第一章

為什麼要用 Python 寫 Excel？

Excel 使用者經常在遇到瓶頸時，不斷質疑試算表工具。最典型的例子是，當活頁簿包含太多的資料和公式，Excel 的運算速度會很慢，最糟的情況是直接閃退。在事情變糟之前質疑 Excel 設置，確實有其道理：假如你正在處理一份至關重要的報表，任何錯誤都可能導致財務上或名譽上的損失，或者你每天花費數小時手動更新 Excel 試算表，你應該學會運用程式語言，將流程自動化。自動化可以將人為錯誤的風險降至最低，讓你可以把時間投注在更有生產力的任務，不再是單純地將資料複製、貼上表單。

本章將告訴你，為何 Python 是在 Excel 上工作的不二選擇，以及 Python 相對於 Excel 內建的自動化語言 VBA 的優勢。先認識 Excel 如何作為一種程式語言，並瞭解其特別之處後，我會點出 Python 相對於 VBA 的強大功能。話不多說，讓我們先來認識本書兩位主角吧！

在電腦科技領域，Excel 和 Python 已經存在多年：Microsoft 公司於 1985 年發表 Excel，令許多人驚豔的是，當時 Excel 只能在 Apple 公司的麥金塔電腦上執行。直到 1987 年，Windows 系統終於取得 Excel 版本：Excel 2.0。Microsoft 並非電子試算表軟體的先行者，VisiCorp 公司早在 1979 年就發布了 VisiCalc 軟體，而後 Lotus Software 公司也發行了 Lotus 1-2-3。Microsoft 公司並沒有憑藉 Excel 取得市場領導地位：1982 年，Microsoft 公司先推出了 Multiplan，這是一款可用於 MS-DOS 系統與其他作業系統的電子試算表軟體，不過它不適用於 Windows 系統。

碁峰 www.gotop.com.tw

Excel 問世的六年後，Python 在 1991 年誕生了。Excel 進入市場後很快地得到大眾接納，Python 花了比較久時間才在網頁開發或系統管理等特定領域普及。2005 年，當 *Numpy* 這個基於陣列的運算與線性代數的套件出現後，Python 很快地成為了科學計算的優異選擇。NumPy 結合了兩個先驅節點套件，將科學計算的所有開發流程整合成單一專案。今日，NumPy 更是由無數的科學套件而組成，其中包括 2008 年問世的 *pandas*，促成了 2010 年後 Python 在資料科學與金融運算領域的廣泛採用。Pandas 套件的出現，讓 Python 與 R 成為資料科學領域中最被使用的程式設計語言，廣泛用於如資料分析、統計與機器學習等工作。

Python 和 Excel 出現時間很早，並不是兩者唯一的共通點：Excel 和 Python 都是一種程式語言。儘管 Python 是一種程式語言並不會讓你驚訝，但將 Excel 視為一種程式語言，可能需要一番解釋，且聽我娓娓道來。

Excel 是一種程式語言

本節介紹 Excel 如何作為一種程式語言，幫助你瞭解試算表相關的議題為何經常上新聞。我們將檢視在軟體開發社群中幾個最佳實踐，避免犯下使用 Excel 的常見錯誤。最後介紹 Power Query 和 Power Pivot，這兩個現代 Excel 工具的多數功能都可以 pandas 替代。

假如對你而言 Excel 的用途不只是列購物清單，你一定使用過類似 `=SUM(A1:A4)` 來加總儲存格範圍。如果你要花幾秒想想這個函數如何運作，你會發現儲存格的值經常仰賴於一個或幾個儲存格，而這些儲存格也可能取決於其他儲存格的值。這類的嵌套函式呼叫和其他程式語言的運作方式相差無幾，差別只在於你將程式碼寫在儲存格裡而不是文字編輯器。假如這一點還無法說服你：在 2020 年底，Microsoft 宣布發布 *lambda 函式*，讓使用者能以 Excel 的自有公式語言編寫可重複使用的函式，也就是說，使用者不再需要仰賴除了 Excel 以外的語言（如 VBA）。根據 Excel 產品長 Brian Jones 所言，這正是讓 Excel 成為一個「真正的」程式語言的最後一塊拼圖[1]。這意味著，Excel 使用者應該被稱呼為「Excel 程式設計師」！

Excel 程式設計師有一個特別之處：他們多數人是商務使用者或領域專家，但不曾接受過正統的電腦科學教育。他們可能是交易員、會計師或工程師。他們所使用的試算表工具被設計來解決商業問題，經常忽略軟體開發的最佳實踐。有鑑於此，這些試算表工具

1 你可以在 Excel Blog 閱讀 lambda 函式相關聲明（*https://oreil.ly/4-0y2*）。

碁峰 www.gotop.com.tw

經常在同一份表單中混雜了輸入值、計算式和輸出值，經常需要拐彎抹角才能讓它們好好運作，甚至可以隨意對表單進行修改，沒有任何防呆機制。換句話說，這些試算表工具缺少了牢固可靠的應用基礎設施，說明文件缺三漏四，系統缺乏測試。有時候，這些問題可能帶來毀滅性災難：假如你在下單前忘記重新計算交易報表，你可能會購買或售出錯誤數量的股票，導致鉅額虧損。假如你不只是用自己的錢進行交易，那麼極有可能上新聞，例如下節所述。

新聞中的 Excel

Excel 是新聞裡的常客，在本書創作之時，就有兩個事件上了頭條。第一則新聞來自人類基因組組織命名委員會（HUGO Gene Nomenclature Committee），該委員會將幾個人類基因組重新命名，好讓 Excel 不再將基因名稱讀取為日期。舉例來說，為了避免 `MARCH1` 被判別為 `1-Mar`，該基因組被重新命名為 `MARCHF1`[2]。第二則新聞則是，Excel 被控耽誤了英國 16,000 例 COVID-19 測試報告。該事件是由於這些測試結果以舊的 Excel 檔案格式（*.xls*）寫成，此格式最多只能存放約 65,000 筆資料。這意味著，大一點的資料集就會受到這個資料筆數限制的影響[3]。儘管這兩則新聞皆證明了 Excel 對於現代社會舉足輕重的地位，但遠遠無法比得上「倫敦鯨事件」受到矚目的程度。

倫敦鯨事件發生於 2012 年，「倫敦鯨」是摩根大通銀行一位交易員的綽號，他的交易失誤導致該行發布超過 20 億美元交易損失。事件爆發源頭是一份以 Excel 設計的風險估值模型，這個模型大大低估了投資組合的真實虧損風險。事後的檢討報告中[4]指出「以一連串 Excel 試算表運作的模型，完全仰賴人工作業，手動將一份試算表的資料複製貼上到另一份」。除了工作流程問題，該試算表還存在一個邏輯上的謬誤：在某個計算式中，他們除以總和而不是平均值。

假如你對這類故事感興趣，歡迎參閱由 European Spreadsheet Risks Interest Group（EuSpRIG）經營維護的 Horror Stories 網站（*https://oreil.ly/WLO-I*）。

為了避免你的公司陷於類似上述新聞事件的窘境，我們來看看下列最佳實踐，讓你在使用 Excel 時更加安全。

---

2 James Vincent, "Scientists rename human genes to stop Microsoft Excel from misreading them as dates," TheVerge, August 6, 2020, *https://oreil.ly/0qo-n*.

3 Leo Kelion, "Excel: Why using Microsoft's tool caused COVID-19 results to be lost," BBC News, October 5, 2020, *https://oreil.ly/vvB6o*.

4 維基百科中有關於此事件的參考文獻（*https://oreil.ly/0uUj9*）。

碁峰 www.gotop.com.tw

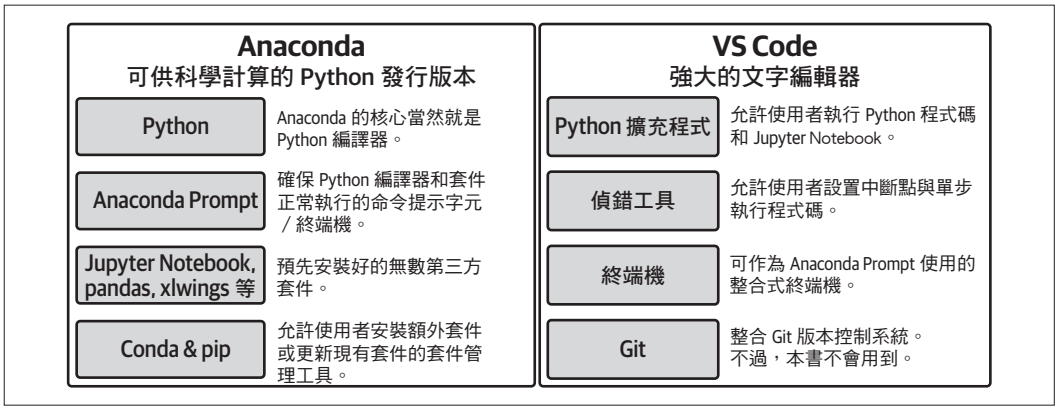

圖 2-1　開發環境

Anaconda（Python 發行版）

Anaconda 可以說是資料科學界最受歡迎的 Python 發行版本，預先安裝好數百種第三方套件：它擁有 Jupyter Notebook 以及本書密集使用的其他套件如 pandas、OpenPyXL 和 xlwings。Anaconda 個人版可以免費下載，所有軟體套件都可兼容。Anaconda 會安裝為一份資料夾，也可以輕鬆解除安裝。在成功安裝之後，我們將在 Anaconda Prompt 上學習幾個常用命令，並執行互動式視窗。接著會介紹 Conda 和 pip 這些軟體套件管理工具，最後以 Conda 虛擬環境收尾。就從下載與安裝 Anaconda 開始吧！

安裝

前往 Anaconda 官方網站（*https://oreil.ly/QV7Na*），下載最新版 Anaconda 安裝程式（個人版）。如使用 Python 3.x 版本，請選擇 64-bit graphical installer[1]。完成下載後，請按兩下安裝程式，啟動安裝程序，同意勾選所有預設設置。關於更詳細的安裝指示，請參考官方說明文件（*https://oreil.ly/r01wn*）。

1　32-bit 版本只存在於 Windows 作業系統，目前已越來越少見。找出你的 Windows 版本很簡單，請在檔案總管中前往 C 槽。如果你同時看到 *Program Files* 和 *Program Files (x86)* 資料夾，這表示你使用的是 Windows 64-bit 版本。如果你只看到 *Program Files* 資料夾，則你使用的是 Windows 32-bit 系統。

碁峰 www.gotop.com.tw

pandas 就是這麼一種套件，我將在第 5 章好好介紹它。pandas 已預先安裝於 Anaconda Python 發行版中，使用者無須另行手動安裝。

Conda vs. pip

在 Anaconda 中，你應該透過 Conda 安裝任何所需套件，並僅使用 pip 來安裝 Conda 無法找到的其餘套件。否則，Conda 可能會覆寫那些以 pip 安裝的檔案。

表 2-2 整理了一些最常使用到的指令。將這些指令輸入 Anaconda Prompt 中，就能安裝、更新或解除安裝第三方套件。

表 2-2　Conda 和 pip 指令

動作	Conda	pip
列出所有已安裝的套件	`conda list`	`pip freeze`
安裝最新套件版本	`conda install` *`package`*	`pip install` *`package`*
安裝特定套件版本	`conda install` *`package=1.0.0`*	`pip install` *`package==1.0.0`*
更新套件	`conda update` *`package`*	`pip install --upgrade` *`package`*
解除安裝套件	`conda remove` *`package`*	`pip uninstall` *`package`*

舉例來說，如果你想查看 Anaconda 發行版中目前有哪些可用套件，請輸入：

```
(base)> conda list
```

當本書需要使用到 Anaconda 安裝程序未包含的套件時，我會特別註明並示範如何安裝。不過，不妨現在就將一些套件安裝好，以防不時之需。我們先來安裝 plotly 和 xlutils 吧，這兩個套件可在 Conda 上找到：

```
(base)> conda install plotly xlutils
```

執行這則指令後，Conda 將會示範它將進行何種動作，並要求你進行確認，這時請輸入 **y** 然後按下 Enter 鍵。接下來，請透過 pip 安裝 pyxlsb 和 pytrends，這兩個套件在 Conda 上沒有：

```
(base)> pip install pyxlsb pytrends
```

不同於 Conda，pip 會在你按下 Enter 鍵的時候立刻安裝，無須額外確認。

碁峰 www.gotop.com.tw

第四章

NumPy 基礎

我們在第 1 章提過，NumPy 是 Python 語言中支援科學計算的核心套件，針對陣列運算和線性代數提供一系列功能。NumPy 是 pandas 函式庫的主心骨幹，本章將首先介紹 NumPy 的基礎知識：在瞭解什麼是 NumPy 陣列後，我們會學習向量化（vectorization）和廣播（broadcasting）機制，這兩個關鍵概念將幫助你撰寫簡潔的數學運算式，而你也將在 pandas 再次遇上它們。接著，瞭解為什麼 NumPy 提供了名為「通用函式」的特殊函數，最後我們會實際練習如何取得與設定陣列的值，並且解釋 NumPy 陣列的檢視表和副本模式差異。儘管我們很少在本書內容直接使用 NumPy，掌握基本原理有助於我們在下一章學習 pandas 函式庫。

開始使用 NumPy

在本節內容中，我們會學習一維和二維 NumPy 陣列，以及「向量化」（vectorization）、「廣播」（broadcasting）和「通用函式」（universal function）等技術名詞背後的意涵。

NumPy 陣列

想以巢狀串列執行陣列運算，如同上一章的例子，你需要編寫一些迴圈。舉例來說，想對一個巢狀串列的所有要素新增一個數值，你可以使用下列巢狀串列運算式：

```
In [1]: matrix = [[1, 2, 3],
                  [4, 5, 6],
                  [7, 8, 9]]

In [2]: [[i + 1 for i in row] for row in matrix]

Out[2]: [[2, 3, 4], [5, 6, 7], [8, 9, 10]]
```


碁峰 www.gotop.com.tw

然而這樣的程式碼可讀性並不高，況且，如果是大型陣列的情況，循環查看每個要素會大幅降低運算速度。根據用例和陣列大小，以 NumPy 陣列取代 Python 串列可以讓運算效率大幅提高。NumPy 以 C 或 Fortan 寫成的程式碼締造高效運算——這是比 Python 運算速度更快的編譯語言。NumPy 陣列是一種「同質性資料」（homogenous data）的 N 維陣列。同質性指陣列中的所有要素都必須屬於同一種資料型態。大多數時候，你會處理以浮點數組成的一維或二維陣列，如圖 4-1 所示的結構。

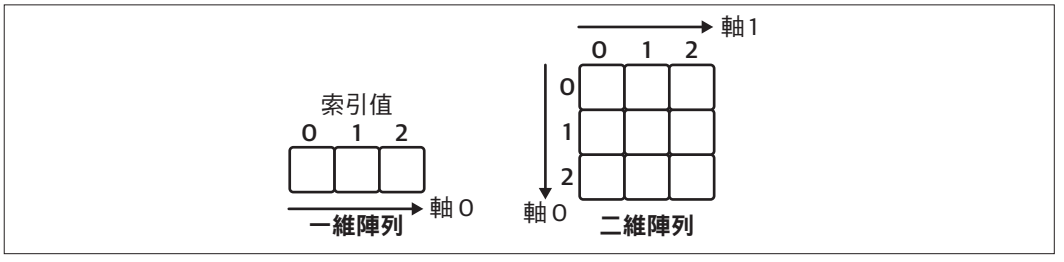

圖 4-1　一維陣列和二維陣列

我們分別建立一維陣列和二維陣列：

```
In [3]: # First, let's import NumPy
        import numpy as np

In [4]: # Constructing an array with a simple list results in a 1d array
        array1 = np.array([10, 100, 1000.])

In [5]: # Constructing an array with a nested list results in a 2d array
        array2 = np.array([[1., 2., 3.],
                           [4., 5., 6.]])
```

陣列維度

注意一維陣列和二維陣列的差異：一維陣列只有一個軸，因此不存在外顯的列（row）或行（column）。雖然這和 VBA 陣列的情況相同，但如果你以前使用的是 MATLAB，大概需要重新適應，因為 MATLAB 語言的一維陣列永遠都會有行或列的方向性。

即使 `array1` 除了最後一個要素（它是浮點數）以外都由整數組成，NumPy 陣列的同質性特點會強制指定該陣列的資料型態為 `float64`，此資料型態可接納所有要素。想瞭解陣列資料型態的更多資訊，請查找 `dtype` 屬性：

碁峰 www.gotop.com.tw

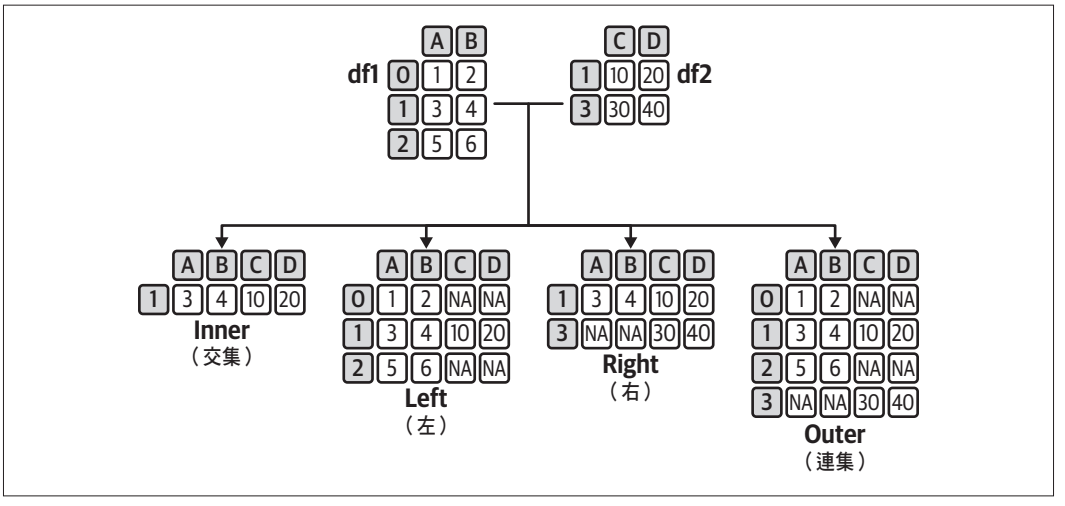

圖 5-3　Join 類型

在 `join` 的情況下，pandas 使用兩個 DataFrame 的 index 來對齊資料列。*inner join* 傳回的 DataFrame 是 index 重疊的資料列。*left join* 會取用左側 DataFrame `df1` 的所有 row，並按 index 配對右側 DataFrame `df2`。當 `df2` 的 row 不一致時，pandas 會填入 `NaN`。left join 對應的是 Excel 的 `VLOOKUP` 函數。*right join* 取用右側表格 `df2` 的所有 row，並按 index 配對左側 DataFrame `df1`。最後，*outer join*，這是 *full outer join*，這個函數會取用兩個 DataFrame 中 index 的集合，並盡可能配對值。表 5-5 是圖 5-3 的文字形式。

表 5-5　Join 類型

類型	描述
`inner`	同時存在於兩個 DataFrame 的 row index
`left`	左側 DataFrame 的所有 row，配對右側 DataFrame 的 row
`right`	右側 DataFrame 的所有 row，配對左側 DataFrame 的 row
`outer`	兩個 DataFrame 的 row index 交集

我們用實際例子來看看這些函數如何運作，以程式碼呈現圖 5-3 的範例：

```
In [74]: df1 = pd.DataFrame(data=[[1, 2], [3, 4], [5, 6]],
                            columns=["A", "B"])
         df1

Out[74]:    A  B
         0  1  2
         1  3  4
```


碁峰 www.gotop.com.tw

```
Q4        64933.848496   7600.277035  55001.831706  86248.512650

In [96]: data.plot()  # Shortcut for data.plot.line()

Out[96]: <AxesSubplot:xlabel='Quarters'>
```

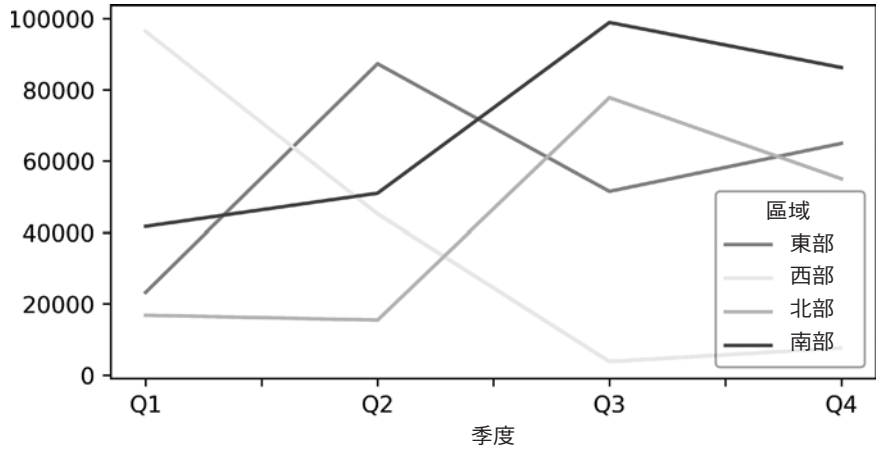

圖 5-4　Matplotlib 圖表

在這個例子中，我使用了 NumPy 陣列來建構一個 pandas DataFrame。使用 NumPy 陣列的好處是可以借助上一章提到的 NumPy 建構子；此處，我們使用了 NumPy 來產生一個基於偽隨機數的 pandas DataFrame。因此，當你以本例進行實際演練時，你會得到不同的值。

神奇指令

讓 Matplotlib 可執行於 Juptyer Notebook 的 `%matplotlib inline` 指令是一個「神奇指令」（magic command）。神奇指令是一套簡單指令，可以讓 Jupyter Notebook 的儲存格以特定方式執行動作，或是讓一些麻煩的任務變得更容易處理，讓人感覺像是施展了魔法一樣。這些神奇指令看起來和寫在儲存格裡的 Python 程式碼一樣，不過它們以 `%%` 或 `%` 開頭。影響整個儲存格的指令以 `%%` 開頭，而以 `%` 開頭的指令只會影響儲存格裡的單行程式碼。

我們會在下一章認識更多神奇指令，如果你想搶先看看可用的神奇指令，不妨執行 `%lsmagic`，如果想取得更詳細的指令描述，請執行 `%magic`。

碁峰 www.gotop.com.tw

即便使用了 `%matplotlib notebook` 這個神奇指令，你大概會發現 Matplotlib 的設計初衷是靜態圖表，而不是為了在網頁上呈現互動式體驗。為此，我們接下來要認識 Plotly，這是一個專為網頁設計的函式庫。

Plotly

Plotly 是一個基於 JavaScript 的函式庫，自 4.8.0 以後的版本可作為 pandas 繪圖後端使用，提供優秀的互動性：使用者可以輕鬆縮放、在 legend（圖示說明）上選取或取消選取特定類別，還能取得關於資料點的提示說明。Plotly 沒有內建於 Anaconda 發行版中，如果你尚未安裝 Plotly，可以執行以下指令：

```
(base)> conda install plotly
```

執行以下儲存格，將 Plotly 設定為整個 Notebook 的繪圖後端，如果你重新執行一次該儲存格，它也會被輸出為一個 Plotly 圖表。在 Plotly 的情況下，你不需要執行神奇指令，只需要將它設定為後端，就能輕鬆繪製出像圖 5-5 和圖 5-6 的圖表：

```
In [97]: # Set the plotting backend to Plotly
         pd.options.plotting.backend = "plotly"

In [98]: data.plot()

In [99]: # Display the same data as bar plot
         data.plot.bar(barmode="group")
```

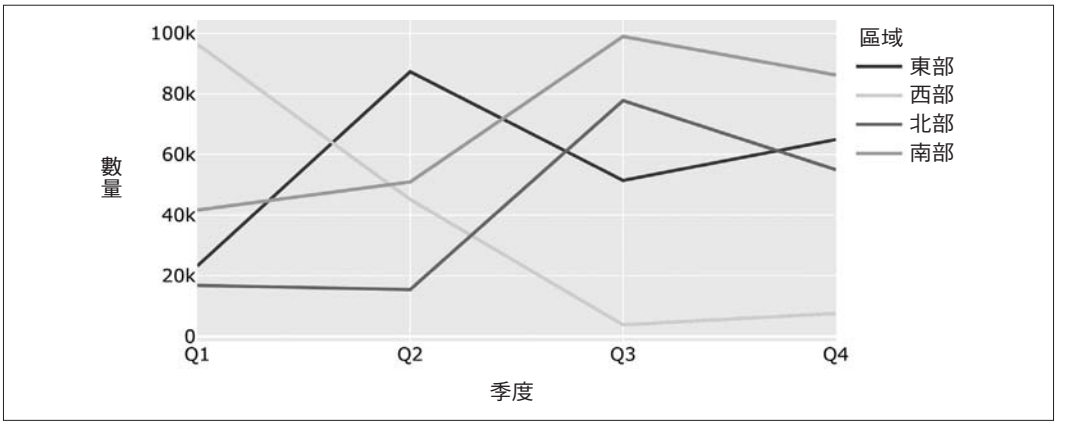

圖 5-5　Plotly 折線圖

碁峰 www.gotop.com.tw

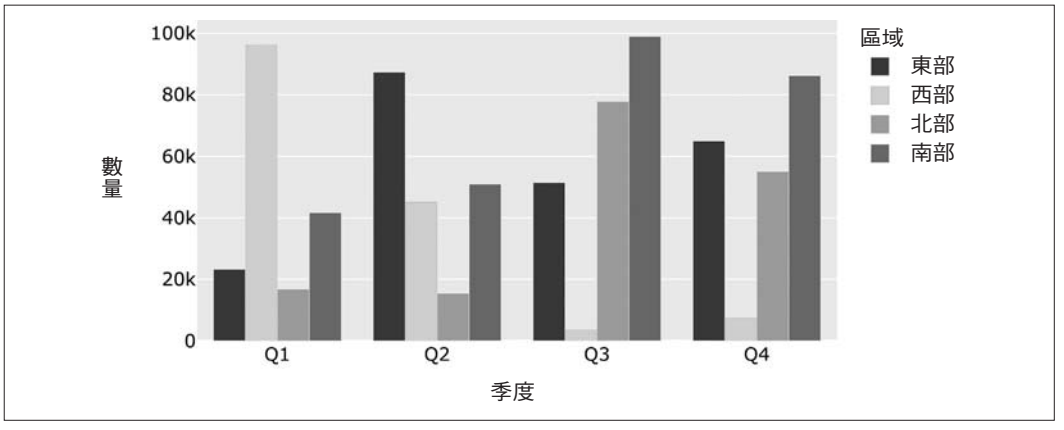

圖 5-6　Plotly 柱狀圖

繪圖後端的差異

如果將 Plotly 設定為繪圖後端，你需要參考 Plotly 說明文件，檢查一下 plot 方法接受哪些引數。舉例來說，你可以到 Plotly 柱狀圖說明文件閱讀 `barmode=group` 引數的相關說明（*https://oreil.ly/Ekurd*）。

pandas 與繪圖函式庫提供了多樣化的圖表類型和格式化選項，幫助使用者以心儀的方式呈現資料。你也可以將多個圖表排列成一系列子圖表。表 5-6 列示了可用的圖表類型以供參考。

表 5-6　pandas 圖表類型

類型	描述
`line`	折線圖，執行 `df.plot` 的預設圖表
`bar`	柱狀圖
`barh`	橫向柱狀圖
`hist`	直方圖
`box`	箱形圖
`kde`	KDE 圖（核密度估計），也可以透過 `density` 取用
`area`	面積圖
`scatter`	散布圖
`hexbin`	六角形箱形圖
`pie`	圓餅圖

碁峰 www.gotop.com.tw

除此之外，pandas 提供了一些更廣泛的繪圖工具和技法，這些工具由數個獨立元件組成。關於更多內容，請參考 pandas 視覺化處理說明文件（*https://oreil.ly/FxYg9*）。

其他繪圖函式庫

Python 在科學運算領域的應用相當活躍廣泛，除了 Matplotlib 和 Plotly 以外，還有許多優秀選項可供不同情境使用：

Seaborn

Seaborn（*https://oreil.ly/a3U1t*）是基於 Matplotlib 的繪圖套件。它對預設樣式進行改進，並額外增加了圖表如 heatmap，有助於簡化你的工作：編寫少少幾行程式碼就能建立進階統計圖表。

Bokeh

Bokeh（*https://docs.bokeh.org*）在技術和功能方面和 Plotly 相似：這是基於 JavaScript 的繪圖套件，因此能夠在 Jupyter Notebook 中創造優秀的互動式圖表。Bokeh 隨附於 Anaconda 發行版中。

Altair

Altair（*https://oreil.ly/t06t7*）是基於 Vega 專案的統計視覺化函式庫（*https://oreil.ly/RN6A7*）。Altair 也是基於 JavaScript 的套件，支援如縮放圖表等互動式功能。

HoloViews

HoloViews（*https://holoviews.org*）是另一個基於 JavaScript 的套件，旨在簡化資料分析和視覺化處理。只需簡單幾行程式碼，使用者就能創造出複雜的統計圖表。

我們會在下一章建立更多圖表，對時間序列（time series）進行分析。在此之前，先來學習如何用 pandas 匯入和匯出我們的資料吧！

碁峰 www.gotop.com.tw