**R語言集訓班 Quick R Introduction**

**Week 01: R language 101**

**第一周課程目標**

上課時間：18:30~21:30，地點：資訊大樓四樓微五電腦教室（140406）

|  |  |
| --- | --- |
| 課程 | 預計時間 |
| * 建立R 的使用環境 * 熟悉R 語言基礎操作 * 了解R 語言的物件結構 | 1小時 |
| * 操作R 語言進行資料整理 * 運用R語言做文字處理(基礎) | 1小時 |
| * 集結R社群─使用套件 * 執行R 語言做資料的讀寫 | 1小時 |

**第一周教學大綱**

零、為何要學R語言/Chapter 0: Why R?

(一) 起源與使用社群

(二) 優點

(三) 缺點

(四) 目前發展趨勢

壹、R語言入門須知/Chapter 1: An Introduction to R

(一) 從入門到精熟

(二) 環境安裝

(三) Rstudio介面說明

(四) Rstudio編寫環境說明

1. R語言的數據型態與基本操作
2. 註釋語法
3. 向量：數值向量(numeric）、整數向量(integer）、文字向量(character）、邏輯值向量(logical）
4. 向量取值、取代、新增
5. 數列操作seq與運算
6. 求助、說明文件相關
7. 其他數據型態：因子factor、列表list、矩陣matrix、資料框data.frame
8. 常用函數一print、rep、rm

參、R語言內建常用的函式

(一) 確認型態type、class、str、is.XXX系列

(二) 型態轉換as.XXX系列

(三) 基本統計函數min、max、sum、var、sd

(四) 常用函數二table、prop.table

肆、使用R整理數據(基礎)

(一) 使用R語言內部資料集─鳶尾花資料集iris

(二) 表格的取值─座標

(三) 表格的取值─列、欄

(四) 表格的取值─條件篩選

【動動手時間─鳶尾花資料集iris】

伍、R語言做文字處理(基礎)

(一) 字串處理paste、nchar、substring、strsplit

(二) 關鍵字的取代與找查gsub、grep、grepl、gregecpr

【動動手時間─信箱帳號】

陸、套件安裝

(一) 下載套件

(二) 載入套件

(三) 第一次使用套件

【動動手時間─套件使用】

柒、資料讀取與匯出(英文資料)

(一) 工作目錄設定

(二) 資料集介紹

(三) 資料讀取

(四) 資料匯出

(五) 讀取方式百百種─條條大路通羅馬

【動動手時間─歌詞讀取】

**附錄**

壹、關於講師

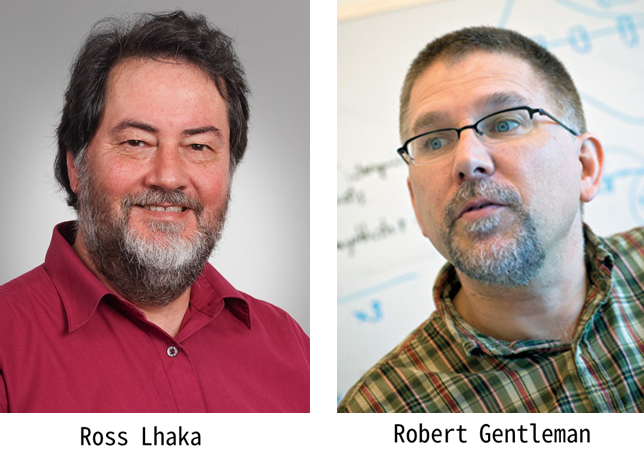
貳、引用的參考資料

參、推薦自學網站、社群、社團

**零、為何要學R語言Why R?**

~~因為講師比較會R(笑~)~~

1. 起源與使用社群

一切起源於1976年的Bell實驗室，為了節省使用SAS和SPSS軟體的經費，而開發了S語言(R語言的前身)，主要是處理向量(Vector)、矩陣(Matrix)、陣列組(Array)以及進行圖表和統計分析。

後來1993年，兩位統計學家Ross Lhaka和Robert Gentleman，為了方便教授統計學，以S語言為基礎，合作開發完成出一個新的程式語言；因為兩位教授的名字皆為R開頭，所以此語言就稱為R語言。

**R語言是開源的!!**意思是不但每個人都可以免費使用，大家也可以針對一些特殊的需求，編寫出特定的函式，分享到平台上(開源到社群)，為這個語言貢獻一份心力。

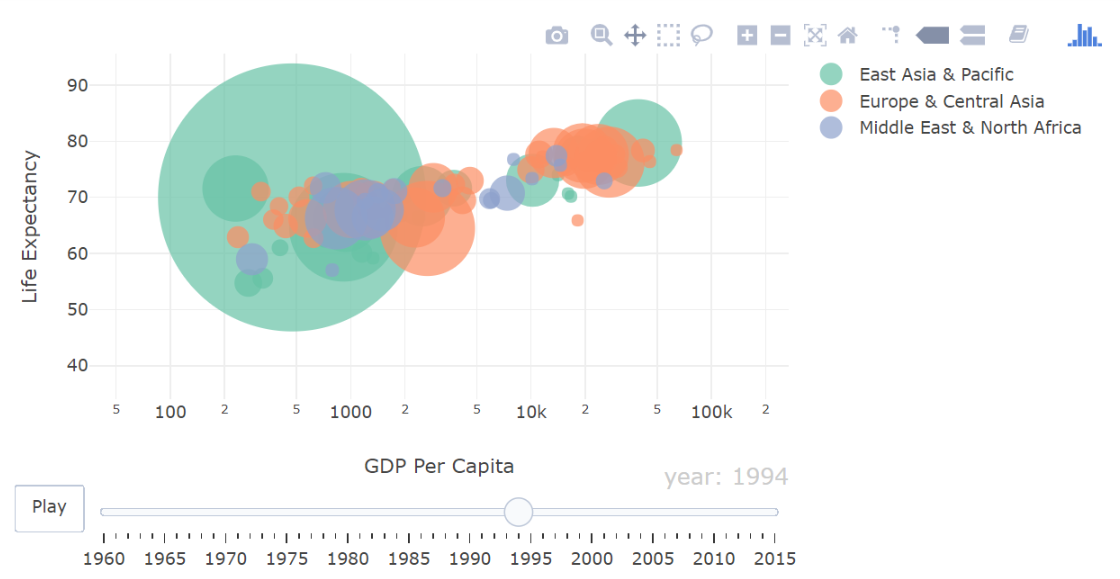
R語言的社群主要為統計相關領域。除了一般我們傳統上的統計分析、推論之外，目前R語言也在資料科學的領域蓬勃發展。

1. 優點

想要進行數據探索與分析，許多語言如R、Python、Java、C++、SQL、Julia等等都可以做到(先不論實用與否)，畢竟操作數據本來就是許多程式語言的基本功能，所以是什麼讓R語言異軍突起呢？以下列出講師覺得R語言在使用上四個特色之處：

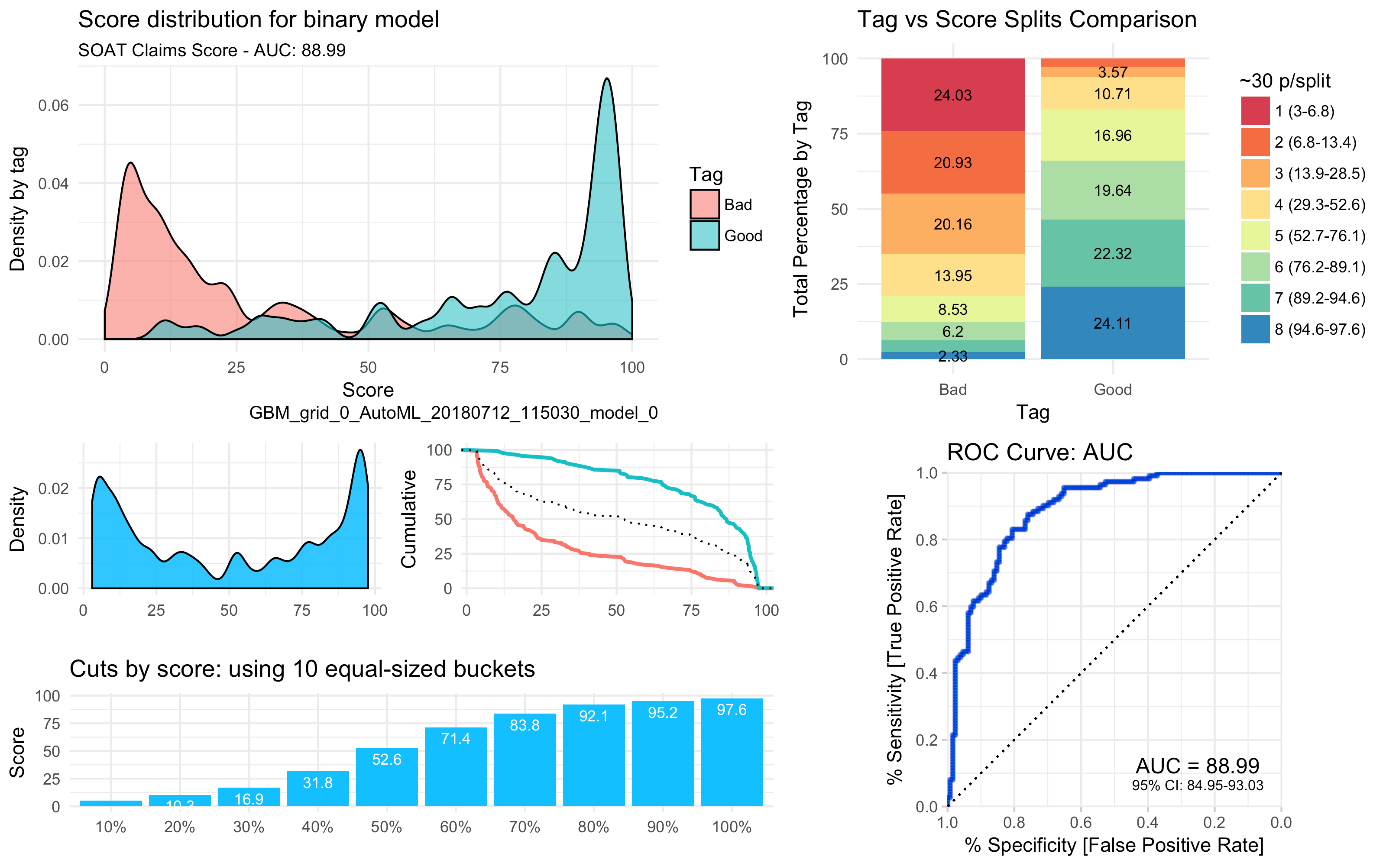
1. 物美價廉─視覺化的顏值高，且完全免費

在數據探索與分析結束後，可以迅速的視覺化是R語言的一大優點。包含了各種資料型態的圖形呈現，除了例如傳統的直方圖、散佈圖、泡泡圖等等，也涵蓋了地理圖資的呈現；另一方面，視覺化設計的自由度很高，亦可讓使用者與視覺化結果產生互動。正是易於操作與多元呈現的設計，成為R語言的一大特色。



圖一、GDP國內生產總值vs預期壽命，泡泡圖，模仿Hans Rosling的數據視覺化(註一)

圖片連結：<https://yaojenkuo.shinyapps.io/gapminder_replica/>



圖二、Machine Learning Results in R: one plot to rule them all!(註二)

Shiny互動式介面：

範例01：<https://littlefish0331.shinyapps.io/106354005_hw3/>

範例02：<https://littlefish0331.shinyapps.io/honey_ClassGroup_random/>

範例03：<https://en9515.shinyapps.io/PUBLISH/>

1. 兼容並蓄─演算法覆蓋廣，擴展易

作為學術界發起的統計分析工具，它幾乎覆蓋整個統計領域的前沿技術。甚至國外許多研究，在論文完成的時就會將研究結果製作成一個套件，一同發表。

因此，不管是熱門的神經網絡，或是經典了一百多年還熠熠生輝的基礎迴歸，都可以在R語言上面找到相對應的套件做使用。截至2018.11為止，R的開源社群已經有13398個套件(package)，在Github上面未發布的更多，所以幾乎你想對數據做的事，都可以找到能直接調用的演算法與函數做實現。

1. 軟體擴展性易

該點要特別注意，這裡的擴展性指的是數據處理端(擷取、轉換和載入，即為extract、transform、load，ETL)的擴展性。

如果數據是放在資料庫裡，沒問題，R語言有套件可以輕鬆擷取；不同類型的資料也能做轉換，例如文字、音樂等等；更可以在不同語言之間調節載出與載入，例如Hapdoop、H2O、Python等等。

1. 強大的社群支持─集眾人力量於一身

面對大數據的時代，資料科學的發展是來自各種面向的努力才得以達成，不管是程式語言的編碼、演算能力，統計學家的數據直覺、模型配飾能力，抑或是專業領域的分析技巧；都要彼此互相尊重、合作，才能在大數據的浪潮下存活。

常見的社群有GitHub、Stack overflow、PPT、Reddit、Google、R語言論壇、統計之都等等。

1. 缺點

沒有完美的程式語言，有優點必有缺點。R語言有幾個為人詬病的缺點，或者是說在眾多程式語言中，無法突出的幾項表現。

1. 速度：<https://github.com/GiovineItalia/Gadfly.jl/issues/110>

R語言的速度在程式語言中並非前段班，這一點是大家都認可的缺點。

當然處理速度除了軟體底層的設計外，也與硬體設備有關。依照目前講師經驗，處理過最大的數據約為90GB，R語言還尚可使用，但實際上，在超過20GB後便可感覺到電腦卡卡，計算出現稍微延遲。

整題來講，R語言是可以處理龐大資料的，也許在機器學習應用上，R語言表現不佳，但是，畢竟R語言是從統計領域開發，而統計擅長的是對大數據趨勢做出評估與預測，也就是大資料化簡為繁成小資料呈現的動作，這部分仍然表現優異。

但這也引發另一個問題，真的會處理到這麼龐大的數據嗎？就像如果市面上沒有售價兩千元的商品，有必要發行面額兩千元的貨幣嗎？又或者該思考的是，針對這個大數據時代，我們是否該進化，開發下一階段符合需求的硬體與軟體了。

1. 領域特定

R 語言在統計和數據科學等領域如魚得水，但是在程式語言的其他應用上就有些力不從心了，例如嵌入式系統、網頁、App、遊戲應用等等。等等~那其他程式語言不也這樣嗎？當然如此，只是其他前段班的程式語言比其R語言，在功能上更包山包海。

1. 既直覺又不直覺的語法

R 語言的一些語法可能會讓擅長其他語言的工程師「渾身不舒服」。例如在索引上，R語言是從 1 開始索引，非傳統數據結構從0開始。

1. 目前發展趨勢

以2018年最新的RedMonk、TIOBE、IEEE、PYPL以及KDnuggets排行調查作為參考。

1. RedMonk排行榜：前三名JavaScript、Java、Python，R語言排第十四名(註三)

根據GitHub內部排名(以開發人員為出發點)，以及Stackoverflow 本身的標籤關聯(以用戶討論作為出發點)作為排名指標，RedMonk 是希望把關於語言的討論、使用及開發聯繫起來，以深入挖掘未來它們潛在的採用趨勢。

1. TIOBE排行榜：前三名Java、C、C++，Python第四名，R語言第十四名(註四)

每月更新，是程式語言流行趨勢的一個指標。排名是基於網路上的工程師、課程和第三方廠商的數量。要注意的是，這個排行榜只是反映某一門編程語言的熱門程度，並不能説明一門編程語言好不好，或者一門語言所編寫的代碼數量多少。

1. IEEE排行榜：前三名Python、C++、C，R語言為第七名(註五)

此為電機電子工程師學會的排行榜，綜合許多的指標，顯示在該領域下的程式語言排名。

1. PYPL排行榜：前三名Python、Java、JavaScript，R語言第七名(註六)

利用Google搜索活動來對語言的流行度進行排名。

1. [KDnuggets排行榜](https://www.kdnuggets.com/2018/05/poll-tools-analytics-data-science-machine-learning-results.html)：前三名為Python、RapidMiner、R(註7)

此為數據科學網站。針對數據科學與機器學習的工具，每年都有進行調查，其調查結果被視為這個領域的重要指標，許多專業書籍都參考此網站做出的結果。

在2016年以前，R語言是佔據該調查的第一名，也就是數據科學與機器學習的語言首選，但在2017年被Python追過，2018年更是退居到第三名

**總結來說，Python持續強勢，R語言稍有消退。**主要有兩個原因，分別是嵌入式開發的變化，以及R語言在機器學習應用較弱勢所造成的。而正確的媒體判讀，應該要去思考這些指標背後代表的意義，以及該語言的設計背景與使用社群，正如同R語言與Python的使用社群不同，設計面向不同，硬要說誰好誰壞，其實並沒有意義。

**壹、R語言入門須知Chapter 1: An Introduction to R**

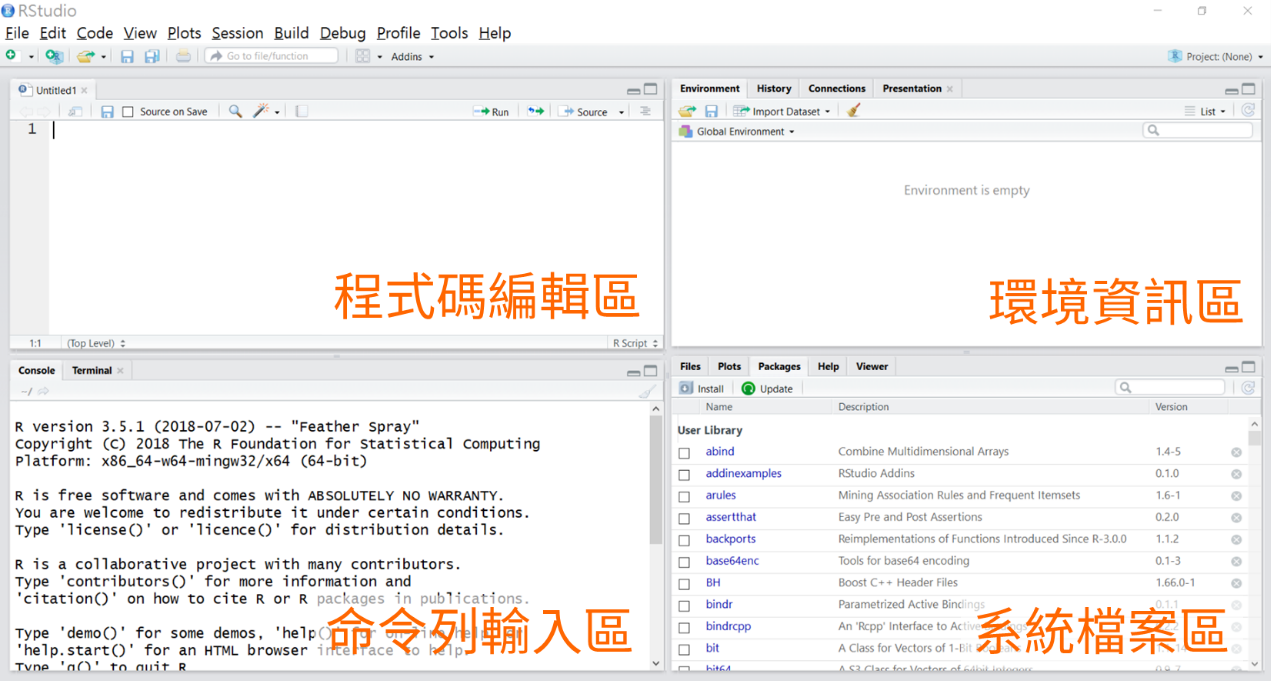
1. 從入門到精熟
2. 搭訕它：要有一個好的切入點，買本件單易懂的書，選個環境優、燈光美、氣氛佳的場所，簡單的學習領會R語言的美好。
3. 認識它：遇到疑問就勇於提問，問題越多代表要求越多、代表創意越多。
4. 攻略它：做專案、參加比賽，在實際解決問題精熟R語言。
5. 環境安裝

參考《2018\_R語言集訓班\_安裝教學.pdf》

1. Rstudio介面說明

Rstudio主要分成四個區域，分別為：程式碼編輯區、環境資訊區、命令列輸入區、系統檔案區。以下我們一一介紹。

* 快捷鍵成癮者請進：[Keyboard Shortcuts](https://support.rstudio.com/hc/en-us/articles/200711853-Keyboard-Shortcuts)



1. 命令列輸入區(左下)：程式的輸入、輸出、中斷

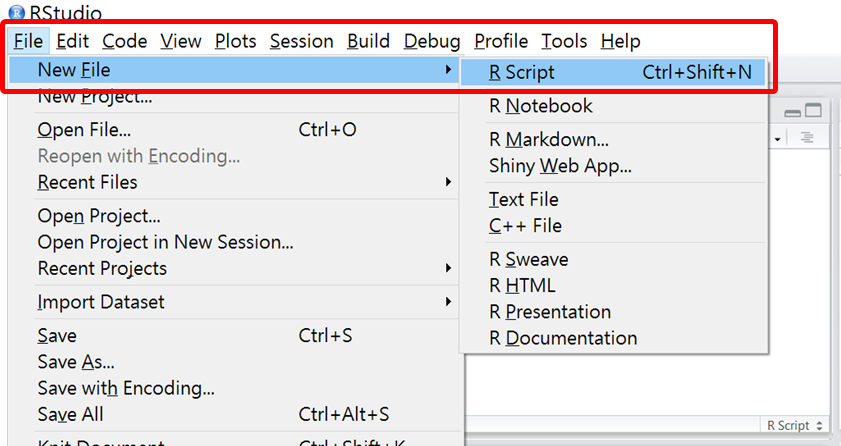
此區域為R語言的直譯器(Interpretor)，直接輸入指令就可以獲得執行結果。

1. 左下角當符號 > 表示可以輸入指令

* 輸入 1 + 1 後按下 Enter，檢查螢幕輸出
* 輸入 1 + 後按下 Enter，檢查螢幕輸出

最左下角的開頭變成 + 表示指令尚未輸入完成，應繼續輸入

1. 按下 ESC，會中斷執行中的程式 (左下角回復成 > 開頭)
2. 程式碼編輯區(左上)：撰寫程式的地方，也就是停留時間最長的地方。
3. 要撰寫一個新的腳本(script)，請點選上方功能列表的「File」→「New File」→「R Script」，即可開啟一個空白的腳本(script)。



1. 欲執行程式碼請大膽按下：

Control + Enter (Windows) or Command + Enter (Mac)

1. 在程式碼編輯區中輸入 1 + 1 後按下 Control + Enter，檢查 命令列區。
2. 在程式碼編輯區中輸入 1 + 後按下 Control + Enter，檢查 命令列區，在命令列區按下 ESC 中斷程式。
3. 如果要儲存撰寫好的程式碼，請點選上方功能列表的「File」→「Save」

第一是存檔時，可能要選擇編碼方式，請點選「UTF-8」

1. 環境資訊區(右上)：儲存你的變數與物件；紀錄曾經執行過的指令。

有兩個子標籤，Environment與History

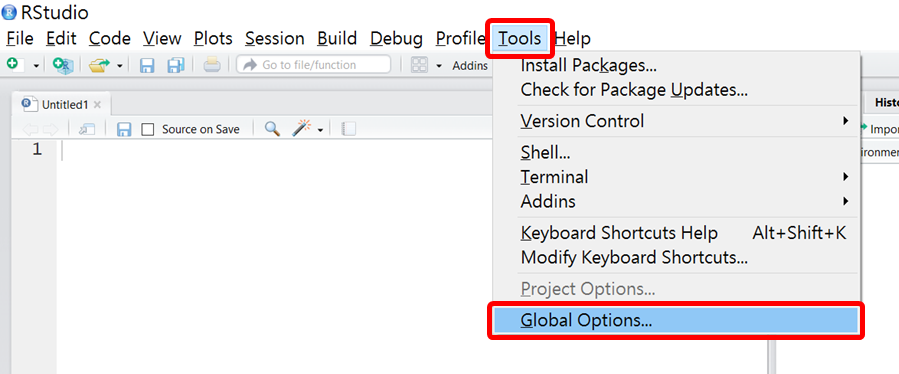
1. Environment：紀錄使用者設定的變數、資料與物件，以及其所代表的值。
2. History：紀錄Console視窗執行過的所有指令。
3. 系統檔案區(右下)

此區域有四個主要的子標籤：Files、Plots、Packages、Help。

1. Files：在此可以查看整個資料夾的內容
2. Plots：在此可以呈現圖表
3. Packages：在此可以看到已安裝的R擴充套件
4. Help：可以瀏覽輔助說明文件內容。
5. Rstudio編寫環境說明

良好的編寫環境可以讓人打程式時，心情好一點。以下講解如何調整RStudio的環境。

請點選上方功能列表的「Tools」→「Global Options…」，在Options的視窗下，選擇Appearance的頁面。一共有五個參數可做調整：Rstudio主題、縮放比例以及程式碼編輯區字型、字體大小、主題。





可以依照自己的喜好做調整，講師的調整為

* Rstudio theme：Classic Editor theme：Monokai
* Zoom：140% Editor Font：Lucida Console Editor Font size：12

**貳、R語言的數據型態與基本操作**

我們將從R語言最基本的組成─向量開始，然後學習操作，並延伸出其他資料型態，一步一步認ˋ是R語言世界的架構。

1. 註釋語法

在撰寫程式的過程中，一定要養成寫註釋的好習慣，這樣未來在回頭復習程式碼，才不會看不懂，或是花費很多時間在閱讀。

R語言的註釋語法為「#」，只要是在「#」後面撰寫的文字都不會執行

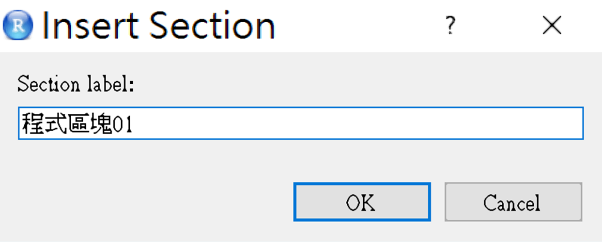
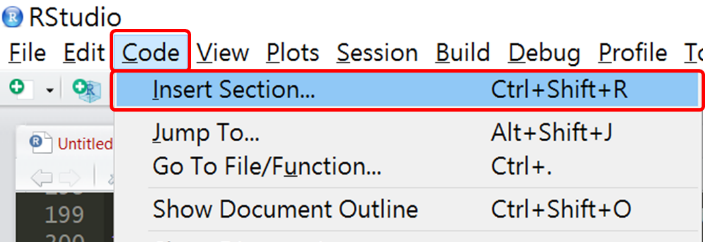
*# 這是單行註解*

*# 多行註解*

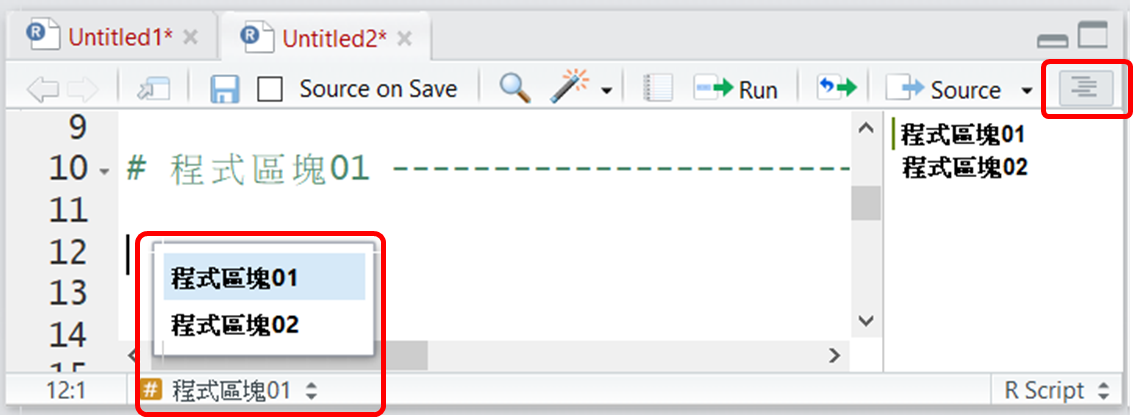
*# 只能*

*# 每行都加#*

另外，RStudio有一個程式區塊的貼心設計，點選上方通能列表的「Code」→點擊「Insert Section…」，輸入程式區塊的名稱。



接著使用者便能用類似超連結的方式快速抵達程式區塊，這對於撰寫大量code的人，提升很多效率。



1. 向量：數值向量(numeric）、整數向量(integer）、文字向量(character）、邏輯值向量(logical)

在R語言的世界中，所有的資料(文字、數值或邏輯等)，都**不是**以純量（Scalar）形式存在，而是以向量(Vector)形式存在，這樣的設計理念使得元素級別運算(element-wise operation)對 R 語言使用者而言是理所當然的。

以下依序呈現數值向量(numeric）、整數向量(integer）、文字向量(character）、邏輯值向量(logical)：

*# class(變數)，可以幫助我們檢視這個變數的資料型態*

*# 數值向量(numeric)*

> num\_vector <- 170

> class(num\_vector )

[1] "numeric"

*# 整數向量(integer)*

> int\_vector <- 170L

> class(int\_vector )

[1] "integer"

*# 文字向量(character)*

> chr\_vector <- "Learn R the easy way"

> class(chr\_vector)

[1] "character"

*# 邏輯值向量(logical)*

> logi\_vector <- TRUE

> class(logi\_vector)

[1] "logical"

如果想要創建長度大於 1 的向量，最常使用的方法是利用 c() 函數(combine 的縮寫），在 c()函數中可以將多筆資料以逗號分隔存入一個物件之中。

> nums <- c(170, 180)

> nums

[1] 170 180

> ints <- c(170L, 180L)

> ints

[1] 170 180

> chars <- c("Learn", "R", "the", "easy", "way")

> chars

[1] "Learn" "R" "the" "easy" "way"

> logicals <- c(TRUE, FALSE)

> logicals

[1] TRUE FALSE

1. 向量取值、取代、新增
2. 向量取值

* 在[ ] (中括號) 中輸入元素的位置進行取值，特別注意起始位置是從1開始計算
* 使用負號(-)移除給定位置元素，也稱為反向選取。

*# 在[ ](中括號)中輸入元素的位置進行取值*

> x <- c(174, 158, 160, 168, 173)

> x[1] # 選取第1個位置的元素

[1] 174

> x[c(1, 3)] # 選取第1, 3個位置的元素

[1] 174 160

> x[c(2, 3, 1)] # 依序取值

[1] 158 160 174

*# 在[ ]中使用負號 (-) 做反向選取*

> x[-1]

[1] 158 160 168 173

> x[-c(1, 3, 4)]

[1] 158 173

如果想給定條件進行取值

* 比較運算子(>, <, >=, <=, ==, !=)
* 邏輯運算子 (&, |)

x <- c(174, 158, 160, 168, 173)

> x > 160

[1] TRUE FALSE FALSE TRUE TRUE

*# 使用比較運算子 加上 `which` 函數進行取值*

> index <- which(x > 160)

> index

[1] 1 4 5

> x[index]

[1] 174 168 173

指令壓縮，將指令寫在 中，以達到縮短程式碼的功效

> x[which(x > 160)]

[1] 174 168 173

*# 配合邏輯算子進行多條件的取值*

x[which(x > 160 & x < 175)]

[1] 174 168 173

*# 可以省略 which*

> x[x > 160 & x < 175]

[1] 174 168 173

1. 向量取代

利用 [ ] (中括號) 與 <- (箭號) 進行取代與新增元素

> x <- c(174, 158, 160, 168, 173)

> x

[1] 174 158 160 168 173

*# 取代特定位置的元素*

> x[2] <- 158.5 # 取代x物件的第二個元素

> x

[1] 174.0 158.5 160.0 168.0 173.0

> x[c(1, 5)] <- 175

> x

[1] 175.0 158.5 160.0 168.0 175.0

*# 也可以用條件篩選做取代*

> x[x > 160] <- 170 # 取代大於160的值為170

> x

[1] 170.0 158.5 160.0 170.0 170.0

1. 向量新增

可用 [ ] (中括號) 與 <- (箭號) 進行新增元素

NA 為系統保留字，表示Not Available / Missing Values

> x <- c(174, 158, 160, 168, 173)

> x

[1] 174 158 160 168 173

# 在 [ ] 中新增元素

> x[6] <- 168

> x

[1] 174 158 160 168 173 168

> x[8] <- 147

> x # 未指定的元素值預設為NA

[1] 174 158 160 168 173 168 NA 147

> length(x) # 查看向量物件的長度

[1] 8

> x[length(x) + 1] <- 166 # 接續增加新元素

> x

[1] 174 158 160 168 173 168 NA 147 166

1. 數列操作seq與運算

要製造一個數列有很多種方式。

* 連續整數向量可以利用 : (冒號) 簡記，譬如數列 (1, 2, 3, 4) 在 R 語言中可利用 1:4 簡記
* 也可以使用 4:1 的簡記方式來產生 (4, 3, 2, 1) 的向量

*# 生成連續整數向量*

> 1:4

[1] 1 2 3 4

> 4:1

[1] 4 3 2 1

除了冒號簡記法外，可以透過seq函數生成有規則的數值向量(序列)

在seq()函數中有一些參數可以使用

* by 表示數列間隔，預設為1
* length.out 表示數列長度

> seq(1, 4)

[1] 1 2 3 4

> seq(1, 9, by = 2) # 間隔為2

[1] 1 3 5 7 9

> seq(1, 9, length.out = 5) # 分割長度為5

[1] 1 3 5 7 9

【動動手時間─數列生成】

1. 利用簡記法列出 1 ~ 10的數列
2. 利用 seq 函數列出偶數數列: 2, 4, 6, 8, 10
3. 觀察 seq(1, 10, length.out=5) 的輸出結果

【解答】【動動手時間─數列生成】

> 1:10

[1] 1 2 3 4 5 6 7 8 9 10

> seq(2, 10, by = 2)

[1] 2 4 6 8 10

> seq(2, 10, length.out = 5)

[1] 2 4 6 8 10

> seq(1, 10, length.out=5)

[1] 1.00 3.25 5.50 7.75 10.00

R語言的數列運算滿足 recycling properties

> c(1, 2, 3) \* c(2, 2, 2)

[1] 2 4 6

*# shorter arguments are recycled*

> 1:3 \* 2

[1] 2 4 6

> c(0.5, 1.5, 2.5, 3.5) \* c(2, 1)

[1] 1.0 1.5 5.0 3.5

【動動手時間─數字向量的四則運算】

1. 向量的四則運算，請計算以下五位女藝人的BMI

Hint: 。(身高的單位為公尺)

height <- c(174, 158, 160, 168, 173)

weight <- c(52, 39, 42, 46, 48)

【解答】【動動手時間─數字向量的四則運算】

weight/(height/100)^2

※補充：基本運算順序

括號 → 指數 →乘法 → 除法 → 加減

有關於R語言其他的運算操作，例如取餘數、絕對值、對數，可以自行Google研究。

1. 求助、說明文件相關

有時遇到不會用的函式function、或是套件(第六章)，除了上網Google外，在R語言中，我們也可以查詢說明文件。

以下是各種自救方法

|  |  |
| --- | --- |
| > help.start()  > ab # 輸入`ab`後 按下tab  > ?abs # 等同於 help(abs)  > ??abs  > apropos("abs")  > vignette()  > vignette("Introduction", "Matrix") | * help.start: 說明檔首頁 * 自動完成 (tab鍵):列出所有ab開頭的函數 * ? (一個問號):查詢特定函數的說明檔 * ?? (兩個問號):查詢包含特定關鍵字的說明檔 * apropos:查詢包含特定關鍵字的函數 * example:執行特定函數的使用範例 * vignette:查詢R環境中各種介紹文件 |

1. 其他數據型態：因子factor、矩陣matrix、資料表/框data.frame、列表list

除了用c()，把許多可以把多筆資料合併成一個向量之外，還有不同的資料型態。這邊提及的每一種都是一個很大的主題，除了資料表data.fram在第四章有詳細說明之外，其他的部分，將會以下列程式碼的方式，做初步的認識。

1. 因子factor

這種數據型態，多了levels屬性

在記錄數據時，電腦會記錄levels的順序，所以減少許多記憶空間

*# 因子factor*

> temp1 <- factor(c(10, 22, 28, 10, 19, 30))

> temp1

[1] 10 22 28 10 19 30

Levels: 10 19 22 28 30

*# 電腦紀錄的方式為1,3,4,1,2,5*

*# 在對應回去levels就可以的到原始的紀錄*

> temp2 <- factor(c("低溫", "舒適", "高溫", "低溫", "舒適", "高溫"))

> temp2

[1] 低溫 舒適 高溫 低溫 舒適 高溫

Levels: 低溫 高溫 舒適

*# 電腦紀錄的方式為1,3,2,1,3,2*

*# 在對應回去levels就可以的到原始的紀錄*

1. 矩陣matrix

如同數學上的矩陣，一模模一樣樣XD~就連元素取得也是相同的方式！

可以設定有多少row、多少column、以及是否先填滿row

* matrix(data, nrow = ？, ncol = ？, byrow = logical)

*# 矩陣matrix*

> matrix(1:12, nrow = 3, ncol = 4, byrow = TRUE)

[,1] [,2] [,3] [,4]

[1,] 1 2 3 4

[2,] 5 6 7 8

[3,] 9 10 11 12

> matrix(1:12, nrow = 3, ncol = 4, byrow = FALSE)

[,1] [,2] [,3] [,4]

[1,] 1 4 7 10

[2,] 2 5 8 11

[3,] 3 6 9 12

*# 矩陣元素取得*

> A <- matrix(1:12, nrow = 3, ncol = 4, byrow = TRUE)

> A[3,1]

[1] 9

> A[1:3, 2]

[1] 2 6 10

1. 資料表/框data.frame

也有人稱之為數據框，它是由一系列欄向量(column vector)所組成，可以把它看作是矩陣的擴充。

對單獨的向量或是矩陣而言，它們之中所記錄的元素必須相同，但是對資料表(data.frame) 而言，不同的欄向量的元素類別可以不同。資料表有幾個特色：

1. 每個欄(column)皆有一個名稱，如果沒有設定，R語言會自動取名。我們可以使用names()或是colnames()函式查詢或設定欄位名稱
2. 每個列(row)皆有一個名稱，R語言預設該列名稱為"1"、"2"、"3"…等，相當於數字編號。可以使用row.names函式查詢或設定列名稱

*# 資料表/框data.frame*

> num <- 1:3

> eng <- letters[1:3]

> df <- data.frame(num, eng)

> df

num eng

1 1 a

2 2 b

3 3 c

*# 查詢欄名稱*

> names(df) # 亦可用colnames(df)

[1] "num" "eng"

*# 設定欄名稱*

> colnames(df) <- c("數字", "字母")

*# 查詢列名稱*

> row.names(df)

[1] "1" "2" "3"

1. 列表/串立/清單list：存放異質性資料的容器

可以想像list是一個大抽屜，每一個抽屜可以存放不同的物件，也就是說每一個抽屜可以存放不同的數據型態。

*# 清單list*

> x1 <- c("林志玲", 174, 52, TRUE) *# 所有元素都被轉換成字串*

> x1

[1] "林志玲" "174" "52" "TRUE"

> x2 <- list("林志玲", 174, 52, TRUE) *# 保留資料型態*

> str(x2) *#str()可以檢視資料結構中的數據型態*

List of 4

$ : chr "林志玲"

$ : num 174

$ : num 52

$ : logi TRUE

> x3 <- list(name=c("林志玲", "隋棠", "蔡依林"),

height=c(174, 173, 158),

model=c(TRUE, TRUE, FALSE))

> x3

$name

[1] "林志玲" "隋棠" "蔡依林"

$height

[1] 174 173 158

$model

[1] TRUE TRUE FALSE

如果想要取出list中的資料

x3[[1]]

[1] "林志玲" "隋棠" "蔡依林"

x3$name

[1] "林志玲" "隋棠" "蔡依林"

x3[["name"]]

[1] "林志玲" "隋棠" "蔡依林"

names(x3)

[1] "name" "height" "model"

names(x3) <- c("Name", "Height", "Model")

names(x3)

[1] "Name" "Height" "Model"

1. 常用函數一print、rep、rm

以下介紹幾種常用的函數。

1. 印出print()

> chars <- c("Learn", "R", "the", "easy", "way")

> print(chars)

[1] "Learn" "R" "the" "easy" "way"

*# 或是直接用()，把整個包起來*

> (chars <- c("Learn", "R", "the", "easy", "way"))

[1] "Learn" "R" "the" "easy" "way"

1. 重複rep()

如果有相同的東西要重複出現，又不想打太多次，可以使用rep()函式

*# 重複元素，參數times為重複次數。*

rep(x = 1:5, times = 3)

1. 移除物件rm()

如果右上角的環境資訊區Environment，有太多的變數，想要做清理，可以使用rm()函式。

*# 利用rm()清除變數、物件等*

a <- c(1,2,3) *# 這時Environment區域會出現a物件*

rm(a) *# 如此就可清除a變數*

*# 利用ls()，可以得知目前Environment的所有變數*

ls()

rm(list = ls()) *# 可以清空Environment*

**參、R語言內建常用的函式**

這一章，會教導許多的內建、好用的函式，熟悉並熟練他們，對於資料的操作，可以更有效率。

1. 確認型態class、str、is.XXX系列

* class可以檢視資料型態(資料結構)
* str總覽資料，可以檢視資料結構中的數據型態
* is.XXX系列可以確認各種資料結構，包括是否為向量is.vector()、是否為數值向量is.numeric()、是否是字串is.character、是否是清單is.list()、是否是資料表is.data.frame()

1. 型態轉換as.XXX系列

有時為了順利的操作數據，比如計算、條件篩選、字詞統計等等，我們會需要將資料型態做轉換。包括as.numeric()、as.character()、as.list()、as.data.frame()

> (num <- 1:5)

[1] 1 2 3 4 5

> as.character(num)

[1] "1" "2" "3" "4" "5"

> temp2 <- factor(c("低溫", "舒適", "高溫", "低溫", "舒適", "高溫"))

> temp2

[1] 低溫 舒適 高溫 低溫 舒適 高溫

Levels: 低溫 高溫 舒適

> as.numeric(temp2)

[1] 1 3 2 1 3 2

1. 基本統計函數min、max、sum、var、sd

R語言一開始就是統計學家合作開發完成，所以有內建的許多統計函式。

其中包含最小值min()、最大值max()、總和sum()、變異數var()、標準差sd()、四捨五入round()

num <- c(1:3, 5:7)

num

[1] 1 2 3 5 6 7

min(num)

[1] 1

max(num)

[1] 7

sum(num)

[1] 24

var(num)

[1] 5.6

sd(num)

[1] 2.366432

round(x = sd(num), digits = 2)

[1] 2.37

關於統計圖表，例如直方圖等等，會在之後其他周做教學。

1. 常用函數二table、prop.table

要計算出現次數，可以利用table()

如果要計算出現的比例，可以使用prop.table()

> dayoff <- c("星期一", "星期三", "星期四", "星期五", "星期二",

"星期二", "星期三", "星期一", "星期一", "星期五")

> table(dayoff)

星期一 星期二 星期三 星期五 星期四

3 2 2 2 1

> prop.table(table(dayoff))

星期一 星期二 星期三 星期五 星期四

0.3 0.2 0.2 0.2 0.1

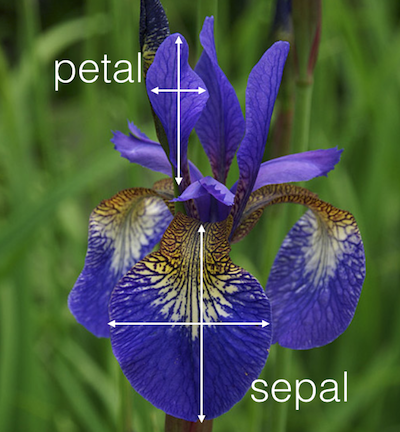
**肆、使用R整理數據**

資料表data.frame是資料分析時最基本的物件，R 提供將外部資料轉成 data.frame 的功能

透過data.frame 可以進行以下功能：

* 資料的整理與擷取(本章重點)
* 圖形的繪製
* 模型的配適與預測

接下來我們將使用R語言的內部資料─鳶尾花資料集iris，並學習如何提取欄、列，以及透過座標和條件篩選資料。

1. 使用R語言內部資料集─鳶尾花資料集iris

iris是世界上最常見的範例資料之一，也是非常著名的生物資訊資料集之一，資料的筆數為150筆，共有五個欄位：

1. 花萼長度(Sepal Length)：計算單位是公分。

2. 花萼寬度(Sepal Width)：計算單位是公分。

3. 花瓣長度(Petal Length) ：計算單位是公分。

4. 花瓣寬度(Petal Width)：計算單位是公分。

5. 類別(Class)：可分為Setosa，Versicolor和Virginica三個品種。

讓我們在R語言裡面實際看看這些資料吧!

> data("iris")

> head(iris) *# 列出前幾筆資料, 預設6筆*

Sepal.Length Sepal.Width Petal.Length Petal.Width Species

1 5.1 3.5 1.4 0.2 setosa

2 4.9 3.0 1.4 0.2 setosa

3 4.7 3.2 1.3 0.2 setosa

4 4.6 3.1 1.5 0.2 setosa

5 5.0 3.6 1.4 0.2 setosa

6 5.4 3.9 1.7 0.4 setosa

> dim(iris) *# 列出資料表的 rows and columns 數量*

[1] 150 5

1. 表格的取值─座標

類似於向量取值，在中括號 [i, j] 中進行取值

逗號的前後分別表示資料表的 row and column

> iris[2, 3]

[1] 1.4

> iris[1:6, 1:3]

Sepal.Length Sepal.Width Petal.Length

1 5.1 3.5 1.4

2 4.9 3.0 1.4

3 4.7 3.2 1.3

4 4.6 3.1 1.5

5 5.0 3.6 1.4

6 5.4 3.9 1.7

1. 表格的取值─列、欄

欲選取第i筆觀察資料(row)時，使用 [i, ] 在column欄位留白

> iris[2, ]

Sepal.Length Sepal.Width Petal.Length Petal.Width Species

2 4.9 3 1.4 0.2 setosa

> iris[c(1, 51, 101),]

Sepal.Length Sepal.Width Petal.Length Petal.Width Species

1 5.1 3.5 1.4 0.2 setosa

51 7.0 3.2 4.7 1.4 versicolor

101 6.3 3.3 6.0 2.5 virginica

欲選取整欄資料(column)時，有三種常用方法

> iris[,1] *# 欄位名稱未知*

> iris$Sepal.Length *# 已知欄位名稱*

> iris[["Sepal.Length"]] *# 已知欄位名稱*

[1] 5.1 4.9 4.7 4.6 5.0 5.4 4.6 5.0 4.4 4.9 5.4 4.8 4.8 4.3 5.8 5.7 5.4

[18] 5.1 5.7 5.1 5.4 5.1 4.6 5.1 4.8 5.0 5.0 5.2 5.2 4.7 4.8 5.4 5.2 5.5

[35] 4.9 5.0 5.5 4.9 4.4 5.1 5.0 4.5 4.4 5.0 5.1 4.8 5.1 4.6 5.3 5.0 7.0

[52] 6.4 6.9 5.5 6.5 5.7 6.3 4.9 6.6 5.2 5.0 5.9 6.0 6.1 5.6 6.7 5.6 5.8

[69] 6.2 5.6 5.9 6.1 6.3 6.1 6.4 6.6 6.8 6.7 6.0 5.7 5.5 5.5 5.8 6.0 5.4

[86] 6.0 6.7 6.3 5.6 5.5 5.5 6.1 5.8 5.0 5.6 5.7 5.7 6.2 5.1 5.7 6.3 5.8

[103] 7.1 6.3 6.5 7.6 4.9 7.3 6.7 7.2 6.5 6.4 6.8 5.7 5.8 6.4 6.5 7.7 7.7

[120] 6.0 6.9 5.6 7.7 6.3 6.7 7.2 6.2 6.1 6.4 7.2 7.4 7.9 6.4 6.3 6.1 7.7

[137] 6.3 6.4 6.0 6.9 6.7 6.9 5.8 6.8 6.7 6.7 6.3 6.5 6.2 5.9

1. 表格的取值─條件篩選

利用條件式做篩選

> iris[iris$Sepal.Length > 5.5 & iris$Species=="setosa", ]

Sepal.Length Sepal.Width Petal.Length Petal.Width Species

15 5.8 4.0 1.2 0.2 setosa

16 5.7 4.4 1.5 0.4 setosa

19 5.7 3.8 1.7 0.3 setosa

【動動手時間─鳶尾花資料集iris】

* 1. 請幫我統計Sepal.Length > 4.5的鳶尾花種類個數。

【解答】【動動手時間─鳶尾花資料集iris】

aim <- iris[iris$Sepal.Length > 4.5, "Species"]

table(aim)

## setosa versicolor virginica

## 45 50 50

**伍、R語言做文字處理(基礎)**

這一章中，我們將講解基礎的文字處理，包含字串(Character)的連接、長度、擷取、分割，以及進階的關鍵字取代、找查(Pattern Matching and Replacement)。

在R語言中，字串的輸入要用""或是''來包覆。

常用的字串處理函數有：字串的連接paste、計算字串長度nchar、截取子字串substring、字串的切割strsplit

1. 字串處理paste、nchar、substring、strsplit
2. 字串的連接paste

*# 兩個字串做連接*

> a <- "year"

> b <- "2018"

> paste(a, b, sep = ":")

[1] "year:2018"

*# 即使不是字串，連接完之後也會變成字串*

> c <- "date"

> d <- 20181119

> paste(c, d, sep = ":")

[1] "date:20181119"

1. 計算字串長度nchar

*# 計算字串的長度*

> word <- c("R語言集訓班")

> nchar(word) # 試著與 length 函數比較

[1] 6

> length(word) # 這是在計算向量的長度

[1] 1

1. 截取子字串substring

*# 擷取子字串*

> word <- c("R語言集訓班")

> substring(text = word, first = 1, last = 3)

[1] "R語言"

1. 字串的切割strsplit

*# 字串的切割*

> test <- c("language")

> strsplit(x = test, split = "a")

[[1]]

[1] "l" "ngu" "ge"

【動動手時間─字串的連接、長度、擷取、分割】

以下為生日快樂歌的部分歌詞

song <- c("Happy Birthday to You. Happy Birthday Dear Littlefish. Happy Birthday to You.")

1. 計算這段歌詞song的長度(含空白與標點符號)
2. 取出字串 timedate <- "2018-11-19" 的月份
3. 請幫我取出「Happy Birthday to Dear Littlefish」這一段
4. 請取出「Happy Birthday to Dear」，並利用paste的方式，對自己說生日快樂

【解答】【動動手時間─字串的連接、長度、擷取、分割】

*# Question 01*

*# 計算本首歌的長度(含空白與標點符號)*

> nchar(song)

[1] 77

*# Question 02*

*# 取出字串 timedate <- "2018-11-19" 的月份*

> timedate <- "2018-11-19"

> tmp <- strsplit(timedate, "-")

> tmp[[1]][2]

[1] "11"

*# Question 03*

*# 請幫我擷取Happy Birthday to Dear Littlefish*

> substring(text = song, first = 24, last = 54)

[1] "Happy Birthday Dear Littlefish."

*# Question 04*

*# 請取出「Happy Birthday to Dear」，並利用paste的方式，對自己說生日快樂*

> happy <- substring(text = song, first = 24, last = 42)

> name <- "Steve"

> paste(happy, name)

[1] "Happy Birthday to Steve"

(二) 關鍵字的取代與找查gsub、grep、grepl、gregexpr

1. 關鍵字的取代gsub

*# 關鍵字的取代gsub*

*# gsub(pattern, replacement, x)*

> year <- c("民國99", "民國100", "民國101")

> gsub("民國", "", year)

[1] "99" "100" "101"

*# 加上1911，轉換成西元*

> as.numeric(gsub("民國", "", year)) + 1911

[1] 2010 2011 2012

1. 關鍵字的找查grep、grepl

grep第一個參數是pattern，第二個是data。output為有符合這個pattern是第幾筆資料。與grep用法相似,差異在於其output是TRUE/FALSE

*# 關鍵字的找查grep、grepl*

*# grep(pattern, x, value=TRUE)*

*# grepl(pattern, x)*

title <- c("馬習會前交涉祕辛曝光", "馬說明馬習會：公布逐字稿不可思議的透明", "談22K政策朱立倫：不幸被企業濫用")

grep("馬習會", title)

[1] 1 2

grepl("馬習會", title)

[1] TRUE TRUE FALSE

grep("馬習會", title, value = TRUE)

[1] "馬習會前交涉祕辛曝光"

[2] "馬說明馬習會：公布逐字稿不可思議的透明"

1. 關鍵字的找查gregexpr

找出所有符合pattern的字串在哪個位置及長度

> txt <- c("名模林志玲身高有174公分，體重52公斤", "女神蔡依林身高158公分，體重只有39公斤")

> matches <- gregexpr("[0-9]+", txt)

> regmatches(txt, matches)

[[1]]

[1] "174" "52"

[[2]]

[1] "158" "39"

※補充：pattern表示方式

* [0-9]，代表pattern符合0|1|2|3|4|5|6|7|8|9
* [0-9]+，代表pattern符合一或多個0|1|2|3|4|5|6|7|8|9
* 更多pattern表示方式，可以Google搜尋「正規表示法」<https://atedev.wordpress.com/2007/11/23/%E6%AD%A3%E8%A6%8F%E8%A1%A8%E7%A4%BA%E5%BC%8F-regular-expression/>

小結論：

* 連結：paste
* 計算長度：nchar
* 擷取字串：substring
* 分割字串：strsplot
* 取代：gsub
* 查找位置：grep(value=TRUE), grepl
* 查找結果：grepexpr

**陸、套件安裝**

R語言是開源的，也就是說每一位R語言的使用者，都可以依照自己特殊需求，撰寫獨特的功能包，並透過官方認證後，發佈到R語言的官方社群，我們稱這些功能包之為套件。目前R語言官方認證安全可使用的套件一共有13398個，目前數量還在成長當中。

因為官方認證的程序相當嚴謹，所以許多厲害的人，在開發完一個套件後，會先放在Github上與社群討論修改，等到版本穩定之後再上傳到官方社群。(又或者套件有經常改動的需求，所以就不會上傳到官方社群。)

所以，知識的累積造成人類文明不斷地進步，學會使用別人撰寫好的套件，可以使我們有如站在巨人的肩膀上，可以看得更高、更遠。

以下逐一說明如何下載套件、載入套件，以及使用套件。

我們以praise套件做為範例，正如其名，該套件的功能是會不斷地讚美你，讓人心情變得愉悅。

觀看目前有多少R語言套件：<https://www.r-pkg.org/>

1. 下載套件

R語言套件安裝主要來源有二：1. 官方社群。 2. GitHub、Gitlab、BitBucket等社群

1. 套件來自官方社群

在命令列輸入區(Console)或是程式碼編輯區(R script)輸入下列指令，並執行。

*# 使用方法*

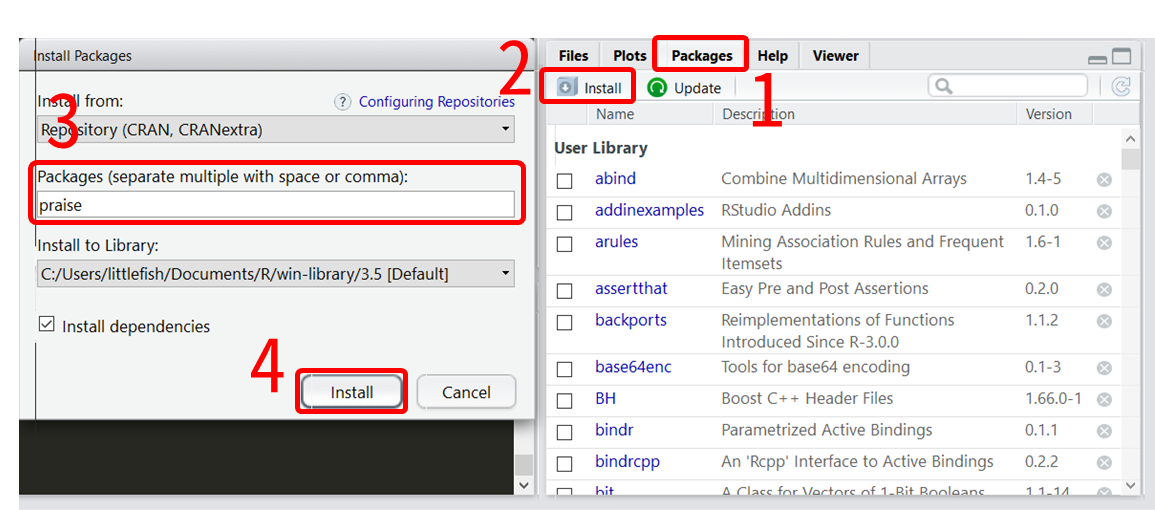
*# install.packages("套件名稱")*

*# 安裝套件praise*

install.packages("praise")

※補充：使用圖形化介面安裝方式

如果不習慣使用程式語言安裝，可以用右下角「系統檔案區」的「Packages」子標籤下 → 點擊「Install」，並在跳出的視窗中，輸入想下載的套件名稱 → 點擊「Insatll」即可。可以觀察命令列輸入區(Console)，會看到如同程式碼安裝的過程。



1. 套件來自非官方社群，如GitHub、GitLab等等

在官方社群，我們利用函式install.package去安裝，但是現在套件放在不同的社群上，所以要換另一個函式，下面以下載GitHub社群的套件為範例。(註八)

*# 使用方法*

*# 先安裝一個叫"devtools"的套件，並載入*

*# 再利用套件中的install\_github()函式做安裝*

*# install\_github("GitHub帳號/套件名稱")*

*# 安裝套件praise*

> install.packages(devtools) *# 下載devtools套件*

> library(devtools) *# 載入devtools套件*

> install\_github("cran/praise") *#從GitHub下載cran帳號開發的praise套件*

現在你已經學會了如何下載套件了！至於從非官方的開源社群下載，會依據不同的來源而有不同的函式，可以自行Googleg深入研究。

光是下載是不夠的，我們必須還要載入R語言之中，讓它知道有這個套件可以使用。

1. 載入套件

下載套件，就是因為想使用套件中有許多設計好的功能，但光是下載還不夠，必須告訴R語言現在有這些功能包可以使用了！這就是所謂的「載入套件」。

*# 載入套件praise*

*# 記得要先下載套件唷！*

**>** library(praise)

※補充：為何要分成下載與載入兩個動作

因為不是每一次使用R，都會需要這麼多的功能，如果每次開啟RStudio都要自動載入全部的套件，那將會花費相當多的時間，並不實際；因此習慣上，我們會事需要用到某功能的時候，才載入該套件。

p.s.下載套件只需要一次，只是載入特件的時機自己決定。

※補充：使用圖形化介面載入套件

如果不習慣使用程式語言載入，可以用右下角「系統檔案區」的「Packages」子標籤下 → 輸入想載入的套件名稱「praise」，並將前面的方塊勾選即可。

1. 第一次使用套件

套件中有許多設計好功能，第一次使用時，可能會無從下手。我們可以依從第貳章所教的方法，做簡單的查詢。

*#* 使用？、？？或是help()做查詢

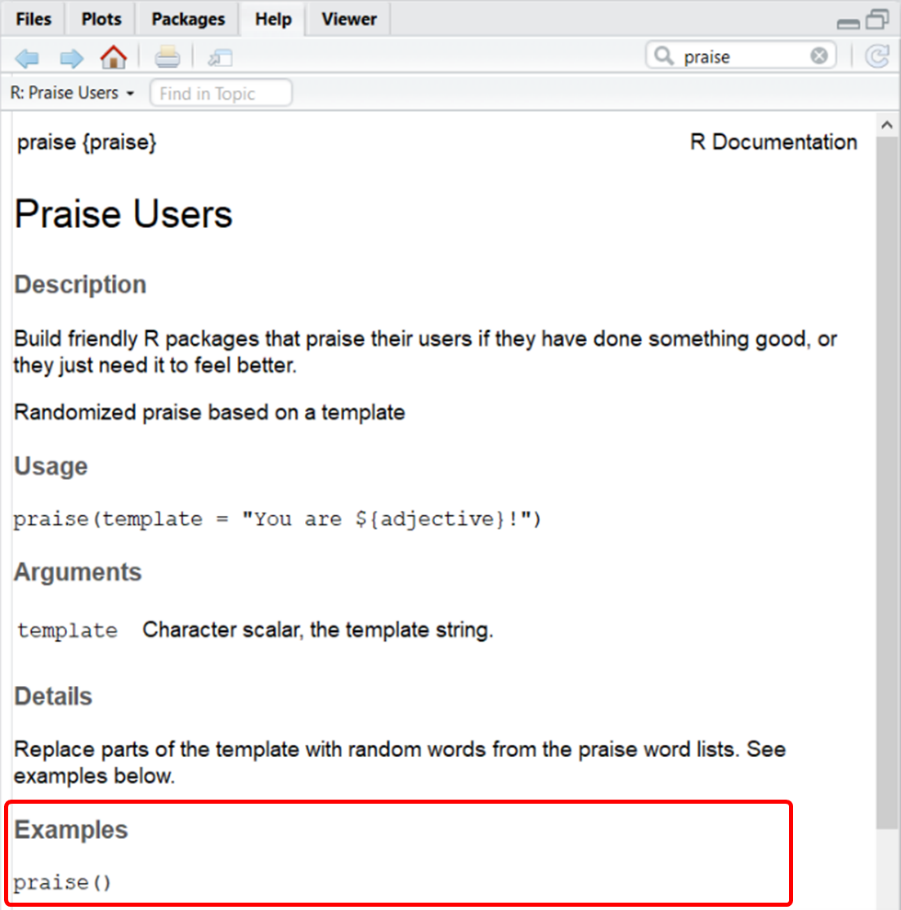
*# 或是在右下角「系統檔案區」的「Help」子標籤下，輸入想搜尋的套件*

> ?praise

> ??praise

> help("praise")

接著我們會在右下角「系統檔案區」的「Help」子標籤下，看到說明文件，滑到最下方通常會有Examples的區域，給出一些範例程式碼讓我們運行。



*#* 使用praise套件的praise()函式

> praise()

[1] "You are stunning!"

如果是功能複雜一點的套件，我們除了上述的方法之外，在正常情況下，套件的開發者會提供自己對於該套件的使用說明(如果有的話)，我們執行下面的程式碼就可以看到這些套件的說明文件

*#* 瀏覽自己電腦中套件的說明文件

> browseVignettes()

因為priase套件的功能過於簡單(笑~)，所以它並沒有提供任何說明文件。

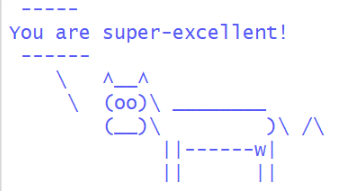
以上使用？、？？、help與vignette都是從自己電腦有安裝的說明文件中來做查詢，如果想要尋找更豐富的資料，可以自行Google深入研究。

【動動手時間─套件使用】

有一個很酷的套件叫做cowsay，它可以幫助我們印出ASCII風格的圖形。請完成下列指定任務。

1. 安裝cowsay套件。
2. 載入cowsay套件。
3. 使用cowsay中的函式印出下列圖形，並結合praise()的讚美文字。

(讚美文字是隨機的，所以和下面截圖不符很正常。)



【解答】【動動手時間─套件使用】

*#* 下載cowsay套件

install.packages("cowsay")

*#* 載入cowsay套件

library(cowsay)

*#* 瀏覽自己電腦中套件的說明文件，並尋找cowsay的說明文件，點選。

> browseVignettes()

*#* 將praise()隨機生成的讚美文字，存入變數sentence中

*# 印出動物，並呈現讚美文字*

> sentence <- praise()

> say(what = sentence, "cow")

**柒、資料讀取與匯出(英文資料)**

學會R語言的基本操作後，離分析真實數據只差一步了。

現在我們要來學習如何讀取自己的數據，並且在整理數據後，如何輸出成一個檔案。

為了要做到這些事情，我們將教導如何設定路徑、下載資料集、資料讀取、整理資料以及資料匯出。其中基礎整理資料的細節與詳細作法可以回顧第肆章。

1. 工作目錄設定(Setting Working Directory)

就像我們會把照片歸檔，分成好幾個資料夾做管理一樣。

當我們在學習程式或是撰寫數據分析報告的時候，也習慣把同一個工作底下的數據、程式檔案、成果報告，放進同一個資料夾。

所以在讀取資料前，我們要先學習如何得知目前R語言的工作目錄，以及如何更改到我們存放資料的目錄底下。

首先，我們在D槽下創建一個新的資料夾，名稱為「week01」

*#* 取得目前工作目錄

> getwd()

[1] "C:/Users/littlefish/Documents" *# 這是講師目前的R語言的工作目錄*

*# 更改工作目錄到D槽*

*# 並確認是否工作目錄切換到D槽之week01資料夾之下*

> setwd("D:/week01")

> getwd()

[1] "D:/week01"

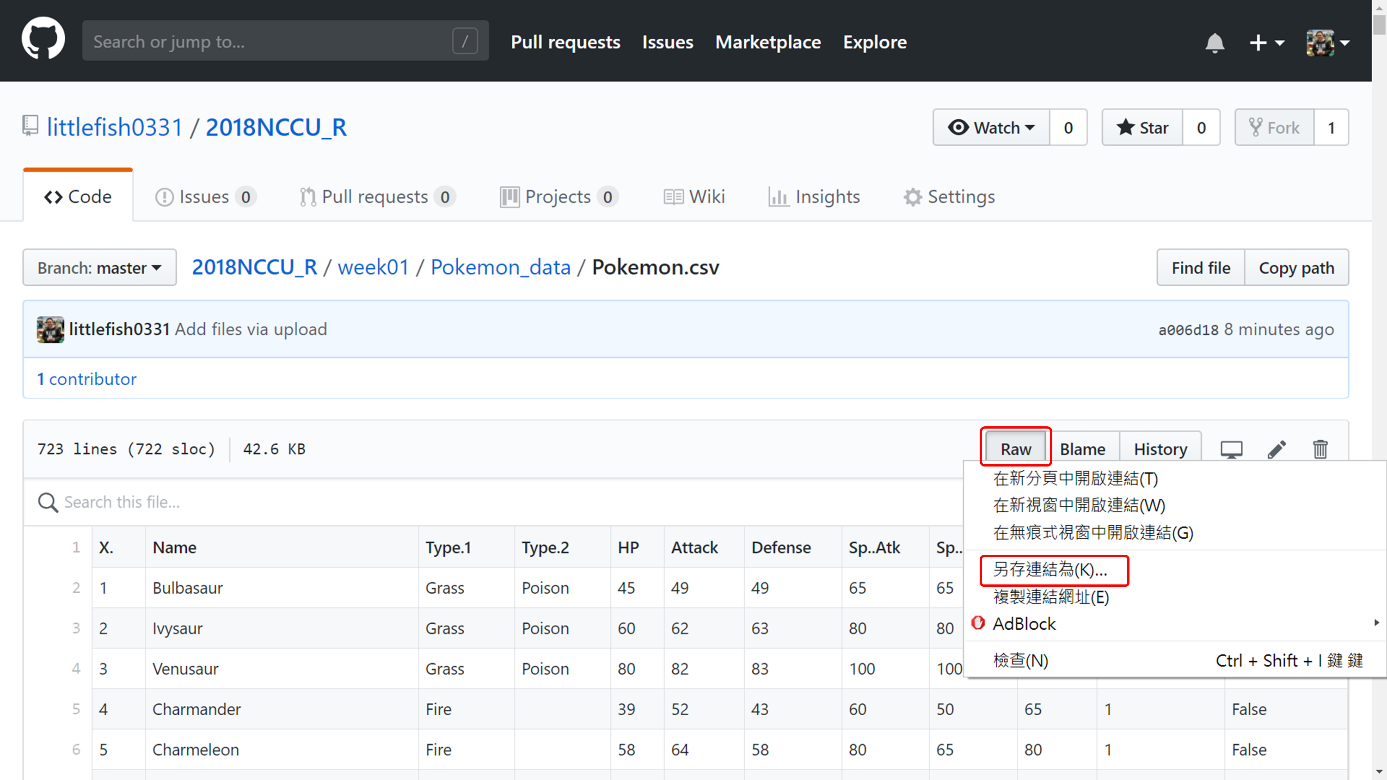
1. 資料集介紹

我們將使用國外知名數據競賽網站上─Kaggle上面提供的開放資料。

資料集名稱「Pokemon with stats」，紀錄721種Pokémon的基本資料，包含Pokémon的變種型態，資料筆數為800筆，13種特徵。

資料來源：<https://www.kaggle.com/gyani95/380000-lyrics-from-metrolyrics>

為了大家方便，講師先行下載經過初步處理後(去除變種型態以及欄位Total，剩餘721筆，12種特徵)，放在GitHub的week01/Pokemon\_data的資料夾中，同學可以在上面觀看資料集，並對「Raw」點擊右鍵，選擇「另存連結為…」，便可將資料夾下載下來，存放到自己電腦D:/week01的資料夾中。



以下是資料的欄位敘述，可以到Kaggle資料集的連結，或是GitHub上做觀看。



那麼接下來我們將學習如何利用R語言，讀取這份Pokémon資料。

1. 資料讀取

讀取資料的方式很多，就像從1加到100有很多方式，可以按計算機也可以用公式。

同樣地，要利用R語言讀取資料進行操作，方法很多種，針對不同的資料型態(csv、xlsx、tsv、json、xml等等)，都有對應不同的函式可以進行讀取；此外，讀取的速度以及對電腦資源的使用，也有所差異。

csv檔(Comma-Separated Values File)是開放資料中，一種常見的數據儲存格式。

以下講解對於csv檔，最常用的讀取方式，也是R語言初學者必備的技能之一。同時介紹在讀取資料時，常常遇見的幾個問題以及解決方式。最後，我們再針對不同的資料型態，所對應的讀取方式做初步認識。

*# 讀取csv檔，並存進變數pokedata中*

*# read.csv("檔案路徑")*

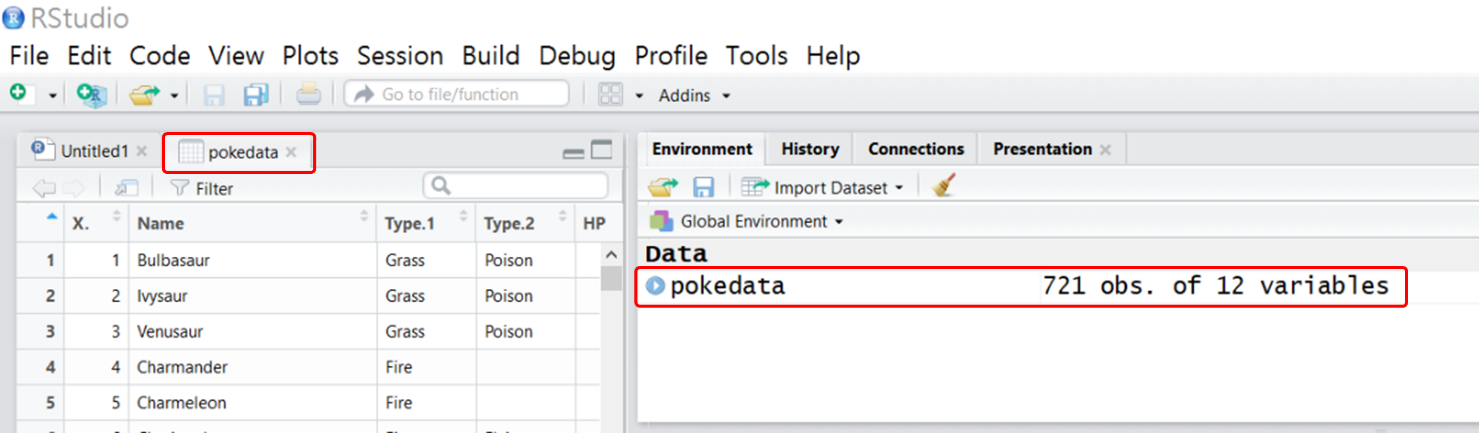
> pokedata <- read.csv("D:/week01/Pokemon\_data/Pokemon.csv")

※補充：Console出現warnings、error

有時候執行上述程式碼，Console會出現一些警告訊息(Warning message)，有時是只是系統在提醒你資料格式有要注意的地方，有時候則是讀取出現問題，如果不確定是哪一種，就再執行一次上面的程式碼，假如警告訊息(Warning message)消失，那麼表示讀取資料沒有問題，如果沒有消失，那可能就要看一下警告訊息了。

如果出現的是錯誤(error)，那通常要看一下錯誤訊息了。

如果讀取成功，點擊右上「環境資訊區」的「Environment」，則在「程式編輯區」就會多出一個數據名稱的子標籤，我們就可以從該子標籤先下觀看數據。



※補充：讀取檔案─編碼問題

電腦編碼指的是電腦內部在儲存檔案時的編寫方式，因為電腦是用0、1代碼在紀錄資料，因此對於數字、英文、中文、標點符號等，都有一套的編碼方式。然而隨著語言的種類變多，大家的編碼方式也不盡相同。

目前最廣為使用的是UTF-8編碼(8-bit Unicode Transformation Format)，另外繁體中文常用的編碼為big5，簡體中文則為GB系列。不同編碼間的轉換常常是讀取資料一個頭痛的問題，我們可以透過read.csv()中的參數fileEncoding做調整。

*# 利用編碼UTF-8讀取資料*

> pokedata <- read.csv("D:/week01/Pokemon\_data/Pokemon.csv",

fileEncoding = "UTF-8")

※補充：read.csv()─參數補充

1. 是否要將第一列(first row)作為標題列，請用參數header。

*# 不讓第一列(first row)做為標題列。(預設為TRUE)*

> pokedata <- read.csv("D:/week01/Pokemon\_data/Pokemon.csv",

header = FALSE)

2. 指定用何種符號做為分開字元，請用參數sep。

*# 使用","作為分隔字元。(預設為",")*

> pokedata <- read.csv("D:/week01/Pokemon\_data/Pokemon.csv", sep = ",")

1. 資料匯出

匯出檔案的方式就和讀取一樣有很多種，通常函式都會成對出現。下面我們以剛剛的資料，匯出成csv檔作為範例，之後在介紹不同的資料型態匯出語法。

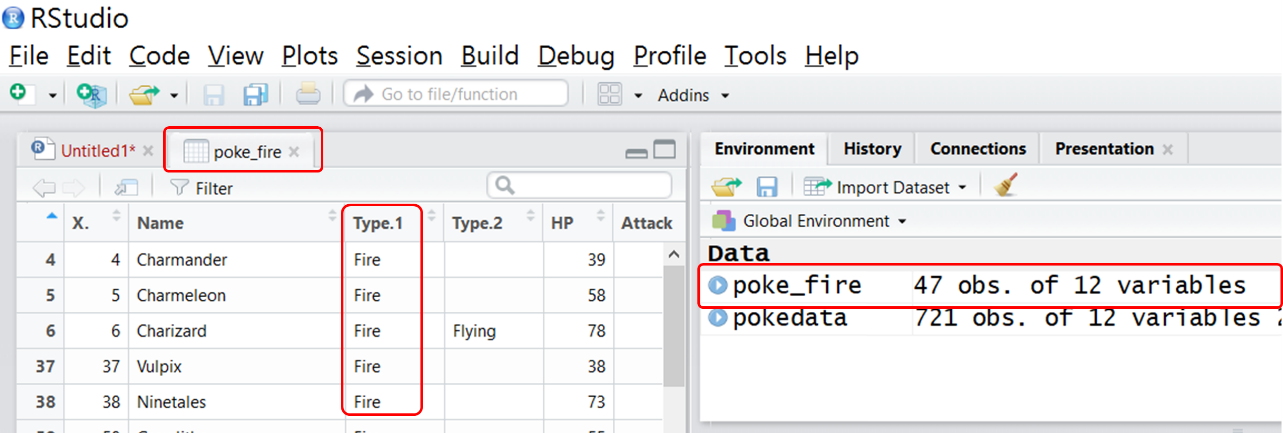
假如針對這721種神奇寶貝，我們只想要火系神奇寶貝，那麼我們可以利用欄位Type.1去做篩選，之後在匯出成一個csv檔案做儲存。

*# 利用欄位Type.1篩選出火系(Fire)神奇寶貝。注意篩選是用兩個等號「==」*

*# 將篩選出的變數，存放到變數poke\_fire中*

> poke\_fire <- pokedata[pokedata$Type.1=="Fire", ]

可以一樣檢查是不是篩選出來的神奇寶貝，確實都是火系的。



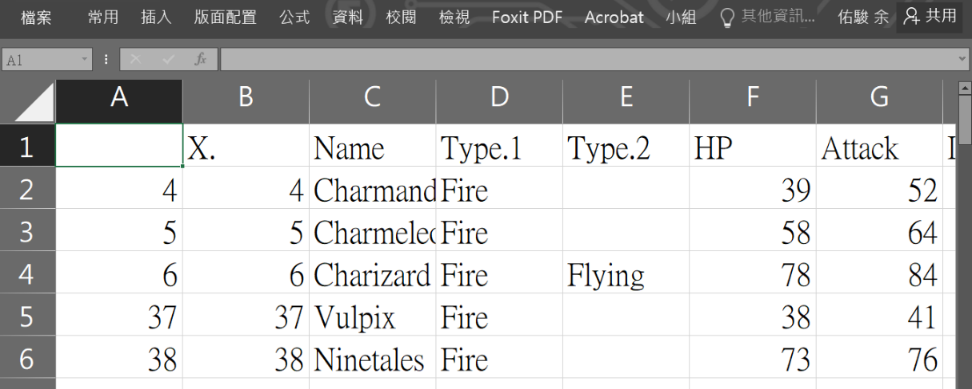
儲存篩選出的火系神奇寶貝，匯出為csv檔，並存到指定的路徑。

*# 儲存poke\_fire為csv檔*

> write.csv("資料集變數", "儲存路徑(函檔案名稱)")

> write.csv(x = poke\_fire, file = "D:/week01/Pokemon\_data/Pokemon\_fire.csv")

那麼我們可以到我們指定的路徑，開啟檔案，看看是否有儲存成功。



可以看到儲存的csv檔中，第一行(first column)多了重複的序號。如果不想要這一行，可以在儲存時，透過參數row.names = FALSE做刪除。

*# 儲存poke\_fire為csv檔*

> write.csv("資料集變數", "儲存路徑(函檔案名稱)")

> write.csv(x = poke\_fire, file = "D:/week01/Pokemon\_data/Pokemon\_fire.csv",

row.names = FALSE)

另外，我們也可以指定儲存時，要用何種編碼方式儲存資料。透過參數fileEncoding

*# 儲存poke\_fire為csv檔*

> write.csv("資料集變數", "儲存路徑(函檔案名稱)")

> write.csv(x = poke\_fire, file = "D:/week01/Pokemon\_data/Pokemon\_fire.csv",

fileEncoding = "UTF-8")

1. 讀取方式百百種─條條大路通羅馬

我們前面提到過，隨著資料型態的不同，會有不同的讀寫方式(讀取與匯出)，以下就介紹幾種常見的資料格式，以及相對應的讀寫方法。可以拿GitHub上面的diff\_type\_data資料夾中的資料做測試。

1. txt檔，文字檔

*# 讀取txt檔案*

> read.table("D:/week01/diff\_type\_data/iris\_txt.txt")

*# 指定路徑，儲存資料集iris，檔案名稱iris\_txt，儲存為txt檔案*

> write.table(x = iris, file = "D:/week01/diff\_type\_data/iris\_txt.txt")

1. csv檔，逗號分隔檔

*# 讀取csv檔案*

read.csv("D:/week01/diff\_type\_data/iris\_csv.csv")

*# 指定路徑，儲存資料集iris，檔案名稱iris\_csv，儲存為csv檔案*

> write.csv(x = iris, file = "D:/week01/diff\_type\_data/iris\_csv.csv")

1. xml檔，Extensible Markup Language，可延伸標記式語言

xml檔，是網頁在儲存資料時一種常用的格式，通常爬蟲時會遇到。有厲害的人已經把相關會用到的功能打包成一個套件XML。

透過這個套件，可以把網頁資料爬梳下來，解析成list格式，在轉換成data.frame，詳細的做法可以Google延伸了解。這邊給一個參考網站，內有範例，歡迎回去嘗試之後在跟講師討論。<https://blog.gtwang.org/r/r-xml-package-parsing-and-generating-xml-tutorial/>

1. json檔，JavaScript Object Notation，JavaScript物件表示法

json格式的資料讀寫，可以用套件jsonlite做完成。

*# 下載並載入套件jsonlite*

install.packages("jsonlite")

library(jsonlite)

*# 讀取json檔案*

fromJSON(txt = "D:/week01/diff\_type\_data/iris\_json.json")

*# 指定路徑，儲存資料集iris，檔案名稱iris\_json，儲存為json檔案*

write\_json(x = iris, path = "D:/week01/diff\_type\_data/iris\_json.json")

另外，還有些套件有優化讀取資料的速度，或是提高對不同資料格式讀取彈性。在此就介紹一個簡單好用的套件readr。

如果有read.csv()讀取不了的csv檔，或許可以試試套件readr中提供的read\_csv()函式；同樣地，如果有read.table()讀取不了的txt檔，或許可以試試套件readr中提供的read\_tsv()函式。

**關於講師**

1. 基本資料

嗨嗨~你好!!我是余佑駿

政大統計不菸不酒研究生，一年前不小心誤入數據分析坑

只是塵世中的迷途小書童，跌跌撞撞地學習中。(單押x4)

喜歡參加比賽，未來想要成為數據分析人員

* 學歷

國立政治大學統計所2017.09~Now

國立清華大學數學系2012.09~2017.08

1. 經歷
2. 實習
3. 國泰世華銀行數位金融─客戶經營部資料分析科分析部門 Intern 2018.7~2018.9
4. 國家高速網路中心 Intern 2018.8~Now
5. 程式教學/講師
6. 2018暑期資料工作坊SQL講師
7. 政大余清祥教授健保資料庫分析團隊
8. 曾在 ladies 閃電秀分享
9. Taiwan Code School新創團隊
10. 比賽/專案
11. 2017 中油大數據CPC Hackathon 第三名
12. D4SG資料英雄專案計畫─第五期
13. 2018 104開放資料黑客松 參與但未入決賽
14. 2018 國泰大數據競賽 進入決賽(ING)
15. T-brain 旅遊訂單成行預測(ING)
16. 2018 新北交通黑客松(ING)
17. 程式技能
18. 資料庫：MSSQL
19. 數據分析：R、Python
20. 演算法：C、C++
21. 與我聯絡：(初次聯絡記得提一下是在哪認識我的!!現在假帳號很多，我有點怕被盜。)

Github：<https://github.com/littlefish0331>

Linkdin：<https://www.linkedin.com/in/佑駿-余-8670b8134/>

FB：<https://www.facebook.com/littlefish0331>

LINE：littlefish0331

Email：[littlefish0331@gmail.com](mailto:littlefish0331@gmail.com)

**貳、引用的參考資料**

註一，圖一參考網址

* + 1. <https://www.ted.com/talks/hans_rosling_shows_the_best_stats_you_ve_ever_seen>
    2. <https://www.gapminder.org/tools/#$state$time$value=1994&delay:100;;&ui$presentation:true;&chart-type=bubbles>
    3. <https://yaojenkuo.shinyapps.io/gapminder_replica/>

註二，圖二參考網址

1. <https://datascienceplus.com/machine-learning-results-one-plot-to-rule-them-all/>

註三~註七

1. <https://redmonk.com/sogrady/2018/08/10/language-rankings-6-18/>
2. <https://www.tiobe.com/tiobe-index/>
3. <https://spectrum.ieee.org/at-work/innovation/the-2018-top-programming-languages>
4. <http://pypl.github.io/PYPL.html>
5. <https://buzzorange.com/techorange/2017/09/14/pytho-win-r-research-by-kdnuggets-2017/>
6. <https://www.kdnuggets.com/2018/05/poll-tools-analytics-data-science-machine-learning-results.html>

註八

從GitHub社群下載套件：<https://cran.r-project.org/web/packages/githubinstall/vignettes/githubinstall.html>

R語言套件一覽表：<https://www.r-pkg.org/>

**參、推薦自學網站與發問社群**

新手有問題請多用Google

高手在民間：Stack overflow是一個程式語言問題發問的平台，上面有許多的人

另外，也有許多平台可以找到不錯的R語言資源。

Rddit、PTT、統計之都、R語言論壇(經管之家)等等

臉書社團推薦：[台灣R語言Club](https://www.facebook.com/groups/1210634969026548/)

R與Python的比較

<https://www.datacamp.com/community/tutorials/r-or-python-for-data-analysis>