第一章

資料管理

R 是免費統計軟體，提供基礎的程式功能以及許多基礎與進階的統計功能，R 的開放性與繪圖能力更是特點。

開放性是 R 的一大特點，每個 R 使用者都可以撰寫自己的函數，組織常用功能，方便程式碼重複使用。很多 R 使用者也將自己撰寫的若干個函數包裝為套件，集中在 R 的官方網站上 CRAN（http://cran.r-project.org/）讓人下載。截至 2023/7/25 為止，CRAN 上有 19881 個程式套件，從資料輸入輸出到進階的統計功能一應俱全。發展新的統計方法的研究者亦常推出 R 套件，因此相較於一般商用統計軟體，R 更能快速跟上資料分析的潮流。

R 可以輕易地畫出品質很好的統計圖形，每個細節幾乎都可以控制。本書的統計圖表完全利用 R 搭配本書所附程式碼繪製而

成，讀者可以更動資料檔案與程式碼設定，快速產生相同品質的統計圖形。

資料管理統計分析或是繪製圖形前置步驟，將資料轉為需要的形式。本章主要目的在介紹，利用 tidyverse 這個超級套件組中的 dplyr 和 tidyr 套件作資料管理。在本書第二版中，我們刻意集中使用 tidyverse 這個超級套件組，或是有關的套件。根據 tidyverse 主要作者 Hadley Wickham，tidy 意為整潔，verse 則是詩篇，tidy-verse 就是希望程式碼或是資料能如詩篇般整潔易讀。越來越多 R 的使用者發現 tidy 風格資料與程式碼的易用與可讀，也越來越多套件撰寫者跟進此風格，tidyverse 這個超級套件組即收集了以此風格撰寫的資料管理以及繪圖等等套件，在此超級套件組之外，很多以此風格撰寫的套件也會將 tidy 這個詞放進套件名稱中。

在資料管理中，我們常常執行「分割-應用-合併」的序列動作。以計算不同性別下各科成績的平均數為例，我們需先將資料「分割」為男、女兩個子資料、在各科成績「應用」計算平均數這個功能，並將各科平均數「合併」成表格；或是，我們想計算不同行業的薪資最大值等等，都需要「分割-應用-合併」的序列動作。dplyr 這個套件，搭配 R 的管道運算子 |>（或 magrittr 的管道運算子 %>%），讓這種「分割-應用-合併」變的非常便利。

給定一筆資料，dplyr 套件可以選擇欄位、切出一部份資料（切片）或取樣若干觀察值，依變數分組，以及轉換和彙總資料，成為後續處理步驟所需的格式。dplyr 用「動詞」表示這些指令（如 select, filter, mutate），讓這些指令非常貼近動作的直覺，而更有趣的是，這些「動詞」指令可以搭配「副詞」修飾，讓指令在某些條件下進行，或用某種方式進行。

本章亦將展示後續各章常用的資料格式轉換，即讓資料在寬格式（wide format）與長格式（long format）中轉換。寬資料中，每橫列代表一個個體（通常是人），每縱行代表一個變項（個體在某個特性上的值），長資料中，每橫列則對應個體與某些變項的組合。在重複測量的研究中，我們可以看到這兩種資料格式，以 100 位參與者，紀錄五年身高為例，我們可以以寬格式紀錄，則包含 100 橫列、6 縱行（參與者 ID 與 5 年身高）；我們也可以以長格式記錄，包含 500 橫列（每列對應一個人-年組合），3 個縱行（參與者 ID、年度、身高）。不同的套件或指令可能偏好某種資料格式，掌握轉換格式技能，我們就可以順利運用各種套件。套件 tidyr 讓格式轉換變得更加容易，且還包含將列（或行）合併（或分離）甚至合併資料集的功能。

本章接下來的內容，我們將展示如何使用 dplyr 提供的「動

詞」和「副詞」來探索資料，tidyr 套件轉換資料格式，以及如何整理研究人員感興趣的特定問題資訊。

本章分成兩個小節，第一節在說明進行「分割-應用-合併」中的各項基本資料操作，包括選取變項、觀察值、篩選變項、變項轉換等等，主要用 dplyr；第二節則是寬格式與長格式間的轉換，同時利用 dplyr 與 tidyr。

資料來自於許功餘、張玉鈴（2015），討論性格向度與青少年問題間的關聯。研究在性格向度方面採用 HEXACO 性格模式，包括六個向度：誠實／謙遜（Honesty-Humility, H），情緒性（Emotionality, E），外向性（Extraversion , X），和悅性（Agreeableness, A），嚴謹性（Conscientiousness, C）與開放性（Openness to Experience, O）。青少年問題則測量焦慮／憂鬱、社會退縮、違反規定以及攻擊行為四類，前兩類是個人內在的情緒或身體問題，稱為內化問題行為，後兩類則是指個人與他人或與他人期望相衝突的問題，稱為外化問題行為。

過去研究發現性格向度影響青少年的問題行為之產生，例如，誠實／謙遜，反映了個體對某些道德原則與社會規範的掌握與堅持，可能使得個體表現出較少違反規定的行為。情緒性可能增加青少年的內化與外化問題行為發生可能性，和悅性、嚴謹性則會

降低發生可能性。此外，可能由於性別角色，內化行為問題的發生在女性居多，男性則較容易發生外化行為問題。

範例資料

資料檔 HEXACO.csv 是一個 CSV 檔，檔案中第一列是變項名稱，總共包含 14 個變項，資料結構如程式報表 1.1。資料中包括 1630 位參與者，記錄參與者的 4 個背景變項（國高中、性別、父親教育程度、母親教育程度），6 個性格向度與 4 個問題行為。

R 程式報表 1.1

```
'data.frame': 1630 obs. of  14 variables:
 $ 國高中       : Factor w/ 2 levels "高中","國中": 1 1 1 1 1 1 1 1 1 1 ...
 $ 性別         : Factor w/ 2 levels "女","男": 2 1 2 2 NA 1 1 1 2 2 ...
 $ 父親教育程度: Factor w/ 5 levels "大學或專科","小學或不識字",..: 5 4 5 4 NA 2
 $ 母親教育程度: Factor w/ 5 levels "大學或專科","小學或不識字",..: 5 5 4 4 NA 4
 $ 誠實.謙遜    : int  48 60 53 48 56 61 53 66 55 58 ...
 $ 情緒性       : int  54 43 50 52 60 65 45 68 52 44 ...
 $ 外向性       : int  44 39 48 46 52 39 51 60 38 55 ...
 $ 和悅性       : int  50 55 47 50 49 47 51 48 55 54 ...
 $ 嚴謹性       : int  43 59 52 46 53 49 48 51 43 43 ...
 $ 開放性       : int  44 57 44 49 44 43 45 51 38 50 ...
 $ 攻擊行為     : int  13 2 1 12 4 NA 21 1 4 2 ...
 $ 焦慮.憂鬱    : int  11 2 0 9 2 6 14 6 5 1 ...
 $ 違反規定     : int  7 1 0 9 1 2 12 0 1 3 ...
 $ 社會退縮     : int  9 4 0 5 3 7 12 2 4 2 ...
```

一、分割-應用-合併

程式碼 1.1 是本書所有章節共用的程式碼，用來設定環境。程式碼前面有 # 符號，則為註解、不會被執行。設定包括四部份，第一部份（前五行）牽涉語系，我們建議設定為中文語系，若為

英文語系而要改為中文語系，可依作業系統別為 Windows 或是 Mac，在 R 的 console 視窗輸入對應指令。第二部份則是管理各章使用的套件，此處程式碼會自動安裝需要的套件。程式碼先載入協助管理套件安裝的 pacman 套件（如果不曾安裝則立即安裝後載入），再用 pacman 套件的 p_load 指令載入其他套件。第三部份是設定輸出時的字型，也需要依作業系統分別設定。第四部份則是將繪圖配色主題預設為極簡主題，以因應本書印刷。四個部份中，只有第二部份因各章使用套件讀同而有差異，其餘各部份程式碼皆相同。

R 程式碼 1.1

```
# 推薦作業系統是設定為中文語系，若為英文語系要改為中文語系：
# Windows 作業系統使用者在 R Console 輸入：
# Sys.setlocale(category = "LC_ALL", locale = "cht")
# Mac 作業系統使用者輸入：
# Sys.setlocale(category = "LC_ALL", locale = "zh_TW.UTF-8")

#確認套件管理軟體 pacman 有載入
if(!require("pacman")){
  install.packages("pacman", repos="https://cran.csie.ntu.edu.tw")
  library(pacman)
}
#載入本章所用的套件供後續使用
pacman::p_load(tidyverse, forcats, printr, flextable, broom,
               ragg, here, jtools, broom, webshot2,
               statmod,ggplot2,GGally,cocor,confintr,
               MASS, magrittr, gtsummary, parameters,
               kableExtra, Hmisc, tidyr)

#使用微軟 Windows 作業系統，請移除以下行的 #
par(family = 'Microsoft Ya Hei')
#使用蘋果 Mac 作業系統，請移除以下行的 #
#par(family='Kaiti TC')
```

```
#底下的圖都用黑白配色（theme_minimal）
ggplot2::theme_set(theme_minimal())
```

程式碼 1.2 讀取資料並顯示資料的初步資訊。程式碼第二列利用 read.csv 指令讀進 CSV 檔，由於檔案的第一列有變項名稱（讀者可以以記事本開啟檔案確認），我們加上 header=TRUE 的指令。指令後面設定，檔案中的 NA 字串表示遺漏值（n.a., not admissible），變項值中如果讀到字串，該變項將被視為類別變項（factor）。請注意，R 的指令與變項名稱，都有區分大小寫。

R 程式碼 1.2

```
#讀檔案
dta <- read.csv("../Data/HEXACO.csv", na.strings='NA', stringsAsFactors = TRUE)

#程式報表1.1，檢視資料結構
str(dta)
```

讀進 CSV 檔後會形成**資料框**（data frame），利用「<-」，我們將它複製到 dta（也可以當成命名此物件）。資料框是分析時最常遇到的物件，它是由變項組成的，包括**數值資料**（numeric）、**類別資料**（factor），也可以在一個資料框內同時包含數值資料與類別資料。

一般的文字檔則可利用 read.table 指令讀進 R 中。利用 foreign 套件，可以利用 read.spss 讀進 SPSS 的 SAV 檔，read.ssd、read.dta

讀進 SAS 與 STATA 的資料檔；如果要讀取 Excel 檔案，則可以利用 openxlsx 套件的 read.xlsx 指令。

指令 str 會呈現資料框的結構，包括有幾筆觀察值，有哪些變項，以及變項是連續變項或是類別變項。如果是類別變項，亦會顯示包括幾個水準等等。程式報表 1.1 呈現 str(dta) 的結果，可以看到資料包括 1630 筆、14 個變項。

第一個變項是「國高中」，背後寫了 Factor，表示是類別變項，有兩個水準，分別是高中與國中，第二個變項「性別」也是類別變項，包括女、男兩個水準。第五個變項「誠實. 謙遜」開始的變項，背後跟著 int，表示是整數的數值資料。

資料框包括多個變項，names(dta) 可以看到變項名稱。head(dta) 可以看到資料 dta 的「頭」六筆（六是預設值），head(dta,10) 則顯示前十筆。依樣畫葫蘆，tail 則看到資料的「尾」。

資料框是一個二維的物件，dta[2,1] 表示第 2 橫列、第 1 縱欄的資料，也就是第 2 位參與者在第 1 個變項（國高中）上的值。利用 dta[9,]，可以得到第九橫列（第九位參與者）的資料，利用 dta[, 1]，我們則得到第一欄資料，也就是第一個變項。我們也可以用排除方式，把其他資料取出來，dta[-9,] 可以取出第九位參與者以外其他參與者的資料，dta[,-1] 則取出第一個變項外，其餘變項的

資料。

取出變項也可以利用名稱，例如，dta[,’國高中’] 可以得到國高中這個變項，或是用 dta$ 國高中也可以。利用 dta[5:7,] 可以擷取第五到第七位參與者資料，dta[5:7,c(’國高中’,’性別’)] 則是取出第五到第七位參與者的國高中與性別資料，這會是較小的資料框。

我們也可以利用條件擷取資料檔中的一部分，例如，dtam <- dta[dta$ 性別 ==’男’,] 會取出男生資料，放在資料框架 dtam 中。要求性別為男生相當於挑選參與者，所以被放在列的座標而非欄的座標中。

雖然基本的 R 已經有擷取變項或觀察值的方式，但會讓程式碼很不好讀，我們以 dplyr 的「動詞」指令，說明如何擷取觀察值或變項。

程式碼 1.3 示範如何利用 dplyr 的 select 選取變項，並搭配使用管道運算子。dplyr 的 select 用於選取變項，需要讀入資料與變項的條件，例如，select(dta, 性別) 會挑出 dta 中的性別變項。由於不同套件可能有同名的指令，因此我們以 dplyr::select，確認呼叫的是 dplyr 套件的 select。我們在 select 設定 where(is.factor)，其中 where 就是用來修飾動詞 select 的副詞，會回傳符合條件的變項，is.factor 則是我們設定的條件，整個指令就是表示要在縱欄中找尋

是類別變項的變項。

管道運算子可以想成水管，讓我們將資料導來導去。我們常常會對資料加工，再對加工後的資料再加工，甚至再對再加工後的資料再再加工；加工常常是呼叫函數，並在其中指定資料。假設我們要抽出類別變項再抓出前六筆，我們可以先以 select 挑出類別變項（注意其中有設定資料檔），再利用 head 抓出求六筆。因為 head 也需要設定資料檔，因此我們需要先將第一個動作結果存成 dta2，以在 head 中設定 dta2。

```
dta2 <- select(data, where(is.factor))
head(dta2)
```

如果動作比較多，我們需要產生很多個中間過程的資料框並取名字。這些資料框常常只需要出現一次，卻佔住了電腦與人腦的記憶體（後者尤為可怕）。為節省記憶體，我們也可以改寫成

```
head(select(dta, where(is.factor)))
```

這樣雖然減省，但就不易讀了；需要熟悉一段時間才能快速意會，資料其實在最裡面，接著一層是第一次的動作，最外層反而是最後一次動作。如果以管道運算子，程式的順序就與運算的順序一致，相當符合直覺，如以程式碼 1.3。

以程式碼 1.3 來說，我們將 dta 導到 select，接下來再把 select

結果導到 head。用管道運算子寫成的程式碼，常常連成一串，如果我們把管道運算子想成水管（pipe），其實就是把水源一次次加工、導來導去。成串的程式碼最前面是資料（水源），接著會通過一連串的動作，最後則是列表、繪圖等等顯示結果的動作；如果不是這類動作，那就會是處理完的資料。在這串程式碼中的順序，就會是資料加工的順序。讀者可以由第二個 |> 開始，選取往前到源頭的程式碼執行，瞭解到該程式碼加工到哪個地步，中間的輸出為何。例如，可以選取「dta |> dplyr::select(where(is.factor))」執行，可以看到就是挑出類別變項。

本書中我們盡可能使用 |> 而非 %>%，前者是 R 基本環境即支援的管道運算子，以便跟更多套件相容。

R 程式碼 1.3

```
#選取變項方式一：利用資料型態
#程式報表1.2
dta |>
  dplyr::select(where(is.factor)) |>
  head()
```

程式報表 1.2 呈現程式碼 1.3 執行結果，選取的資料中的類別變項，並輸出前六筆。

R 程式報表 1.2

```
      國高中  性別  父親教育程度  母親教育程度
1 高中 男 國中 國中
2 高中 女 高中 國中
3 高中 男 國中 高中
4 高中 男 高中 高中
5 高中 NA NA NA
6 高中 女 小學或不識字 高中
```

程式碼 1.4 則搭配 contains 這個「副詞」，用 select 搭配條件（有「誠」或「規」的變項），再進一步算挑出變項的相關（cor），並四捨五入取到小數點第 3 位（round(3)）。結果顯示「誠實. 謙遜」與「違反規定」的相關為-0.38，與先前認為「高誠實／謙遜個體會出現較少違反規定的行為」之看法一致。

R 程式碼 1.4

```
#選取變項方式二：利用變項名稱（部份）文字
dta |>
  dplyr::select(contains(c('誠', '規'))) |>
  cor(use='pair') |>
  round(3)
```

前面說明了可以用變項是否是類別變項、或是包含某些字來選取變項，程式碼 1.5 則利用變項所在位置，指定依 11、13、12、14 順序選取變項。如果要依原順序選取，可以是 c(11:14)，但這邊希望讓相同類別變項重排在一起，因此是 c(11、13、12、14)。其中，c() 可以將裡面的元素合成一個向量。

R 程式碼 1.5

```
#選取變項方式三：利用變項位置
#程式報表1.3
dta |>
  dplyr::select(c(11, 13, 12, 14)) |>
  cor(use='pair') |>
  round(3)
```

程式報表 1.3 呈現四種青少年問題行為的相關，可以看到，同屬外化問題行為，或同屬內化問題行為的變項，相關較高，例如，同屬外化問題行為的攻擊行為與違反規定，相關為 0.696，同屬內化問題行為的焦慮. 憂鬱與社會退縮則為 0.623，都高於其他變項間的相關。

R 程式報表 1.3

```
           攻擊行為 違反規定 焦慮.憂鬱 社會退縮
攻擊行為      1.000    0.696     0.610    0.417
違反規定      0.696    1.000     0.423    0.351
焦慮.憂鬱     0.610    0.423     1.000    0.623
社會退縮      0.417    0.351     0.623    1.000
```

dplyr 的 select 用在選取變項上，slice 則用在觀察值上。程式碼 1.6 用 slice 選取 11:12 列的觀察值，並再用 select 選取 1:4 的變項。程式報表 1.4 因而顯示 2 x 4 的資料。

R 程式碼 1.6

```
#選取觀察值（slice）方式一：利用列的位置
#程式報表1.4
dta |>
  dplyr::slice(11:1) |>
  dplyr::select(1:4)
```

程式報表 1.5 為兩列，四欄的資料框。

R 程式報表 1.4

```
國高中 性別      父親教育程度 母親教育程度
高中 女 研究所以上  大學或專科
```

我們也可以用 slice_sample(n=2)，隨機選取 2 列觀察值，搭配 select(contains(c(’親’))) 選出有「親」這個字的變項。程式報表 1.5 因而顯示 3 x 2 的資料。

R 程式碼 1.7

```
#選取觀察值（slice）方式二：利用列的位置（隨機若干位）
#程式報表1.5
dta |>
  dplyr::slice_sample(n=3) |>
  dplyr::select(contains(c('親')))
```

程式報表 1.5 顯示 3 位參與者，在 2 個變項上的資料。

R 程式報表 1.5

```
父親教育程度 母親教育程度
高中 高中
高中 高中
大學或專科 大學或專科
```

程式碼 1.8 先以 with(dta, levels(性別)) 指令，觀察性別的水準，其中，with(dta) 會讓裡面的指令，限制在 dta 這個資料框中，所以 levels(性別)，就是列出 dta 這個資料框中性別這個變項的水準。接

著，slice_min(性別, n=1) 找出資料中，性別最小的 1 個觀察值；由於「女」是性別最小值，但同分數者會都選入，因此相當於挑出所有女性，再選出「誠實. 謙遜」、「違反規定」兩變項，計算相關，四捨五入至第 3 位。

R 程式碼 1.8

```
#程式報表1.6，看一下性別的兩個水準
with(dta, levels(性別))

#選取觀察值（slice）方式三：利用變項、指定最小值（如果同分數，會都選入）
dta |>
  dplyr::slice_min(性別, n=1) |>
  dplyr::select(contains(c('誠', '規'))) |>
  cor(use='pair') |>
  round(3)
```

程式報表 1.6 呈現性別的水準，依序為「女」、「男」,「女」是最小值。相關部份,「誠實. 謙遜」與「違反規定」相關為-.375。

R 程式報表 1.6

```
[1] "女" "男"
```

除了 slice 外，filter 也可以用來篩選符合條件的觀察值。程式碼 1.9 利用 filter 篩選「性別 == ’男’」的觀察值，再選出「誠實. 謙遜」、「違反規定」兩變項，計算相關，四捨五入至第 3 位。所有男性中,「誠實. 謙遜」與「違反規定」相關為 -.318，比女性相關 -.375 弱。

R 程式碼 1.9

```
#篩選觀察值（filter）方式：指定變項與條件
dta |>
  dplyr::filter(性別 == '男') |>
  dplyr::select(contains(c('誠', '規'))) |>
  cor(use='pair') |>
  round(3)
```

利用 filter 篩選觀察值，也可以設定多個條件。程式碼 1.10 篩選父親與母親教育程度皆為研究所以上者，之後僅選取性別這個變項，再利用 table 指令輸出性別這個變項不同水準之觀察值數目。結果顯示，在本筆資料中，父親與母親教育程度為研究所以上者，女性有 6 位，男性則有 8 位。

R 程式碼 1.10

```
#篩選觀察值（filter）方式：指定變項與與條件
dta |>
  dplyr::filter(母親教育程度=='研究所以上',父親教育程度=='研究所以上') |>
  dplyr::select(性別) |>
  table()
```

程式碼 1.11 與程式碼 1.10 很像，只是不篩選教育程度，因此可以看出資料中，女性與男性的數量。結果顯示，在本筆資料中，女性有 792 位，男性則有 824 位。結合前面的資訊可以知道，女性中父母皆為研究所以上者佔 0.76% (6/792)，男性則為 0.97% (8/824)。

R 程式碼 1.11

```
#對比一下整個樣本的性別人數
dta |>
 dplyr::select(性別) |>
 table()
```

篩選觀察值時，條件未必是等式，也可以是不等式。程式碼 1.12 找攻擊行為之分數在 PR90 以上，「或」違反規定分數在 PR90 以上者。程式碼 1.10 列出母親與父親教育程度，filter 會自動選取兩者皆符合者，相當於「且」，此處則將兩個條件用「｜」（或）連接。其中，quantile(攻擊行為, probs=.9, na.rm = TRUE)，會計算攻擊行為的 PR90，而遺漏值則予移除不計（na.rm = TRUE）。

結果顯示，在本筆資料中攻擊行為在 PR90 以上「或」違反規定在 PR90 以上者者，女性有 65 位，佔 8.21% (65/792)，男性則有 154 位，佔 18.69% (154/824)。男性在外化問題行為比率較女性高，與預期一致。

R 程式碼 1.12

```
#篩選觀察值（filter）方式：指定變項與與條件
dta |>
  dplyr::filter(攻擊行為 > quantile(攻擊行為, probs=.9, na.rm = TRUE)|
                違反規定 > quantile(違反規定, probs=.9, na.rm = TRUE)) |>
  dplyr::select(性別) |>
  table()
```

前面利用 dplyr 選擇變項或是觀察值，底下則是利用 mutate 對資料加工。mutate 可以利用舊變項轉換成為新變項，也可以修改

變項（當變項名稱相同時即為修改），或是刪除變項（設定變項的值為 NULL）。

程式碼 1.13 先呈現母親教育程度的各水準，發現並未按照順序排，因此利用 mutate 指令，搭配 forcats 套件的 fct_relevel，對類別變項的水準做重新分派。程式製造了「母教育」這個變項，它是「母親教育程度」這個變項重新分派水準後的新變項，水準由小學或不識字、國中、高中、大學或專科到研究所以上排列。同理，也製造了「父變項」這個新變項。製作後，將兩個新變項導入 table 計算觀察值數量。table 這個指令，當收到一個變項時，則計算單一變項的觀察值數量，但若收到兩個以上變項，則計算二維或多維的列聯表。

R 程式碼 1.13

```
#類別變項水準預設排列順序對應編碼大小，常常需要重排
#程式報表1.7前
with(dta, levels(母親教育程度))

#變項轉換一：製造新變項
#程式報表1.7後
dta |>
  dplyr::mutate(母教育 = forcats::fct_relevel(母親教育程度,
                   c("小學或不識字","國中","高中","大學或專科", "研究所以上")),
                父教育 = forcats::fct_relevel(父親教育程度,
                   c("小學或不識字", "國中", "高中","大學或專科", "研究所以上")),
                .keep='none') |>
  table()
```

請特別注意，資料 dta 一路以管道運算子導到 table，並不會

將額外製造的變項或更動的變項更新至 dta，讀者可以檢視 dta，即可以知道資料並未被更動。此一設計可以保護原始資料，而如果需要更新資料，需要回指到資料，後面有程式碼示範如何進行。

程式報表 1.7 前半部呈現母親教育程度的各水準，發現並未按照順序排，後續則利用 mutate 製造「母教育」與「父教育」兩個新變項，並輸出列聯表。在列聯表中，可以看到不同教育程度的母親，對應哪些教育程度的父親。例如，當母親教育程度為高中時，有 475 位父親教育程度同是高中，其次是 147 位為國中，再其次則是 109 位為大學或專科。

R 程式報表 1.7

```
[1] "大學或專科"   "小學或不識字" "研究所以上"   "高中"
[5] "國中"

                  父教育
母教育        小學或不識字 國中 高中 大學或專科 研究所以上
  小學或不識字 18   33   18   2    0
  國中 21   181 106   20    0
  高中 17 147 475 109    5
  大學或專科  2   15   60 245   31
  研究所以上  0    0    2    7   15
```

mutate 也可以修改資料框內現有變項，程式碼 1.14，利用 mutate 加工母親教育程度，加工後仍為同名變項。程式碼 1.14 在連串的管道運算子指令前，有「dta <-」，將變動後資料指回 dta，因此資料將被更新。讀者可以檢視 dta，即可以知道資料已被更動。

R 程式碼 1.14

```
#變項轉換二：修改舊變項
dta <- dta |>
  dplyr::mutate(母親教育程度 = forcats::fct_relevel(母親教育程度,
                   c("小學或不識字","國中","高中","大學或專科","研究所以上")),
                 父親教育程度 = forcats::fct_relevel(父親教育程度,
                   c("小學或不識字", "國中","高中","大學或專科","研究所以上")))
```

程式碼 1.15，示範如何選出父母教育程度相同的觀察值，並製表。首先先選出包含「親」字的變項（即「母親教育程度」與「父親教育程度」），接下來說明要逐列比對（rowwise），並將比對兩者相同與否的結果存入 match 這個變項，再選擇 match 內數值為「TRUE」者（亦即比對結果相同），最後以 ftable 製表。

R 程式碼 1.15

```
#比對父母教育程度是否相同，製造新變項
#程式報表1.8
dta |>
  dplyr::select(contains('親')) |>
  dplyr::rowwise() |>
  dplyr::mutate(match = (母親教育程度 == 父親教育程度)) |>
  dplyr::filter(match == TRUE) |>
  ftable()
```

程式報表 1.8 呈現結果的一部份。請先注意，我們將資料篩選出父親與母親教育程度教育程度相同者（filter(match == TRUE)），所以不含教育程度不相同者。表中，我們看到父親教育程度與母親教育程度，同是小學或不識字者，有 18 名，同是國中者，有

181 名。但父親與母親教育程度不同的組合，就如同前面所言，都是 0 了。

R 程式報表 1.8

```
                              match TRUE
父親教育程度 母親教育程度
小學或不識字 小學或不識字  18
             國中    0
             高中    0
             大學或專科    0
             研究所以上    0
國中         小學或不識字    0
             國中 181
```

程式碼 1.16，則利用連續變項製造出新的類別變項，方便做列聯表。程式碼 1.16 選出「誠實.謙遜」與「違反規定」兩個變項，再以 PR33 與 PR67 當切點，製造出低中高三組的新變項「誠謙」與「違規」。執行到這裡，資料檔應該包含選出的兩個選出的舊變項，與兩個以舊變項製造的新變項。加上「.keep='unused'」，我們告訴 R，僅保留舊變項中沒使用的，亦即，被使用的舊變項就不保留了，因此，後面作表，就只出現「誠謙」與「違規」兩個新變項。

R 程式碼 1.16

```
#可以利用連續變項製造出新的類別變項，方便做列聯表
#程式報表1.9
dta |>
 dplyr::select(誠實.謙遜, 違反規定) |>
 mutate(誠謙 = case_when(誠實.謙遜 < quantile(誠實.謙遜, probs=.33,na.rm = TRUE)
```

```
                            ~ '1_低',
                          誠實.謙遜 > quantile(誠實.謙遜, probs=.67,na.rm = TRUE)
                            ~ '3_高',
                          .default = '2_中'),
       違規 = case_when(違反規定 < quantile(違反規定, probs=.33,na.rm = TRUE)
                            ~ '1_低',
                          違反規定 > quantile(違反規定, probs=.67,na.rm = TRUE)
                            ~ '3_高',
                          .default = '2_中'),
    .keep='unused') |>
table()
```

程式報表 1.9 呈現結果。可以發現，誠謙低分組（誠實.謙遜分數低於 PR33）中，以違規高分為多（違反規定分數高於 PR67）；誠謙高分組中，則以違規低分為多。此一發現支持了研究者關於「誠實／謙遜可能使得個體表現出較少違反規定的行為」的看法。

R 程式報表 1.9

```
        違規
誠謙       1_低 2_中 3_高
  1_低   74  194  265
  2_中  168  293  161
  3_高  189  213   73
```

我們常常需要將資料分組做同樣的運算，這可以用 group_by 作到。程式碼 1.17 幾乎跟程式碼 1.16 相同，但更動了變項，另外比較特別的就是增加 group_by(性別)，這是修飾「動詞」（底下動作）的「副詞」（修飾底下動作），底下的動作都會區分不同性別水準跑。

另外與程式碼 1.16 不同的，是在做完資料處理後，先將資料

導到 tmp 這個資料框，再另外將 tmp 導去製造表格。這樣做的原因，是因為 tmp 這筆資料後面還會利用，所以需要輸出為資料框。

R 程式碼 1.17

```
#分組，對各組做相同動作
tmp <- dta |>
  dplyr::select(性別, 攻擊行為, 違反規定) |>
  dplyr::group_by(性別) |>
  dplyr::mutate(攻擊 = case_when(攻擊行為 < quantile(攻擊行為, prob=.33,
                                 na.rm = TRUE) ~ '1_低',
                           攻擊行為 > quantile(攻擊行為, prob=.67,
                                 na.rm = TRUE) ~ '3_高',
                           .default = '2_中'),
                違規 = case_when(違反規定 < quantile(違反規定, prob=.33,
                                 na.rm = TRUE) ~ '1_低',
                            違反規定 > quantile(違反規定, prob=.67,
                                 na.rm = TRUE) ~ '3_高',
                           .default = '2_中'), .keep='unused') |> as.data.frame()

#程式報表1.10，剛剛資料作列聯表
tmp |> ftable()
```

程式報表 1.10 呈現與程式報表 1.9 類似的表格，但區分為男女。我們可以看到，在男性資料中，違規的低中高組，對應人數最多的也依序是攻擊的低中高組，顯示兩者關係密切；相較於此，女生資料就沒有那麼整齊，攻擊低的組中，違規中的人數較多。違反規定、攻擊行為屬於外化問題行為, 男性較容易發生，可能因而相關較高。

R 程式報表 1.10

```
                違規 1_低 2_中 3_高
性別 攻擊
女   1_低        81  102   14
     2_中        30  237   72
     3_高         7   86  163
男   1_低       165   52   12
     2_中        72  199   81
     3_高        17   75  151
```

為了確認攻擊與違規的相關，在不同性別是否不同，我們將剛剛的資料（tmp）挑出女性，再挑出攻擊、違規兩變項，做出列聯表後，以 sjstats 套件的 xta_statistics 求出適用於類別資料的 phi 相關。結果顯示，卡方值為 293.58，自由度為 4，$p < .001$，卡方檢定顯著，攻擊與違規兩變項並非獨立，其間的 phi 相關是 0.6088。我們也同步分析了男性，phi 相關是 0.6683，確實高於女性。

R 程式碼 1.18

```
#利用 filter 挑出女生樣本，計算攻擊與違規的相關（phi）
tmp |>
  dplyr::filter(性別=='女') |>
  dplyr::select(攻擊, 違規) |>
  table() |>
  sjstats::xtab_statistics(statistics=c('phi'))
```

程式碼 1.19，利用 reframe 這個指令，可將動作後結果存成資料框，讓後續利用更方便。

前面我們是對類別變項求 phi 相關，這邊則是對連續變項求常見的 Pearson 相關，也同時記錄樣本數（n()）。

R 程式碼 1.19

```
#合計：算相關
#程式報表1.17
dta |>
  dplyr::select(性別, 攻擊行為, 違反規定) |>
  dplyr::group_by(性別) |>
  dplyr::reframe(相關係數 = cor(攻擊行為, 違反規定, use='pair'),合計 = n())
```

程式報表 1.11 其實是個資料框，看起來比較整齊。性別有三個水準，除女、男外，還有未填寫者，三個相關分別是 0.6711、0.7097 與 0.6960。男生相關略高於女性。

報表中也顯示了樣本數，女性有 792，確實就是程式報表 1.16 顯示的。

R 程式報表 1.11

```
 性別   相關係數   合計
女 0.6711 792
男 0.7097 824
NA 0.6960  14
```

過去研究發現，違反規定屬於外化問題行為, 男性則較容易發生。程式碼 1.20，再次利用 group_by 搭配 reframe 這個指令，以性別分組計算平均數、標準差、中位數、四分位距與人數。

R 程式碼 1.20

```
#合計：算基本統計量
#程式報表1.12
dta |>
  dplyr::group_by(性別) |>
  dplyr::reframe(違規平均 = mean(違反規定, na.rm = TRUE),
                 違規標準差 = sd(違反規定, na.rm = TRUE),
```

```
                違規中位數 = median(違反規定, na.rm=T),
                違規四分位距 = IQR(違反規定, na.rm=T),
                合計 = n())
```

程式報表 1.12 也是個資料框。可以看到，男性平均數、中位數都高於女性，在表示變異的標準差與四分位距上，也高於女性，顯示男性違規行為較女性多，組內變異也比較大。

R 程式報表 1.12

```
 性別 違規平均 違規標準差 違規中位數 違規四分位距  合計
女 2.863 2.570 2.0    3.00 792
男 4.335 3.211 4.0    4.00 824
NA 5.214 4.475 4.5    4.75  14
```

在前面示範了篩選、轉換變項、分組、合計後，程式碼 1.21 示範如何綜合這些步驟。程式碼 1.21 算出國中生中，不同性別參與者攻擊行為分數偏高（高於平均數 1.96 個標準差）的人數。為了讓表格更易讀，僅保留性別與攻擊並非遺漏的觀察值。結果可以看到，國中生中，女性未偏高者有 294 位，9 位偏高，男性未偏高者有 269 位，16 位偏高。

R 程式碼 1.21

```
#綜合動作：篩選、轉換、分組、合計
dta |>
 dplyr::filter(國高中 == '國中') |>
 dplyr::mutate(攻擊 = 攻擊行為 > (mean(攻擊行為, na.rm=T)
                        + 1.96*sd(攻擊行為, na.rm=T))) |>
 dplyr::group_by(攻擊, 性別) |>
 dplyr::reframe(合計 = n()) |>
 dplyr::filter(攻擊 != 'NA', 性別 != 'NA')
```

二、資料格式轉換

本節展示如何在寬格式與長格式之間轉換。在寬資料中，每個觀察值包含多個變項，以本章來說，有四個背景變項，六個人格分數，四個問題行為分數。有時，我們需要把攻擊行為、違反規定、焦慮.憂鬱、社會退縮分數（四個變項），想成是問題行為分數（單變項），而不同問題行為變成問題行為的四個水準。這樣做，有時是為了在統計上將多變項的問題轉為單變項問題；有時則是將類似變項轉為一個變項的不同水準後，更容易做資料處理或繪圖，比較彼此。

為了示範方便，程式碼 1.22 先選出問題行為的 4 個變項、2 筆觀察值，總共 8 個數值，並製造出識別碼。在寬資料轉成長資料時，最好每筆觀察值都對應單一的識別碼，比較有辦法確認轉換後的長格式資料是否轉換正確。

R 程式碼 1.22

```
#程式報表1.13前
dta |>
  dplyr::select(11:14) |>
  dplyr::slice(1:2) |>
  as_tibble(rownames="識別碼")
```

程式碼 1.23 示範寬格式轉成長格式。程式碼 1.23 也先選出跟前面相同的 4 個變項、2 筆觀察值，製造識別碼，再利用 tidy 套件

中的 pivot_longer 功能，將資料轉成長形。pivot_longer 需要的參數中，cols 用來設定需要寬轉長的變項，預設是所有變項，此處設定「-識別碼」表示除了識別碼外的所有變項，都要寬轉長，由於前面選了四個問題行為變項，這邊即意指這四個變項。寬轉長後，無論原先有幾個變項，都會對應兩個變項，一個用來對應先前變項名稱（names _to），一個對應先前變項數值（也可能是類別，values_to），我們需要在 pivot_longer 設定兩個變項名稱，在此，我們分別設定為「問題行為」以及「分數」。

R 程式碼 1.23

```
#程式報表1.13後
dta |> dplyr::select(攻擊行為:社會退縮) |>
  dplyr::slice(1:2) |>
  as_tibble(rownames="識別碼") |>
  tidyr::pivot_longer(cols = -識別碼, names_to = '行為問題', values_to = '分數')
```

程式報表 1.13 前半中可以看到，資料包含兩筆參與者資料，每一列對應一個參與者（亦對應一個識別碼）。識別碼 1 的參與者，在攻擊行為、焦慮. 憂鬱、違反規定、社會退縮上分別得分是 13、11、7、9。在程式報表 1.20 後半可以看到，前四列的識別碼都是 1，表示此為第一位參與者的資料，行為問題這個變項，則有攻擊行為、焦慮. 憂鬱、違反規定、社會退縮四個水準，在行為問題不同水準下，對應的分數則是 13、11、7、9。由此可以看到，

資料由寬形式換成長形式，資訊保持相同，並沒有增加或減少資訊，而只是同樣資訊的兩種表達形式。

請注意，在寬資料中，一列對應一個參與者（亦對應一個識別碼），但在長資料中，則是若干列對應一個參與者，這些列有共同的識別碼。以程式報表 1.20 來說，前半部一列對應一位參與者，後半部的一列對應一個「參與者-行為問題水準」。

R 程式報表 1.13

```
識別碼 攻擊行為 焦慮.憂鬱 違反規定 社會退縮
1 13 11 7 9
2 2 2 1 4

識別碼    行為問題       分數
1 攻擊行為 13
1 焦慮.憂鬱 11
1 違反規定  7
1 社會退縮  9
2 攻擊行為  2
2 焦慮.憂鬱  2
2 違反規定  1
2 社會退縮  4
```

前面提到，寬格式轉長格式，有時有利於整理資料與比較。程式碼 1.24，即利用長資料中四個行為問題變為四個水準，因而可以用來分類，而有助於資料處理。

程式碼 1.24，先由寬資料中，挑選性別與四個行為問題，製造識別碼，然後轉成長格式。轉成長格式後，行為問題與性別就都是資料的類別變項，可以用來分組（group _by(行為問題, 性別)）

計算平均數，再利用 arrange 指令，依性別排序，並排除性別不明者資料。

R 程式碼 1.24

```
#利用 arrange指令，依性別排序
#程式報表1.14
dta |> dplyr::select(2, 11:14) |>
  as_tibble(rownames="識別碼") |>
  tidyr::pivot_longer(cols = -c(識別碼, 性別),
                      names_to = '行為問題', values_to = '分數') |>
  dplyr::group_by(行為問題, 性別) |>
  dplyr::reframe(平均分數 = mean(分數, na.rm=T)) |>
  dplyr::arrange(性別) |>
  dplyr::filter(性別 != 'NA')
```

程式報表 1.14 先呈現女性在四個行為問題的平均，然後是男性的四個行為問題的平均。可以看到，女性在問題行為分數，較高兩者為攻擊行為、焦慮. 憂鬱，都是 7 分以上，社會退縮則是 4.58 分，違反規定不到 3 分。男性順序也是如此，但攻擊行為分數高於焦慮. 憂鬱，社會退縮與違反規定都較接近。

在其他章節，讀者可以發現用其他套件（例如，gtsummary）可以更快速地得到這張表。不過，這邊一方面是示範寬格式變長格式，另一方面，使用較複雜而設計好的套件功能，當不完全符和需求時，會受限於原先設計而較沒有彈性、不容易更動，這時使用 dplyr 與 tidyr 可能更為方便。讀者可依需求，適當地選用兩種作法。

R 程式報表 1.14

```
行為問題 性別    平均分數
攻擊行為   女 7.991
焦慮.憂鬱 女 7.014
社會退縮   女 4.589
違反規定   女 2.863
攻擊行為   男 9.166
焦慮.憂鬱 男 6.128
社會退縮   男 4.545
違反規定   男 4.335
```

程式碼 1.25，則示範將程式報表 1.14（長格式資料），轉為寬格式資料。程式碼 1.25 前段與程式碼 1.24 完全相同，但在後面利用 pivot_wider，將長格式轉為寬格式。pivot_wider 中，則需要設定 names_from 與 values_from。

R 程式碼 1.25

```
#綜合應用，最後面將資料由長形變成寬形
#程式報表1.15
dta |> dplyr::select(2, c(11:14)) |>
  as_tibble(rownames="id") |>
  tidyr::pivot_longer(cols = -c(id, 性別),
              names_to = '行為問題', values_to = '分數') |>
  dplyr::group_by(行為問題, 性別) |>
  dplyr::reframe(平均分數 = mean(分數, na.rm=T)) |>
  dplyr::filter(性別 != 'NA') |>
  tidyr::pivot_wider(names_from = '行為問題',values_from = '平均分數')
```

程式報表 1.15 與 1.14 數據相同，但轉為寬格式。在寬格式中，男女在不同問題行為的差異很容易看出，我們可以看到，在攻擊行為與違反規定兩個外化行為問題上，男性都比女性高 1 分以上，而在焦慮. 憂鬱，女性高近 1 分，在社會退縮，則幾乎相同。

R 程式報表 1.15

```
性別 攻擊行為 焦慮.憂鬱 社會退縮 違反規定
女 7.991 7.014 4.589 2.863
男 9.166 6.128 4.545 4.335
```

違規行為屬於外化問題行為, 男性分數應該比較高。為檢驗這個想法，程式碼 1.26 計算攻擊行為高分組（高於 PR90）與其餘組中，女性的比率。程式碼 1.26，先以全體資料為參照，計算違反規定是否高於 PR90（違規九變項），再以違規九與性別分組，計算人數（reframe(合計 = n()) ）。讀者可以先執行至此，看到資料會是長格式，較不方便計算女性所佔比率，因此，再將資料轉為寬格式，計算女性百分率，最後篩除無法計算違規九分位的資料（原先違規行為即遺漏者）。

結果顯示，非屬違規行為高於 PR90 的參與者中，女性佔 51.36%，接近五成，但在違規行為高於 PR90 的參與者中，女性則僅佔 27.39%，顯示違規高分組中，女性比率較低，違規高分組較多為男性。

R 程式碼 1.26

```
#綜合應用，計算以違反規定PR90分類學生時，女生所佔各類比率
dta |>
 dplyr::mutate(違規九 = 違反規定 > quantile(違反規定, prob = 0.9, na.rm=T)) |>
 dplyr::group_by(違規九, 性別) |>
 dplyr::reframe(合計 = n()) |>
 pivot_wider(names_from=性別,
             values_from=合計) |>
 dplyr::mutate(違規九分位 = 違規九,
               女性百分率 = 100*(女 / (女+男)), .keep='none')  |>
```

```
dplyr::filter(違規九分位 != 'NA')
```

看完性別的可能影響，我們接著多考慮母親教育程度，在不同母親教育程度、性別下，計算內化問題的平均分數。

程式碼 1.27 先把有關的變項選入（包含性別, 母親教育程度, 社會退縮, 焦慮. 憂鬱四個變項），再利用 pivot_wider 將性別與內化問題轉為寬格式時，一併計算平均數，並刪除母親教育程度遺漏者，以及轉為寬格式時，性別不明產生的變項，並依社會退縮 _ 女（女性在社會退縮平均數）排序。

前面我們長轉寬時，通常都是選取類似的變項，此處是名字來自「性別」，數值來自內化物題，請特別注意程式報表 1.24，看看達成何種效果。

R 程式碼 1.27

```
#綜合應用，不同母親教育程度、性別下，內化問題的平均
#程式報表1.16
dta |>
  dplyr::select(性別, 母親教育程度, 社會退縮, 焦慮.憂鬱) |>
  tidyr::pivot_wider(names_from = '性別',
                     values_from = c('社會退縮', '焦慮.憂鬱'),
                     values_fn = ~ mean(.x, na.rm = TRUE)) |>
  dplyr::filter(母親教育程度 != 'NA') |>
  dplyr::select(-contains('NA')) |>
  dplyr::arrange(社會退縮_女)
```

程式報表 1.16 中，我們可以看到母親教育程度正好按順序排

列，表示不同母親教育程度參與者的社會退縮_女的平均分數，正好與母親教育程度一致。

觀察程式報表 1.16，先看社會退縮，當母親為小學會不識字時，女性社會退縮高於男性，在國中與高中，性別差異不大，但在大學或專科以及研究所以上，男性分數則高於女性。再看焦慮.憂鬱，雖然女性都高於男性，但當母親為小學會不識字時，差異為 1.440，而在國中與高中，下降為.633 與.989，在大學或專科以及研究所以上，再下降為.245 與.100。我們可以發現，內化問題的性別差異，可能隨著母親教育程度而變。

R 程式報表 1.16

```
母親教育程度 社會退縮_男 社會退縮_女   焦慮.憂鬱_男 焦慮.憂鬱_女
研究所以上 4.846   3.800         7.000 7.100
大學或專科 4.611   4.390    6.378 6.623
高中          4.486 4.509    5.942  6.931
國中          4.680 4.647    6.189  6.822
小學或不識字 4.938 5.575    7.613  9.053
```

由前面分析可以看到，本筆資料包含遺漏值，這讓程式碼有時會需要設定排除遺漏值等等，讓程式碼變的比較複雜。雖然如此，由於資料通常都包含遺漏值，程式碼的複雜應屬必要之惡。R 裡面有很多套件提供處理遺漏值的方法，有興趣讀者可以適當查詢。本書不打算著力于此，而僅是利用資料處理，瞭解遺漏情形。

程式碼 1.28 前半，利用本章技巧，示範如何計算每個變項遺

漏值個數，以初步瞭解變項遺漏情形。程式利用 summarise_all 搭配 sum(is.na(.)) 計算每個變項遺漏的個數。

程式碼 1.28 後半計算遺漏題數的比例。R 基本指令 apply 非常方便，可以將一個函數一次用到多個物件中。apply 至少需要設定三個參數，分別是資料框，維度以及函數。資料框可以想成二維的，第一維是橫列，第二維是縱行。當我們將 apply 的維度設定成 1，表示要逐列的應用函數，將 apply 的維度設定成 2，表示要逐行的應用函數。例如，假設 dta 記錄班上同學各個考科的成績，apply(dta,2,mean)，即是要求 R 針對 dta 逐欄計算平均，亦即班上同學在每個考科的平均；apply(dta,2,sum)，即是要求 R 針對 dta 逐列計算總和，亦即每個同學的考科成績總和。

程式碼 1.28 要求 R 針對資料框 dta 逐列地應用函數，而此函數是一個自訂函數。此處的 function(x) 是一個自訂函數，將輸入的 x，逐個看看是否遺漏（is.na(x)），再將是否遺漏（遺漏為 1，未遺漏為 0）加起來。由於是逐列做，apply 一次針對某個參與者的所有變項，因而計算出此位參與者在所有變項上的遺漏題數。接著將遺漏題數依次數製表（table），計算比率（proportions），然後四捨五入到第三位。程式結果顯示，八成五的觀察值完全沒有遺漏，遺漏一題到四題的參與者佔 9.3%、4.5%、1.0%、0.2%。沒有

遺漏五題以上者。

R 程式碼 1.28

```
#程式報表1.25，計算每個變項遺漏值個數
dta |>
  dplyr::summarise_all(~ list(sum(is.na(.)))) |>
  unlist()

#計算遺漏題數之比率
apply(dta, 1, function(x) is.na(x) |> sum()) |>
  table() |>
  proportions() |>
  round(3)
```

程式報表 1.25 可以看到，除了國高中這個變項，其他變項都有遺漏，特別是父親教育程度與母親教育程度。六個性格向度的遺漏值較少，但問題行為的遺漏值較多，可以與回答問題行為較為敏感有關。

R 程式報表 1.25

```
   國高中          性別 父親教育程度 母親教育程度     誠實.謙遜
      0            14         84           84             6
     情緒性        外向性        和悅性        嚴謹性        開放性
      5             6             5             8            11
  攻擊行為     焦慮.憂鬱      違反規定      社會退縮
     41            48            26            21
```

本章小結

進行統計分析前，宜先瞭解資料。本章說明如何利用 dplyr 選取變項或觀察值、轉換舊變項、製造新變項、分組執行，以及利

用 tidyr 做寬格式與長格式間的轉換。透過結合兩者，我們可以探索資料，並將資料轉換為統計程序或繪圖需要的格式。

習題

1、高中成績。

套件 faraway 中的資料 hsb 記錄了 200 名高中生的性別、族裔、閱讀、寫作、數學、科學與社會科學分數。

(1) 請以 p_load(faraway) 載入套件，並執行 data(hsb) 指令，以及 dta <- hsb 指令。利用?hsb 指令，看看資料的描述並摘錄。

(2) 請計算不同性別參與者，在五個學科分數的平均數與標準差。

(3) 請計算不同性別參與者中，閱讀、寫作、數學與社會科學的相關係數。

(4) 請利用 PR67 與 PR33，將閱讀與數學各分為低中高三組，依性別做出列聯表。

(5) 請計算閱讀能力較高者（高於 PR90）與其餘參與者中，女性所佔的比率。

(6) 請仿照程式報表 1.24，計算不同性別與族裔下，科學與社會科學分數的平均數。

2、生活品質。

資料檔 Quality_of_Life.csv 包含性別、年齡層、教育程度三個背景變項，以及視力困難、聽力困難、移動困難與溝通困難四個與老人生活品質有關的分數。本筆資料來自於美國國民健康調查（National Health Interview Survey, NHIS），其中 2010 年資料 55 歲以上參與者。

(1) 請讀進資料檔，說明樣本數，以及三個背景變項的水準。

(2) 請計算不同性別參與者，在四個生活品質分數以及總分的平均數與標準差。

(3) 請利用 PR50，將聽力困難與溝通困難各分為低高兩組，依性別做出列聯表。

(4) 請計算女性樣本中，聽力困難與溝通困難的相關係數。

(5) 請計算移動困難較嚴重者（高於 PR90）與其餘參與者中，女性所佔的比率。

(7) 請計算不同性別與教育程度下，四個生活品質分數的平均數。

參考文獻

許功餘、張玉鈴（2015）。費力控制與 HEXACO 性格向度對青少年內化與外化問題行為之預測效果。中華心理學刊, 57(1), 1-25.