

chapter 5

圖表繪製

/ 蔣宗哲、江政杰

俗話說，「一圖勝千言」。當分析過程中產生的資料趨於龐大繁雜，資料科學家們會尋求更適當的方式來呈現，其中各式圖表就是值得考慮的好工具了。本章將講述如何運用 `matplotlib` 套件來繪製折線圖、散佈圖和長條圖，並接續前一章的臺北市自行車竊案分析應用，展示各種圖表的應用時機。學習完本章後，讀者將可撰寫程式繪製最常見的三種圖表，設定圖表細節並存成圖片檔案以備後續編輯使用。

5-1 matplotlib 套件安裝

5
圖表繪製

俗諺有云，「一圖勝千言」(A picture is worth a thousand words.)，以圖表方式呈現資料，讓人能夠一目瞭然迅速理解資料所訴說的意涵。很多時候，我們希望能以圖表型式來呈現資料分析的結果。這一章我們即將學習以 Python 程式繪製折線圖、散佈圖以及長條圖這三種圖表。

Python 語言有數種繪圖套件，這裡我們使用的是 matplotlib 套件。這一節示範的是以 Windows 作業系統安裝 matplotlib 的流程。其它作業系統的安裝方式可以參考網路上的教學文章。

準備工作：安裝 matplotlib 套件

開啟命令提示字元視窗：點擊左下角搜尋，輸入 cmd 後按 Enter 鍵。接著，在命令提示字元視窗中鍵入 pip install matplotlib，然後按 Enter 鍵。

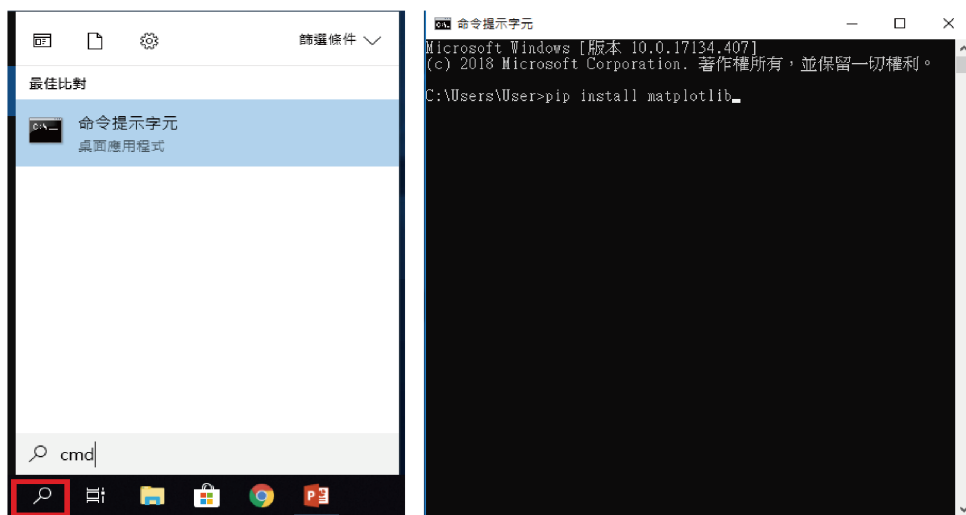


圖 5-1：安裝 matplotlib 套件

若成功安裝，會出現如圖 5-2 的畫面。

```
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 10.0, however version 18.1 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
```

圖 5-2：matplotlib 套件安裝成功

準備工作：中文字顯示問題

matplotlib 套件預設狀態無法顯示中文字型，圖表上顯示中文字時會出現亂碼，因此需要額外進行設定。此部份流程較為繁瑣，建議讀者上網查詢符合己身系統的教學文章。

5-2 折線圖

立馬體驗：plot() 與 show()

請先安裝好 matplotlib 套件，我們來實際體驗一下如何繪製圖表。程式

E5-2-1 的第 01 行帶入 matplotlib 套件的 pyplot 模組，並以縮寫 plt 代表之。程式第 02 行呼叫 plt 模組中的 plot() 函式，傳入一個表示資料數值的列表。第 03 行呼叫 plt 模組中的 show() 函式，顯示繪製結果，如圖 5-3 所示。

E5-2-1.py:

```
01 import matplotlib.pyplot as plt
02 plt.plot([2, 0, 1, 8])
03 plt.show()
```

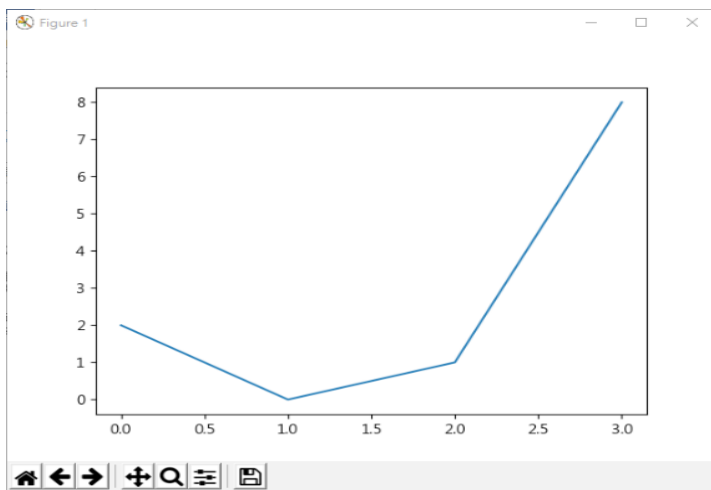


圖 5-3：程式 E5-2-1.py 之繪製結果

當我們只傳入一個列表時，這個列表代表的是 y 軸的資料值，而 x 軸預設為 0, 1, 2, ...。因此圖 5-3 中的第一個資料點是 (0, 2)，第二個資料點是 (1, 0)，以此類推。如果我們只是想觀察一組有序資料的趨勢，只要三行程式碼就能輕鬆完成。

當我們同時擁有 x 和 y 軸的資料值時，則需分別將兩個列表傳入 `plot()` 函式，如程式 E5-2-2 所示。注意，若 x 和 y 的長度不同，會發生 `ValueError` 的錯誤。圖 5-4 為繪製結果，由四個資料點 (1, 2)、(3, 0)、(9, 1) 和 (10, 8) 構成折線。

E5-2-2.py：繪製一條折線

```
01 import matplotlib.pyplot as plt
02 x = [1, 3, 9, 10]
03 y = [2, 0, 1, 8]
04 plt.plot(x, y)
05 plt.show()
```

我們當然也可以在同一張圖上繪製多條折線，只需呼叫 `plot()` 數次，分別傳入資料即可，如程式 E5-2-3 所示。

E5-2-3.py：繪製兩條折線

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

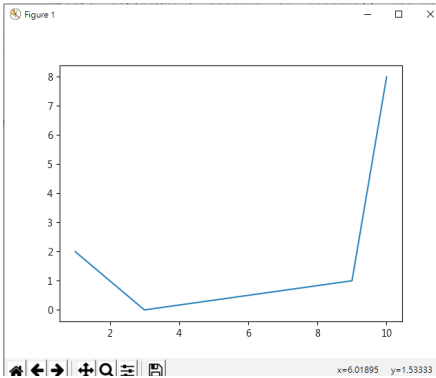


圖 5-4：E5-2-2.py 之繪製結果

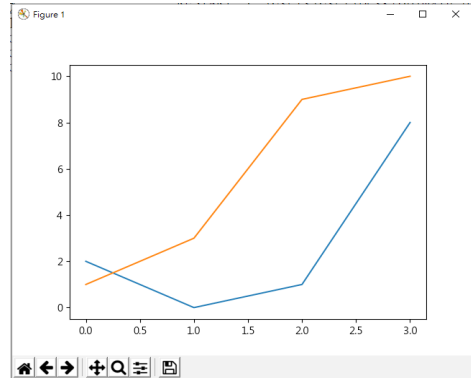


圖 5-5：E5-2-3.py 之繪製結果

plot() 的常用參數

plot() 函式具備有非常多樣的參數設定，讓程式員可以根據自己的需求，繪製出各種變化的圖表。我們先來看線條的顏色怎麼設定。plot() 函式繪製線條時有預設的線條色彩順序，也就是你不必指定線條的顏色，每次呼叫 plot() 函式時，Python 會自動給予線條的顏色。當然，我們也可以自行指定色彩，只需透過 color 參數來指定。表 5-2-1 說明三種指定 color 的方式：以色彩字串、紅綠藍三原色浮點數或紅綠藍三原色十六進位來表示，越接近 00 代表那一項的原色量越少，越接近 FF 代表那一項的原色量越大。

表 5-2-1：plot() 的線條色彩 (color) 參數

參數名稱	可用值
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

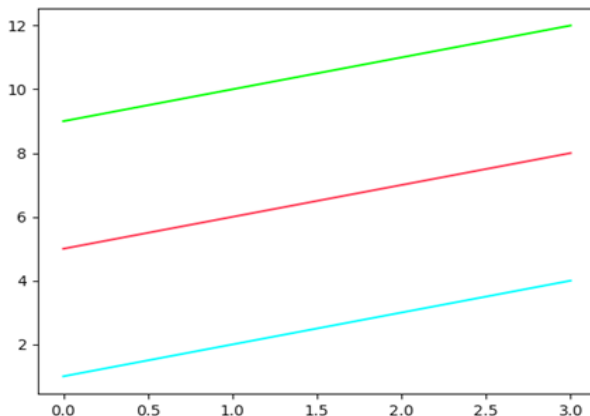


圖 5-6：程式 E5-2-4.py 之繪製結果

線條的樣式及寬度則分別以 linestyle 與 linewidth 表示。程式 E5-2-4 以不同的顏色、樣式和寬度組合繪製四條折線。

表 5-2-2：plot() 的線條樣式 (linestyle) 與寬度 (linewidth) 參數

參數名稱	可用值
linestyle	字串：-、--、-.、:
linewidth	通常在 1-20 之間。

E5-2-4.py：以四種顏色、樣式和寬度繪製四條折線

```

01 import matplotlib.pyplot as plt
02 plt.plot([13, 14, 15, 16], color='green', linestyle=':', linewidth=20)
03 plt.plot([9, 10, 11, 12], color='red', linestyle='-.', linewidth=10)
04 plt.plot([5, 6, 7, 8], color='blue', linestyle='--', linewidth=5)
05 plt.plot([1, 2, 3, 4])

```

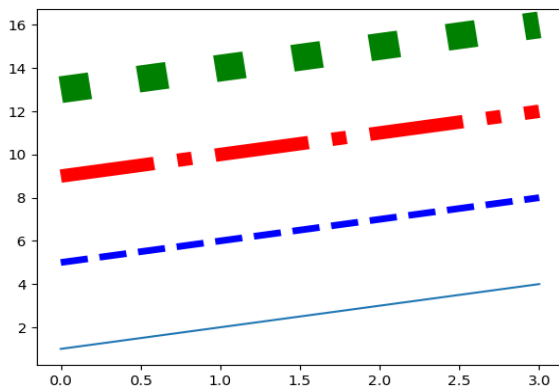


圖 5-7：程式 E5-2-4.py 之繪製結果

為了清楚標記資料點，我們可以指定 marker 和 markersize 兩個參數。

表 5-2-3：plot() 的資料點符號樣式 (marker) 及大小 (markersize) 參數

參數名稱	可用值
marker	字串如 ., ov^<>1234sp*hH+xDd _ 中的單一字元
markersize	通常在 1-10 之間。

E5-2-5.py：以兩種資料點符號繪製兩條折線

```
01 import matplotlib.pyplot as plt
02 plt.plot([5, 6, 7, 8], color='red', marker='v', markersize = 5)
03 plt.plot([1, 2, 3, 4], color='blue', marker='o', markersize = 10)
04 plt.show()
```

E5-2-6.py：多種資料點符號圖示

```
01 import matplotlib.pyplot as plt
02 mstr='.,ov^<>1234sp*hH+xDd|_\'
03 for i in range(len(mstr)):
04     plt.plot([i], marker=mstr[i], markersize=10)
05 plt.show()
```

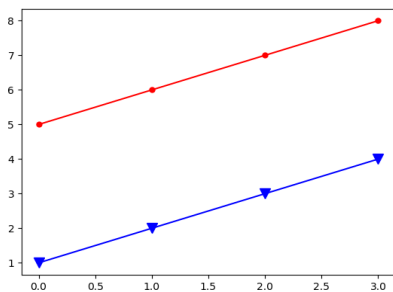


圖 5-8：E5-2-5.py 之繪製結果

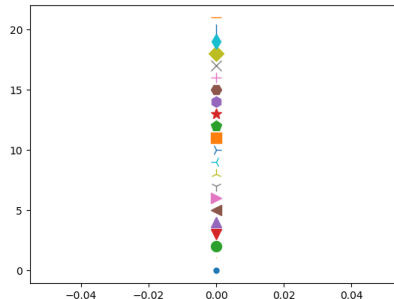


圖 5-9：E5-2-6.py 之繪製結果

程式 **E5-2-7** 示範了一種簡易的設定方法，可以將線條顏色、資料點符號以及線條樣式以一個字串傳入 `plot()`。程式第 02 行的字串 `'bo-'` 代表藍色 ('b')、圓型符號 ('o') 和實線 ('-')。程式第 03 行的字串 `'--g'` 代表虛線 ('--') 和綠色 ('g')。

E5-2-7.py：以簡易字串繪製折線

```
01 import matplotlib.pyplot as plt
02 plt.plot([2, 0, 1, 8], 'bo-')
03 plt.plot([4, 5, 6], [1, 4, 9], '--g')
04 plt.show()
```

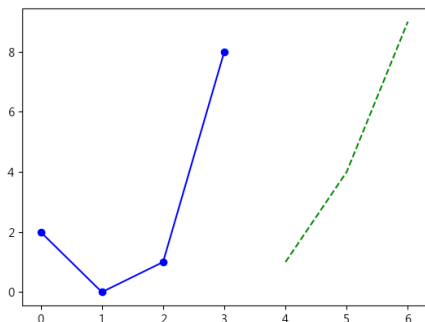



圖 5-10：E5-2-7.py 之繪製結果

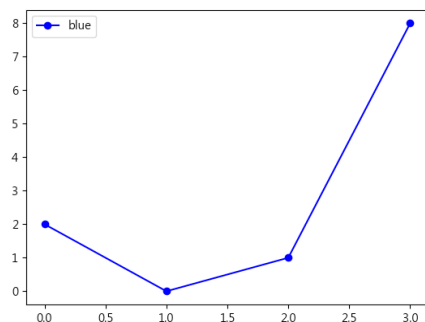


圖 5-11：E5-2-8.py 之繪製結果

最後介紹 `plot()` 函式的 `label` 參數，指定該參數可以設定線條的標籤。如程式 **E5-2-8** 的第 02 行，我們指定 `label` 為 'blue'，這使得該折線的標籤文字為 blue，如圖 5-11 左上角所示。注意，程式第 03 行需要呼叫 `legend()` 函式，否則標籤將不會顯示出來。

E5-2-8.py：為折線添加標籤文字

```
01 import matplotlib.pyplot as plt
02 plt.plot([2, 0, 1, 8], 'bo-', label='blue')
03 plt.legend()
04 plt.show()
```

5-3 圖表常用功能

設定標題文字

pyplot 模組有幾個函式可以用來設定圖表標題以及兩軸標題，如表 5-3-1 所示。這三個函式第一個參數代表標題文字，隨後附帶其它設定的參數，例如字型大小 (fontsize) 與文字位置 (loc) 等等。

表 5-3-1：設定標題文字的函式

函式名稱	說明	附帶參數
title()	圖表標題	fontsize 字型大小 loc 文字位置:left center right
xlabel()	X 軸標題	fontsize 字型大小
ylabel()	Y 軸標題	fontsize 字型大小

E5-3-1.py：設定圖表標題及軸標題文字

```
01 import matplotlib.pyplot as plt
02 plt.plot([2, 0, 1, 8], 'bo-')
03 plt.title('This is title', fontsize=24, loc='left')
04 plt.xlabel('x-axis caption', fontsize=15)
05 plt.ylabel('y-axis caption', fontsize=10)
06 plt.show()
```

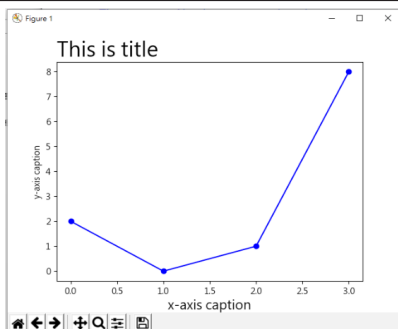


圖 5-12：程式 E5-3-1.py 之繪製結果

設定刻度範圍與刻度值

`plot()` 函式繪圖時會依資料數值的範圍自動設定刻度範圍和刻度值，如果想要自訂刻度的細節，可以使用表 5-3-2 的幾個函式來達成。`xlim()` 和 `ylim()` 函式用來設定刻度範圍，例如程式 E5-3-2 的第 03 行設定 `x` 軸的刻度範圍是 `[-5, 5]`，而第 04 行設定 `y` 軸的刻度範圍是 `[0, 20]`。`xticks()` 和 `yticks()` 函式用來設定刻度值，例如第 05 行設定 `x` 軸的刻度值只顯示了 `-5、0、1、2` 和 `5`，而第 06 行設定 `y` 軸的刻度值只顯示了 `0、2、4、6` 和 `8`。

表 5-3-2：設定刻度的函式

函式名稱	說明
<code>xlim()</code>	<code>x</code> 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>	<code>y</code> 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>xticks()</code>	<code>x</code> 軸刻度，傳入 <code>x</code> 軸要顯示的數值。
<code>yticks()</code>	<code>y</code> 軸刻度，傳入 <code>y</code> 軸要顯示的數值。

E5-3-2.py：設定圖表刻度範圍與刻度值

```

01 import matplotlib.pyplot as plt
02 plt.plot([2, 0, 1, 8], 'bo-')
03 plt.xlim(-5, 5)
04 plt.ylim(0, 20)
05 plt.xticks([-5, 0, 1, 2, 5])
06 plt.yticks(range(0, 10, 2))
07 plt.show()

```

一般來說，傳入 `xlim()` 和 `ylim()` 兩個數值依序是最小值和最大值。如果傳入的第一個值比第二個值大，該軸的範圍就會由大到小。例如我們若把程式 E5-3-2 的第 03 行寫成 `plt.xlim(5, -5)`，繪製結果就會如圖 5-13b 所示。

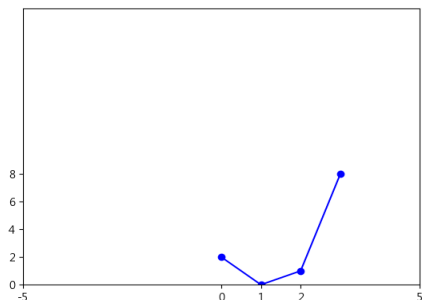


圖 5-13a：程式 5-3-2 之繪製結果

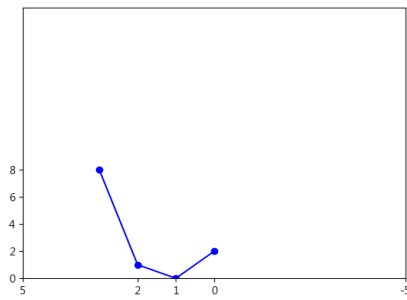


圖 5-13b：程式 5-3-2 之繪製結果 II

儲存圖檔

很多時候我們希望把繪製出來的圖表拿到其它地方使用，例如放入文書報告或簡報中，甚至於儲存下來以備未來使用。`pyplot` 模組中的 `savefig()` 函式這時就派上用場了。

程式 **E5-3-3** 示範了如何用程式繪製出 $y=x^0$ 、 $y=x^1$ 、 $y=x^2$ 和 $y=x^3$ 四個函式圖形並加以存檔。程式第 02 行有一個 `for` 句型，可以看出 `i` 的值會依序以 0、1、2 和 3 來執行以下第 03-08 行。第 03 行設定 `x` 是從 0 到 9 的整數，`y` 則是 `x` 中的每個值 `e` 作 `i` 次方的結果所構成的列表。第 06 行顯示圖表標題，第 07 行繪製圖表，而第 08 行就會將該圖表儲存成名為 `0.png`、`1.png`、`2.png` 和 `3.png` 的檔案。要特別注意的程式第 05 行呼叫了 `clf()` 函式。這個函式會清除目前圖表上的資料（`clf` 就是 **clear figure** 的縮寫）。如果省略了這一行，後續繪製的折線就會不斷疊加在前面折線上。（讀者可以自行嘗試看看。）

表 5-3-3：儲存圖檔的函式

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

E5-3-3.py：繪製四個函式並存檔

```
01 import matplotlib.pyplot as plt
02 for i in range(4):
03     x = range(10)
04     y = [e**i for e in x]
05     plt.clf()
06     plt.title('y = x**' + str(i))
07     plt.plot(x, y)
08     plt.savefig(str(i) + '.png')
```

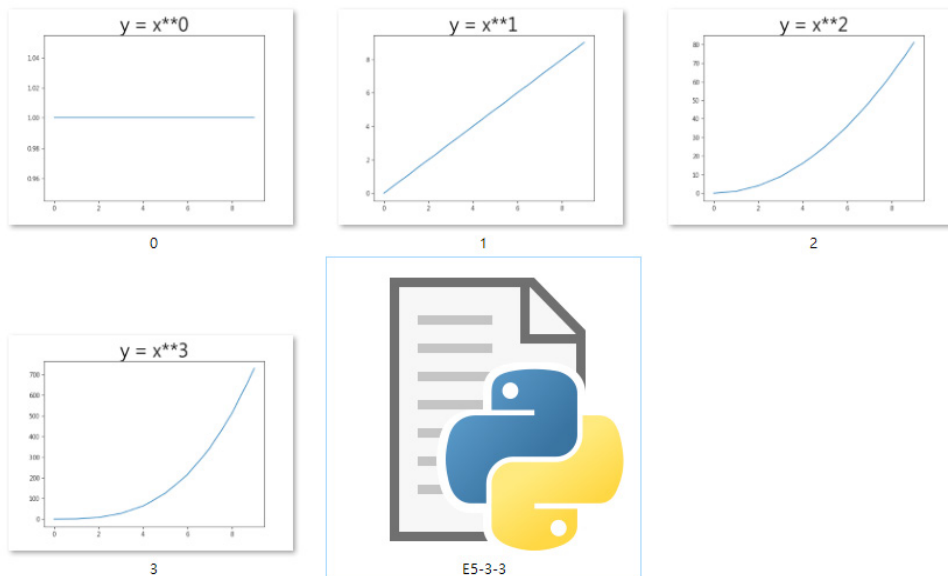


圖 5-14：程式 E5-3-3.py 之執行成果

執行程式 E5-3-3 後，我們會在程式檔的同一個目錄下看到四個圖檔，如圖 5-14 所示。學會存檔後，我們可以更方便地把圖表儲存下來，以作更廣泛的運用囉！

隨堂練習

5.3.1 請使用自行車竊盜資料檔，繪製如下的折線圖，用以觀察各行政區在不同時刻的竊案數變化。

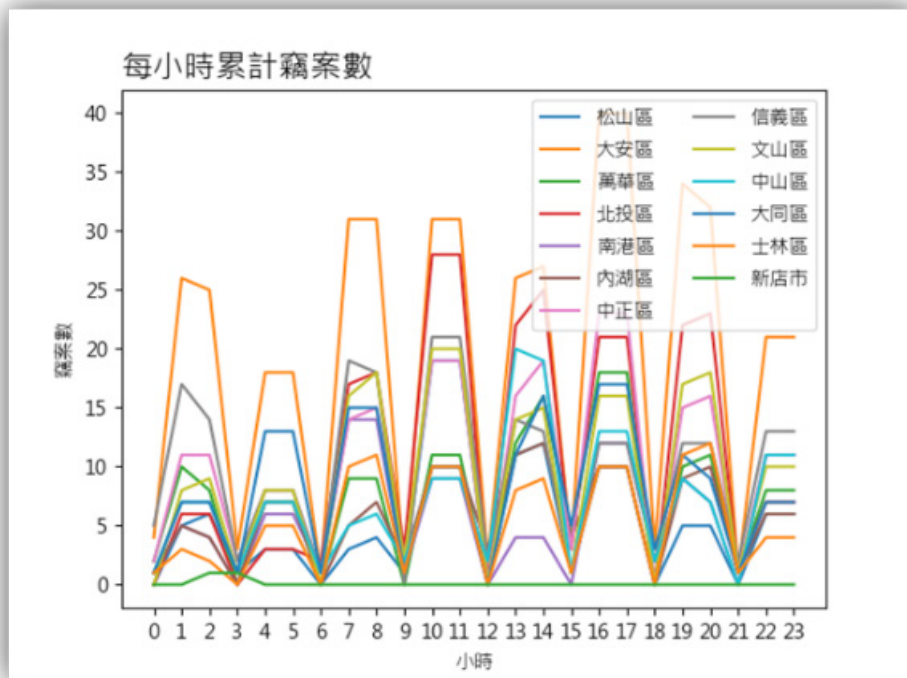


圖 5-15：自行車竊案之折線圖範例

5-4 散佈圖

折線圖常用來觀察有序數列的變化趨勢，而散佈圖可用於觀察兩個維度資料的關係。前一節我們學習了用 `plot()` 函式繪製折線圖，事實上，`plot()` 函式也可以繪製散佈圖，只要呼叫時指定資料點符號樣式但不要指定線條樣式就可以了。程式 **E5-4-1** 的第 04 行以綠色 (g) 倒三角形 (v) 繪製十個資料點 (0, 10)、(1, 11)、.....、(9, 19)。第 05 行以藍色 (b) 和星號 (*) 繪製十個資料點 (10, 0)、(11, 1)、.....、(19, 9)。

E5-4-1.py：繪製兩組資料形成的散佈圖

```
01 import matplotlib.pyplot as plt
02 x = range(10)
03 y = range(10, 20)
04 plt.plot(x, y, 'gv')
05 plt.plot(y, x, 'b*', markersize=20)
06 plt.show()
```

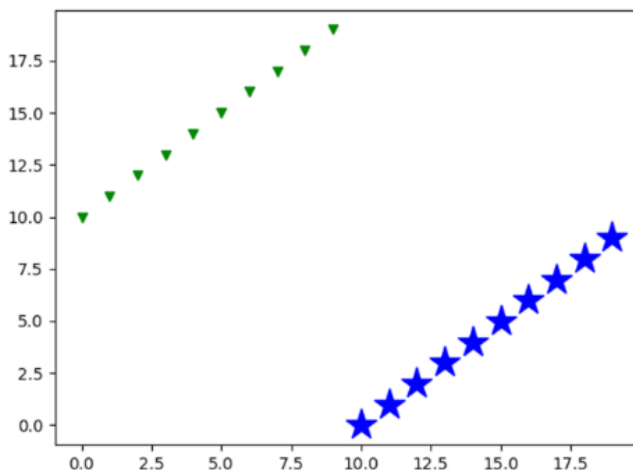


圖 5-16：程式 **E5-4-1.py** 繪製散佈圖之結果

散佈圖函式：scatter()

pyplot 模組中有一個專門用來繪製散佈圖的函式 `scatter()`。（散佈圖的英文即為 **scatter plot**。）呼叫 `scatter()` 時，基本需要傳入的是 **x** 和 **y** 軸的資料，可以是單值（如 `scatter(2, 4)`），也可以是一組值（如 `scatter([1, 2], [3, 4])`），**x** 和 **y** 的值個數需相同。表 5-4-1 列出此函式的常用參數，而程式 E5-4-2 示範了如何呼叫 `scatter()`。第 04 行設定資料點符號為菱形（`'d'`），標籤文字為 `'default'`。未指定的符號相關屬性（如符號大小與顏色等等）都會有預設值。程式第 05 行設定符號大小 200，顏色為紅色，邊框寬度 3，邊框顏色為黑色，標籤文字為 `'red'`。圖 5-16 呈現程式 E5-4-2 的執行結果。

表 5-4-1：散佈圖函式 `scatter()` 的常用參數

參數名稱	可用值
<code>x / y</code>	資料點的 <code>x / y</code> 值
<code>marker</code>	字串 <code>.,ov^<>1234sp*hH+xDd _</code> 中的單一字元
<code>s / c</code>	符號大小 / 符號顏色
<code>linewidths</code>	符號邊框寬度
<code>edgecolors</code>	符號邊框顏色
<code>label</code>	符號標籤

E5-4-2.py：設定散佈圖的符號屬性

```

01 import matplotlib.pyplot as plt
02 x = range(10)
03 y = range(10, 20)
04 plt.scatter(x, y, marker='d', label='default')
05 plt.scatter(y, x, s=200, c='red', linewidths=3,
06             edgecolors='black', label='red')
07 plt.legend()
08 plt.show()

```

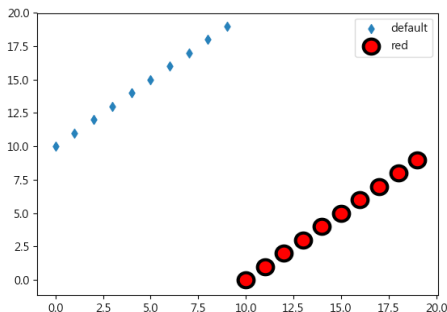



圖 5-17：程式 5-4-2 繪製之結果

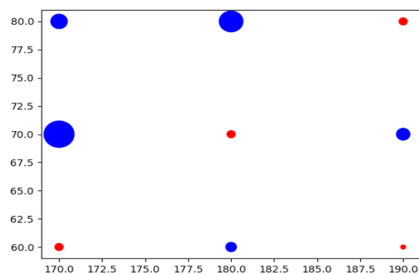


圖 5-18：程式 5-4-3 繪製之結果

在程式 E5-4-2 中我們使用了 `s` 和 `c` 參數來指定大小和顏色。作為一個專門的散佈圖繪製函式，`scatter()` 可以指定個別資料點符號的大小與顏色。只要我們傳入多個 `s` 和 `c` 的值，`scatter()` 便會以傳入的值來設定。例如 `scatter([1, 2], [3, 4], c=['blue', 'red'], s=[10, 20])` 所繪製的兩個資料分別為座標 (1, 3)、大小為 10 的藍色符號以及座標 (2, 4)、大小為 20 的紅色符號。程式 E5-4-3 繪製九組 (身高，體重) 的人數的散佈圖。第 03 行的 `h_w_c` 的每一個元素代表身高 (如 170)、體重 (如 80) 和人數 (如 12)。程式第 08 行依人數決定符號大小，第 09 行設定超過 3 人的符號為藍色。

E5-4-3.py：繪製 (身高，體重) 的人數散佈圖

```
01 import matplotlib.pyplot as plt
02 h_w_c = [[170, 80, 12], [180, 80, 25], [190, 80, 3],
03          [170, 70, 40], [180, 70, 3], [190, 70, 8],
04          [170, 60, 3], [180, 60, 5], [190, 60, 1]]
05 x = [e[0] for e in h_w_c]      # 以身高作為 x 軸
06 y = [e[1] for e in h_w_c]      # 以體重作為 y 軸
07 s = [e[2]*20 for e in h_w_c]   # 以人數作為符號大小
08 c = ['blue' if e[2]>3 else 'red' for e in h_w_c]
09 plt.scatter(x, y, s, c)
10 plt.show()
```

隨堂練習

5.4.1 請使用自行車竊盜資料檔，繪製如下的散佈圖，用以觀察竊案數在各月份及各時刻的分佈情形。

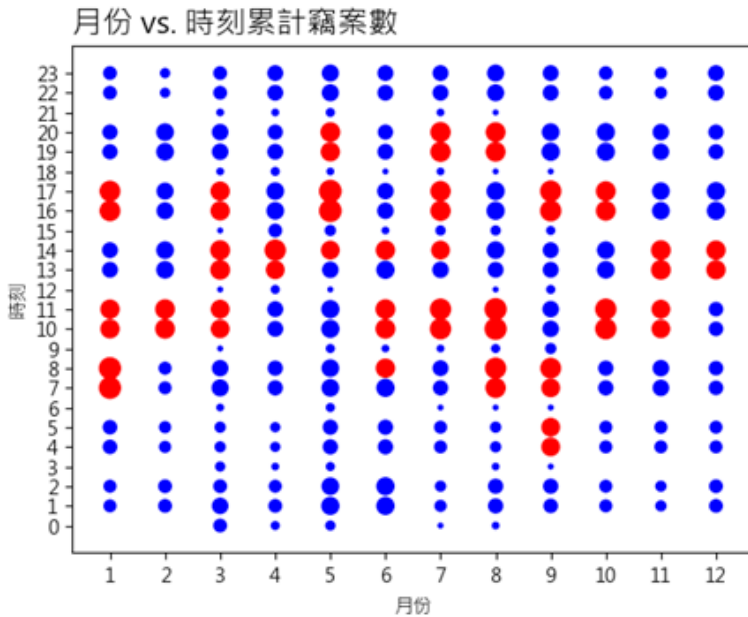


圖 5-19：自行車竊案之散佈圖範例

5-5 長條圖

本章最後一節要介紹的是長條圖，長條圖常用來比較多組資料的數值大小。

長條圖函式：bar()

呼叫 `bar()` 函式時，基本傳入的是 `x` 和 `y` 軸（`y` 在長條圖中特別稱為 `height`）的資料。此外，也可以指定長條的寬度與顏色。

表 5-5-1：長條圖函式 `bar()` 的常用參數

參數名稱	可用值與範例
<code>x/height</code>	<code>x / y</code> 軸座標值
<code>width</code>	長條寬度（預設 <code>0.8</code> ）
<code>color</code>	長條顏色
<code>label</code>	標籤

程式 E5-5-1 呼叫了 `bar()` 函式兩次。第 02 行在 `x` 軸位置 0、2 和 4 處分別繪製高度為 10、15 和 8 的長條，標籤文字為 `blue`。第 03 行在 `x` 軸位置 1、3 和 5 處分別繪製高度為 7、12 和 11 的長條，寬度為 0.2，顏色為粉紅色（`pink`），標籤文字為 `orange`。圖 5-20 為此程式繪製之結果。

E5-5-1.py：繪製兩組資料形成的長條圖

```
01 import matplotlib.pyplot as plt
02 plt.bar([0, 2, 4], [10, 15, 8], label='blue')
03 plt.bar([1, 3, 5], [7, 12, 11], width=0.2,
04         color='pink', label='orange')
05 plt.legend()
06 plt.show()
```

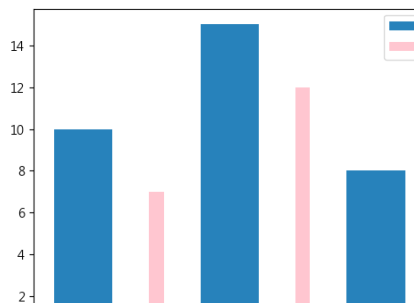


圖 5-20：程式 5-5-1 繪製之結果

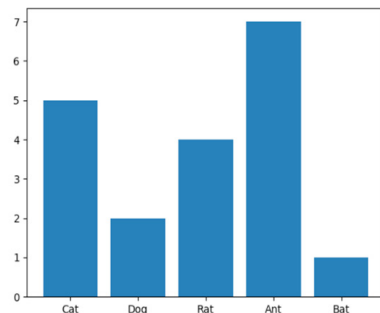


圖 5-21：程式 5-5-2 繪製之結果

在本章 5-3 節我們曾介紹 `xticks()` 可以用來指定 `x` 軸要顯示的刻度，例如 `xticks([1, 3, 5])` 代表 `x` 軸只會顯示刻度 1、3 和 5。在繪製長條圖時，`x` 軸經常需要顯示每個長條代表的意義（文字）而不是刻度的數值。呼叫 `xticks()` 時可以傳入第二組資料來代表在指定刻度顯示的文字，例如 `xticks([1, 3, 5], ['a', 'b', 'c'])` 會在刻度 1、3 和 5 處分別顯示文字 `a`、`b` 和 `c`。

程式 E5-5-2 中第 02 行為欲繪製成長條圖的資料，共有五組資料。第 04 行在 `x` 軸位置 0、1、2、3 和 4 處繪製高度為 5、2、4、7 和 1 的長條。第 05 行在相同的 `x` 軸位置顯示了 `Cat`、`Dog`、`Rat`、`Ant` 和 `Bat` 的文字。圖 5-21 為本程式繪製的結果。

E5-5-2.py：設定長條圖 `x` 軸的文字

```
01 import matplotlib.pyplot as plt
02 data = [['Cat', 5], ['Dog', 2], ['Rat', 4], ['Ant', 7], ['Bat', 1]]
03 plt.bar(range(len(data)), [e[1] for e in data])
04 plt.xticks(range(len(data)), [e[0] for e in data])
05 plt.show()
```

隨堂練習

5.5.1 請使用自行車竊盜資料檔，繪製如下的長條圖，用以比較各行政區的竊案數。

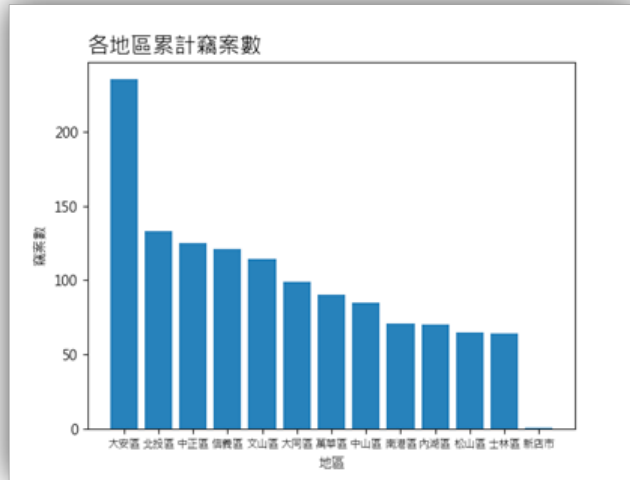


圖 5-22：自行車竊案之長條圖範例