

R 語言基礎課程

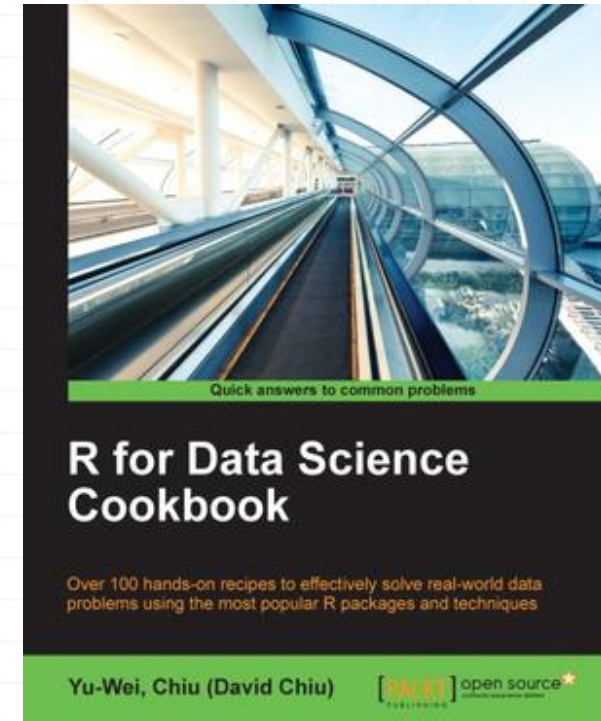
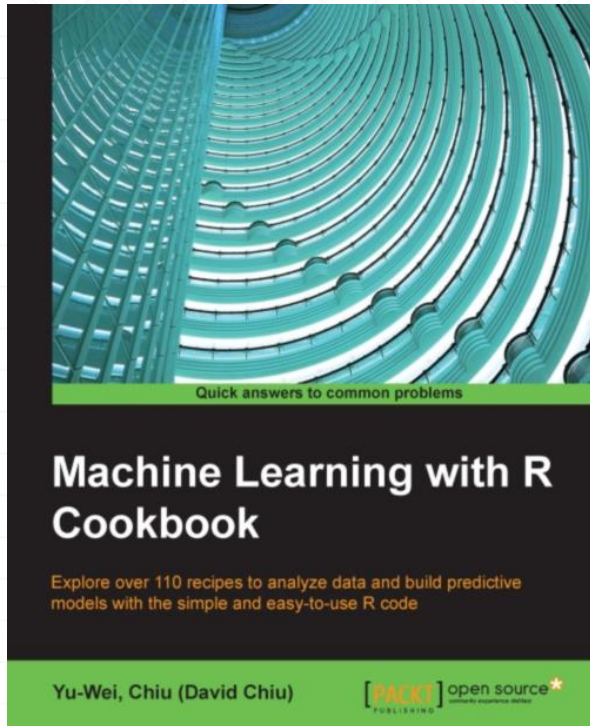
丘祐瑋
David Chiu

關於我



- 大數軟體有限公司創辦人
- 前趨勢科技工程師
- ywchiu.com
- 大數學堂
<http://www.largitdata.com/>
- 粉絲頁
<https://www.facebook.com/largitdata>
- R for Data Science Cookbook
<https://www.packtpub.com/big-data-and-business-intelligence/r-data-science-cookbook>
- Machine Learning With R Cookbook
<https://www.packtpub.com/big-data-and-business-intelligence/machine-learning-r-cookbook>

Machine Learning With R Cookbook (机器学习与R语言实战) & R for Data Science Cookbook (数据科学 R语言实现)



Author: David (YU-WEI CHIU) Chiu

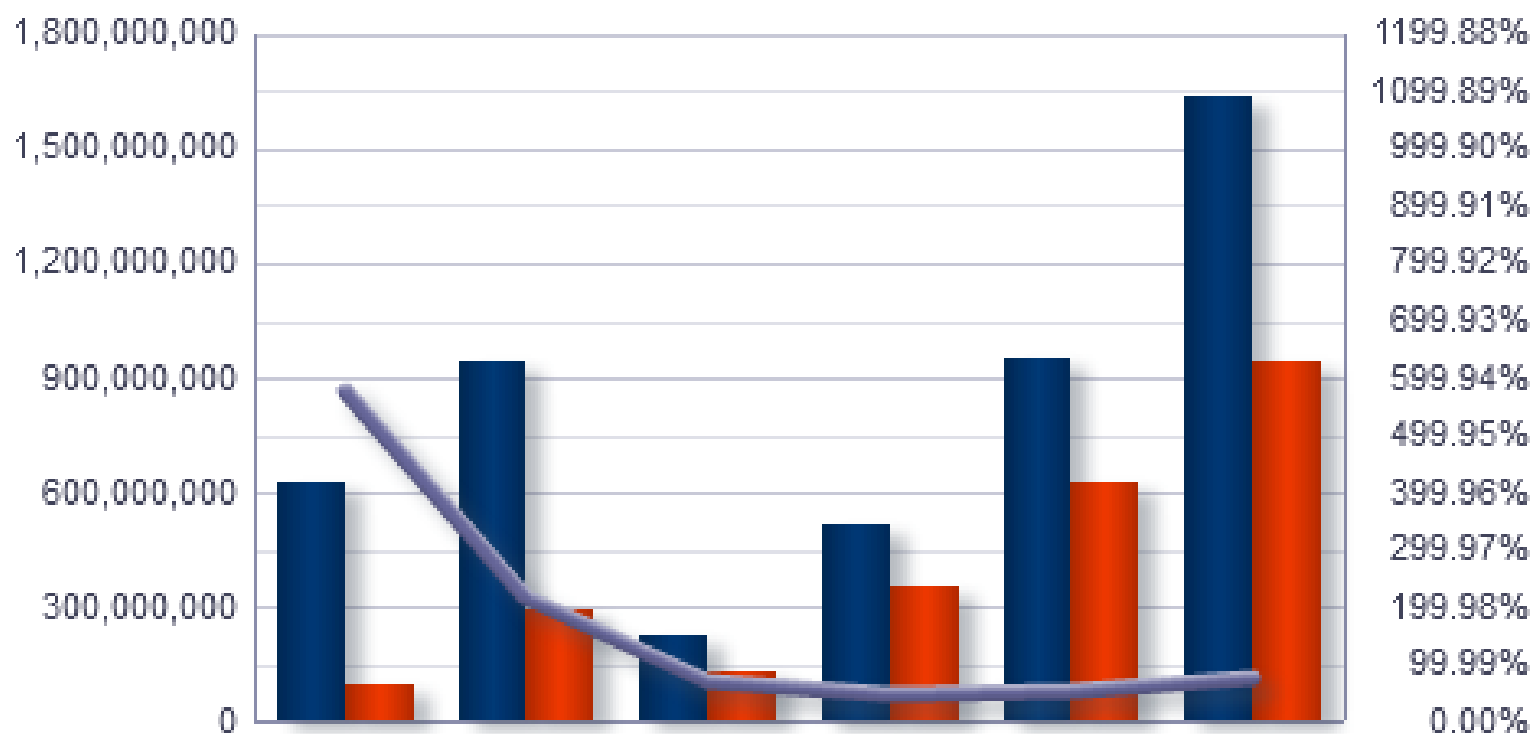
環境資訊頁面

- 所有課程補充資料、投影片皆位於
 - https://github.com/ywchiu/cdc_course

R語言與資料分析

資料分析實作 - 一個簡單的問題

- 試想如果今天主管要你找出哪個年齡層需要接種疫苗的民眾最多，並畫出資料分佈圖的話，該怎麼做？



不同派別的做法

■ 資料庫派的

- 先下個SQL 做個資料聚合
- 使用視覺化工具呈現到報表上
- 或許使用Excel 比較容易些



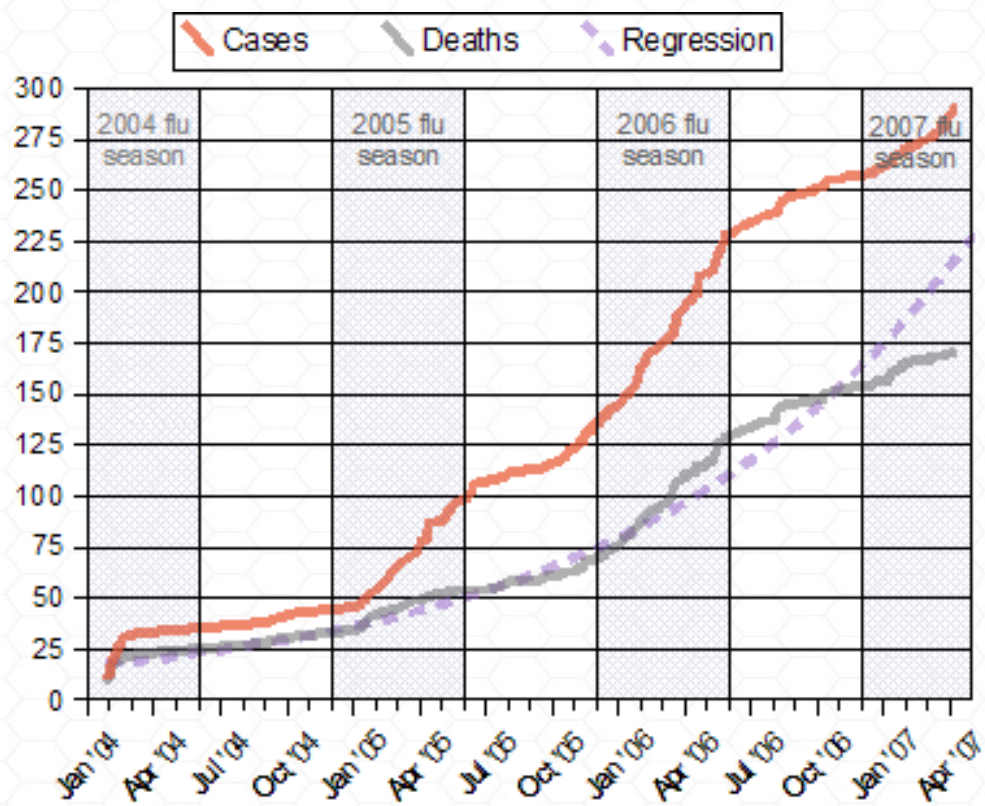
■ 軟體工程師派的

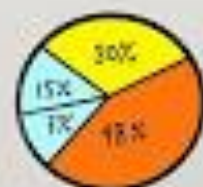
- 寫一個For迴圈掃過資料後，依條件規則進行分組統計
- 使用圖表套件呈現圖表



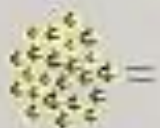
相關性分析 - 更複雜的問題

e.g. 分析H5N1 散播趨勢





idea 45% is SALE!



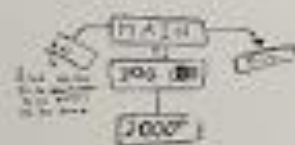
\$



25%
Flow



TEAM



MARKETING

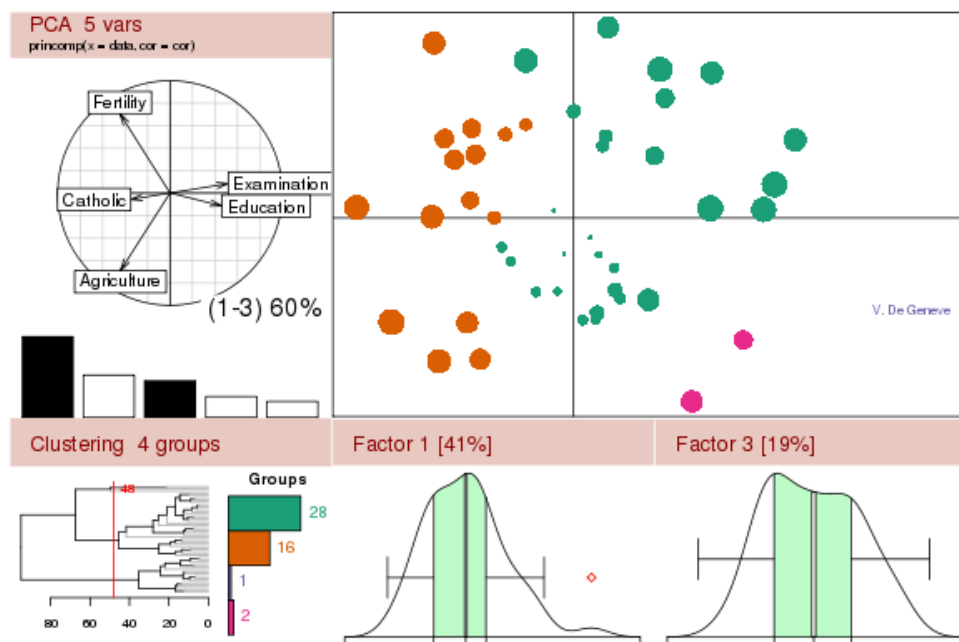


CLOUD



什麼是R

- AT&T貝爾實驗室暨S語言所發展出來的GNU 專案
- 提供統計分析與圖形視覺化功能的開源程式語言
- 使用C, Fortran 做為編程底層的函式(Functional)語言



S 語言

- 1976 年 John Chambers 在貝爾實驗室開發出 S，用來取代SAS 與 SPSS
- 1976 年使用Fortran 實現的第一代 (S Version 1)
- 1978 年支援Linux 系統 (S Version 2)
- 1983 ~ 1992 年引入萬物皆物件的概念 (S version 3)
- 1993 年被MathSoft 買斷，改版為 S-PLUS(當時三大統計軟體之一)
- 1995 年更新後變為 (S Version 4)
- 1998 年S 獲得ACM 的軟體系統獎
- 2008 年S-PLUS 被TIBCO收購

R 語言

- S 語言的方言 (分支)
- 受到函數式編程(Functional Programming)語言Scheme 的啟發，因而想將函數式編程概念加入到 S 語言當中
- 1992年Ross Ihaka 與 Robert Gentleman 為了教授統計，因此開發出了 R語言
- 除了R 以外，還有S-Plus，但兩個分支走向不同，一個走向社群，一個走向商業

為什麼使用R

- 立即完成統計分析
 - 快速資料
 - 簡化資料分析
 - 輕鬆製作報表
- 內建許多數學函式及圖形套件(也可安裝第三方套件)
 - 可以結合其他語言：如Java, C++
- 免費且開源
 - <http://cran.r-project.org/src/base/>
 - 容易擴充和客製化



應用範圍

- 統計分析
- 迴歸分析
- 資料分群
- 資料分類
- 推薦系統
- 文字探勘
- 深度學習

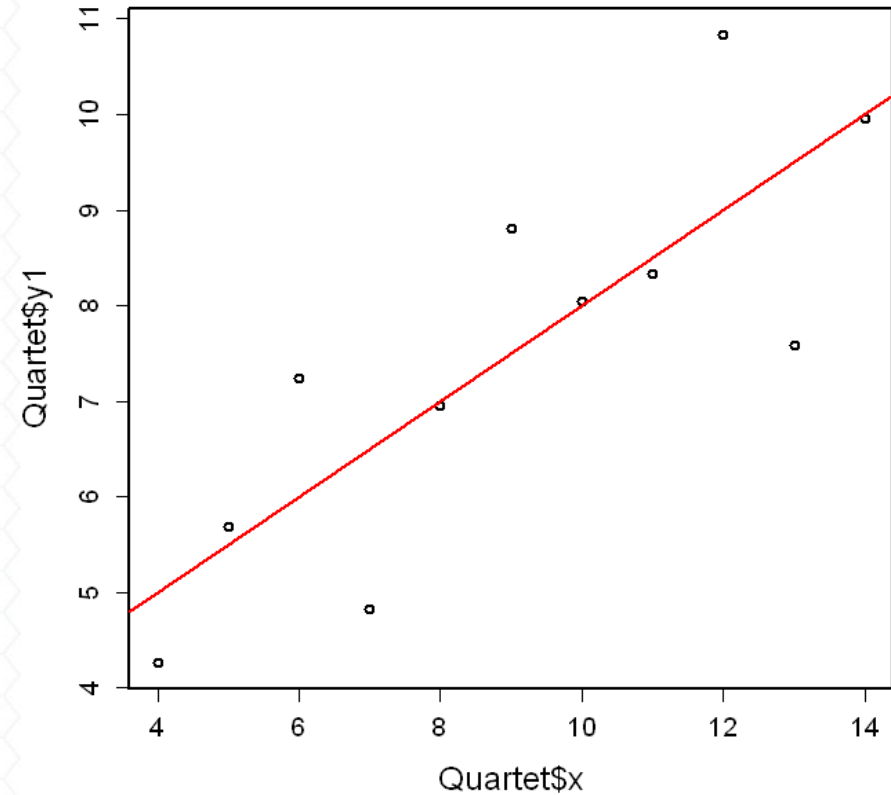


影像辨識

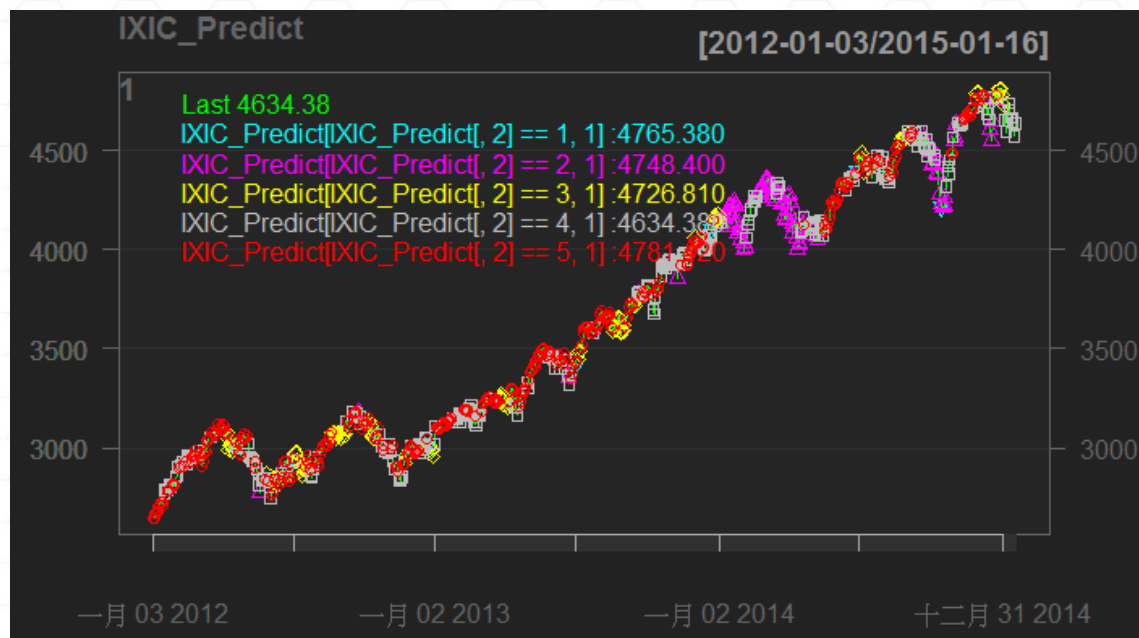


範例 - 用R做簡單迴歸分析

```
data(anscombe)
plot(y1 ~ x1, data = anscombe)
lmfit <- lm(y1~x1, data=anscombe)
abline(lmfit, col="red")
```



更複雜的分析



預測股票漲跌

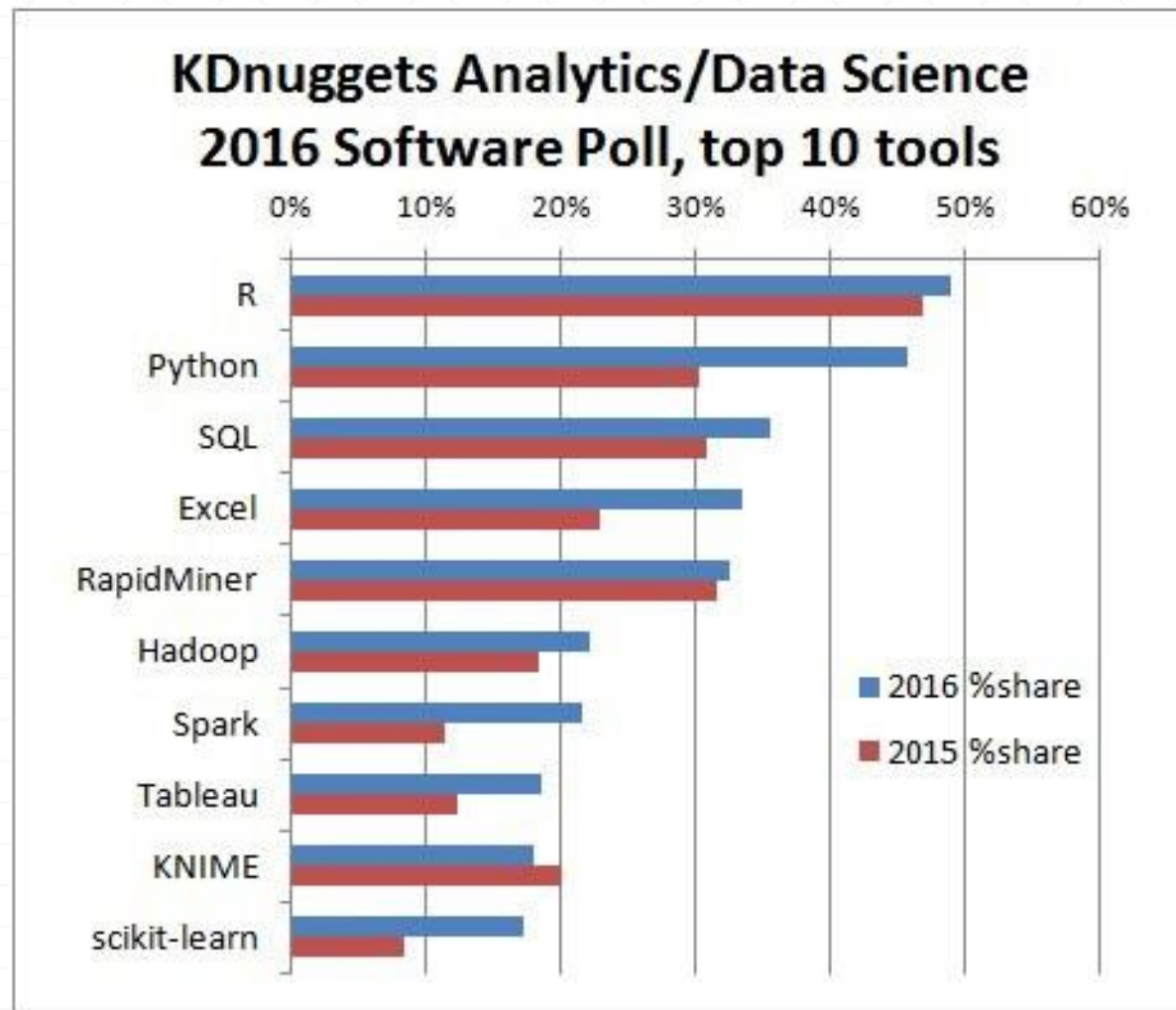
人臉辨識



最廣泛被用來做資料分析的語言

最受歡迎的語言排行為 R,
Python 及 SQL

By KDnuggets 2017.



R語言環境設定

下載R

■ <https://cran.r-project.org/bin/windows/base/>

R-3.4.3 for Windows (32/64 bit)

[Download R 3.4.3 for Windows](#) (62 megabytes, 32/64 bit)

[Installation and other instructions](#)

[New features in this version](#)

If you want to double-check that the package you have downloaded matches the package distributed by CRAN, you can compare the [md5sum](#) of the .exe to the [fingerprint](#) on the master server. You will need a version of md5sum for windows: both [graphical](#) and [command line versions](#) are available.

Frequently asked questions

- [Does R run under my version of Windows?](#)
- [How do I update packages in my previous version of R?](#)
- [Should I run 32-bit or 64-bit R?](#)

Please see the [R FAQ](#) for general information about R and the [R Windows FAQ](#) for Windows-specific information.

Other builds

- Patches to this release are incorporated in the [r-patched snapshot build](#).
- A build of the development version (which will eventually become the next major release of R) is available in the [r-devel snapshot build](#).
- [Previous releases](#)

Note to webmasters: A stable link which will redirect to the current Windows binary release is [<CRAN MIRROR>/bin/windows/base/release.htm](https://cran.r-project.org/bin/windows/base/release.htm).


Last change: 2017-12-06

(選項)下載 Microsoft R Open

■ <https://mran.microsoft.com/download/download-platforms>

Microsoft R Application Network

[Home](#) [About R](#) [Microsoft R Open](#) [R Packages](#) [R Community](#) [R Tools](#)




Find an R Package 

Download Microsoft R Open 3.4.3


Microsoft R Open, **the enhanced distribution of R** from Microsoft, is a complete and free open source platform for statistical analysis and data science. R Open 3.4.3 is based on (and 100% compatible with) the statistical language, R-3.4.3. It includes additional capabilities for performance, reproducibility and platform support. [Learn more...](#)

Microsoft R Open & MKL Downloads

While the install of MKL, used for multithreaded performance, is **optional**, we recommend both Microsoft R Open & MKL for optimal performance on Windows and Linux. Mac OS X includes Math Libraries by default.

Platforms (64-Bit only)	Downloads
Windows - Windows - Windows 7.0 (SP1), 8.1, 10 and Windows Server®, 2008 R2 (SP1), 2012 SHA 256: C8D50172EC8173CEF503A616F9EF310DC5B274AC045EA71B9C8AED248098CBDD	 Download
Ubuntu - Ubuntu - 14.04, 16.04 SHA 256: BF2CD35A11DB604B1FA8F5F0C0ACF0AE05756020D90FB3F6CBB639337EFCEE5B	 Download
RHEL/CentOS - Red Hat Enterprise Linux - 6.5, 7.1 SHA 256: BF2CD35A11DB604B1FA8F5F0C0ACF0AE05756020D90FB3F6CBB639337EFCEE5B	 Download

[Prerequisites & Install Docs](#) | [Forum](#) | [News](#) | [Past Releases](#)



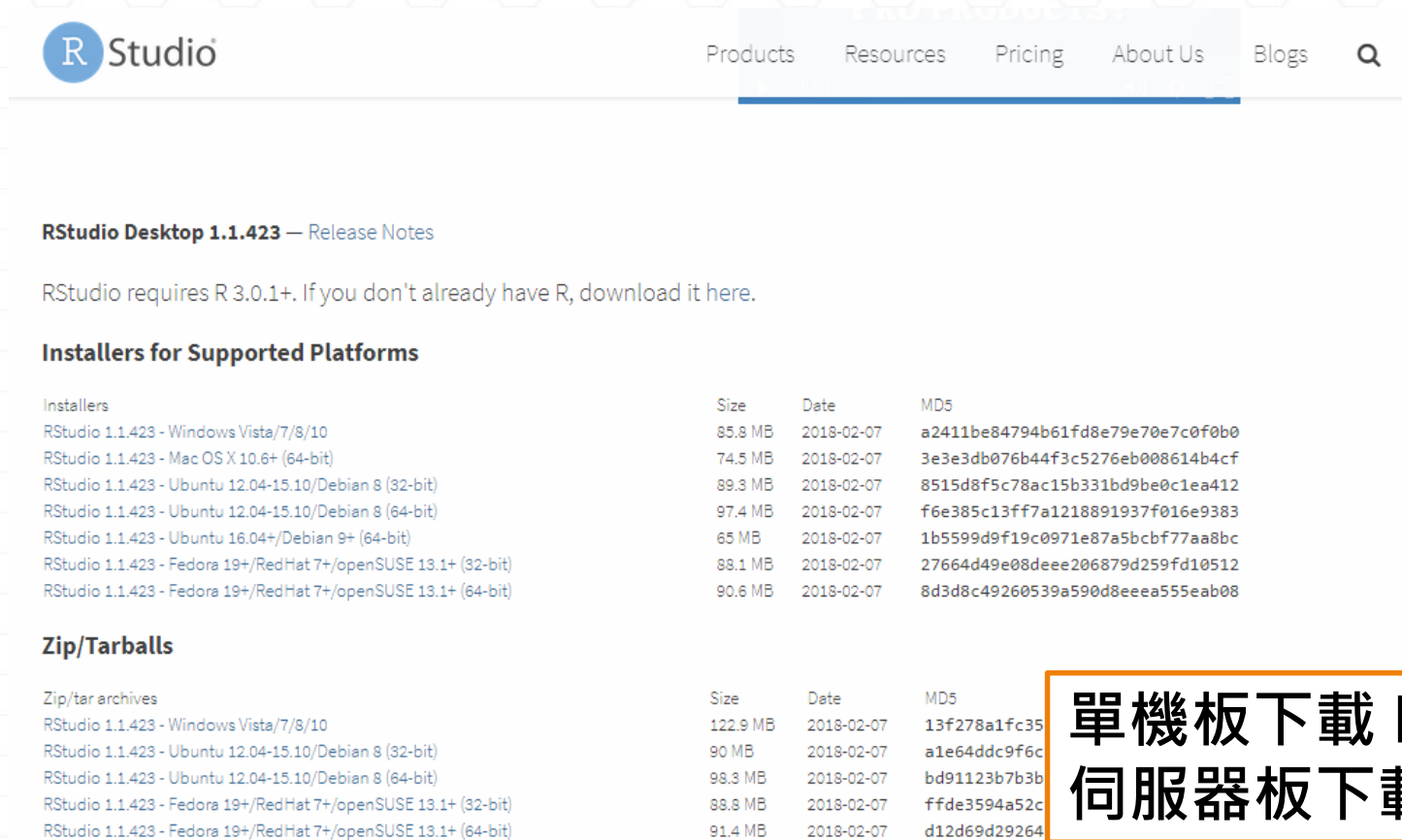
Licensing of R Open and MKL

Microsoft R Open and Revolution R Open are distributed by Microsoft Corporation under the terms of the [General Public License version 2](#).

Use of the Intel Math Kernel Library (**Intel MKL**) is governed by the terms of the [Microsoft R Services MKL End User License Agreement](#).

下載RStudio

■ <https://www.rstudio.com/products/rstudio/download/#download>



The screenshot shows the RStudio website's download page. At the top is the RStudio logo and a navigation bar with links for Products, Resources, Pricing, About Us, and Blogs. The main heading is "RStudio Desktop 1.1.423 — Release Notes". Below this, a paragraph states: "RStudio requires R 3.0.1+. If you don't already have R, download it here." The page is divided into two main sections: "Installers for Supported Platforms" and "Zip/Tarballs". Each section contains a table with columns for the installer name, size, date, and MD5 hash.

RStudio Desktop 1.1.423 — Release Notes

RStudio requires R 3.0.1+. If you don't already have R, download it here.

Installers for Supported Platforms

Installers	Size	Date	MD5
RStudio 1.1.423 - Windows Vista/7/8/10	85.8 MB	2018-02-07	a2411be84794b61fd8e79e70e7c0f0b0
RStudio 1.1.423 - Mac OS X 10.6+ (64-bit)	74.5 MB	2018-02-07	3e3e3db076b44f3c5276eb008614b4cf
RStudio 1.1.423 - Ubuntu 12.04-15.10/Debian 8 (32-bit)	89.3 MB	2018-02-07	8515d8f5c78ac15b331bd9be0c1ea412
RStudio 1.1.423 - Ubuntu 12.04-15.10/Debian 8 (64-bit)	97.4 MB	2018-02-07	f6e385c13ff7a1218891937f016e9383
RStudio 1.1.423 - Ubuntu 16.04+/Debian 9+ (64-bit)	65 MB	2018-02-07	1b5599d9f19c0971e87a5bcbf77aa8bc
RStudio 1.1.423 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (32-bit)	88.1 MB	2018-02-07	27664d49e08deee206879d259fd10512
RStudio 1.1.423 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (64-bit)	90.6 MB	2018-02-07	8d3d8c49260539a590d8eeea555eab08

Zip/Tarballs

Zip/tar archives	Size	Date	MD5
RStudio 1.1.423 - Windows Vista/7/8/10	122.9 MB	2018-02-07	13f278a1fc35
RStudio 1.1.423 - Ubuntu 12.04-15.10/Debian 8 (32-bit)	90 MB	2018-02-07	a1e64ddc9f6c
RStudio 1.1.423 - Ubuntu 12.04-15.10/Debian 8 (64-bit)	98.3 MB	2018-02-07	bd91123b7b3b
RStudio 1.1.423 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (32-bit)	88.8 MB	2018-02-07	ffde3594a52c
RStudio 1.1.423 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (64-bit)	91.4 MB	2018-02-07	d12d69d29264

單機板下載 Desktop 版本
伺服器板下載 Server 版本

Rstudio

編輯區

```
1 library(rvest)
2 appledaily <- html("http://www.berich.com.tw/DP/Cn
3 article <- appledaily %>% html_nodes("table") %>%
```

歷史&環境

```
table(tw2330$tf)
hist(tw2330$Close)
pairs(iris)
pairs(iris, col="iris$Species")
pairs(iris, col=iris$Species)
```

控制臺

```
[Workspace loaded from ~/.RData]
> pairs(iris)
> pairs(iris, col="iris$Species")
Error in plot.xy(xy, type, ...) : invalid color name 'iris$Species'
> pairs(iris, col=iris$Species)
>
```

繪圖&套件&檔案

The Plots/Packages/Help/Viewer pane displays a 5x5 grid of plots for the iris dataset. The diagonal elements are histograms for each variable: Sepal.Length, Sepal.Width, Petal.Length, Petal.Width, and Species. The off-diagonal elements are scatter plots showing the relationships between pairs of these variables. The plots are color-coded by species, with black, red, and green points representing the three species in the dataset.

R 語言基礎

數學運算

數字相加

$3 + 8$

數字相減

$3 - 8$

數字相乘

$5 * 5$

數字相除

$11 / 2$

取指數

2^10

取餘數

$11 \% 2$

可以將R 當成計算機使用



設定變數

指定變數

`a <- 3`

`a`

變數相加

`b = 5`

`c <- a + b`

`c`

可以使用 = 或 <- 指定變數
為了提高程式的可讀性，建議使用 <- 為主

基礎資料型態

數值型態

numer <- 17.8

字串型態

char <- "hello world"

布林邏輯

logic <- TRUE

使用 class 檢查資料型態

class(logic)

不同型態資料做運算

```
card_length <- 3
```

```
card_width <- "5 inches"
```

```
card_length * card_width
```

```
Error in card_length * card_width :  
  non-numeric argument to binary operator
```

```
#重新將card_width 指到5
```

```
card_width <- 5
```

```
card_length * card_width
```

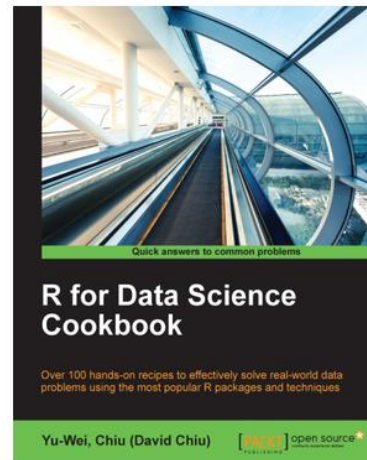

範例:計算一本書的價錢

RRP <- 35.99

Exchange <- 31.74

NTD <- RRP * Exchange

NTD



R for Data Science Cookbook

Yu-Wei, Chiu (David Chiu)
July 2016



★★★★★ feefo
1 customer reviews

Over 100 hands-on recipes to effectively solve real-world data problems using the most popular R packages and techniques

\$35.99

RRP \$35.99

eBook

Print + eBook



Add to Cart

向量 (Vector)

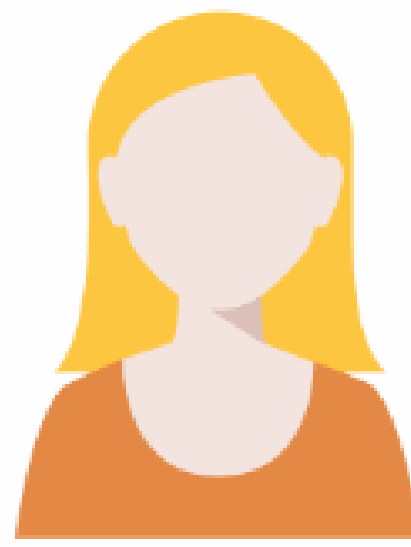
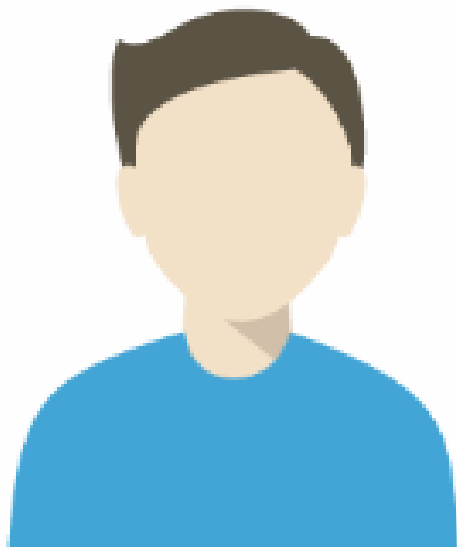
使用向量存放多個變數的資料

不同型態的向量

```
height_vec <- c(180,169,173)
```

```
name_vec <- c("Brian", "Toby", "Sherry")
```

向量表示方式: c()



向量運算

產生兩個向量

```
x <- c(1,2,3,4)
```

```
y <- c(2,3,4,5)
```

將兩個向量進行數學運算

```
x+y
```

```
x*y
```

```
x - y
```

```
x/y
```

在R 裡面，所有資料都是以向量表示
因此當我們指 $a <- 1$ 時，其實是代表 $a <- c(1)$

產生包含連續數字的向量

但如果要產生 1 ~ 20 呢?
除了用 `c(1,2,3,4...)` 下外有其他方法嗎?

■ 產生1到20

```
x <- 1:20
```

```
x
```

■ 或使用seq

```
y <- seq(1,20)
```

```
y
```

那什麼是 `seq()` ？

■ 使用? 或help 去觀看seq 的用法

?seq

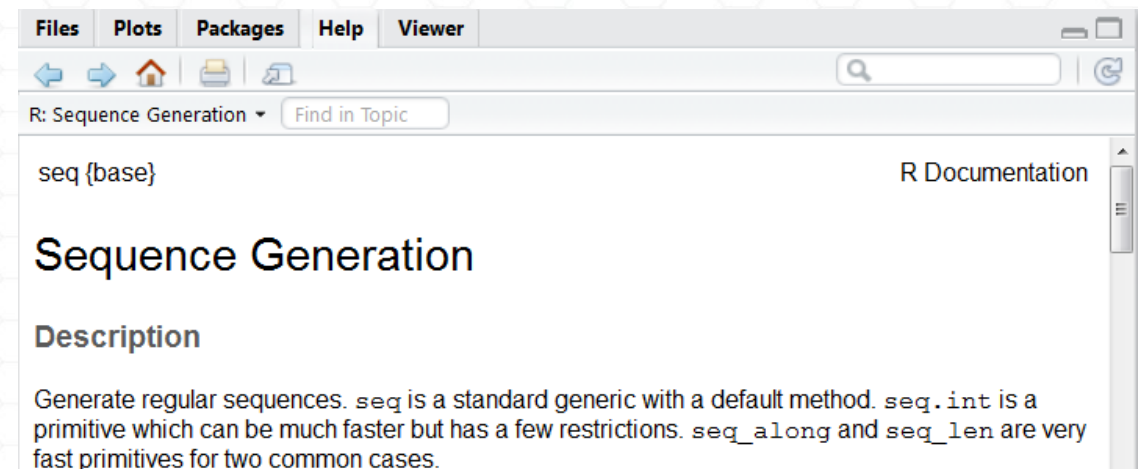
help(seq)

■ 試著使用seq 產生不同類型向量

`seq(1,20,2)`

`seq(1,3.5, by =0.5)`

`seq(1,10,length=2)`



加總向量內的所有元素

透過sum 將向量資料作加總

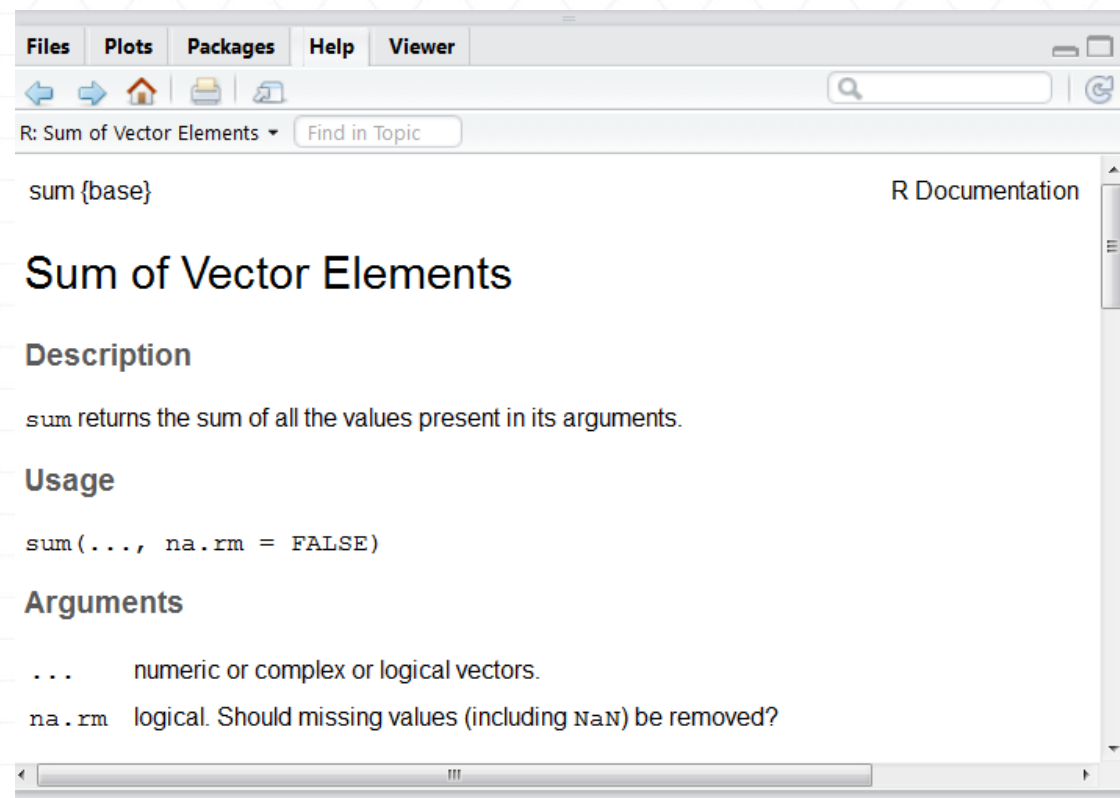
```
x <- c(1,2,3,5,7)
```

```
sum(x)
```

查詢該如何使用sum函式

```
?sum
```

```
help(sum)
```



指定向量名稱

- 可以使用names 指定向量名稱

```
height_vec <- c(180,169,173)
```

```
height_vec
```

```
names(height_vec) <- c("Brian", "Toby", "Sherry")
```

```
height_vec
```

```
name_vec <- c("Brian", "Toby", "Sherry")
```

```
names(height_vec) <- name_vec
```

如果我要知道每個身高的
主人名稱?

判斷向量內容是否符合條件

- 找出哪些資料符合條件 (TRUE/FALSE)

`height_vec > 175`

`height_vec < 175`

`height_vec >= 175`

`height_vec <= 175`

`height_vec == 180`

`height_vec != 180`

- 使用 & (and) 找出符合條件的資料

`height_vec >= 170 & height_vec < 180`

- 使用 | (or) 找出符合條件的資料

`height_vec >= 180 | height_vec < 170`

- 篩選出符合條件的值

`height_vec[height_vec > 175]`

練習題：使用向量計算BMI

- Brian的身高為180, 體重是73公斤;Toby身高是169公分, 體重是87公斤; Sherry 身高為173公分,體重是 43公斤。請用Vector找出誰的BMI是異常的?
- BMI值計算公式: $BMI = \text{體重(公斤)} / \text{身高}^2(\text{公尺}^2)$

	身體質量指數(BMI) (kg/m ²)
體重過輕	BMI < 18.5
正常範圍	18.5 ≤ BMI < 24
異常範圍	過重: 24 ≤ BMI < 27
	輕度肥胖: 27 ≤ BMI < 30
	中度肥胖: 30 ≤ BMI < 35
	重度肥胖: BMI ≥ 35

矩陣 (Matrix)

建立矩陣

■ 學生兩次考試的成績

```
kevin <- c(85,73)
```

```
marry <- c(72,64)
```

```
jerry <- c(59,66)
```

■ 從向量中建立矩陣

```
mat <- matrix(c(kevin, marry, jerry), nrow=3, byrow= TRUE)
```

如果要表示多個人的多次考試成績，除了為每個人建立各自的向量外，還有其他方式嗎？

依不同方向產生矩陣

■ 依欄(column)產生3列矩陣

`matrix(1:9, nrow=3)`

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

■ 依列(row)產生3列矩陣

`matrix(1:9, byrow=TRUE, nrow=3)`

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

新增欄位與列的名稱

```
colnames(mat) <- c('first', 'second')  
rownames(mat) <- c('kevin', 'marry', 'jerry')
```

或是

```
mat <- matrix(c(kevin, marry, jerry), nrow=3, byrow=TRUE,  
dimnames=list(c('kevin', 'marry', 'jerry'), c('first', 'second')))
```

如果要增加學生名稱與考試次數等文字敘述？

取矩陣維度、列與欄數

■ 取維度

`dim(mat)`

■ 取列數

`nrow(mat)`

■ 取行數

`ncol(mat)`

3 列 (row)

2 欄 (Col)

	first	second
kevin	85	73
marry	72	64
jerry	59	66

3 X 2 矩陣

依欄或列取矩陣資料

- 取第一列

`mat[1,]`

- 取第一行

`mat[,1]`

- 取第二、三列

`mat[2:3,]`

- 取第二列第一行的元素

`mat[2,1]`

利用[] 分別取得列與欄的資訊
，前面是列
，後面是欄

[列, 欄]

新增列與欄

- 新增學生資料
(增加第四個學生資料)

```
mat2 <- rbind(mat, c(78,63))  
rownames(mat2)[4] <- 'sam'  
mat2
```

	first	second
kevin	85	73
marry	72	64
jerry	59	66
sam	78	63

- 新增考試分數
(增加第三次考試成績)

```
mat3 <- cbind(mat, c(82,77,70))  
colnames(mat3)[3] <- 'third'  
mat3
```

	first	second	third
kevin	85	73	82
marry	72	64	77
jerry	59	66	70

矩陣運算

■ 矩陣宣告

```
m1 <- matrix(1:4, byrow=TRUE, nrow=2)
```

```
m2 <- matrix(5:8, byrow=TRUE, nrow=2)
```

■ 矩陣運算

```
m1 + m2
```

```
m1 - m2
```

```
m1 * m2
```

```
m1 / m2
```

如同向量一樣，可以直接對矩陣進行加減乘除

使用rowSums 及colSums

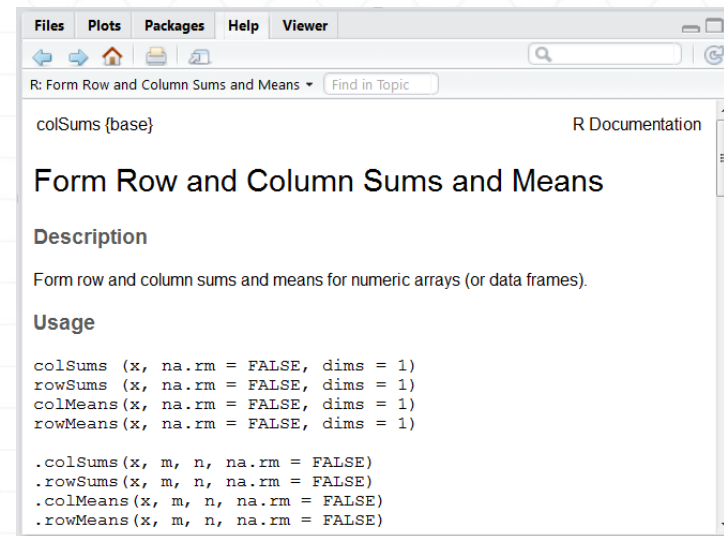
- 使用rowSums 及 colSums 針對列及欄加總

rowSums(mat2)

colSums(mat2)

- 善用? 查詢 rowSums 及 colSums 的用法

?rowSums



矩陣乘積

■ m1 X m2

m1 %*% m2

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} = \begin{bmatrix} 1 * 5 + 2 * 7 & 1 * 6 + 2 * 8 \\ 3 * 5 + 4 * 7 & 3 * 6 + 4 * 8 \end{bmatrix}$$

2 X 2 矩陣

2 X 2 矩陣

=

2 X 2 矩陣

那如果要產生一個九九乘法表呢？

使用矩陣計算考試成績

- 學生兩次考試的成績

```
kevin <- c(85,73)
```

```
marry <- c(72,64)
```

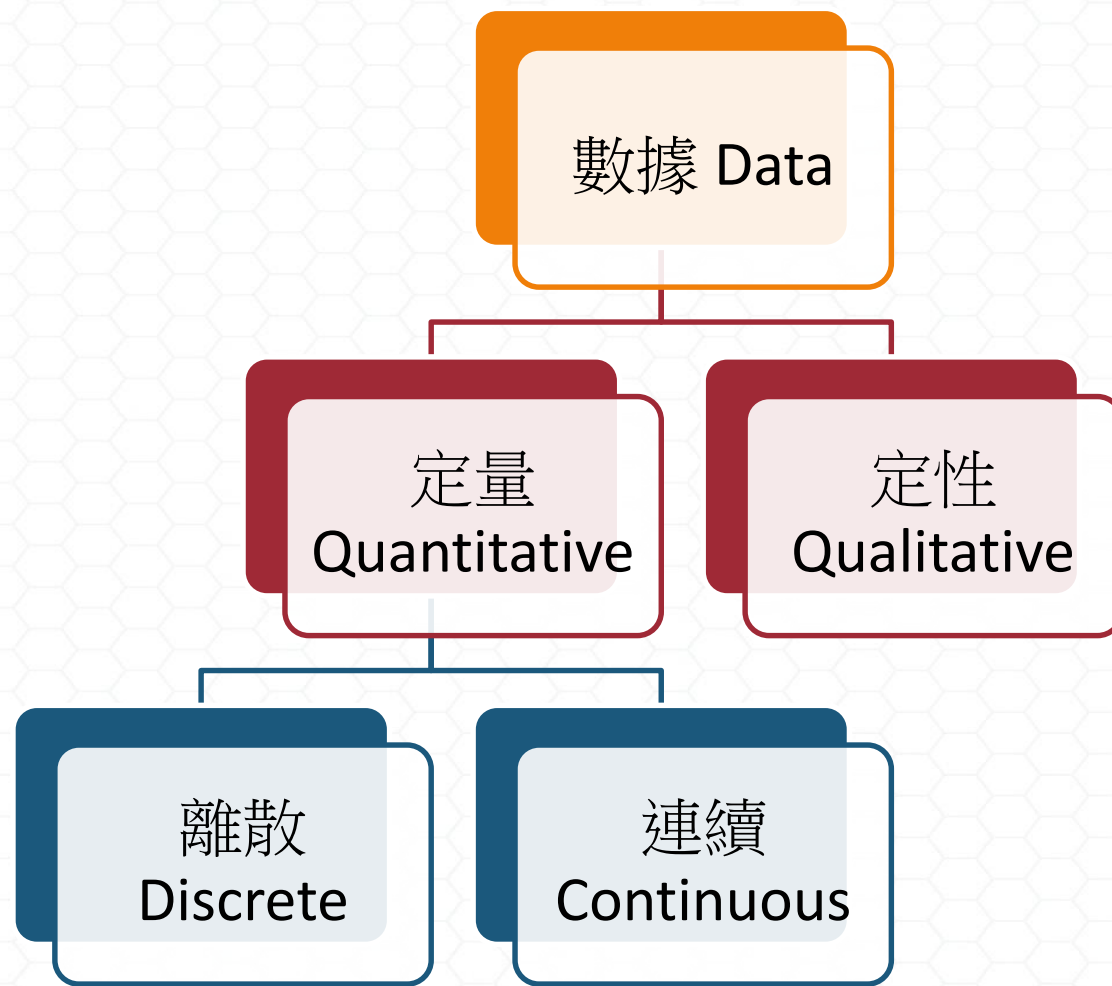
```
jerry <- c(59,66)
```

```
mat <- matrix(c(kevin, marry, jerry), nrow=3, byrow= TRUE)
```

- 如果老師希望給每個人最後總成績，以加權為第一次考試佔40%，第二次佔60%；請問該怎麼用矩陣運算達成？

階層 (Factor)

數據的種類



數據型態

■ 定性資料 (Qualitative or Categorical Data)

- 敘述特性或種類

- e.g. 住在哪一區, 哪個種族的人

■ 定量資料 (Quantitative or Numerical Data)

- 可以被計數或測量

- e.g. 身高、消費金額

定量資料類型

■ 離散數據 (Discrete Data)

- 只能用自然數或整數單位計算
- 只能按計量單位數計數，可由一般計數方法取得
- e.g. 員工人數

■ 連續資料 (Continuous Data)

- 一定區間內可以任意取值的數據，其數值是連續不斷的，相鄰兩個數值可取無限個數值
- 其數值只能用測量或計量的方法取得
- e.g. 零件的規格尺寸

使用階層將資料轉換為定性數據(Factor)

- 使用Factor 將字串轉換成階層

```
weather <- c("sunny","rainy", "cloudy", "rainy", "cloudy")
```

```
weather_category <- factor(weather)
```

```
weather_category
```

- 善用levels 檢查有哪些類別

```
levels(weather_category)
```

character 跟 Factor 屬於不同東西
請善用class 檢查資料型態

有順序的階層

■ 產生可比較的類別資訊

```
temperature <- c("Low", "High", "High", "Medium", "Low", "Medium")  
temperature_category <- factor(temperature, order = TRUE, levels =  
c("Low", "Medium", "High"))
```

■ 產生類別大小

```
temperature_category[3] > temperature_category[1]  
temperature_category[4] > temperature_category[3]
```

■ 檢查類別

```
levels(temperature_category)
```

Data Frame

建立Data Frame

■ 建立 Vector

```
days <- c('mon','tue','wed','thu','fri')
```

```
temp <- c(22.2,21,23,24.3,25)
```

```
rain <- c(TRUE, TRUE, FALSE, FALSE, TRUE)
```

■ 使用 Vector 建立Data Frame

```
df <- data.frame(days,temp,rain)
```

```
df
```

在Matrix中，所有資料必須是同一資料型態，但如果要混雜不同型態資料呢？Data Frame

檢視 Data Frame

檢視資料形態

`class(df)`

檢視架構

`str(df)`

檢視資料摘要

`summary(df)`

使用R 內建的資料集

■ 表列資料集

`data()`

■ 使用資料集

`data(iris)`

■ 觀察讀取到的資料集型態

`class(iris)`

Sepal length ↕	Sepal width ↕	Petal length ↕	Petal width ↕	Species ↕
5.1	3.5	1.4	0.2	<i>I. setosa</i>
4.9	3.0	1.4	0.2	<i>I. setosa</i>
4.7	3.2	1.3	0.2	<i>I. setosa</i>
4.6	3.1	1.5	0.2	<i>I. setosa</i>
5.0	3.6	1.4	0.2	<i>I. setosa</i>
5.4	3.9	1.7	0.4	<i>I. setosa</i>
4.6	3.4	1.4	0.3	<i>I. setosa</i>
5.0	3.4	1.5	0.2	<i>I. setosa</i>



Iris 資料集

■ http://en.wikipedia.org/wiki/Iris_flower_data_set



Iris setosa



Iris versicolor



Iris virginica

觀看資料集的前幾筆資料與後幾筆資料

■ 觀看前幾筆資料

`head(iris)`

`head(iris, 10)`

■ 觀看後幾筆資料

`tail(iris)`

`tail(iris, 10)`

請善用?檢視
函式說明

取得指定列與行的部分資料集

- 取前三列資料

```
iris[1:3,]
```

- 取前三列第一行的資料

```
iris[1:3,1]
```

- 也可以用欄位名稱取值

```
iris[1:3,"Sepal.Length"]
```

- 取前兩行資料

```
iris[,1:2]
```

取特定欄位向量值

```
iris$"Sepal.Length"
```

df[列, 欄]

資料篩選

- 取前五筆包含length 及 width 的資料

```
five.Sepal.iris <- iris[1:5, c("Sepal.Length", "Sepal.Width")]
```

- 可以用條件做篩選

```
setosa.data <- iris[iris$Species=="setosa", 1:5]
```

- 使用which 取得符合資料的位置

```
which(iris$Species=="setosa")
```

資料排序

- 用Sort 作資料排序

```
sort(iris$Sepal.Length, decreasing = TRUE)
```

- 用order做資料排序 (order 可以取得排序後的位置)

```
iris[order(iris$Sepal.Length, decreasing = TRUE),]
```

範例：健保門診與住院人數 (腸病毒) 分析

■ 分析健保門診及住院就診人次統計-腸病毒

□ <https://data.cdc.gov.tw/dataset/hi-outpatient-emergency-visit-enteroviral-infection>



清單(Lists)

清單(Lists)

- 如果要混雜不同的資料型態

```
phone <- list(thing="iphone X" , height=5.65, width=2.79 )
```

```
phone
```

- 使用\$取得內容物

```
student <- list(name="Toby", score = c(87,57,72))
```

```
student$score
```

清單(Lists) (續)

- 沒有名字的清單

```
li <- list(c(3,5,12), c(2,4,5,8,10))
```

```
li
```

```
[[1]]  
[1] 3 5 12  
  
[[2]]  
[1] 2 4 5 8 10
```

- 取得第一筆資料

```
li[[1]]
```

- 使用lapply將函式套用到list 上 (迴圈函數)

```
lapply(li, sum)
```

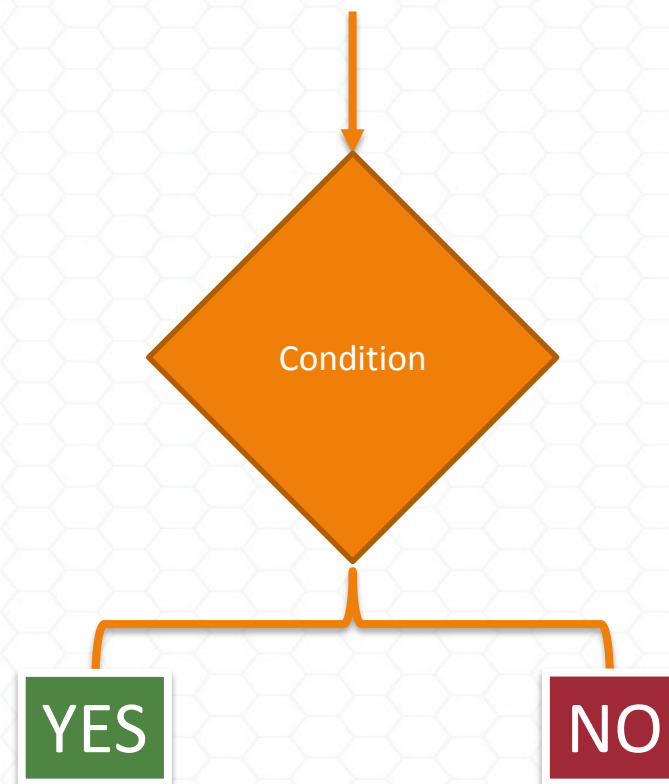

流程控制(Flow Control)

IF...ELSE...

■ If 及 else 的判斷

$x = 5$

```
if(x > 3){  
    print("x > 3")  
}else{  
    print("x <= 3")  
}
```



IF...ELSE IF...ELSE

■ 使用else if

```
x <- 5
```

```
if(x > 3){  
  print("x > 3");  
} else if(x == 3){  
  print("x == 3")  
} else{  
  print("x < 3")  
}
```


FOR 迴圈

■ For 迴圈

```
for(i in 1:10){  
  print (i)  
}
```

■ 1~100的總和 (盡量避免這樣做)

```
s <- 0  
for(i in 1:100){  
  s <- s + i  
}  
s
```

R 在For 迴圈的運算特別緩慢
盡量使用內建函式與向量化計算
`sum(1:100)`

三種FOR 迴圈

```
x <- c("sunny", "rainy", "cloudy", "rainy", "cloudy")
```

```
for(i in 1:length(x)) {  
  print(x[i])  
}
```

```
for(i in seq_along(x)) {  
  print(x[i])  
}
```

```
for(letter in x) {  
  print(letter)  
}
```

使用while 迴圈

- 當不滿足while中定義的條件時，才會跳出迴圈

```
s <- 0;  
cnt <- 0;  
while(cnt <= 100){  
  s <- s + cnt;  
  cnt <- cnt + 1;  
}  
s
```

盡量使用內建函式與向量化計算
`sum(1:100)`

範例：產生多筆頁面連結

```
url <- 'https://tw.appledaily.com/new/realtime/'
```

```
for (i in seq(1,10)){  
  print(paste0(url, i))  
}
```

如何產生每頁的連結？

e.g. <https://tw.appledaily.com/new/realtime/2>

1 2 3 4 5 6 7 8 9 10 下10頁

函式 (Function)

函式 (Function)

- 回傳值為最後被執行的語句

```
f = function(<arguments>) {  
    #任何腳本  
}
```

- 可帶預設參數

```
f = function(a, b = 2, c = NULL) {  
}
```

DRY: Don't Repeat Yourself



建立一個簡單函式

- 將參數a 與 b 相加後回傳加總後的值

```
addNum <- function(a = 2, b = 3) {  
  s <- a + b  
  s  
}
```

- 帶參數a 與 b 的運行結果

```
addNum(3,5)
```

- 不帶參數的運行結果

```
addNum()
```

Lazy Function

```
f <- function(a, b) {  
  a * 2  
}  
f(3)
```

```
[1] 6
```

```
f <- function(a, b) {  
  print(a+ b)  
}  
f(3)
```

Error in print(a + b) : argument
"b" is missing, with no default

範例：撰寫函式計算文章詞頻

■ 計算新聞中各個詞出現的次數

```
f <- file('https://raw.githubusercontent.com/ywchiu/cdc_course/master/data/disease.txt')
article <- readLines(f)
close(f)
```

```
wordcount <- function(article){
  article.split <- strsplit(article, ' ')
  article.vec <- unlist(article.split)
  table(article.vec)
}
```

```
wordcount(article)
```



迴圈函式

使用迴圈函式

- `lapply`: 將函式套用在清單(List)上的每一元素
- `sapply`: 產生較`lapply`簡化的結果
- `apply`: 將函式套用在陣列(array)中
- `tapply`: 套用函式在向量(vector)的部分子集合

lapply

```
x <- list(c(1,2,3,4), c(5,6,7,8))  
lapply(x, sum)
```


套用在陣列清單中

```
m1 <- matrix(1:4, byrow=TRUE, nrow=2)
```

```
m2 <- matrix(5:8, byrow=TRUE, nrow=2)
```

```
li <- list(m1, m2)
```

```
lapply(li, mean)
```

串接匿名函式

- 可串接匿名函式於輸入參數中

```
lapply(li,function(e) e[1,])
```

apply

```
x <- list(c(1,2,3,4), c(5,6,7,8))  
apply(x, sum)
```

■ 與lapply 做比較

```
x <- list(c(1,2,3,4), c(5,6,7,8))  
lapply(x, sum)
```


更多supply

```
m1 <- matrix(1:4, byrow=TRUE, nrow=2)
```

```
m2 <- matrix(5:8, byrow=TRUE, nrow=2)
```

```
li <- list(m1, m2)
```

```
supply(li, mean)
```

```
supply(li,function(e) e[1,])
```

使用 apply

```
m <- matrix(1:4, byrow=TRUE, nrow=2)
```

```
apply(m, 1, sum) # rowSums
```

```
apply(m, 2, sum) # colSums
```

tapply

```
x <- c(80,70,59,88,72,57)
```

```
t <- c(1,1,2,1,1,2)
```

```
tapply(x,t, mean)
```


使用 `tapply` 進行分組計算

```
data(iris)
```

```
tapply(iris$Sepal.Length, iris$Species, mean)
```

The background features a light blue hexagonal grid pattern. Overlaid on this are several concentric, semi-transparent circles in shades of blue and teal. A dark blue horizontal line runs across the top of the image, and a darker teal horizontal band is at the bottom. The text "THANK YOU" is centered in a bold, dark blue font.

THANK YOU