1.3.6 相關關係與因果關係

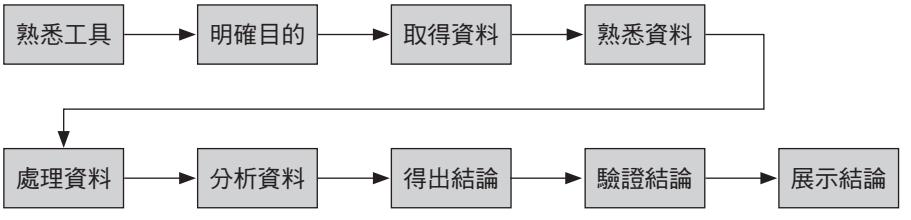
相關關係不等於因果關係，相關關係只能說明兩件事情有關聯，而因果關係是說明一件事情導致了另一件事情的發生，不要把這兩種關係混淆使用。

例如，啤酒和尿布是具有相關關係的，但是不具有因果關係；而流感疾病和關鍵字搜尋量上漲是具有因果關係的。

在實際業務中會遇到很多相關關係，但是具有相關關係的兩者不一定有因果關係，一定要注意區分。

1.4 資料分析的常規流程

我們再來回顧一下資料分析的概念，資料分析是借助合適的工具去幫助公司發現資料背後隱藏的資訊，對這些隱藏的資訊進行挖掘，從而促進業務發展。基於此，可以將資料分析分為以下幾個步驟：



1.4.1 熟悉工具

資料分析是利用合適的工具和合適的理論挖掘隱藏在資料背後的資訊，因此資料分析的第一步就是要熟悉工具。工欲善其事，必先利其器，只有熟練使用工具，才能更有效率地處理資料、分析資料。

1.4.2 明確目的

做任何事情都要目的明確，資料分析也一樣，首先要明確資料分析的目的，即希望透過資料分析得出什麼。例如，希望透過資料分析發現流失的使用者都有哪些特徵，希望透過資料分析找到銷量上漲的原因。

1.4.3 取得資料

目的明確後我們就要取得資料，在取得資料之前還需要確定以下幾點：

- 需要什麼指標？
- 需要什麼時段的資料？
- 這些資料都存在哪個資料庫或哪個表中？
- 如何提取，是自己寫 SQL 還是可以直接從 ERP 系統中下載？

1.4.4 熟悉資料

拿到資料以後，我們要先熟悉資料。熟悉資料就是看一下有多少資料、這些資料是類別型還是數值型的、每個指標大概有哪些值，以及這些資料能不能滿足我們的需求，如果不夠，那麼還需要哪些資料。

取得資料和熟悉資料是一個雙向的過程，當你熟悉完資料以後發現當前資料維度不夠，那就需要重新取得；當你取得到新的資料以後，需要再去熟悉，所以取得資料和熟悉資料會貫穿在整個資料分析過程中。

1.4.5 處理資料

取得的資料是原始資料，這些資料中一般會有一些特殊資料，我們需要對這些資料進行前置處理，常見的特殊資料主要有以下幾種：

- 異常資料。
- 重複資料。
- 缺失資料。
- 測試資料。

重複資料、測試資料一般都是直接刪除。

缺失資料的部分，如果缺失比例高於 30%，那麼我們會選擇放棄這個指標，即做刪除處理。而對於缺失比例低於 30% 的指標，我們一般進行填充處理，即使用 0、均值或眾數等進行填充。

實踐篇

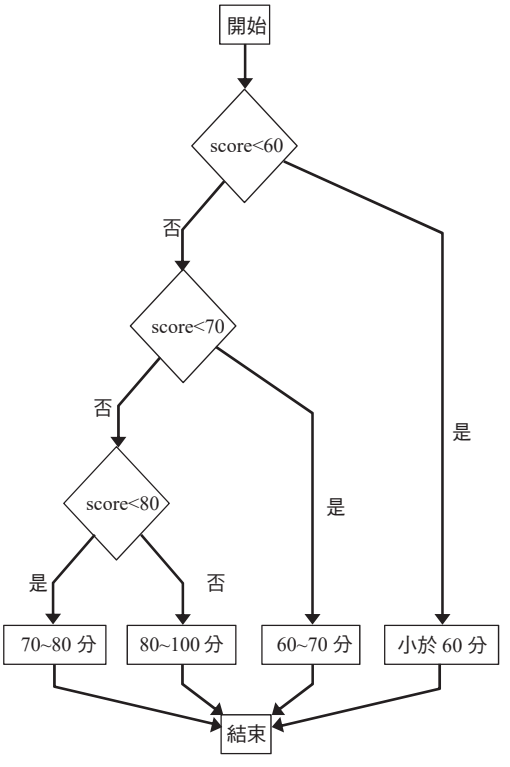
實踐篇是本書的重點，主要圍繞資料分析的各個流程展開，介紹每一個流程中都會有什麼操作，這些操作用 Excel 如何實現，用 Python 又該如何實現。

資料分析的整個流程其實和炒菜做飯的原理一樣，都是將一堆原材料整理分配成不同的成品：首先要瞭解鍋（Python 基礎知識）；然後要買米、菜等原材料（取得資料來源）；菜買回來了，需要淘米洗菜（資料預處理）；菜品洗好後是放在一起的，這個時候你要做什麼菜，就把什麼菜挑出來（資料篩選）；菜挑出來以後就可以進行切配了（數值操作）；菜切好了，就可以下鍋烹調（資料運算）；不同菜品需要烹調的時間不同，你需要有一個炒菜計時器（時間序列）；菜全部做好了，冷盤和熱菜不能放一起，必須要分開放（資料分組）；除了常規菜，還可以做一個水果拼盤（多表拼接）；所有的都做好了，就可以端上桌了（結果匯出）。

菜全部做好後，第一件事情是什麼？就是拍照上傳。拍照時要先將菜品擺盤，然後開啟相機的美顏、濾鏡拍照，拍完後將照片上傳和朋友分享，這一過程就是資料視覺化的過程。

elif 中可以有 else，也可以沒有，但是一定要有 if，具體執行順序是先判斷 if 後面的條件是否滿足，如果滿足則執行 if 為真時的程式，結束迴圈；如果 if 條件不滿足就去判斷 elif。可以有多個 elif，但是只有 0 個或 1 個 elif 語句會被執行。

舉例來說，你要猜某個人的考試分數，你該怎麼猜？先判斷這個人是否及格（60 分為準），如果不及格，分數範圍直接猜一個小於 60 分的即可，如果及格了，再去判斷他的分數到底在哪個區間，具體流程如下圖所示。



程式如下所示。

```
>>>if score < 60:
      print(" 小於 60 分 ")
>>>elif score < 70:
      print("60~70 分 ")
>>>elif score < 80:
      print("70~80 分 ")
>>>else:
      print("80~100 分 ")
```

3.2 DataFrame 表格型資料結構

3.2.1 DataFrame 是什麼

Series 是由一組資料與一組索引（行索引）組成的資料結構，而 DataFrame 是由一組資料與一對索引（列索引和欄索引）組成的表格型資料結構。之所以叫做表格型資料結構，是因為 DataFrame 的資料形式和 Excel 的資料儲存形式很相近，接下來的章節主要圍繞 DataFrame 這種表格型資料結構展開。以下就是一個簡單的 DataFrame 資料結構：

```
      技能
第一  Excel
第二  SQL
第三  Python
第四  PPT
```

上面這種資料結構和 Excel 的資料結構很像，既有欄索引又有列索引，由列索引和欄索引確定唯一值。如果把上面這種結構用 Excel 表展示如右表所示。

	技能
第一	Excel
第二	SQL
第三	Python
第四	PPT

3.2.2 建立一個 DataFrame

建立 DataFrame 使用的方法是 pd.DataFrame()，透過將不同的物件傳入 DataFrame() 方法即可實現。

傳入一個列表

傳入一個列表的實現如下所示。

```
>>>import pandas as pd
>>>df1 = pd.DataFrame(["a","b","c","d"])
>>>df1
```

4.1.2 匯入 .csv

在 Excel 中匯入 .csv 格式的檔案和開啟 .xlsx 格式的檔案一樣，按兩下即可。而在 Python 中匯入 .csv 檔用的方法是 read_csv()。

直接匯入

只需要指明檔案路徑即可。

```
>>>import pandas as pd
>>>df = pd.read_csv(r"C:\ACD019600\test.csv")
>>>df
    編號    年齡    性別    註冊時間
0   A1      54      男      2018/8/8
1   A2      16      女      2018/8/9
2   A3      47      女      2018/8/10
3   A4      41      男      2018/8/11
```

指明分隔符號

Excel 和 DataFrame 中的資料排列是整齊有規則的，這都是工具在後台根據某條規則進行切分的。read_csv() 預設檔中的資料都是以逗號分開，但是有的檔案不是用逗號分開的，這個時候就需要人為指定分隔符號，否則就會出現錯誤。

新增一個以空格作為分隔符號的檔案，如下所示：

```
編號 年齡  性別  註冊時間
A1   54    男    2018/8/8
A2   16    女    2018/8/9
A3   47    女    2018/8/10
A4   41    男    2018/8/11
```

如果用預設的逗號作為分隔符號，看看匯入的結果如何。

```
>>>df = pd.read_csv(r"C:\ACD019600\test1.csv")
>>>df
    編號年齡性別註冊時間
0   A1 54 男 2018/8/8
1   A2 16 女 2018/8/9
2   A3 47 女 2018/8/10
3   A4 41 男 2018/8/11
```

8.3.6 median 求中位數

中位數就是將一組含有 n 個資料的序列 X 按從小到大排列後，位於中間位置的那個數。

中位數是以中間位置的數來反映資料的一般情況，不容易受到極大值、極小值的影響，因而在反映資料分佈情況上要比平均值更有代表性。

現有序列為 X：{X_1、X_2、X_3、.....、X_n}。

如果 n 為奇數，則中位數：

$$m = X_{\frac{n+1}{2}}$$

如果 n 為偶數，則中位數：

$$m = \frac{X_{\frac{n}{2}} + X_{\frac{n}{2}+1}}{2}$$

例如，1、3、5、7、9 的中位數為 5，而 1、3、5、7 的中位數為 (3+5)/2=4。

在 Excel 和 Python 中求一組資料的中位數，都是使用 median() 函式來實現的。

以下為在 Excel 中求中位數的範例：

```
median(D2:D6) # 表示求 D2:D6 區域內的中位數
```

在 Python 中，median() 函式的使用原則和其他函式的一致。

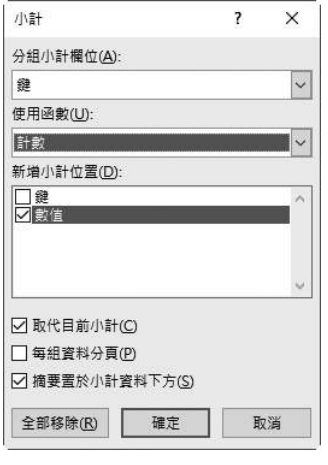
```
# 對整個表呼叫 median() 函式
>>>data = {"C1":[1,4,7],"C2":[2,5,8],"C3":[3,6,9]}
>>>df = pd.DataFrame(data,index = ["S1","S2","S3"])
>>>df
    C1  C2  C3
S1  1   2   3
S2  4   5   6
S3  7   8   9
>>>df.median()
C1    4.0
C2    5.0
C3    6.0
dtype: float64
# 求取每一列的中位數
```

鍵	數值
A	1
B	3
C	5
A	7
B	9
C	2
A	4
B	6
C	8

排序 →

鍵	數值
A	1
A	7
A	4
B	3
B	9
B	6
C	5
C	2
C	8

鍵值排序完成後，選取待分組區域，然後依次按一下功能表中的資料 > 小計即可。分類欄位、彙總方式都可以根據需求選擇。彙總方式就是對分組後的資料進行什麼樣的運算，我們這裡進行的是計數運算，因此在「新增小計位置」中勾選數值核取方塊。小計對話方塊及分組結果如下圖所示。



分組結果 →

鍵	數值
A	1
A	7
A	4
A 計數	3
B	3
B	9
B	6
B 計數	3
C	5
C	2
C	8
C 計數	3
總計數	9

Excel 中常見的彙總方式如下表所示。

彙總方式	含義
求和	對分組後的資料進行求和
計數	對分組後的資料進行計數
平均值	對分組後的資料求平均值

13.2.2 明確目的

知道了要把哪些資料圖表化以後，就需要明確目的。我們前面說了，視覺化是用來表達資訊的一種方式，既然是用來表達資訊的，就應該明確要表達什麼，要傳遞哪些資訊給看圖人。例如，要表達最近幾個月的銷量呈上漲趨勢，還是要表達用戶中有超過 50% 的用戶是 1990 年以後出生的。

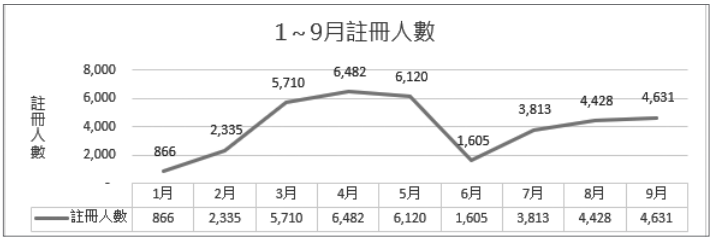
13.2.3 尋找合適的表現形式

明確了要表達什麼資訊以後，就可以選擇合適的表現形式了。不同的目的使用的表現形式是不一樣的。

繼續用前面的例子。若要說明最近幾個月的銷量趨勢首選折線圖，透過折線圖的走勢，可以很清楚地看出最近幾個月銷量是上升還是下降的；如果要說明不同年齡層用戶的占比首選圓形圖，這樣我們能很清楚地看出哪個年齡層占比最大，哪個占比最小。

13.3 圖表的基本組成元素

一個正規的視覺化圖表如下圖所示，該表包含了一個圖表中的基本組成元素。



畫布

畫布就是字面意思，你首先需要找到一塊“布”，即繪圖介面，然後在這塊“布”上繪製圖表。

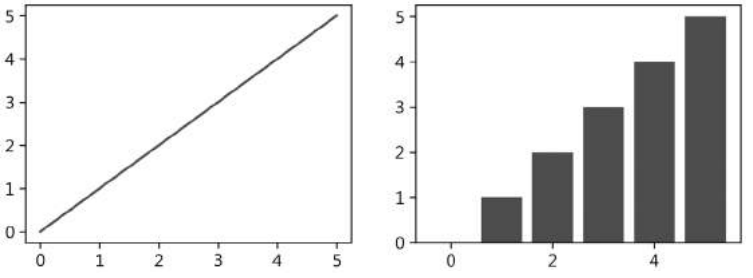
用這種方式建立座標系時，同樣需要將具體的繪圖代碼跟在建立座標系語句後面。將圖表的整個區域分成 2 行 2 列，並在第 1 個座標系上做折線圖，在第 4 個座標系上做直條圖，具體實現如下所示。

```
>>>import numpy as np
>>>x = np.arange(6)
>>>y = np.arange(6)

#將圖表的整個區域分成 2 行 2 列，且在第 1 個座標系上做折線圖
>>>plt.subplot(2,2,1)
>>>plt.plot(x,y)

#將圖表的整個區域分成 2 行 2 列，且在第 4 個座標系上做直條圖
>>>plt.subplot(2,2,4)
>>>plt.bar(x,y)
```

執行結果如下圖所示。



13.5.5 用 plt.subplots 函式建立座標系

plt.subplots 函式也是 plt 函式庫中的一個函式，它與 subplot2grid 函式和 subplot 函式的不同之處是，subplot2grid 函式和 subplot 函式每次只傳回一個座標系，而 subplots 函式一次可以傳回多個座標系。

```
>>>fig,axes = plt.subplots(2,2)
```

上面的程式指定將圖表的整個區域分成 2 行 2 列，並將 4 個座標系全部傳回，執行結果如下圖所示。

13.5.6 幾種建立座標系方法的區別

第一種建立座標系的方法 add_subplot 屬於物件式程式設計，所有的操作都是針對某個物件進行的。比如，先建立一塊畫布，然後在這塊畫布上建立座標系，進而在座標系上繪圖。而後三種建立座標系的方法屬於函式式程式設計，都是直接呼叫 plt 函式庫裡面的某個函式或者方法來達到建立座標系的目的。

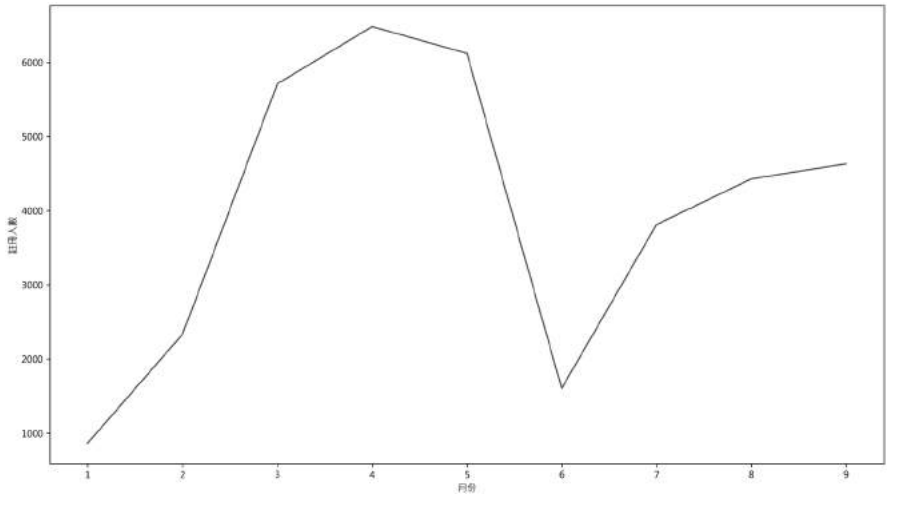
物件式程式設計的代碼比較煩瑣，但是便於理解；函式式程式設計雖然代碼簡潔，但是不利於新人掌握整體的繪圖原理，因此建議讀者剛開始的時候多使用物件式程式設計，等到對整個繪圖原理都十分熟悉後，再嘗試使用函式式程式設計。

這兩種程式設計方式不僅體現在建立座標系中，在接下來的一些操作中也會有涉及。有的時候兩者會交叉使用，也就是在一段代碼中既有函式式程式設計，也有物件式程式設計。

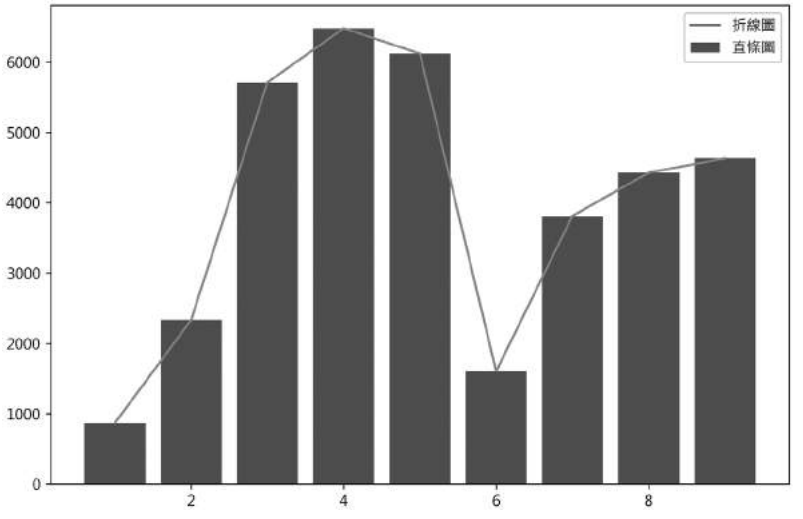
13.6 設定座標軸

13.6.1 設定座標軸的標題

下圖中橫軸的標題為月份，縱軸的標題為註冊人數。



折線圖和直條圖的圖例如下圖所示。



還可以透過修改 loc 參數的參數值來調整圖例的顯示位置，loc 參數的參數值及說明如下表所示。

字串	位置代碼	說明
best	0	根據圖表區域自動選擇最合適的展示位置
upper right	1	圖例顯示在右上角
upper left	2	圖例顯示在左上角
lower left	3	圖例顯示在左下角
lower right	4	圖例顯示在右下角
right	5	圖例顯示在右側
center left	6	圖例顯示在左側中心位置
center right	7	圖例顯示在右側中心位置
lower center	8	圖例顯示在底部中心位置
upper center	9	圖例顯示在頂部中心位置
center	10	圖例顯示在正中心位置