第一章

探索式資料分析

本章重點是介紹所有資料分析的第一步：探索資料。

古典統計學幾乎都只注重推論，推論就是基於小樣本得出關於整體資料的結論。1962 年，John W. Tukey（*https://oreil.ly/LQw6q*）（圖 1-1）發表了一篇開創性的論文「The Future of Data Analysis」[Tukey-1962]，他呼籲對統計學進行重新設計。在論文中，他提出了一門稱為**資料分析**的新學科，並將統計推論包括在內，由此 Tukey 建立了與工程和電腦科學界的聯繫（他創造了術語「軟體」和「bit」，「binary digit」的縮寫）。出乎意料的，這個原始理念被延續下來，並且成為資料科學基礎的一部分。Tukey 於 1977 年出版經典書籍《*Exploratory Data Analysis*》[Tukey-1977]，此書開創了探索式資料分析的研究領域。在書中，Tukey 展示了有助於描繪資料集樣貌的圖表（例如，箱形圖、散佈圖）及摘要統計資訊（平均數、中位數、分位數等）。

隨著計算能力和表達性資料分析軟體的可用性提升，探索式資料分析已遠遠超出了其原始範圍。該學科的主要驅動力來自於新技術的快速發展、更多更大的可獲取資料，以及定量分析更廣泛地應用於各種學科中。Stanford 大學統計學教授 David Donoho 曾在美國紐澤西州普林斯頓召開的圖基百年紀念研討會上演講並發表了一篇出色的文章 [Donoho-2015]，文中他將資料科學的起源追溯到 Tukey 在資料分析方面開拓性的工作。

圖 1-1　John Tukey，著名的統計學家，他在 50 多年前提出的理論構成了資料科學的基礎

結構化資料的組成

資料的來源有很多，例如感應器的測量值、事件、文字、圖片和影片等，並且當今**物聯網**正產生出大量的資訊流。這些資料大部分都是非結構化的：圖片是像素的集合，每個像素包含了 RGB（紅色、綠色、藍色）顏色的資訊；文字是單詞和非單詞字的排序，通常會按段落和小節等來組成。點擊是使用者在 App 或 Web 頁面上的一系列動作。事實上，如何將原始資料轉化成可操作的資訊，才是資料科學主要面臨的挑戰。如果要使用本書中所介紹的統計學概念，就必須將非結構化的原始資料處理並轉換成結構化資料。最常見的結構化資料形式之一，是具有欄和列的表格，就像是從關聯式資料庫取出的資料或被收集用於研究的資料。

結構化資料有兩種基本類型：數值型和類別型。數值資料有兩種形式：**連續型**（例如風速或持續時間）和**離散型**（例如事件的發生次數）。**類別**資料僅採用一組固定的值，例如電視螢幕的類型（等離子、LCD、LED 等）或州名（阿拉巴馬州、阿拉斯加、……等等）。**二元**資料是類別資料的特殊情況，它只能是兩個值的其中一種，例如：0/1、yes/no 或 true/false。類別資料的另一種形式是排序類別的**順序**資料。例如數字評級（1、2、3、4 或 5）。

為什麼我們要關注資料類型的分類呢？事實證明，在資料分析和預測模型中，資料類型對於確定可視化類型、資料分析或統計模型是非常重要的。*R* 和 *Python* 之類的資料科學軟體也使用這些資料類型來提高運算效能。更重要的是，變數的資料類型決定了軟體將如何處理該變數計算的方法。

重要術語

數值資料（*Numeric*）

可以數字刻度表示的資料。

連續型（*Continuous*）

可以在一個區間內取任何值的資料。

同義詞

區間資料、浮點數、數值資料

離散型（*Discrete*）

只能取整數的資料，例如計數。

同義詞

整數、計數

類別資料（*Categorical*）

只能從特定集合中選擇一種代表分類的資料。

同義詞

枚舉資料、列舉資料、因子、名目資料

二元資料（*Binary*）

一種特殊的類別資料，只能從兩個類別中取其一（例如：0 或 1、True 或 False）。

同義詞

二分資料、邏輯型資料、指標和布林

順序資料（*Ordinal*）

具有明確排序的類別資料。

同義詞

有序因子資料

軟體工程師或資料庫工程師可能對此會產生疑問：為什麼我們在資料分析中也要了解**類別資料**和**順序資料**呢？畢竟類別資料只是一組文字或數值，而資料庫會自動處理資料的內在表徵。但是相較於文字，將資料標示成類別資料確實有以下優點：

- 當我們知道資料是類別資料時，統計軟體可就此確定統計運算過程的處理方式，例如圖表生成或模型套入。具體來說，*R* 語言的順序資料可用 `ordered.factor` 來表示，這樣使用者指定的順序就能保持在圖、表和模型中；在 *Python* 中，`scikit-learn` 透過 `sklearn.preprocessing.OrdinalEncoder` 來支持順序資料。
- 可以優化資料儲存和索引（如同關聯式資料庫）。
- 限定了在給定類別變數中可以採用的可能取值（例如枚舉）。

第三個優點可能會導致一些意想不到的行為。*R* 語言的資料導入函式（例如 `read.csv`）預設將一欄文字自動轉換成 `factor`，之後在操作到該欄位資料時，會假定所允許的值侷限在先前已導入的值，若在此時再給予一個新的文字值，將會觸發警告並產生 `NA`（缺失值）。而 *Python* 的 `pandas` 套件不會自動轉換，但你可以在使用 `read_csv` 函式時指定一欄作為分類的類別。

本節重點

- 在軟體中，資料通常按類型分類。
- 資料類型包含了數值型（連續型資料及離散型資料）和類別型（二元資料和順序資料）。
- 資料分類為軟體指明了處理資料的方式。

延伸閱讀

- 資料類型有時會令人困惑，因為各個類型間會有一些重疊，而且不同軟體的資料分類可能各有不同。R Tutorial 網站（*https://oreil.ly/2YUoA*）有介紹 *R* 語言使用的資料分類方式；`pandas` 套件說明（*https://oreil.ly/UGX-4*）介紹了不同的資料類型，以及如何在 *Python* 中操作。
- 資料庫有更詳細的資料分類方式，其中考慮了精確率、固定長度、可變長度字段等因素，請參考 W3Schools 的 SQL 指南（*https://oreil.ly/cThTM*）。

矩形資料

資料科學中用於分析的典型參考框架是**矩形資料**，例如試算表或是資料庫的資料表。

矩形資料是二維矩陣的總稱，其中列（Row）表示紀錄（案例），而欄（Column）表示特徵（變數）；**資料框**（*Data frame*）是 *R* 和 *Python* 中的特定資料格式。然而資料並不總是以這種形式開始，非結構化資料（例如文字 text）必須先經過處理和整理，才能作為一組能用於矩形資料結構的特徵（請參見第 2 頁的「結構化資料的組成」）。對於大部分的情形，關聯式資料庫中的資料必須被提取出來並放進單個表格中，以完成資料分析和建模的工作項目。

重要術語

資料框（*Data frame*）

矩形資料（如試算表）是對於統計及機器學習模型最基本的資料結構。

特徵（*Feature*）

表格中的一欄通常當作一個**特徵**。

同義字

屬性、輸入、預測變數、變數

結果（*Outcome*）

許多資料科學專案都會包含預測**結果**，通常是 yes/no 結果（在表 1-1 中，預測的結果是「拍賣是否競爭」）。**特徵**有時會用來預測實驗或研究的結果。

同義字

依變數、回應、目標、輸出

紀錄（*Records*）

表格中的一列通常當作一筆**紀錄**。

同義字

案例、舉例、實例、觀察、模式、樣本

表 1-1　典型的資料框格式

類別	幣別	賣家等級	持續時間	截止日期	成交價	起標價	是否競爭
音樂 / 電影 / 遊戲	美元	3249	5	週一	0.01	0.01	0
音樂 / 電影 / 遊戲	美元	3249	5	週一	0.01	0.01	0
汽車	美元	3115	7	週二	0.01	0.01	0
汽車	美元	3115	7	週二	0.01	0.01	0
汽車	美元	3115	7	週二	0.01	0.01	0
汽車	美元	3115	7	週二	0.01	0.01	0
汽車	美元	3115	7	週二	0.01	0.01	1
汽車	美元	3115	7	週二	0.01	0.01	1

表 1-1 混合了測量和計數的資料（例如持續時間和價格）及分類資料（例如類別和幣別）。如前所述，二元資料（yes/no 或 0/1）是類別變數中的一種特殊形式，如表 1-1 的最右列，此變數顯示拍賣是否具有競爭力（是否有多個競標者）；當需要預測拍賣是否具有競爭力時，此指標變數（Indicator Variable）也剛好是**結果**變數（Outcome Variable）。

資料框和索引

傳統資料庫表會指定一個或多個欄位作為索引，本質上是一個列的編號，可以大幅地提升某些資料庫查詢的效率。在 *Python* 中，透過 `pandas` 程式庫，基本的矩形資料結構是 `DataFrame` 物件，並且在預設的情形下，*Python* 會依據 `DataFrame` 中列的順序自動產生一個整數索引。在 `pandas` 程式庫也支援多層或多階層索引，以提高特定操作的效率。

在 *R* 語言中，基本的矩形資料結構是 `data.frame` 物件。`data.frame` 物件隱含著基於列順序的整數索引。儘管使用者可以使用 `row.names` 屬性來創建自定義鍵，但 *R* 語言原生的 `data.frame` 物件並不支援自定義索引或多層索引，兩個新的套件 `data.table` 和 `dplyr` 解決了這個缺陷，因此被廣泛使用；這二者都支援多層索引，可以顯著地提升 `data.frame` 的使用效率。

術語上的差異

矩形資料的術語可能令人困惑，對於同件事物，統計學家和資料科學家會使用不同的術語來表達。統計學家在模型中使用**預測變數**來預測一個**回應**或**依變數**；資料科學家使用**特徵**來預測**目標**。另一個同義詞尤其令人困惑：**樣本**，電腦科學家使用**樣本**表示一列數據，而對於統計學家，一個**樣本**則表示一個包含許多列的集合。

非矩形資料結構

除了矩形資料之外，還有另一種資料結構。

時間序列資料紀錄了同一個變數的連續測量值，它是統計預測方法的原始材料，也是物聯網設備所產生的資料關鍵的組成成分。

空間資料結構用於地圖和區位分析，比起矩形資料結構更為複雜和多變。就**物件**觀點的表示，空間資料著重於物件（如一棟房子）及其空間座標；相反地，**場**觀點則關注空間中的小單位及相關指標（例如像素的亮度）。

圖形（或網路）資料結構用來表示實體、社交網路上及抽象的關係。舉例來說，Facebook 或 LinkedIn 的社交網路圖表示人們在網路上的相互關係；由道路連結而成的匯聚點分布則是實體網路的例子。圖形資料結構對於某些類型的問題非常有用，像是網路優化及推薦系統。

在資料科學中，每種資料類型都有其專門的理論，而本書所關注的是矩形資料，它是建立預測模型的基礎成分。

統計學中的圖形

在電腦科學及資訊科技中，**圖形**一詞通常是指對於實體間關聯的描述，以及底層的資料結構。在統計學中，**圖形**則用來代表各種圖表或**視覺化**結果，而不僅指實體間的關聯情形；此術語只用於代表視覺化結果，而非資料結構。

本節重點

- 矩形矩陣是資料科學中的基本資料結構，在矩陣中列是紀錄，而欄是變數（特徵）。
- 不同的術語可能令人困惑，在資料科學相關的學科之間（例如統計學、電腦科學及資訊科技），有著一系列的同義詞。

延伸閱讀

- *R* 語言中關於資料框的文件：（*https://oreil.ly/NsONR*）
- *Python* 關於資料框的文件：（*https://oreil.ly/oxDKQ*）

位置估計

測量或計數的變數可能會有上千種不同的值。探索資料的一個基本步驟是取得每個特徵（變數）的「典型值」：對資料最常出現位置的估計（即資料的集中趨勢）。

重要術語

平均數（*Mean*）

所有數值的總和除以數值的個數。

同義詞

平均

加權平均數（*Weighted mean*）

所有數值乘以權重後的總和，再除以權重之和。

同義詞

加權平均

中位數（*Median*）

在排序資料中，位於正中間可將資料分為上下兩等分的數值。

同義詞

第 50 百分位數

百分位數（*Percentile*）

將資料由小到大排序並計算相應的累計百分比，位於百分之 *P* 的數值。

同義詞

分位數（Quantile）

加權中位數（*Weighted median*）

在排序資料中，可將權重總和平分為上下兩等分的數值。

截尾平均數（*Trimmed mean*）

排除排序資料中兩端固定數量的極端值後計算的平均數。

同義詞

截尾均值 （Truncated mean）

穩健（*Robust*）

對極端值不敏感。

同義詞

耐抗性（Resistant）

離群值（*Outlier*）

與大部分資料差異甚大的數值。

同義詞

極端值

彙總資料乍看之下似乎非常簡單，像是計算資料的**平均數**即可。事實上，雖然平均數易於計算且方便使用，但它不一定總是最佳用來測量中心值的方法；因此，統計學家研究並提出了幾種能替代平均數的估計量。

指標和估計量

統計學家通常將**估計**（*estimate*）一詞用於手上已有的數值，以區分我們所看到的資料情形與確切的狀態或理論的真實性。資料科學家和商業分析師更傾向於將這樣的數值稱為一個**指標**。這術語上的差異反映了統計方法和資料科學方法的不同：統計學的核心在於解釋不確定性，而資料科學則注重具體業務或組織的目標。因此統計學家「估計」，而資料科學家「測量」。

平均數

平均數或稱為**平均**，最基本的位置估計量，等於數值的總和除以數值的個數。例如一組數字：{3 5 1 2} 的平均數是 (3 + 5 + 1 + 2) / 4 = 11 / 4 = 2.75。一般會使用符號 $\bar{x}$（讀作「x bar」）表示母體樣本的平均數。給定一組 n 個數值：x_1, x_2, ..., x_n，平均數的公式為：

$$\text{平均數} = \bar{x} = \frac{\sum_{i=1}^{n} x_i}{n}$$

通常使用 N（或者 n）表示紀錄值或觀察值的總數。在統計學中，大寫字母 N 代表母體，而小寫字母 n 代表母體中的一個樣本；但在資料科學中，這個區別不太重要，因此兩種表示方法皆可見。

截尾平均數是平均數的一個變形，在計算截尾平均數時，會去除排序資料兩端一定數量的值，再計算剩餘數值的平均數。若使用 $x_{(1)}$, $x_{(2)}$, ..., $x_{(n)}$ 作為一組排序資料，其中 $x_{(1)}$ 為最小值，$x_{(n)}$ 為最大值，則去除 p 個最大值和 p 個最小值的截尾平均數的計算公式為：

$$\text{截尾平均數} = \bar{x} = \frac{\sum_{i=p+1}^{n-p} x_{(i)}}{n - 2p}$$

截尾平均數能消除極端值對平均數的影響。舉例來說，在國際跳水比賽中，有五位裁判評分，而一位選手最終的分數會排除五個評分中最高分和最低分後，以剩餘三個分數取平均，這樣能確保裁判難以操縱選手的分數，因為裁判有可能會偏向自己國家的選手。截尾平均數被廣泛的運用，在很多情形下，比起一般的平均數，人們更傾向於使用截尾平均數。第 11 頁的「中位數與穩健估計」將會有更詳細的介紹。

另外一種平均數是**加權平均數**，在計算時會將每個數值 x_i 乘以一個權重 w_i 後，再將總和除以權重的總和。計算公式為：

$$\text{加權平均數} = \bar{x}_w = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i}$$

使用加權平均數主要是基於以下兩點的考量：

- 有些數值在本質上較其他值多變，因此須將多變的觀察值予以較低的權重；例如，當我們要對來自多個感應器的數據計算平均時，但其中一個感應器的數據不是很準確，那麼我們可以降低該感應器數據的權重。
- 所收集的資料可能無法準確代表我們想要測量的不同群體；例如，由於線上實驗的進行方式，我們可能無法得到一組能精確反映所有使用者群體的數據；為了修正這一點，我們可以對來自代表性不足的群體予以較高的權重。

中位數與穩健估計

中位數是位於一組排序資料的中間數值；如果這組數值個數為偶數，則中位數不是這組資料中的數值，而是位於中間的兩個數值的平均。相較於平均數使用所有觀察值，中位數只取決於排序資料的中間值。雖然這樣看起來中位數存在著缺點，但考慮到平均數對資料更敏感，不少案例能證明，中位數仍是更好的位置指標。舉例來說，我們想了解西雅圖華盛頓湖周邊地區典型家庭的收入情形。在比較麥地那地區和溫德米爾地區時，使用平均數會產生很不同的結果，因為比爾蓋茲就住在麥地那地區；但我們若使用中位數，統計結果就不會受到比爾蓋茲的影響，因為位於中間的觀察值沒有改變。

基於使用加權平均數的相同原因，我們有時也需要使用**加權中位數**。雖然每個數值有對應的權重，我們需要和計算中位數的方法一樣，先將資料排序。加權中位數不是取位於中間的數值，而是取可以使權重總和平分為上下兩部分的數值。和中位數一樣，加權中位數也對極端值不敏感。

離群值

中位數是一種**穩健**的位置估計量，因為它不會受到會造成統計結果偏差的**離群值**（極端個案）影響。離群值是指距離資料集中其他值很遠的數值。儘管在各種資料匯總和圖表繪製中對離群值已有一些慣例（參見第 20 頁的「百分位數與箱形圖」），但對於離群值的確切定義還是有點主觀。離群值本身並不一定是無效或是錯誤的資料（如前面例子中

比爾蓋茲的收入），但通常是由於資料錯誤所導致的。例如，資料混淆了不同的指標單位（混用了公里和公尺），或是感應器讀數不準確；當錯誤的資料導致了離群值，使用平均數就會是不好的位置估計，然而中位數的估計仍然有效。無論如何，我們都應該找出離群值，而它們通常值得進一步研究。

異常檢測（*Anomaly Detection*）

在典型的資料分析中，離群值有時能提供資訊，有時令人討厭；相反的是，**異常檢測**所關注的就是離群值，而其他大部分的資料則用於定義與異常相對的「正常」情形。

中位數並非是唯一穩健的位置估計量；事實上，截尾平均數亦被廣泛運用來消除離群值的影響。例如，對於非小型的資料集，去掉頭尾各 10% 的資料（是個常見的選擇），可以保護資料免於極端值的影響。截尾平均數可視為一種介於平均數和中位數之間的折衷方案：它對於極端值非常穩健，並且能使用較多資料來計算位置估計量。

其他的穩健位置估計

統計學家還提出了其他的位置估計量，主要的目的是能提出更穩健、更有效率的估計量（即能更好地辨別出資料集之間微小的位置差異）。這些估計量通常適用於小型資料集，而對於大規模至於中規模的資料集，可能無法提供幫助。

範例：人口與謀殺率的位置估計

表 1-2 顯示了一個資料集的前幾列，包含了美國各州的人口數及謀殺率，單位為每年每十萬人中被謀殺的人數。

表 1-2　*data.frame* 中的幾行資料，列出了美國各州的人口數和謀殺率

	州	人口	謀殺率	縮寫
1	阿拉巴馬州	4,779,736	5.7	AL
2	阿拉斯加州	710,231	5.6	AK
3	亞利桑那州	6,392,017	4.7	AZ
4	阿肯色州	2,915,918	5.6	AR
5	加利福尼亞洲	37,253,956	4.4	CA
6	科羅拉多州	5,029,196	2.8	CO

	州	人口	謀殺率	縮寫
7	康乃狄克州	3,574,097	2.4	CT
8	德拉瓦州	897,934	5.8	DE

使用 *R* 語言計算人口數的平均數、截尾平均數和中位數如下：

```
> state <- read.csv('state.csv')
> mean(state[['Population']])
[1] 6162876
> mean(state[['Population']], trim=0.1)
[1] 4783697
> median(state[['Population']])
[1] 4436370
```

在 *Python* 中我們可以使用 pandas 對資料框方法來計算平均數和中位數，而截尾平均數則需要用到 scipy.stats 的 trim_mean 函式：

```
state = pd.read_csv('state.csv')
state['Population'].mean()
trim_mean(state['Population'], 0.1)
state['Population'].median()
```

平均數大於截尾平均數，而截尾平均數則大於中位數。

這是因為截尾平均數分別排除了五個最大和最小數值（trim=0.1 表示去除兩端各 10% 的資料）。若要計算美國的平均謀殺率，我們需要考慮到各州人口數的差異，而使用平均數或中位數。*R* 語言沒有內建加權平均數的函式，因此我們需要安裝外部套件，像是 matrixStats：

```
> weighted.mean(state[['Murder.Rate']], w=state[['Population']])
[1] 4.445834
> library('matrixStats')
> weightedMedian(state[['Murder.Rate']], w=state[['Population']])
[1] 4.4
```

在 *Python* 中，計算加權平均數可以使用 NumPy，而計算加權中位數則可使用特殊的套件 wquantiles（*https://oreil.ly/4SIPQ*）：

```
np.average(state['Murder.Rate'], weights=state['Population'])
wquantiles.median(state['Murder.Rate'], weights=state['Population'])
```

在本案例中，加權平均數和加權中位數大致相同。

本節重點

- 平均數是一種基本的位置指標，但對於極端值（離群值）敏感，易受影響。
- 其他位置指標（中位數、截尾平均數）對離群值和異常分布較不敏感，因此也較穩健。

延伸閱讀

- 維基百科上關於集中趨勢（Central tendency）的文章（*https://oreil.ly/qUW2i*）包含對於各種位置測量值的廣泛討論。
- John Tukey 於 1977 年發表的經典著作《*Exploratory Data Analysis*》（Pearson）至今仍廣為閱讀。

變異性估計

位置只是一個總結特徵值的維度。第二個維度是**變異性**（Variability），又稱**離散程度**，用於測量資料數值是緊密聚集或是發散的。變異性是統計學的核心概念：如何測量變異性、如何降低變異性、如何識別真實變異性中的隨機性、如何辨識真實變異性的各種來源，以及如何在存有變異性的情況下做出決定。

重要術語

偏差（*Deviations*）

位置的觀察值與估計量之間的差異。

同義詞

誤差、殘差

變異數（*Variance*）

對於 n 個數值，變異數是每個數值與平均數之間的偏差值，平方後加總，再除以 $n - 1$。

同義詞

均方誤差（mean-squared-error）

標準差（*Standard deviation*）

變異數的平方根。

平均絕對偏差（*Mean absolute deviation*）

每個數值與平均數之間的偏差值取絕對值後的平均數。

同義詞

L1 範數、曼哈頓範數

中位數絕對偏差（*Median absolute deviation from the median*）

每個數值與平均數之間的偏差值取絕對值後的中位數。

全距（*Range*）

又稱極差，是資料集中最大值與最小值之間的差額。

順序統計量（*Order Statistics*）

根據數值大小排序的指標。

同義詞

排名

百分位數（*Percentile*）

第 *P* 百分位數，表示在資料集中，有 *P*% 的值等於或在其下，而有 (*100-P*)% 的值等於或在其上。

同義詞

四分位數

四分位距（*Interquartile range*）

第三四分位數和第一四分位數之間的差距。

同義詞

四分位差

正如同有平均數、中位數等不同的方式來測量位置，亦有多種不同的方式來測量變異性。

標準差與相關估計數

最廣泛使用於估計變異量的方法是基於位置估計量與觀察值之間的差異或**偏差**。給定一組資料 {1, 4, 4}，平均數為 3，而中位數為 4；每個數值與平均數之間的差距分別為：1 – 3 = –2、4 – 3 = 1、4 – 3 = 1。透過這些偏差，我們可得知資料與中心值之間的分散程度。

測量變異性的一種方法是去估計這些偏差的典型值；對偏差值取平均，無法提供我們太多資訊，因為負的偏差值會和正的偏差值抵銷。事實上，相對於平均的偏差值之總和恰好為零；因此另一種簡單的方法是將數值與平均數之差取絕對值後計算平均。上面的例子，各個偏差的絕對值分別為 {2 1 1}，而這些絕對值的平均數為 (2 + 1 + 1) / 3 = 1.33。這就是**平均絕對偏差**，計算公式如下：

$$\text{平均絕對偏差} = \frac{\sum_{i=1}^{n} |x_i - \bar{x}|}{n}$$

其中 $\bar{x}$ 是樣本的平均數。

最廣為人知的變異性估計量為**變異數**和**標準差**，這二者是基於偏差的平方。變異數是偏差平方的平均，而標準差是變異數的平方根。

$$\text{變異數} = s^2 = \frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{n-1}$$

$$\text{標準差} = s = \sqrt{\text{Variance}}$$

標準差與原始資料有相同的尺度，因此它比變異數更易於解讀；但其實標準差的公式更複雜也不太直覺，所以我們可能會覺得奇怪，為什麼在統計上更傾向使用標準差而非平均絕對偏差？這是由於標準差在統計學理論中的卓越之處：從數學的角度來看，使用平方比使用絕對值更方便，尤其是對統計模型而言。

自由度是 *n*，還是 *n* – 1？

在統計學書籍中，對於計算變異數時，被除數為什麼要用 $n - 1$ 而非使用 n，總是多有討論；這討論也帶出了**自由度**的概念。這兩者的計算結果差異不大，這是因為 n 通常夠大，以致於使用 n 或 $n - 1$，結果不會有太大的差別。但若你有興趣了解這個問題，我們以你想要根據樣本來估計母體為前提，解釋一下原因。

如果在計算變異數的公式中使用了直觀的除數 n，就會低估變異數的真實值和母體的標準差；這被稱為**有偏估計量**（*Biased estimate*）。然而若除以 $n-1$，此時變異數就成了**無偏估計量**（*Unbiased estimate*）。

要完整解釋為什麼使用 n 會導致有偏估計量，就涉及了自由度的概念。自由度考慮了計算估計量中的限制個數；在這種情形下，自由度是 $n-1$，因為其中有一個限制：標準差取決於計算樣本的平均數。對於大部分的情形，資料科學家不需要考慮自由度的問題。

不論是變異數、標準差，還是平均絕對偏差，三者對於極端值和離群值都不穩健（參閱第 11 頁的「中位數與穩健估計」關於穩健位置估計量的討論）。變異數和標準差對離群值都極度敏感，因為它們是基於偏差的平方值。

中位數絕對偏差（*median absolute deviation from the median*, MAD）是一個穩健的變異性估計量。計算公式為：

$$\text{中位數絕對偏差} = \text{Median}\left(|x_1 - m|, |x_2 - m|, \ldots, |x_N - m|\right)$$

其中 m 是中位數。如同中位數，中位數絕對偏差也不會受到極端值的影響。我們亦可以參考截尾平均數的算法來計算截尾標準差（參見第 10 頁的「平均數」）。

即使資料符合常態分布，變異數、標準差、平均絕對偏差及中位數絕對偏差並非是相等的估計量。事實上，標準差總是大於平均絕對偏差，而平均絕對偏差總是大於中位數絕對偏差。在常態分布的情形下，有時中位數絕對偏差會乘上一個常數尺度因子，以讓中位數絕對偏差與標準差有相同的尺度。常數尺度因子通常使用 1.4826，表示常態分布的 50% 會落在正負中位數絕對偏差之間的範圍內。（請見：*https://oreil.ly/SfDk2*）

基於百分位數的估計量

另一種估計離散程度的方法是基於觀察排序資料的分布情形。基於排序資料的統計量被稱為**順序統計量**。最基本的測量值是**全距**，即為資料中最大值與最小值的差異。能知道最大值與最小值是十分有用的，有助於辨別出離群值；然而全距對於離群值非常敏感，因此對於測量資料的離散程度助益不大。

為了避免受到離群值的影響，我們可以先排除兩端的資料再計算全距。正式而言，此估計量是基於**百分位數**之間的差異。在一組資料中，第 P 百分位數表示至少有 P% 的數值小於或等於此數，而至少有（100 – P）% 的數值大於或等於此數。例如，要找到第 80 百分位數，我們要先將資料排序，再從小到大的順序找出佔 80% 的數值。請注意，中位數等同於第 50 百分位數；百分位數在本質上等同於**四分位數**，而四分位數是以分數作為索引，因此 0.8 四分位數等同於第 80 百分位數。

一種測量變異性的常用方法是計算第 25 百分位與第 75 百分位之差，稱為**四分位距**（IQR）。舉例來說，對於資料集 {3,1,5,3,6,7,2,9}，我們先排序成 {1,2,3,3,5,6,7,9}，第 25 百分位是 2.5，而第 75 百分位是 6.5，因此四分位距為 6.5 – 2.5 = 4。不同的軟體在計算方法上可能有所不同，因此會得出不同的答案，但通常差異很小。（請參見本節末段的小祕技）

對於非常大的資料集，準確計算百分位的成本是非常高的，因為需要對所有資料做排序；因此機器學習和統計軟體會利用特別的演算法，例如 [Zhang-Wang-2007]，快速計算出有一定準確度的近似百分位數。

四分位距的準確定義

如果我們有一組偶數個數的資料，那麼根據前述定義，百分位數不一定是唯一的。事實上我們可以取任一位於順序統計量 $x_{(j)}$ 和 $x_{(j+1)}$ 之間的數值，只要 j 滿足：

$$100 * \frac{j}{n} \leq P < 100 * \frac{j+1}{n}$$

正式的說法是，百分位數是一種加權平均：

$$\text{Percentile}(P) = (1 - w)x_{(j)} + wx_{(j+1)}$$

其中權重值 w 介於 0 到 1 之間。不同的統計軟體選取 w 的方法略有不同。*R* 語言的函式 `quantile` 提供了九種計算百分位數的方法。除非是小型的資料集，否則我們通常不需要煩惱百分位數的準確計算方法。目前，*Python* 的 `numpy.quantile` 套件只支援 1 種方法：線性內插法（linear interpolation）。

範例：美國各州人口數的變異性估計

表 1-3（為求方便，我們重複表 1-2）顯示了資料集的前幾行資料，列出美國各州的人口數和謀殺率。

表 1-3　*data.frame* 中的幾行資料，列出了美國各州的人口數和謀殺率

	州	人口	謀殺率	縮寫
1	阿拉巴馬州	4,779,736	5.7	AL
2	阿拉斯加州	710,231	5.6	AK
3	亞利桑那州	6,392,017	4.7	AZ
4	阿肯色州	2,915,918	5.6	AR
5	加利福尼亞洲	37,253,956	4.4	CA
6	科羅拉多州	5,029,196	2.8	CO
7	康乃狄克州	3,574,097	2.4	CT
8	德拉瓦州	897,934	5.8	DE

我們可以使用 *R* 語言內建的標準差、四分位距（IQR）及中位數絕對偏差（MAD）函式，來計算人口數的變異數估計量：

```
> sd(state[['Population']])
[1] 6848235
> IQR(state[['Population']])
[1] 4847308
> mad(state[['Population']])
[1] 3849870
```

pandas 的資料框提供計算標準差和四分位數的方法，使用百分位數可以確定四分位距。而穩健的中位數絕對偏差則可以用 statsmodels 套件的 robust.scale.mad 函式：

```
state['Population'].std()
state['Population'].quantile(0.75) - state['Population'].quantile(0.25)
robust.scale.mad(state['Population'])
```

標準差幾乎是兩倍的中位數絕對偏差 MAD（*R* 語言預設將 MAD 的尺度調整成與平均數相同）。這並不奇怪，因為標準差對離群值很敏感。

本節重點

- 變異數和標準差是被廣泛慣用的變異性統計量。
- 二者皆對離群值敏感。
- 較穩健的指標包含平均絕對偏差、中位數標準偏差和百分位數（四分位數）。

延伸閱讀

- David Lane 的線上統計學課程有個章節關於百分位數的介紹。（*https://oreil.ly/o2fBI*）
- Kevin Davenport 在 *R*-Bloggers 上發表了一篇有用的文章，介紹了中位數的偏差及其穩健的特性。（*https://oreil.ly/E7zcG*）

探索資料的分布

我們提到的每個估計值都會加總成一個數字，以描述資料的位置或是變異性，這對探索資料的分布也十分有用。

重要術語

箱形圖（*Boxplot*）
Tukey 所發明的一個圖表，能快速的將資料分布視覺化。

同義詞
盒鬚圖

次數表（*Frequency table*）
數值資料之值落在各組區間內的個數。

直方圖（*Histogram*）
次數表的圖表，各組區間在 x 軸，而計數或比率在 y 軸。雖然看起來很相近，長條圖不應該和直方圖搞混。有關這二者的差異，參見第 27 頁的「探索二元資料及類別資料」。