫 pyplot 模組的函式

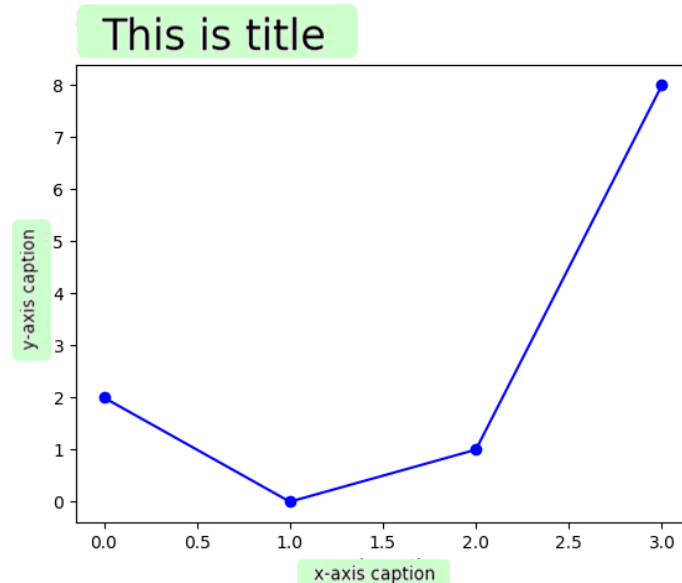
函式名稱	說明	參數 (還有很多問 google 大神)
<code>title()</code>	圖表標題	<code>fontsize</code> 字型大小 <code>loc</code> 文字位置: left center right
<code>xlabel()</code>	X 軸標題	<code>fontsize</code> 字型大小
<code>ylabel()</code>	y 軸標題	<code>fontsize</code> 字型大小

```
import matplotlib.pyplot as plt

y1 = [2, 0, 1, 8]
plt.plot(y1, 'bo-')

plt.title('This is title', fontsize=24, loc='left')
plt.xlabel('x-axis caption', fontsize=10)
plt.ylabel('y-axis caption', fontsize=10)

plt.show()
```



通用：圖表刻度

▶ 圖表刻度

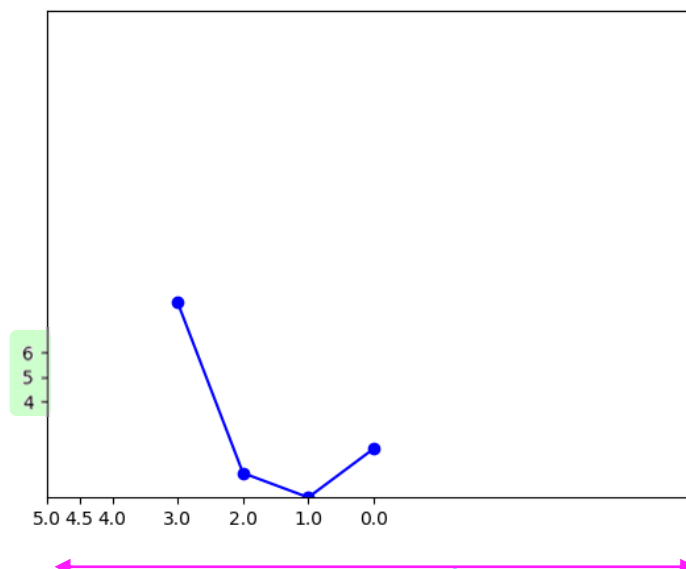
函式名稱	說明
<code>xticks()</code>	x 軸刻度，傳入 x 軸要顯示的數值。
<code>yticks()</code>	y 軸刻度，傳入 y 軸要顯示的數值。
<code>xlim()</code>	x 軸邊界，傳入最小和最大值。（反過來傳的話，圖也會反過來畫。）
<code>ylim()</code>	y 軸邊界，傳入最小和最大值。（反過來傳的話，圖也會反過來畫。）

```
import matplotlib.pyplot as plt

y1 = [2, 0, 1, 8]
plt.plot(y1, 'bo-', label='y1')

plt.xticks(list(range(10)) + [4.5])
plt.yticks(range(4, 7))

plt.xlim(5, -5)
plt.ylim(0, 20)
plt.show()
```



通用：儲存圖檔

▶ 呼叫 pyplot 模組的函式

函式名稱	說明
savefig()	傳入圖檔檔名，支援 png, pdf, eps, pgf, ps, raw, rgba, svg, svgz 格式。

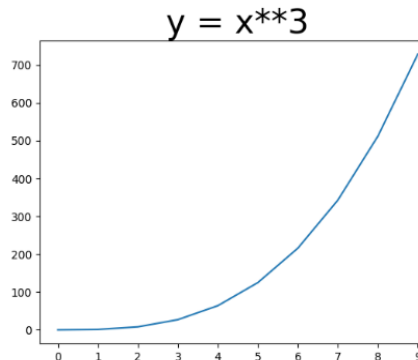
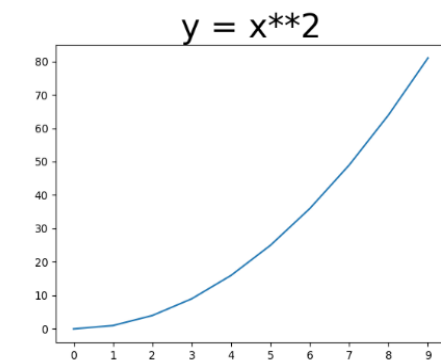
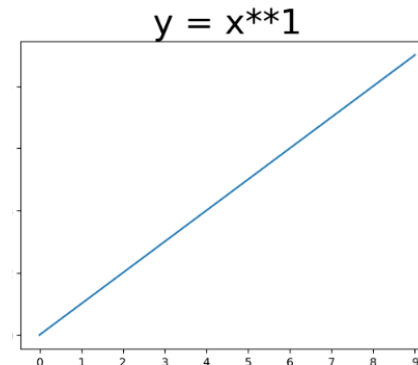
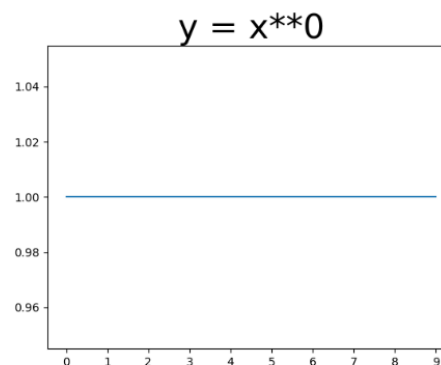
```
import matplotlib.pyplot as plt

for i in range(4):
    x = range(10)
    y = [e**i for e in x]
    plt.clf() #清除圖表
    plt.title('y = x**' + str(i))
    plt.plot(x, y)
    plt.savefig(str(i) + '.png')
```



撰寫程式快速繪製多個圖檔來放進書面報告或投影片簡報，是不是很過癮呢！

以前要針對 Excel 裡的資料一組一組慢慢拉，學會程式後，一切更有效率了！



實務應用：自行車竊案任務

```
import bike_crime as bkc
import matplotlib.pyplot as plt
```



請下載 `bike_crime.py`，並另開一個 `vis.py` 來撰寫本頁的程式碼。

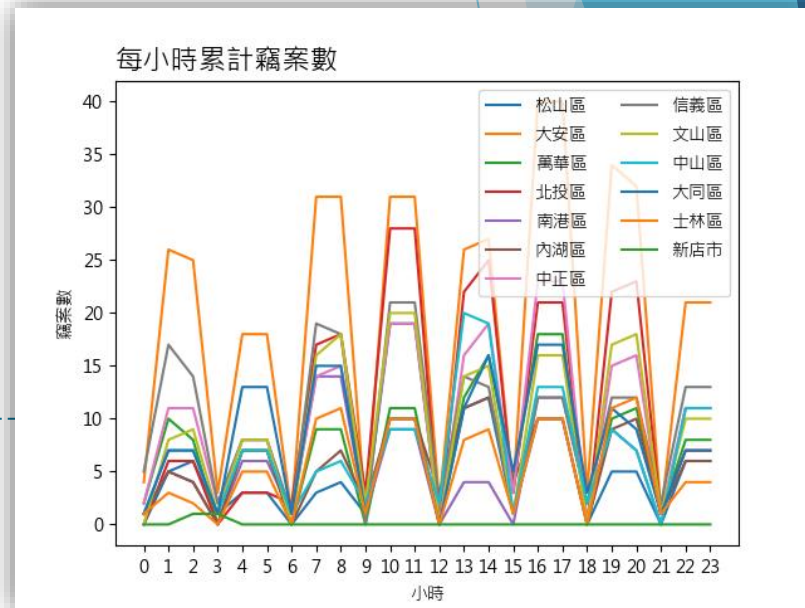
```
def main():
    data = bkc.GetCrimeData('臺北市10401-10709自行車竊盜點位資訊.txt')
    phc = bkc.GetPlaceHourCounts(data)

    #各地區每小時竊案累積數
    for e in phc:
        plt.plot(e[1], label=e[0])
    plt.xticks(range(24))
    plt.title('每小時累計竊案數', fontsize=16, loc='left')
    plt.xlabel('小時')
    plt.ylabel('竊案數')
    plt.legend(ncol=2, loc='upper right')
    plt.show()

main()
```

在程式中加上 `print(phc[:4])` 就能看到 `phc` 長這樣

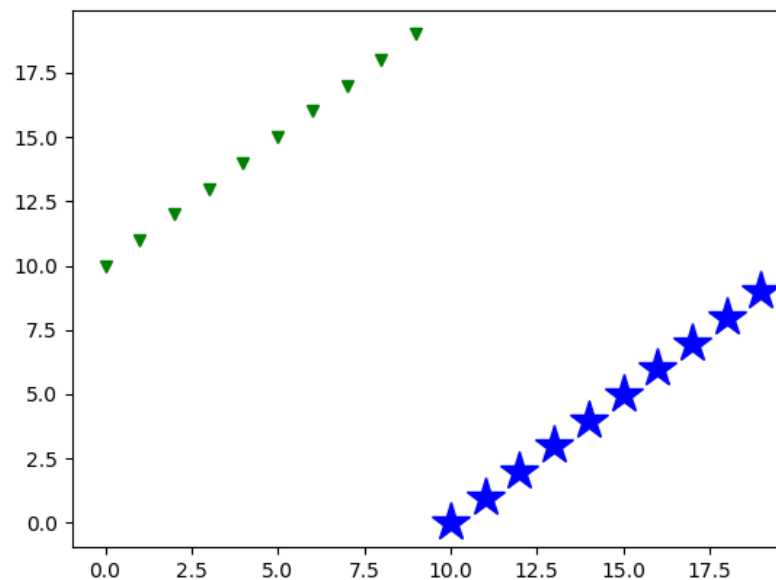
```
[
['松山區', [0, 5, 6, 1, 3, 3, 0, 3, 4, 1, 19, 19, 0, 11, 12, 1, 10, 10, 0, 5, 5, 0, 6, 6]],
['大安區', [4, 26, 25, 3, 18, 18, 1, 31, 31, 1, 31, 31, 2, 26, 27, 3, 40, 40, 3, 34, 32, 1, 21, 21]],
['萬華區', [2, 10, 8, 0, 7, 7, 0, 9, 9, 0, 11, 11, 0, 12, 16, 4, 18, 18, 0, 10, 11, 1, 8, 8]],
['北投區', [0, 6, 6, 0, 3, 3, 2, 17, 18, 3, 28, 28, 0, 22, 25, 3, 21, 21, 0, 22, 23, 1, 7, 7]],
...
]
```



散佈圖：plot()

- ▶ 散佈圖可用於觀察兩個維度資料的關係。
- ▶ 使用 plot() 便可繪製。

```
import matplotlib.pyplot as plt  
  
x = range(10)  
y = range(10, 20)  
  
plt.plot(x, y, 'gv')  
plt.plot(y, x, 'b*', markersize=20)  
plt.show()
```



不要傳入線條樣式，plot() 就不會連線，自然而然就變成散佈圖的樣子囉。

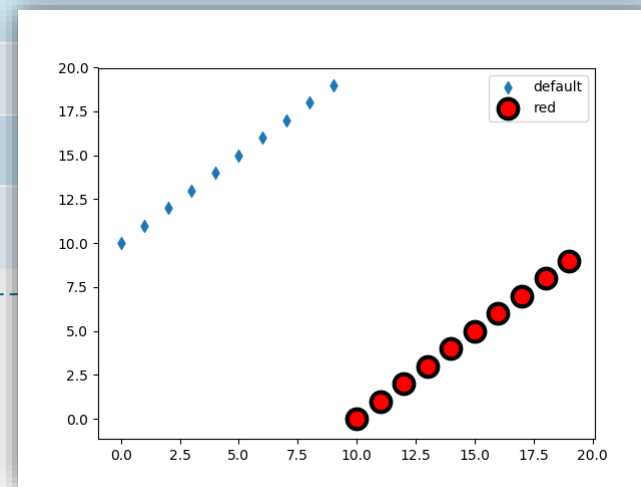
散佈圖：scatter()

▶ scatter() 是一個專門繪製散佈圖的函式。

參數名稱	可用值與範例
marker	字串：. , o v ^ < > 1 2 3 4 8 s p * h H + x D d _
s	符號大小
c	符號顏色
linewidths	符號邊框寬度
edgecolors	符號邊框顏色
label	符號標籤

```
import matplotlib.pyplot as plt
x = range(10)
y = range(10, 20)
```

```
plt.scatter(x, y, marker='d', label='default')
plt.scatter(y, x, s=200, c='red', linewidths=3, edgecolors='black',
            label='red')
plt.legend()
plt.show()
```



散佈圖：scatter()

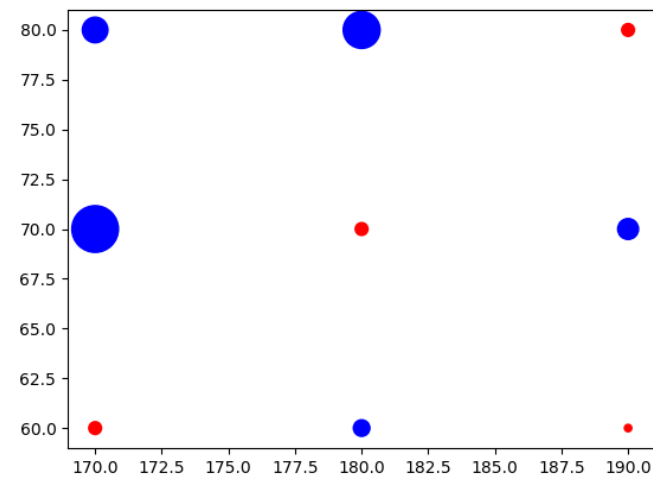
▶ scatter() 可以指定個別資料點的顏色與大小。

```
import matplotlib.pyplot as plt

#[身高, 體重, 人數]
h_w_c = [[170, 80, 12], [180, 80, 25], [190, 80, 3],
          [170, 70, 40], [180, 70, 3], [190, 70, 8],
          [170, 60, 3], [180, 60, 5], [190, 60, 1]]

x = [e[0] for e in h_w_c] # 以身高作為 x 軸
y = [e[1] for e in h_w_c] # 以體重作為 y 軸
s = [e[2]*20 for e in h_w_c] # 以人數作為符號大小
c = ['blue' if e[2]>3 else 'red' for e in h_w_c]

plt.scatter(x, y, s, c)
plt.show()
```



再加上 alpha 透明度，傳說中帥氣的泡泡圖就可以這樣畫出來囉！
(試試在上面的 scatter 呼叫多加 alpha=0.3 。)

實務應用：自行車竊案任務

```
import bike_crime as bkc
import matplotlib.pyplot as plt
```



請從 moodle 下載 bike_crime.py，並
另開一個 vis.py 來撰寫本頁程式碼。

```
def GetMonthHour(record):
    return bkc.GetMonth(record), bkc.GetHour(record)

def main():
    data = bkc.GetCrimeData('臺北市10401-10709自行車竊盜點位資訊.txt')

    pmhc = bkc.GetGroupCounts(data, GetMonthHour)

    counts = [[0]*24 for m in range(13)] #counts[m][h] 記錄 m 月份 h 時刻的竊案總數
    for e in pmhc:
        counts[e[0][0]][e[0][1][0]] += e[1]
        counts[e[0][0]][e[0][1][0]+1] += e[1]
```

在程式中加上 `print(pmhc[:4])` 就能看到 pmhc 長這樣

```
[
[(6, (13, 15)), 15],
[(1, (16, 18)), 20],
[(1, (7, 9)), 23],
[(1, (13, 15)), 11]
...
]
```

接下一頁

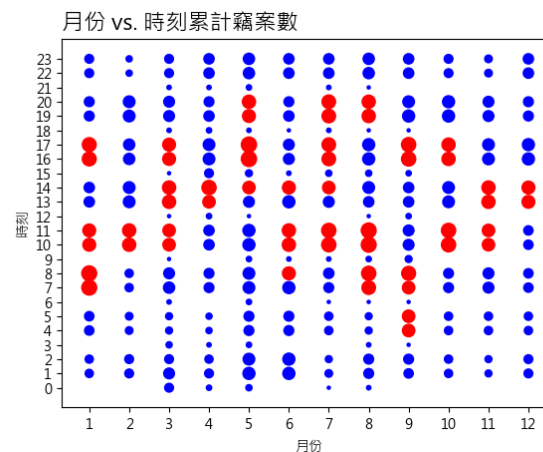
實務應用：自行車竊案任務

接上一頁

```
x = [] #以月份作 x 座標
y = [] #以時刻作 y 座標
s = [] #以竊案數作大小 (size)
c = [] #以是否超過 15 件區別藍紅色
for m in range(1, 13):
    for h in range(24):
        x.append(m)
        y.append(h)
        s.append(counts[m][h]*5)
        c.append('blue' if counts[m][h]<=15 else 'red')

plt.title('月份 vs. 時刻累計竊案數', fontsize=16, loc='left')
plt.xlabel('月份')
plt.xticks(range(1, 13))
plt.ylabel('時刻')
plt.yticks(range(0, 24))
plt.scatter(x, y, s, c)
plt.show()

main()
```



長條圖：bar()

- ▶ 長條圖用以比較多組資料的數值大小。
- ▶ bar() 函式可以繪製長條圖。

參數名稱	可用值與範例
x	x軸座標值
height	y軸座標值
width	長條寬度 (預設 0.8)
label	標籤

```
import matplotlib.pyplot as plt
```

```
counts = [5, 2, 4, 7, 1]
```

```
x = range(0, 10, 2) #0, 2, 4, 6, 8
```

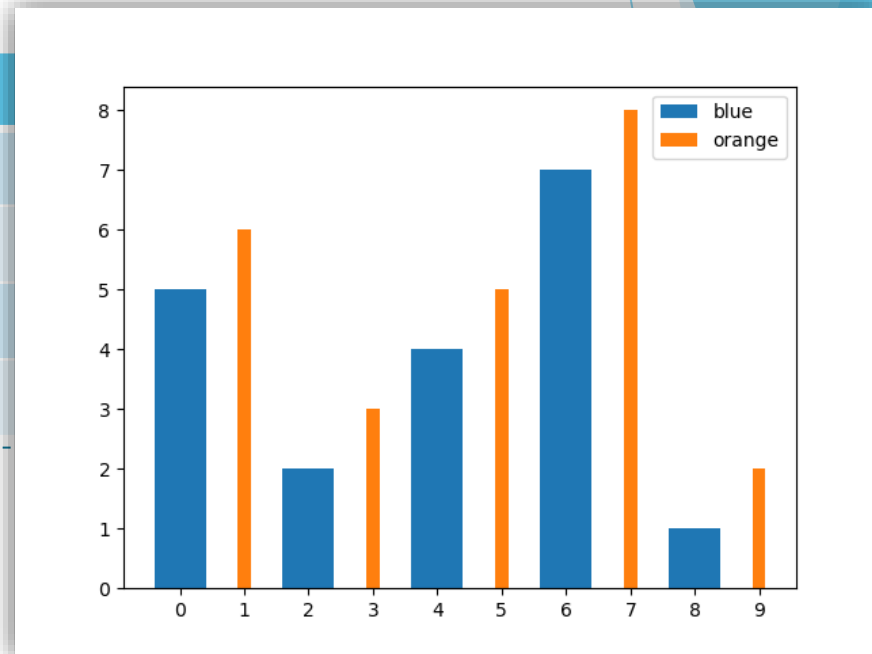
```
plt.bar(x, counts, label='blue')
```

```
plt.bar(x=[e+1 for e in x], height=[e+1 for e in counts], width=0.2, label='orange')
```

```
plt.xticks(range(10))
```

```
plt.legend()
```

```
plt.show()
```

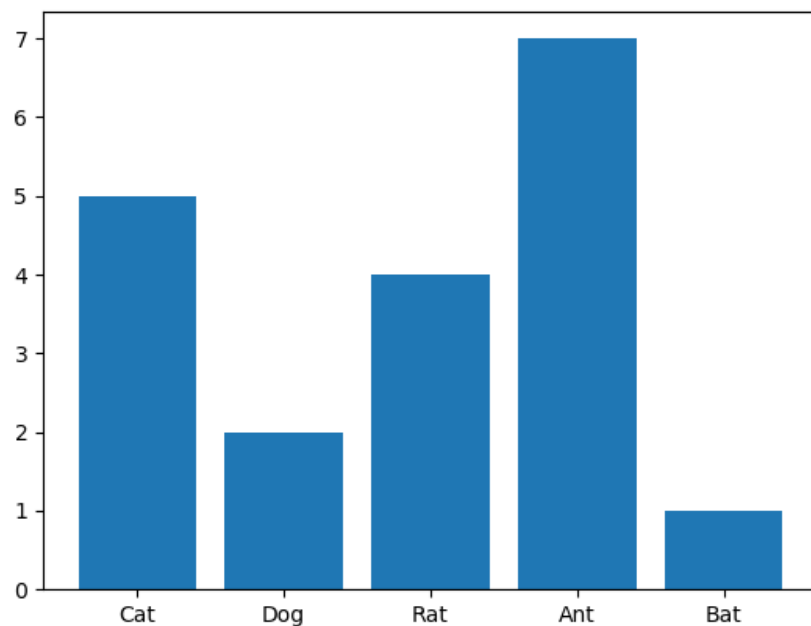


長條圖：bar()

▶ 使用 `xticks()` 來指定位置上的文字。

```
import matplotlib.pyplot as plt

data = [['Cat', 5], ['Dog', 2], ['Rat', 4], ['Ant', 7], ['Bat', 1]]
plt.bar(range(len(data)), [e[1] for e in data])
plt.xticks(range(len(data)), [e[0] for e in data])
plt.show()
```



實務應用：自行車竊案任務

```
import bike_crime as bkc
import matplotlib.pyplot as plt
```



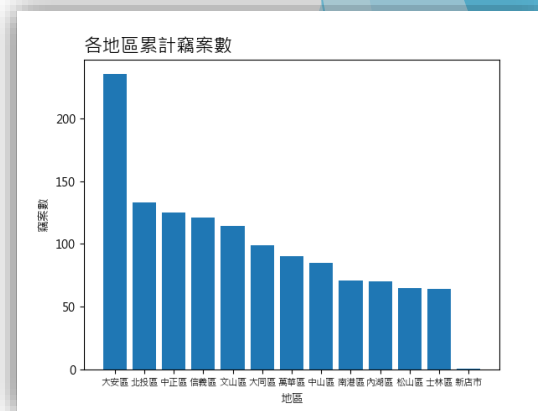
請從 moodle 下載 bike_crime.py，並另開一個 vis.py 來撰寫本頁程式碼。

```
def main():
    data = bkc.GetCrimeData('臺北市10401-10709自行車竊盜點位資訊.txt')

    #各地區竊案累積總數
    pc = bkc.GetGroupCounts(data, bkc.GetPlace)
    def cmp(x):
        return x[1]
    pc = bkc.GetGroupCounts(data, bkc.GetPlace)
    pc.sort(key=cmp, reverse=True)

    plt.bar(range(len(pc)), [e[1] for e in pc])
    plt.xticks(range(len(pc)), [e[0] for e in pc], fontsize=8)
    plt.title('各地區累計竊案數', fontsize=16, loc='left')
    plt.xlabel('地區')
    plt.ylabel('竊案數')
    plt.show()

main()
```



快速檢索

#折線圖

[#基本用法](#) [#顏色](#) [#符號](#) [#線條樣式](#) [#標籤](#)

#散佈圖

[#plot\(\)](#) [#scatter\(\)](#)

#長條圖

[#基本用法](#) [#加註軸文字](#)

#通用

[#資料標籤](#) [#標題](#) [#軸刻度](#) [#存檔](#)

#自行車竊案範例

[#折線圖](#) [#散佈圖](#) [#長條圖](#)