



---

# 遠東國際商業銀行 R 語言教育訓練

丘祐瑋 – David Chiu

EMAIL: [david@largitdata.com](mailto:david@largitdata.com)

網站: [www.largitdata.com](http://www.largitdata.com)

電話: +886929094381

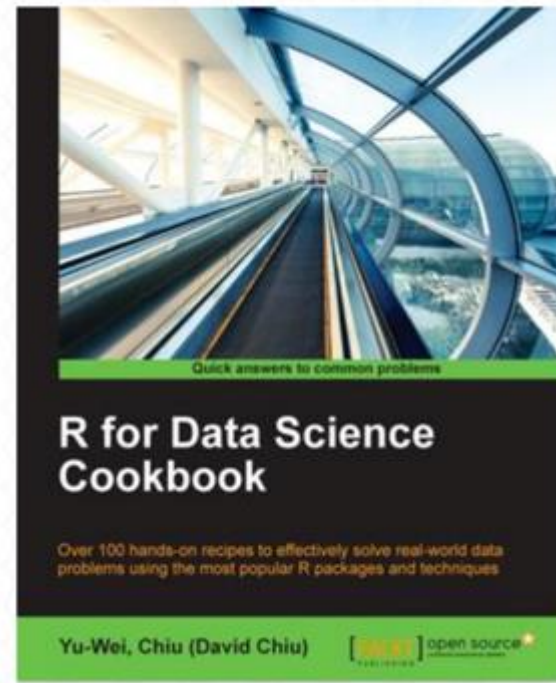
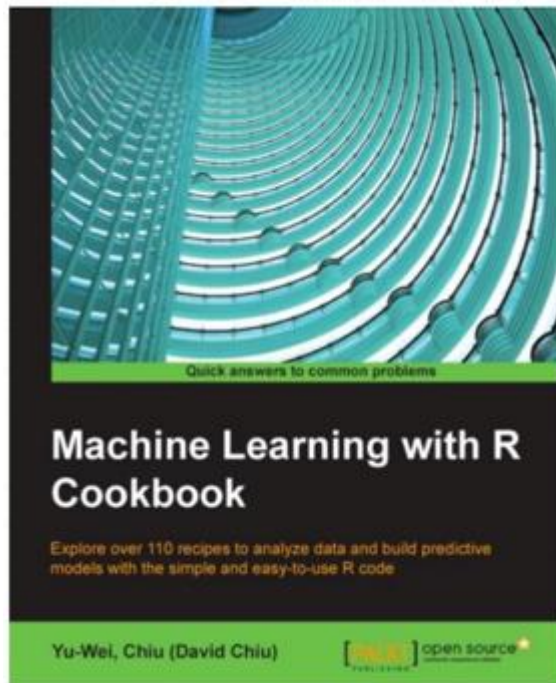
# 關於我

---



- 大數軟體有限公司創辦人
- 前趨勢科技工程師
- [ywchiu.com](http://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 (機器學習與R語言實戰) & R for Data Science Cookbook



Author: Yu-Wei (David) Chiu

# 課程資料

---

本日課程資料放置在:

<https://github.com/ywchiu/feibr>

本日課程程式碼放置在:

<http://rpubs.com/ywchiu/feibr>



# 什麼是資料科學？

---

# 資 料 科 學

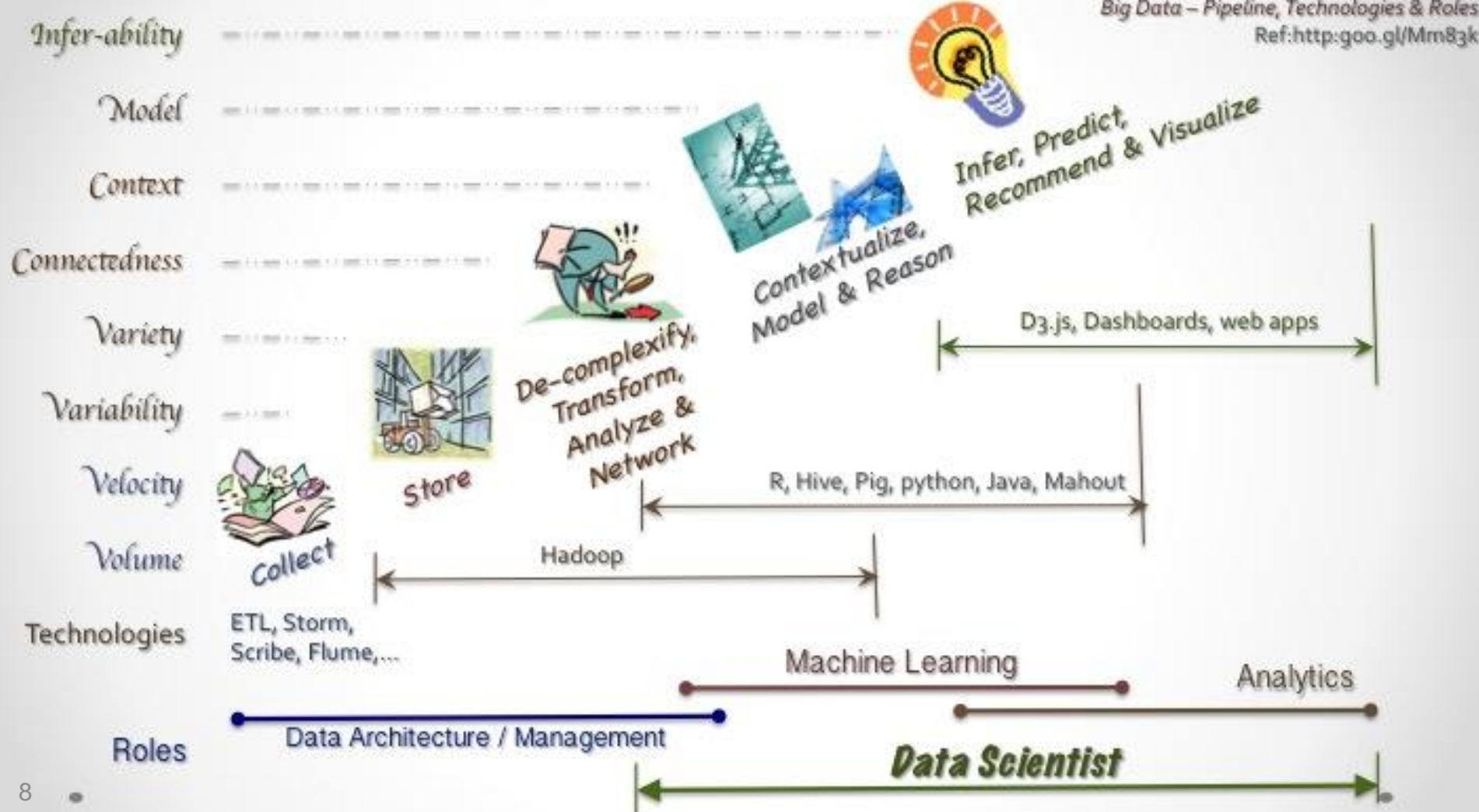
從資料鑒往

從資料知來

# 使用資料擬定策略









# 資料科學家主要工作內容

---

**“80% 都在做加總與平均”**

## 工作內容

資料處理 (*Data Munging*)

真正能用在資料分析的時間很少，必須要  
能善用工具

資料分析 (*Data Analysis*)

詮釋結果 (*Interpret Result*)

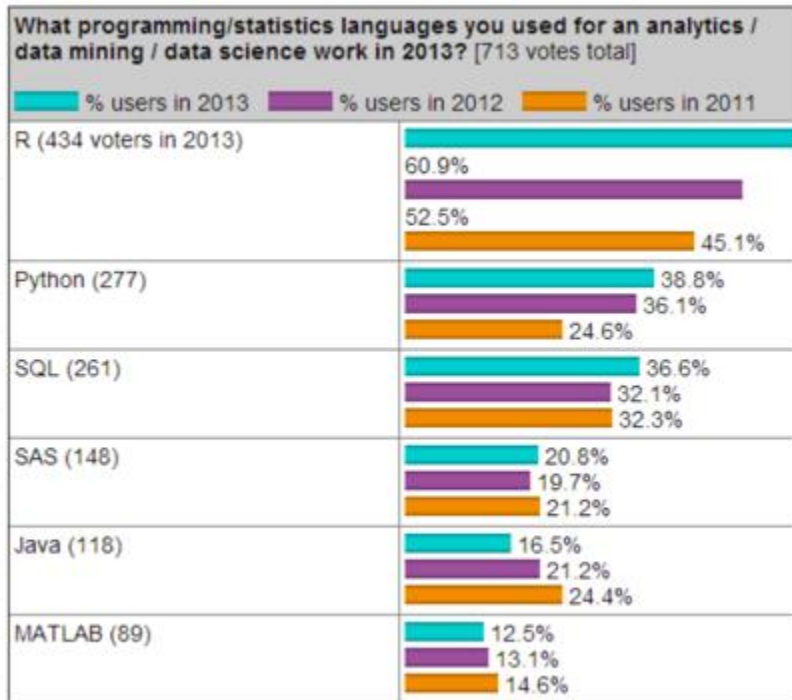
# 資料分析語言



# 資料分析語言

最受歡迎的語言持續為 R, Python (39%), 及 SQL (37%). SAS 大約在 20% 上下.

By Gregory Piatetsky, Aug 27, 2013.



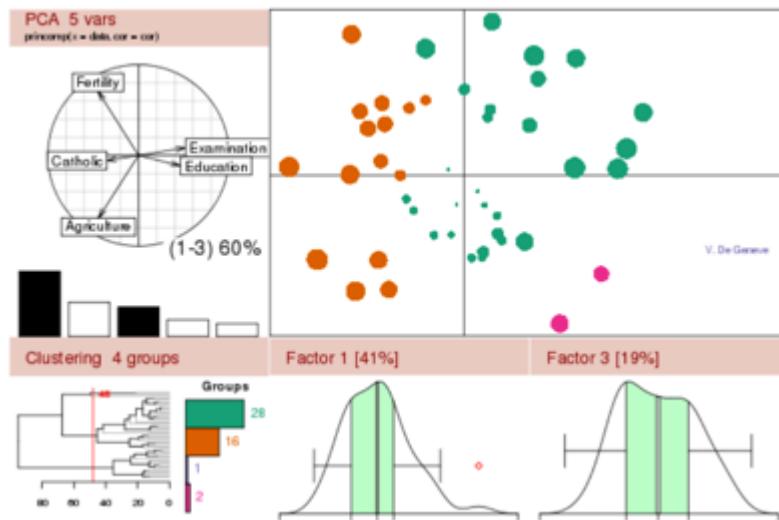


# R 語言與數據分析

---

# R 語言

- 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來源程式語言
- 使用C, Fortran 程式設計的函式語言



# R 語言

---

- S 語言的方言 (分支)
- 受到函數式程式設計語言Scheme 的啟發，因而想將該功能加入到 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

# R 語言

---

立即完成統計分析

- 數據處理
- 資料分析
- 報表製作



內建許多數學函式及圖形套件(也可安裝協力廠商套件)

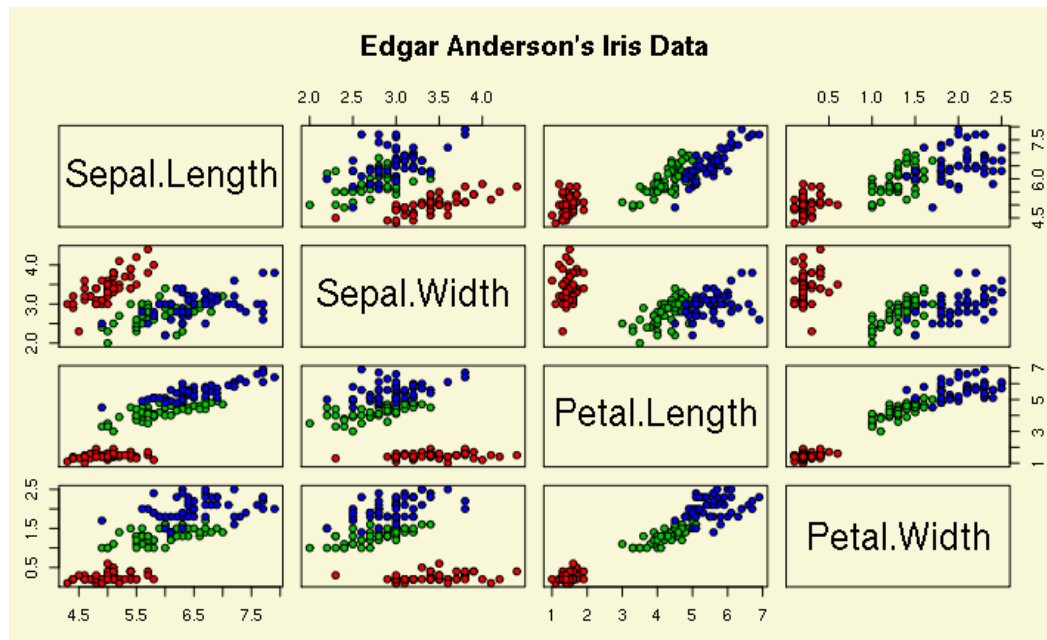
- 可以結合其他語言：如Java, C++
- 免費且開源 <http://cran.r-project.org/src/base/>

容易擴充和客制化



# 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



# 影像辨識

---





# R 語言基礎

---

# 下載R

---

## R-3.4.1 for Windows (32/64 bit)

[Download R 3.4.1 for Windows](#) (75 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: 2017-06-30, by Duncan Murdoch

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

# 下載RStudio

[rstudio::conf](#)[Products](#)[Resources](#)[Pricing](#)[About Us](#)[Blogs](#)

## RStudio Desktop 1.0.153 — Release Notes

RStudio requires R 2.11.1+. If you don't already have R, download it [here](#).

### Installers for Supported Platforms

Installers	Size	Date	MD5
RStudio 1.0.153 - Windows Vista/7/8/10	81.9 MB	2017-07-20	b3b4bbc82865ab105c21cb70b17271b3
RStudio 1.0.153 - Mac OS X 10.6+ (64-bit)	71.2 MB	2017-07-20	8773610566b74ec3e1a88b2fdb10c8b5
RStudio 1.0.153 - Ubuntu 12.04-15.10/Debian 8 (32-bit)	85.5 MB	2017-07-20	981be44f91fc07e5f69f52330da32659
RStudio 1.0.153 - Ubuntu 12.04-15.10/Debian 8 (64-bit)	91.7 MB	2017-07-20	2d0769bea2bf6041511d6901a1cf69c3
RStudio 1.0.153 - Ubuntu 16.04+/Debian 9+ (64-bit)	61.9 MB	2017-07-20	d584cbab01041777a15d62cbef69a976
RStudio 1.0.153 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (32-bit)	84.7 MB	2017-07-20	8dfce96059b05a063c49b705eca0ceb4
RStudio 1.0.153 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (64-bit)	85.7 MB	2017-07-20	16c2c8334f961c65d9bfa8fb813ad7e7

### Zip/Tarballs

Zip/tar archives	Size	Date	MD5
RStudio 1.0.153 - Windows Vista/7/8/10	117.6 MB	2017-07-20	024b5714fa6ef337fe0c6f5e2894cbcb
RStudio 1.0.153 - Ubuntu 12.04-15.10/Debian 8 (32-bit)	86.2 MB	2017-07-20	f8e0ffa7ec62665524f9e2477facd346
RStudio 1.0.153 - Ubuntu 12.04-15.10/Debian 8 (64-bit)	92.7 MB	2017-07-20	2077c181311d1aad6fb8d435f8f1f45f
RStudio 1.0.153 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (32-bit)	85.4 MB	2017-07-20	92e1a22d14952273ec389e5a55be614f
RStudio 1.0.153 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (64-bit)	86.6 MB	2017-07-20	0b71c5a7fc53c84b3fe67242240b3531

<https://www.rstudio.com/products/rstudio/download3/#download>

# 使用 Rstudio

The screenshot shows the RStudio interface with four panels highlighted by orange dashed boxes and labels:

- 編輯區 (Editor):** Contains the R script editor with the following code:

```
1 library(rvest)
2 appledaily <- html("http://www.berich.com.tw/DP/Cr
3 article <- appledaily %>% html_nodes("table") %>%
```
- 歷史&環境 (History & Environment):** Contains the Environment and History tabs. The Environment tab shows the following objects:

```
table(tw2330$tf)
hist(tw2330$Close)
pairs(iris)
pairs(iris, col="iris$Species")
pairs(iris, col=iris$Species)
```
- 控制臺 (Console):** Contains the R console output and input. The output shows the workspace loaded from `~/RData` and the execution of the `pairs(iris)` function, which results in an error:

```
[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 (Plots, Packages, Help, Viewer):** Contains the Plots, Packages, Help, and Viewer tabs. The Plots tab shows a grid of plots for the `iris` dataset, including `Sepal.Length`, `Sepal.Width`, `Petal.Length`, `Petal.Width`, and `Species`.

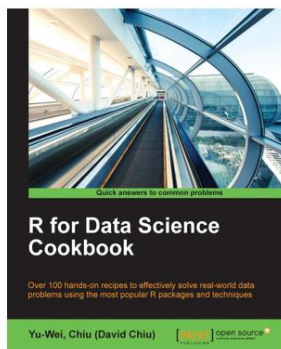
# 使用向量存放多個變數的資料

RRP <- 35.99

Exchange <- 31.74

NTD <- RRP \* Exchange  
NTD

1. 可以使用 <- 或 = 進行賦值
2. 每次賦值會產生一個物件
3. R語言中每個變數都是物件



## 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



# 變數與數學運算

---

# 不同型態的向量

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

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

使用c () 宣告向量



# 使用向量計算

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?

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

```
weight_vec <- c(73, 87, 43)
```

```
names_vec <- c('Brian', 'Toby', 'Sherry')
```

```
bmi_vec <- weight_vec / (height_vec / 100) ^ 2  
names(bmi_vec) = names_vec
```

```
bmi_vec[bmi_vec < 18.5 | bmi_vec >= 24]
```

**BMI值計算公式:**

**BMI = 體重(公斤) / 身高<sup>2</sup>(公尺<sup>2</sup>)**

	身體質量指數(BMI) (kg/m <sup>2</sup> )
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

# 矩陣可以存放二維向量

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？

```
kevin <- c(85,73)
marry <- c(72,64)
jerry <- c(59,66)
mat <- matrix(c(kevin, marry, jerry), nrow=3, byrow= TRUE)
weighted_score <- mat[,1] * 0.4 + mat[,2] * 0.6
```

可以分別取矩陣各列(或行)進行運算

# 表示類別資料(Factor)

---

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

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

# 使用list 包裝不同類型資料

---

使用list 包裝類型不同的資料

```
person <- list(name='James', height=180, Employ=TRUE)  
person
```

使用lapply 套用函式到list 裡面的元素

```
li = list( c(98,82,66,54), c(83,72,77))  
lapply(li, sum)
```

# Data Frame

---

當我希望能夠存放不同類型的資料在一表格(Tabulated Data)之中時，並且可以根據列與欄操作與分析資料時，可以建立Data Frame

使用資料集

`data(iris)`

`class(iris)`



*Iris setosa*



*Iris versicolor*

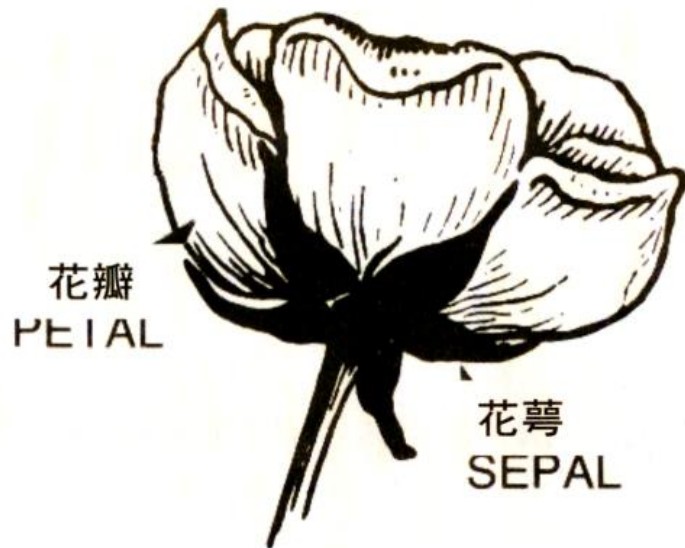


*Iris virginica*

# 如何從花萼與花瓣長寬分辨花種？

Fisher's Iris Data

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>





# 檢視 Data Frame

---

# 檢視資料形態

`class(iris)`

# 檢視架構

`str(iris)`

# 檢視資料摘要

`summary(iris)`

# 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"]
```

df[列, 欄]

取前兩行資料

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

取特定欄位向量值

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

# 使用條件篩選資料

---

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]
```

# 資料排序

---

用Sort 作資料排序

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

用order做資料排序

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

# 使用繪圖元件探索資料

---

#Pie Chart

```
table.iris = table(iris$Species)  
pie(table.iris)
```

#Histogram

```
hist(iris$Sepal.Length)
```

#Box Plot

```
boxplot(Petal.Width ~ Species, data = iris)
```

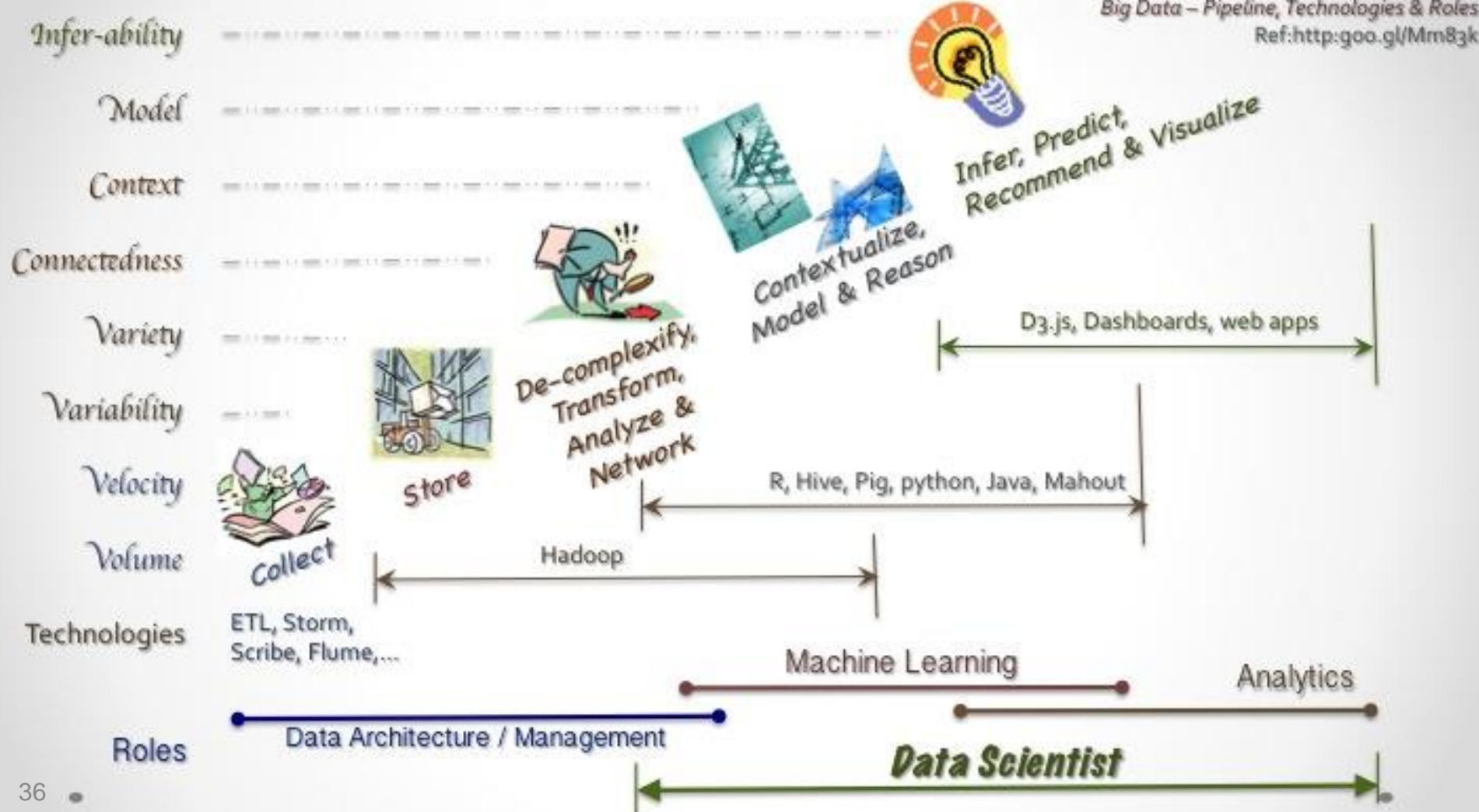
#Scatter Plot

```
plot(x=iris$Petal.Length, y=iris$Petal.Width, col=iris$Species)
```



# 使用 R 語言探討資料科學流程

---





# 不動產交易實價資料



NEWS ▶ 內政部105年幸福列車單身聯誼開始報名，內政部補助活動費用2成，歡迎單身未婚男女踴躍參加。

## 系統訊息

## 系統維護訊息

- 9月1日提供7月1-15日成交案件查詢及下載，歡迎查詢利用。 105.09.01
- 內政部105年幸福列車單身聯誼開始報名，內政部補助活動費用2成，第一場苗栗花露農場 105.06.29  
新竹南寮漁港，第2場高雄農場高雄外港，6月26日截止報名；第3場新店碧潭，7月3日截止報名，歡迎參加。...[活動連結](#)
- 105年3月15日起opendata及付費提供批次資料原提供「交易年月」增提供「交易日期」。 105.03.15
- 本部實施「不動產實價登錄」制度，推動各項地政便民服務措施，及財政部實行網路報稅等措施，降低申請不動產移轉登記所需稅捐成本。因此世界銀行於經商環境(Doing 104.12.02

## 重要公告

提醒民眾及房仲業者：不動產仲介經紀業服務報酬之計收，主管機關並未規定固定收費比率，消費者可與房仲業者自行議訂，其向買賣或租賃之一方或雙方收取報酬之總額，合計不得超過該不動產實際成交價金6%或1.5個月租金。



App及文件下載



查詢須知

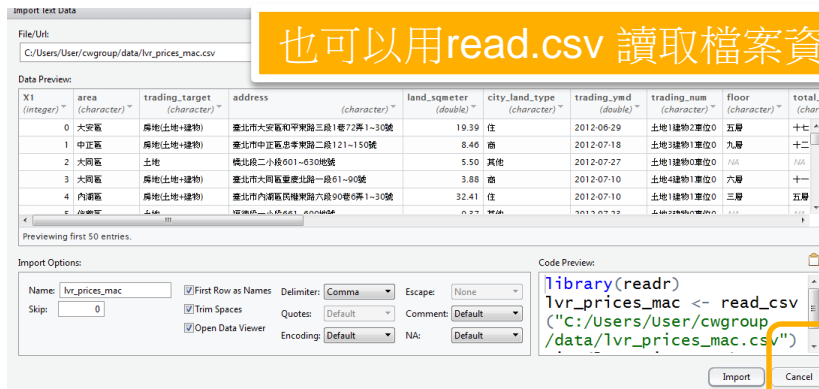
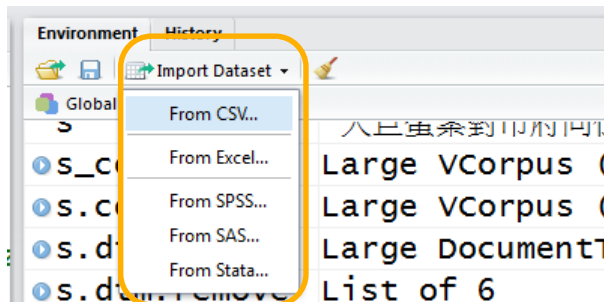
# 使用R 讀取csv檔案

檢視目錄所在  
`getwd()`

可以使用`setwd` 修改目錄位置

下載檔案資料

```
download.file('https://raw.githubusercontent.com/ywchiu/cathayr/master/data/lvr_prices.csv', 'lvr_prices.csv')
```



也可以用`read.csv` 讀取檔案資料

# 資料探索

---

計算大安區所有成交物件的的總價(total\_price)總合

```
daan <- lvr_prices[lvr_prices$area == '大安區',]  
sum(as.numeric(daan$total_price), na.rm = TRUE)
```

計算總價(total\_price)平均

```
mean(as.numeric(daan$total_price), na.rm = TRUE)
```

## 資料探索(II)

列舉中山區所有成交物件總價(total\_price)前三名的地址(address)以及(total\_price)

```
zhongshan <- lvr_prices[lvr_prices$area == '中山區',  
  c('address', 'total_price')]  
idx <- order(zhongshan$total_price, decreasing = TRUE)  
res <- zhongshan[idx,]  
res[1:3,]
```

也可以換成head(res,3)

那如果想要換區統計呢? 是否可以將動作包裝成函式?

# 函式 (Function)

```
getTopThree <- function(area){  
  zhongshan <- lvr_prices[lvr_prices$area == area,]  
  idx <- order(zhongshan$total_price, decreasing = TRUE)  
  res <- zhongshan[idx,c('area', 'address',  
    'total_price')]  
  return(res[1:3,])  
}
```

```
getTopThree('大安區')
```

那如果不想一一打入台北市12區的話  
如何該列舉各區的統計數據？

# tapply

計算屬性

分組條件

```
tapply(lvr_prices$total_price, lvr_prices$area,  
function(e)mean(e,na.rm=TRUE))
```

匿名函式

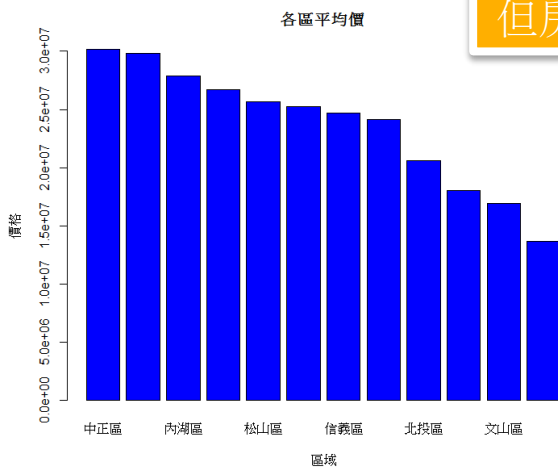
士林區	大同區	大安區	中山區	中正區	內湖區	文山區
24139903	18063872	29798170	26708805	30154011	27905514	16953869
北投區	松山區	信義區	南港區	萬華區		
20626410	25652125	24725051	25235793	13642289		

該如何將這些資料視覺化？

# 使用長條圖比較平均房價高低

```
price_per_sec <- tapply(lvr_prices$total_price, lvr_prices$area,  
function(e)mean(e,na.rm=TRUE))
```

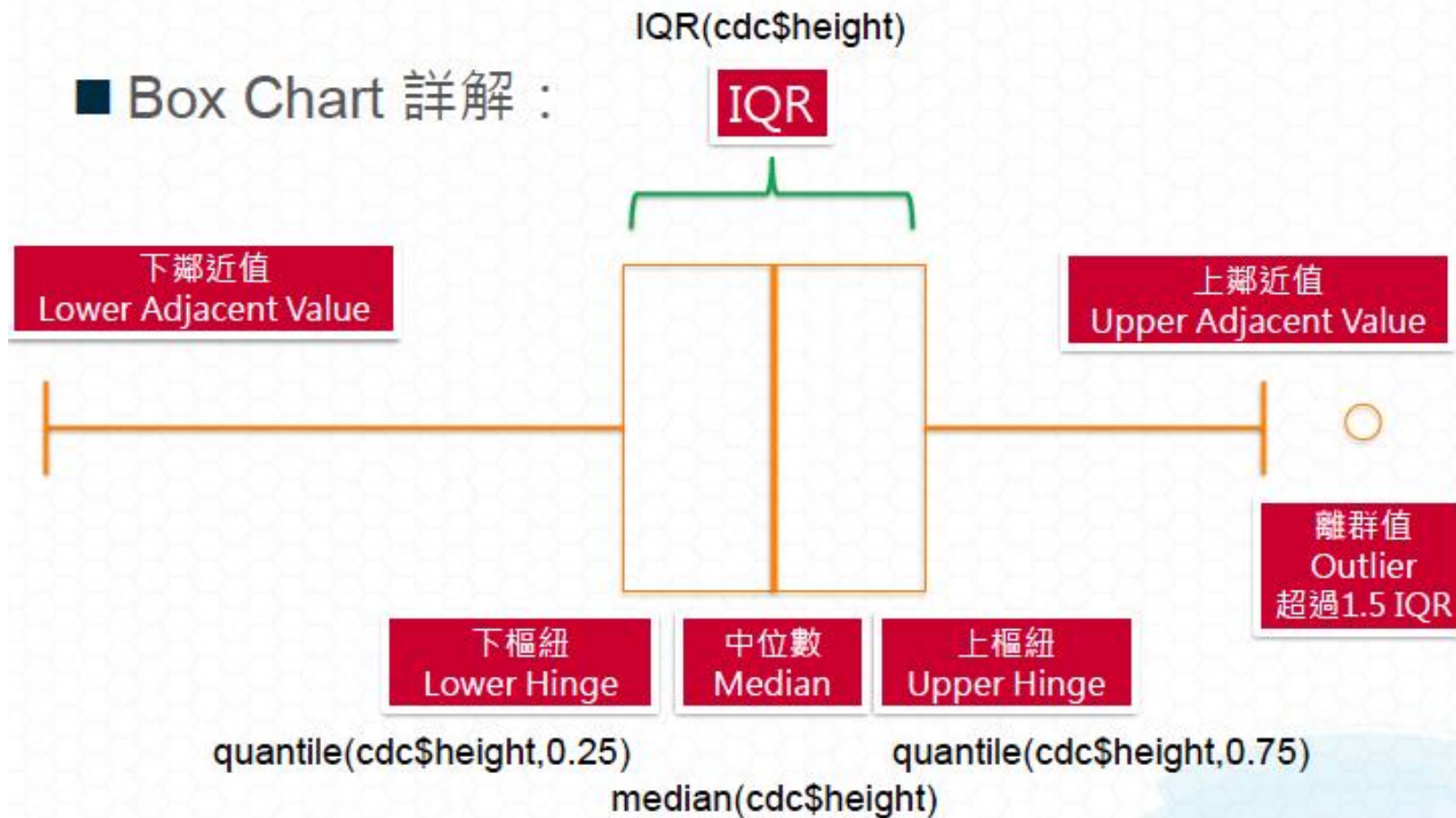
```
barplot(sort(price_per_sec, decreasing = TRUE), main= "各區  
平均價", xlab = "區域", ylab = "價格", col="blue")
```



但房屋買賣價格很大，要考慮離群值

# Box Chart

## ■ Box Chart 詳解：

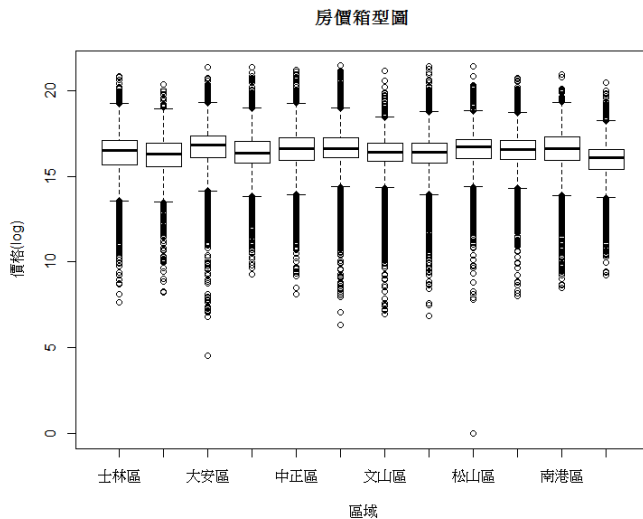




# 繪製 Box Chart

繪製台北市各區域(根據Area 區分)房價的箱型圖 (Y軸為總價 total\_price)

```
boxplot(log(total_price) ~ area, data = lvr_prices, main= "房價箱型圖",  
xlab = "區域", ylab = "價格(log)")
```



# 安裝與使用dplyr

---

- 但 `tapply` 只能針對一個變數(區域或時間)做聚合，是否有其他解決方案？
- 使用dplyr
  - 提供操作資料的基本語法  
`filter`, `select`, `arrange`, `mutate`, `summarise`, `group_by`
  - 提供資料合併功能(JOIN)  
`Inner join`, `left join`
  - 可以操作資料表(data table) 或資料庫 (Database)的資料

# 如果想要繪製各區不同時間點的房價走勢？

---

安裝dplyr

```
install.packages("dplyr")
```

使用dplyr

```
library(dplyr)
```

觀看說明頁

```
help(package='dplyr')
```

dplyr 的過濾功能

```
filter(lvr_prices, area == '中山區')
```

dplyr 的欄位選取

```
select(lvr_prices, total_price)
```

## 但如果我想同時選擇欄位又過濾資料呢？

---

鏈接(Chaining)

%>% (Then)

來自 magrittr

使用Then (%>%)

lvr\_prices %>%

select(area, total\_price) %>%

filter(area == '中山區')

%>%  
magrittr

*Ceci n'est pas un pipe.*

## 分組計算 (group\_by, summarise)

---

使用Arrange 可以將資料做排序

```
lvr_prices %>%  
  select(area, total_price) %>%  
  filter(area == '中山區') %>%  
  arrange(total_price)
```

由大到小排序 (desc)

```
lvr_prices %>%  
  select(area, total_price) %>%  
  filter(area == '中山區') %>%  
  arrange(desc(total_price))
```

如同

```
SELECT total_price, area  
FROM lvr_prices  
WHERE area = "中山區"  
ORDER BY total_price
```

# 資料做排序

---

## 分組計算函式

**group\_by:** 分組依據

**summarise:** 依組別計算結果

統計各區域各年月(2012年1月1日後)的價格總和

```
lvr_prices$trading_ym <- as.Date(format(lvr_prices$trading_ymd, '%Y-%m-01'))
lvr_stat <- lvr_prices %>%
  select(trading_ym, area, total_price) %>%
  filter(trading_ym >= '2012-01-01') %>%
  group_by(trading_ym, area) %>%
  summarise(overall_price = sum(as.numeric(total_price), na.rm=TRUE))
```

# 繪製房價變化

使用折線圖繪製台北市各區域(根據Area 區分)從2012 年至今每月的房價。(X軸為交易月，Y軸為總價total\_price)

```
lvr_stat$area <- as.factor(lvr_stat$area)
```

```
par(mfrow=c(3,4))
```

```
for (a in levels(lvr_stat$area)){  
  plot(overall_price ~ trading_ym  
    ,lvr_stat[lvr_stat$area == a,]  
    , type='l', main = a)  
}
```

產生 3 X 4 的圖表

使用for 迴圈繪製各區域圖

type	description
p	點。
l	直線。
o	點+直線。(兩者重疊)
b	點+直線。(不重疊)
c	點+直線。(點為空白)
S/s	階梯狀。
h	Histogram狀。
n	空樣式。



# 產生pivot table

可以使用tidyr 套件產生pivot table

`library(tidyr)`

```
price_pivot <- spread(lvr_stat, trading_ym,  
overall_price, fill=0)
```

`View(price_pivot)`

	area	2012-01-01	2012-02-01	2012-03-01	2012-04-01	2012-05-01	2012-06-01	2012-07-01	2012-08-01	2012-09-01	201
1	士林區	661140000	231680000	359891504	205481036	2539010528	514470692	1612810630	2627628833	4072187784	
2	大同區	0	180150000	525210000	87110000	74806000	173360800	302473961	2056051662	1168712672	
3	大安區	139136600	123870000	64470991	127736080	49052000	668836500	2953852056	4914767352	5015582531	
4	中山區	17601653	140250000	176022439	3156689238	2374390095	925510000	3036981800	4692015079	7291463744	
5	中正區	258200000	112690000	660420000	935400000	550190000	364040000	1463004513	4498343873	3952174637	
6	內湖區	349930000	216810000	299907515	944724354	1444681765	615189000	2529917498	5319499128	8621316230	
7	文山區	166887497	147810000	553478681	739757581	192890000	297841800	1245614572	2878918358	2197557340	
8	北投區	43850000	68000	82490000	494942526	899718610	732233963	2308572984	4394034989	3995621228	
9	松山區	0	0	405003695	554400	405400000	270720000	1216371195	2230302414	3043203633	
10	信義區	40800000	177020000	1269564574	2517094802	1241890000	647360000	2809067000	2790734498	4376895822	

# 將結果存回檔案中

使用write.csv

write.csv(price\_pivot, 'taipei\_house\_price.csv')

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1		area	2012/1/1	2012/2/1	2012/3/1	2012/4/1	2012/5/1	2012/6/1	2012/7/1	2012/8/1	2012/9/1	2012/10/1	2012/11/1	2012/12/1	2013/1/1	2013/2/1	2013/3/1
2	1	士林區	6.61E+08	2.32E+08	3.6E+08	2.05E+08	2.54E+09	5.14E+08	1.61E+09	2.63E+09	4.07E+09	4.35E+09	5.26E+09	9.21E+09	4.07E+09	3.01E+09	6.5E+09
3	2	大同區	0	1.8E+08	5.25E+08	87110000	74806000	1.73E+08	3.02E+08	2.06E+09	1.17E+09	3.74E+09	2.36E+09	2.2E+09	1.51E+09	8.8E+08	2.08E+09
4	3	大安區	1.39E+08	1.24E+08	64470991	1.28E+08	49052000	6.69E+08	2.95E+09	4.91E+09	5.02E+09	7.11E+09	5.24E+09	7.4E+09	5.08E+09	4.66E+09	7.33E+09
5	4	中山區	17601653	1.4E+08	1.76E+08	3.16E+09	2.37E+09	9.26E+08	3.04E+09	4.69E+09	7.29E+09	8.88E+09	8.14E+09	1.12E+10	8.48E+09	5.1E+09	1.22E+10
6	5	中正區	2.58E+08	1.13E+08	6.6E+08	9.35E+08	5.5E+08	3.64E+08	1.46E+09	4.5E+09	3.95E+09	7.33E+09	4.2E+09	5.67E+09	4.18E+09	2.47E+09	3.7E+09
7	6	內湖區	3.5E+08	2.17E+08	3E+08	9.45E+08	1.44E+09	6.15E+08	2.53E+09	5.32E+09	8.62E+09	7.08E+09	1.32E+10	1.11E+10	5.84E+09	6.36E+09	8.71E+09
8	7	文山區	1.67E+08	1.48E+08	5.53E+08	7.4E+08	1.93E+08	2.98E+08	1.25E+09	2.88E+09	2.2E+09	4.29E+09	3.56E+09	5.15E+09	2.64E+09	3.15E+09	4.16E+09
9	8	北投區	43850000	68000	82490000	4.95E+08	9E+08	7.32E+08	2.31E+09	4.39E+09	4E+09	3.21E+09	4.49E+09	6.23E+09	5.79E+09	3.97E+09	5.89E+09
10	9	松山區	0	0	4.05E+08	554400	4.05E+08	2.71E+08	1.22E+09	2.23E+09	3.04E+09	5.14E+09	4.11E+09	6.29E+09	2.36E+09	1.76E+09	4.58E+09
11	10	信義區	40800000	1.77E+08	1.27E+09	2.52E+09	1.24E+09	6.47E+08	2.81E+09	2.79E+09	4.38E+09	4.01E+09	6.86E+09	6.35E+09	3.42E+09	2.88E+09	4.89E+09
12	11	南港區	53560259	1.45E+08	4.3E+08	4.53E+08	7.35E+08	2.12E+08	1.43E+09	2.33E+09	4.29E+09	5.97E+09	2.85E+09	5.07E+09	1.95E+09	2.31E+09	3.81E+09
13	12	萬華區	17430000	14800000	7800000	5.14E+08	3.12E+08	88610000	7.59E+08	1.81E+09	1.32E+09	1.77E+09	1.79E+09	3.16E+09	1.19E+09	1.01E+09	2.09E+09



# R 語言與機器學習

---

# 機器學習

---

機器學習的目的是：歸納 ( Induction )

- 從詳細事實到一般通論

*A computer program is said to learn from experience  $E$  with respect to some task  $T$  and some performance measure  $P$ , if its performance on  $T$ , as measured by  $P$ , improves with experience  $E$  -- Tom Mitchell (1998)*

找出有效的預測模型

- 一開始都從一個簡單的模型開始
- 藉由不斷餵入訓練資料，修改模型
- 不斷提升預測績效

# 機器學習步驟

使用者行為



# 機器學習問題分類

---

## 監督式學習 (Supervised Learning)

- 回歸分析 (Regression)
- 分類問題 (Classification)

## 非監督式學習 (Unsupervised Learning)

- 降低維度 (Dimension Reduction)
- 分群問題 (Clustering)

# 使用監督式學習進行預測

---

## 分類問題

根據已知標籤的訓練資料集(Training Set)，產生一個新模型，用以預測測試資料集(Testing Set)的標籤。

e.g. 客戶流失分析

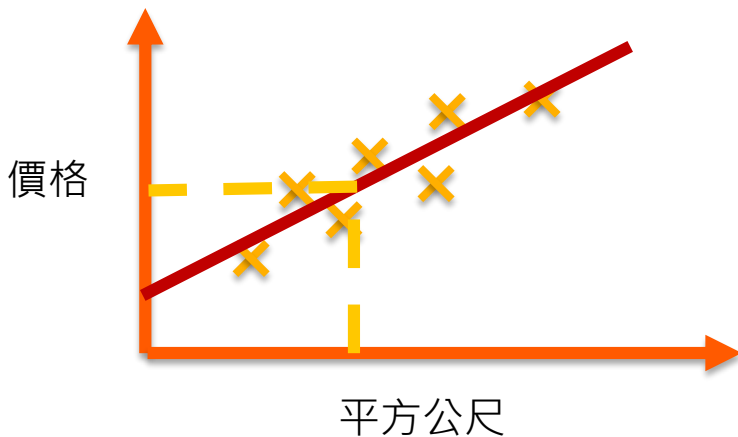
## 回歸分析

使用一組已知對應值的資料產生的模型，預測新資料的對應值

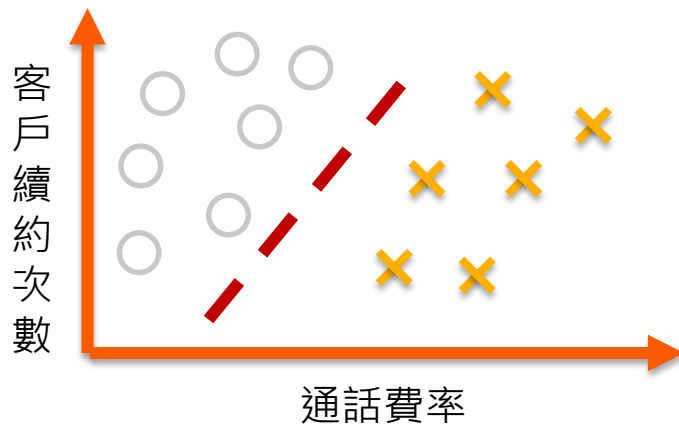
e.g. 股價預測

# 使用監督式學習進行預測

## 回歸分析



## 分類問題





# 使用非監督式學習找出隱藏的架構

---

## 降低維度

產生一有最大變異數的欄位線性組合，可用來降低原本問題的維度與複雜度

e.g. 濃縮用到的特徵，編纂成一個新指標

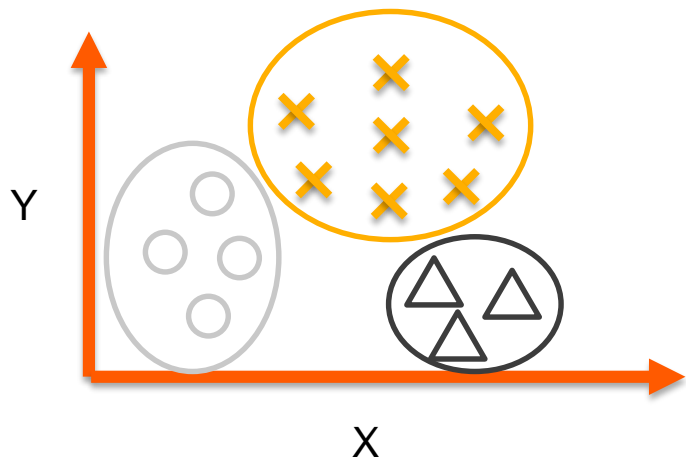
## 分群問題

物以類聚 (近朱者赤、近墨者黑)

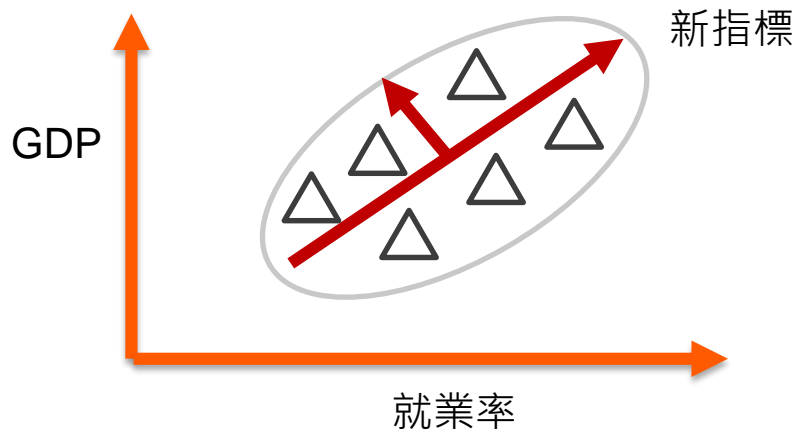
e.g. 將客戶分層

# 使用非監督式學習找出隱藏的架構

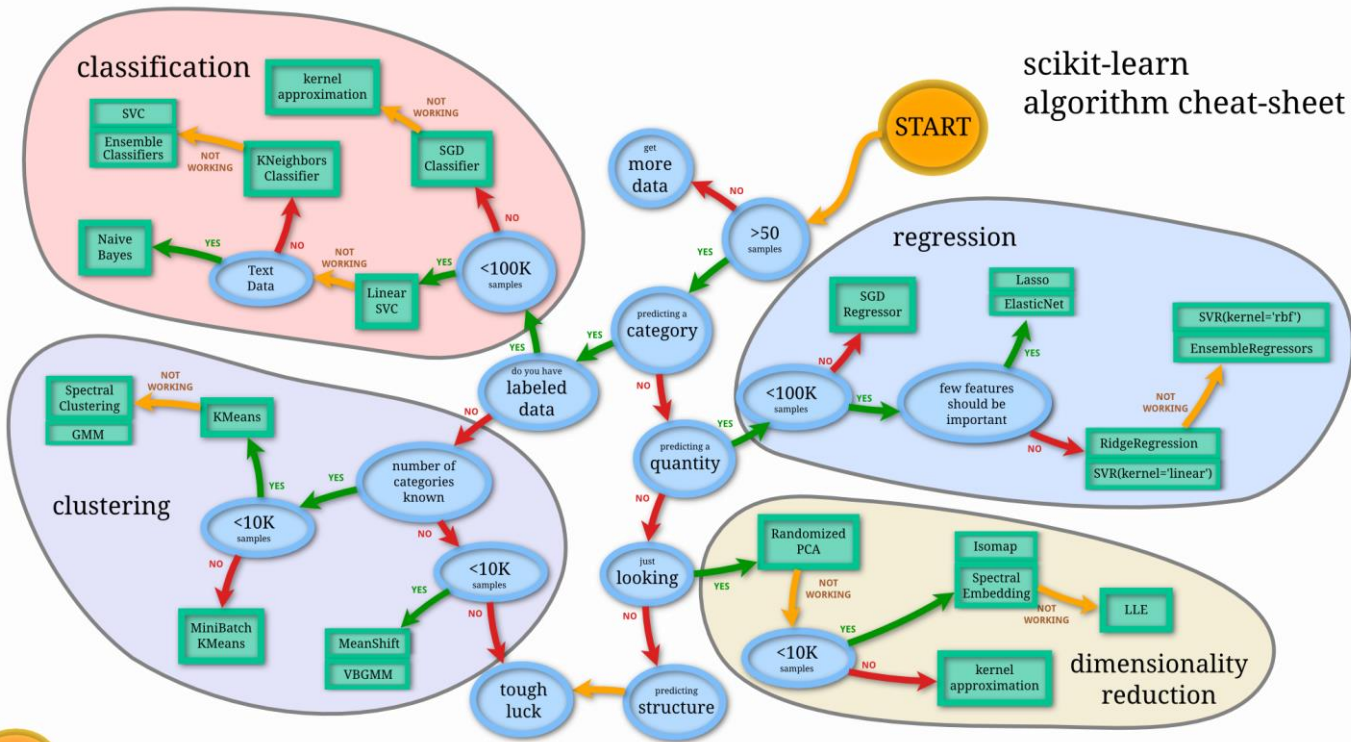
分群問題



降低維度



# 機器學習地圖





# 分類問題

---

# 分類問題

---

根據已知標籤的訓練資料集(Training Set)，產生一個新模型，用以預測測試資料集(Testing Set)的標籤。

給予銀行客戶的特徵, 預測可以批准那些人的貸款申請

# 問題描述

---

**當銀行收到貸款申請時，必須根據客戶的個人檔案決定是否應該要核准貸款**

如果客戶信用良好，批准貸款

如果客戶信用較差，不批准貸款

**分析目的**

**將低銀行風險、提升潛在獲利**

# 抽取特徵資料

從客戶資料表中抽取特徵與預測目標

Predictor (Categorical)	Levels and Proportions				
Account Balance	No Account	None	Below 200 DM	200 DM or Above	
(%)	27.40%	26.90%	6.30%	39.40%	
Payment Status	Delayed	Other Credits	Paid Up	No Problem with current credits	Previous Credits Paid
(%)	4.0%	4.9%	53.0%	8.8%	29.3%
Savings/Stock Value	None	Below 100 DM	[100, 500]	[500, 1000]	Above 1000
(%)	60.3%	10.3%	6.3%	4.8%	18.3%
Length of Current Employment	Unemployed	< 1 Year	[1, 4]	[4, 7]	Above 7
(%)	6.2%	17.2%	33.9%	17.4%	25.3%
Installment %	Above 35%	(25%, 35%]	[20%, 25%]	Below 20%	
(%)	13.6%	23.1%	15.7%	47.6%	
Occupation	Unemployed, unskilled	Unskilled permanent resident	Skilled	Executive	
(%)	2.2%	20%	63%	14.8%	
Sex and Marital Status	Male, Divorced	Male Single	Male Married/Widowed	Female	
(%)	5.0%	31.0%	54.8%	9.2%	
Duration in Current Address	< 1 Year	[1, 4]	[4, 7]	Above 7	
(%)	13.0%	30.8%	14.9%	41.3%	
Type of Apartment	Free	Rented	Owned		
(%)	17.9%	71.4%	10.7%		
Most Valuable Asset	None	Car	Life Insurance	Real Estate	
(%)	28.2%	23.2%	33.2%	15.4%	
No. of Credits at Bank	1	2 or 3	4 or 5	Above 6	
(%)	63.3%	33.3%	2.8%	0.06%	
Guarantor	None	Co-applicant	Guarantor		
(%)	90.7%	4.1%	5.2%		
Concurrent Credits	Other Banks	Dept Stores	None		
(%)	13.9%	4.7%	81.4%		
No of Dependents	3 or More	Less than 3			
(%)	84.5%	15.5%			
Telephone	Yes	No			
(%)	40.4%	59.6%			
Foreign Worker	Yes	No			
(%)	3.7%	96.3%			

# 客戶特徵

---

- 客戶特徵

- Account Balance(帳戶餘額): 沒有帳號(1), 沒有餘額 (2), 部分存款 (3)
- Payment Status(付款狀況): 有問題 (1), 付清 (2), 於該銀行沒有問題(3)
- Savings/Stock Value (存款): 沒有, 少於 100, 100 ~ 1,000, 高於1,000
- Employment Length(工作長短): 少於 1 年, 1 ~ 4年, 4 ~ 7年, 高於 7 年
- Sex/Marital Status(性別與婚姻): 男生/女生, 單身/離婚/已婚/鰥夫
- No of Credits at this bank(貸款數): 1, >1
- Guarantor(保人): None, Yes
- Concurrent Credits(其他貸款數): 其他銀行或百貨, None
- ForeignWorker (外國工作者)
- Purpose of Credit(貸款目的): New car, Used car, Home Related, Other

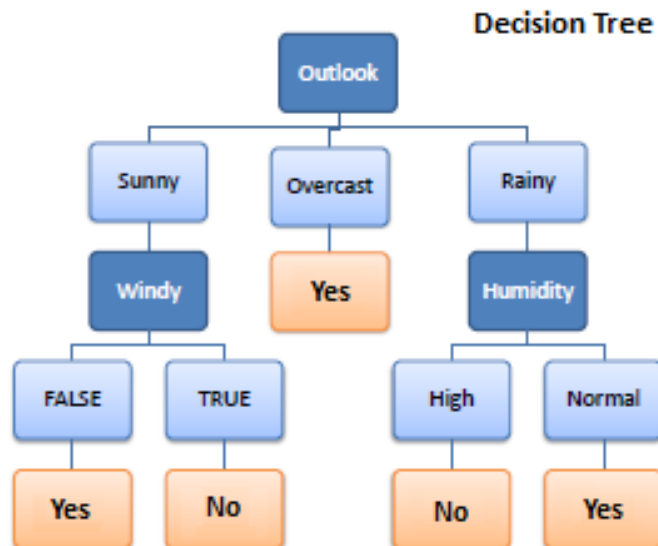
- 預測目標

- Creditability (可信客戶): 1/0



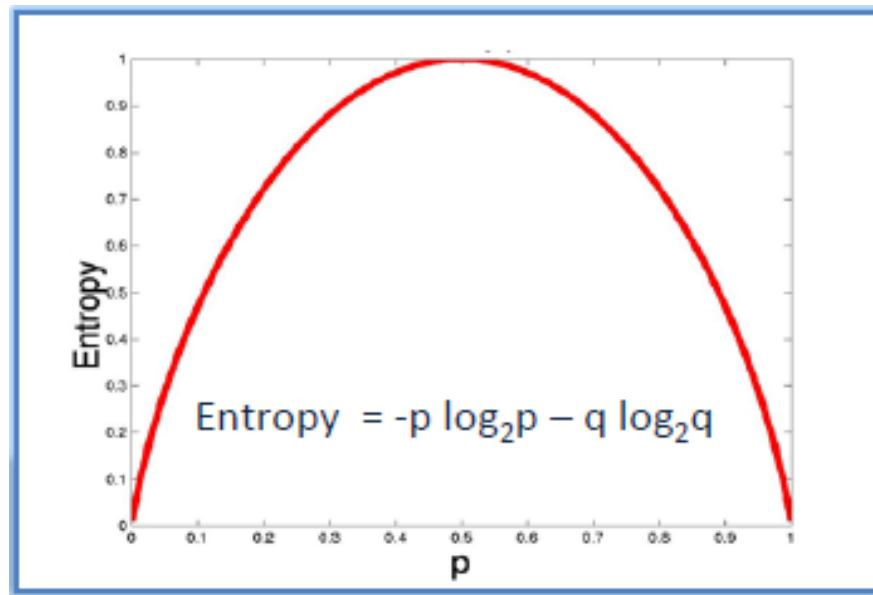
# 決策樹

Predictors				Target
Outlook	Temp.	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No



# Entropy

- 用於計算一個系統中的失序現象，也就是計算該系統混亂的程度



$$\text{Entropy} = -0.5 \log_2 0.5 - 0.5 \log_2 0.5 = 1$$

# 單一變數的計算

$$E(S) = \sum_{i=1}^c -p_i \log_2 p_i$$

Play Golf	
Yes	No
9	5



Entropy(PlayGolf) = Entropy (5,9)  
= Entropy (0.36, 0.64)  
= - (0.36 log<sub>2</sub> 0.36) - (0.64 log<sub>2</sub> 0.64)  
= 0.94

# 多變數的計算

$$E(T, X) = \sum_{c \in X} P(c)E(c)$$

		Play Golf		
		Yes	No	
Outlook	Sunny	3	2	5
	Overcast	4	0	4
	Rainy	2	3	5
				14



$$\begin{aligned} E(\text{PlayGolf}, \text{Outlook}) &= P(\text{Sunny}) * E(3,2) + P(\text{Overcast}) * E(4,0) + P(\text{Rainy}) * E(2,3) \\ &= (5/14) * 0.971 + (4/14) * 0.0 + (5/14) * 0.971 \\ &= 0.693 \end{aligned}$$

# Information Gain

---

- 根據分割(Split)後，所減少的Entropy
- 因此做分割時，會尋找最大的Information Gain

- 1. 計算Entropy

$$\begin{aligned}\text{Entropy}(\text{PlayGolf}) &= \text{Entropy}(5,9) \\ &= \text{Entropy}(0.36, 0.64) \\ &= -(0.36 \log_2 0.36) - (0.64 \log_2 0.64) \\ &= 0.94\end{aligned}$$

# 計算Information Gain

		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3
Gain = 0.247			

		Play Golf	
		Yes	No
Temp.	Hot	2	2
	Mild	4	2
	Cool	3	1
Gain = 0.029			


		Play Golf	
		Yes	No
Humidity	High	3	4
	Normal	6	1
Gain = 0.152			

		Play Golf	
		Yes	No
Windy	False	6	2
	True	3	3
Gain = 0.048			

$$Gain(T, X) = Entropy(T) - Entropy(T, X)$$

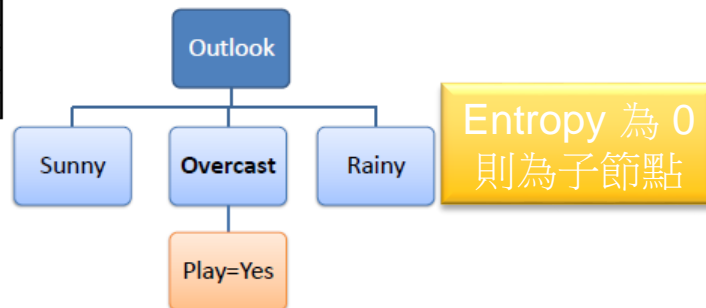
$$\begin{aligned} G(\text{PlayGolf}, \text{Outlook}) &= E(\text{PlayGolf}) - E(\text{PlayGolf}, \text{Outlook}) \\ &= 0.940 - 0.693 = 0.247 \end{aligned}$$

# 選擇有最大Information Gain的屬性

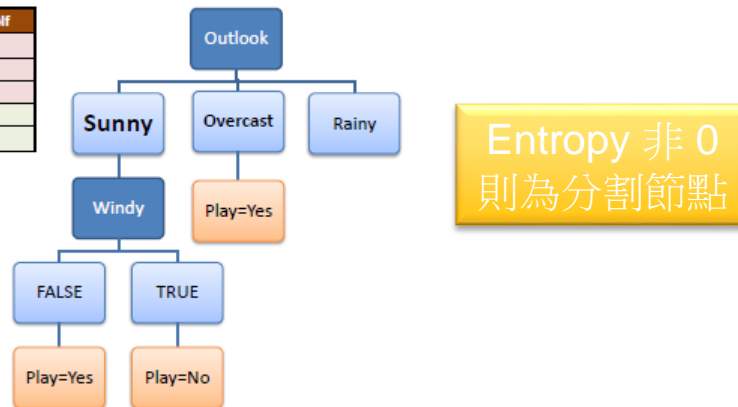
		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3
Gain = 0.247			

# 選擇子節點與分割節點

Temp	Humidity	Windy	Play Golf
Hot	High	FALSE	Yes
Cool	Normal	TRUE	Yes
Mild	High	TRUE	Yes
Hot	Normal	FALSE	Yes
Hot	High	FALSE	Yes



Temp	Humidity	Windy	Play Golf
Mild	High	FALSE	Yes
Cool	Normal	FALSE	Yes
Mild	Normal	FALSE	Yes
Cool	Normal	TRUE	No
Mild	High	TRUE	No





# 決策樹如同IF...ELSE

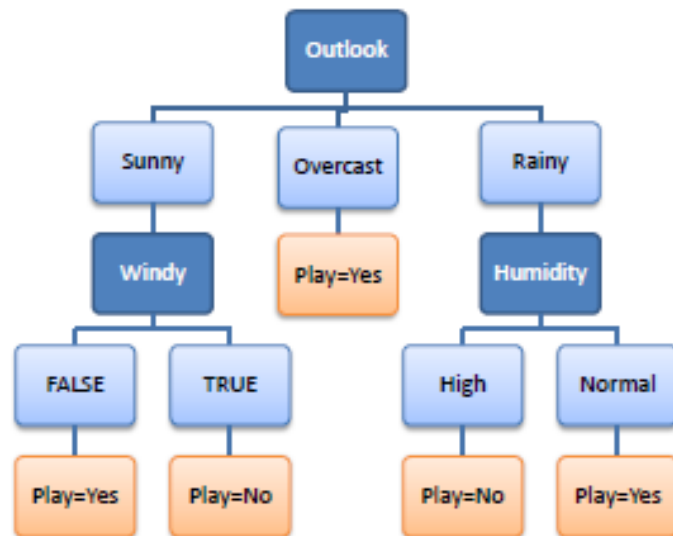
$R_1$ : IF (Outlook=Sunny) AND  
(Windy=FALSE) THEN Play=Yes

$R_2$ : IF (Outlook=Sunny) AND  
(Windy=TRUE) THEN Play=No

$R_3$ : IF (Outlook=Overcast) THEN  
Play=Yes

$R_4$ : IF (Outlook=Rainy) AND  
(Humidity=High) THEN Play=No

$R_5$ : IF (Outlook=Rain) AND  
(Humidity=Normal) THEN  
Play=Yes



# rpart 與遞迴分割法

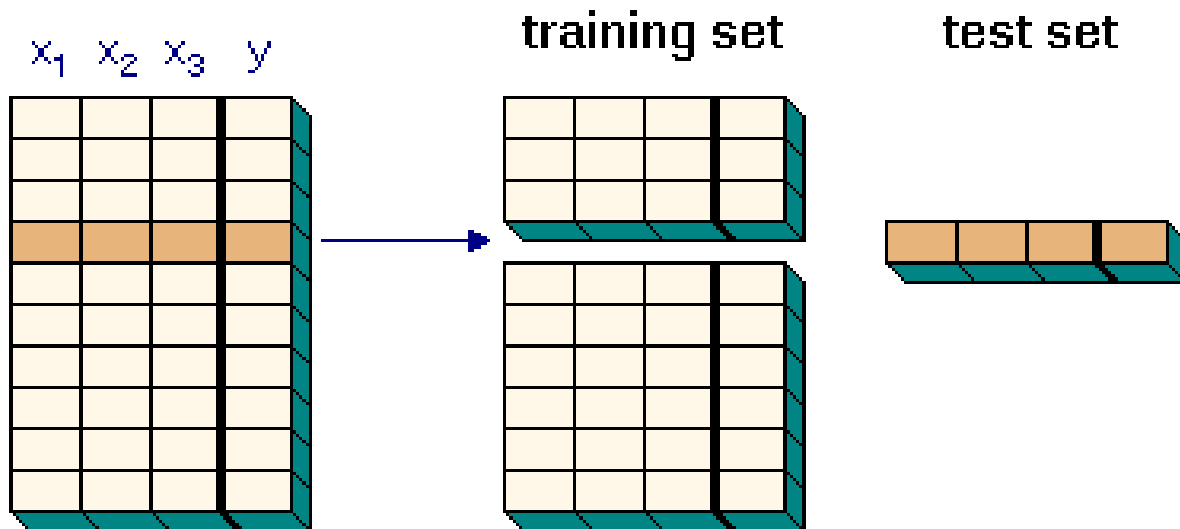
---

- **rpart**

- 對所有參數和所有分割點進行評估
- 最佳的選擇是使分割後組內的資料更為一致(pure)
  - 一致是指組內資料的因變數取值變異較小
- 使用Gini 值量測一致性
- 遞迴分割法 (Recursive Partitioning Tree)
- 使用剪枝 (prune) 方法
  - 先建立一個劃分較細較為複雜的樹模型
  - 根據交叉檢驗(Cross-Validation)的方法來估計不同”剪枝”條件下
  - 選擇誤差最小的樹模型

# 測試模型

- 使用外部資料或是一部分的內部資料來測試資料



訓練模型與測試模型都為同一份  
有球員兼裁判的嫌疑

# 建立分類樹

```
library(rpart)
trainset <- read.csv('Training50.csv')
```

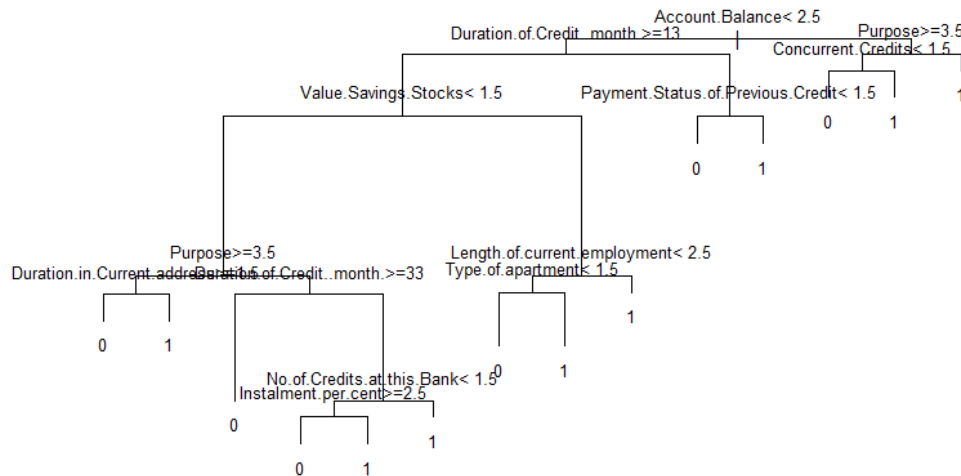
```
model <- rpart(Creditability ~
Account.Balance+Duration.of.Credit..month.+Payment.Status.of.Previous.
Credit+Purpose+Credit.Amount+Value.Savings.Stocks+Length.of.current.em
ployment+Instalment.per.cent+Sex...Marital.Status+Guarantors+Duration.
in.Current.address+Most.valuable.available.asset+Age..years.+Concurren
t.Credits+Type.of.apartment+No.of.Credits.at.this.Bank+Occupation+No.o
f.dependents+Telephone, data=trainset, method = 'class')
```

```
summary(model)
```

機器學習

# 繪製分類樹

```
plot(model, margin = 0.1)  
text(model, pretty=0,cex=0.6)
```



# 評估預測結果

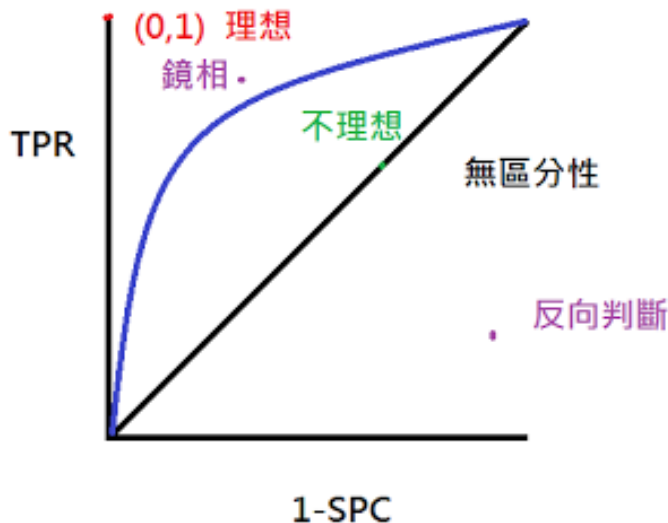
```
trainset <- read.csv('Test50.csv')  
predicted <- predict(model, testset, type = "class")  
table(predicted, testset$Creditability)
```

predicted	0	1
0	64	52
1	93	291

準確率:  $(291 + 64) / (291 + 93 + 52 + 64) = 0.71$

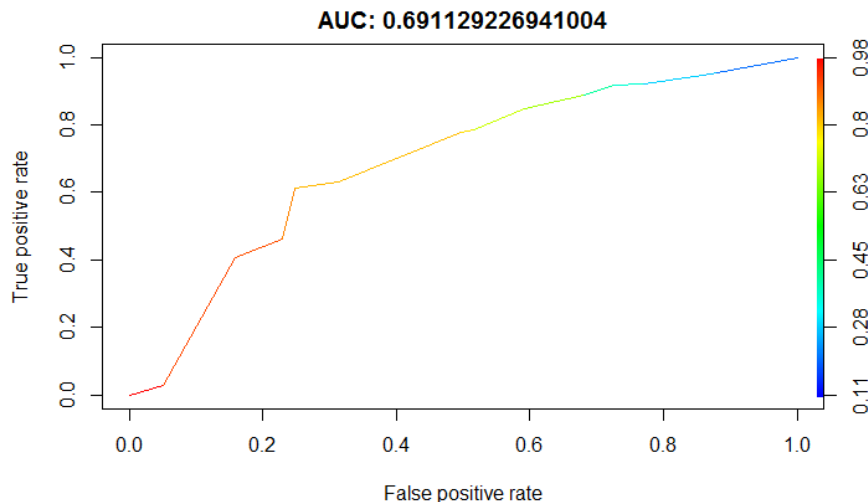
# ROC 曲線

- 接收者操作特徵(receiver operating characteristic, ROC curve)
  - 以假陽性率(False Positive Rate, FPR)為X軸，代表在所有陰性相本中，被判斷為陽性(假陽性)的機率，又寫為(1-特異性)。
  - 以真陽性率(True Positive Rate, TPR)為Y軸，代表在所有陽性樣本中，被判斷為陽性(真陽性)的機率，又稱為敏感性



# 使用 ROCR 套件

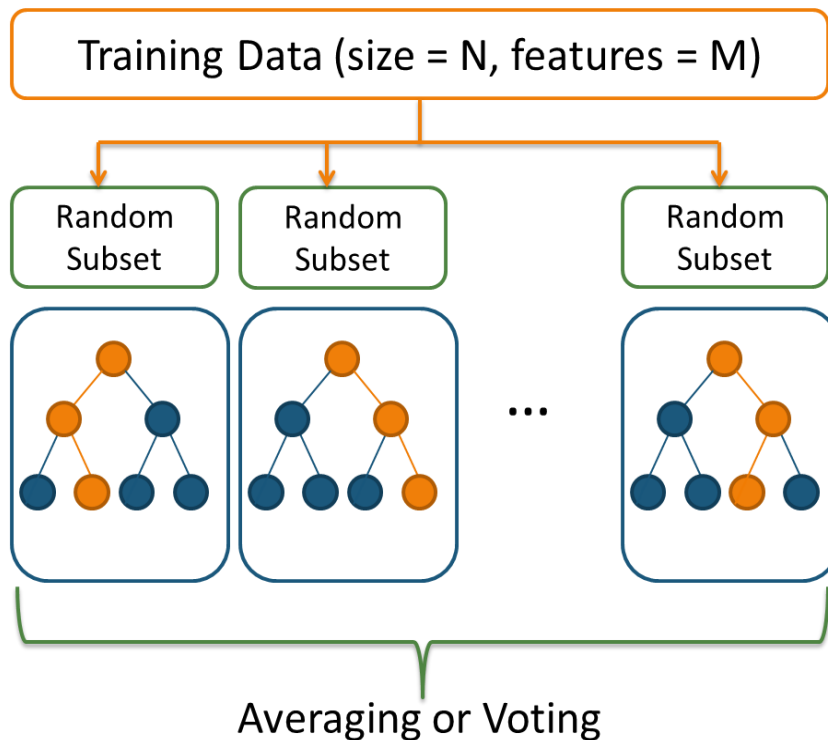
```
library(ROCR)
predictions <- predict(model, testset, type="prob")
pred.to.roc <- predictions[, 2]
pred.rocr <- prediction(pred.to.roc, as.factor(testset$Creditability))
perf.rocr <- performance(pred.rocr, measure = "auc", x.measure = "cutoff")
perf.tpr.rocr <- performance(pred.rocr, "tpr", "fpr")
plot(perf.tpr.rocr, colorize=T, main=paste("AUC:", (perf.rocr@y.values)))
```





# 隨機森林 (Random Forest)

N 多少樹, M 多少個特徵



# 建立隨機森林

```
library(randomForest)
```

替換演算法

```
forest <- randomForest(Creditability ~., data = trainset, ntree=200,  
importance=T, proximity=T)
```

```
forest.predicted <- predict(forest, testset, type = "class")  
table(forest.predicted, testset$Creditability)
```

predicted	0	1
0	53	27
1	104	316

準確率:  $(316 + 53) / (316 + 53 + 27 + 104) = 0.738$

# 各自產生不同成本下的預測結果

---

## # 決策樹

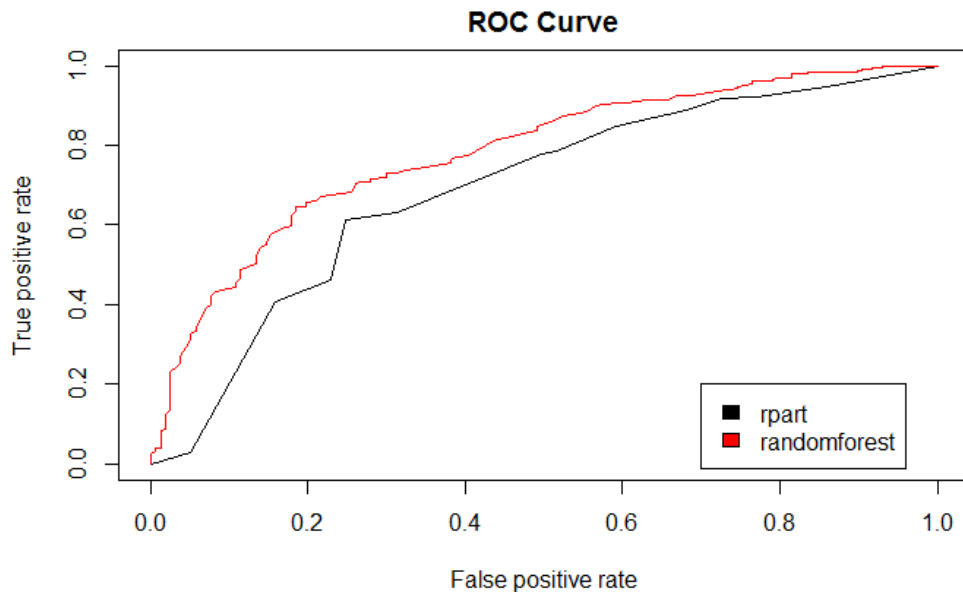
```
predictions1 <- predict(model, testset, type="prob")
pred.to.roc1 <- predictions1[, 2]
pred.rocr1 <- prediction(pred.to.roc1, as.factor(testset$Creditability))
perf.rocr1 <- performance(pred.rocr1, measure = "auc", x.measure = "cutoff")
perf.tpr.rocr1 <- performance(pred.rocr1, "tpr", "fpr")
```

## # 隨機森林

```
predictions2 <- predict(forest, testset, type="prob")
pred.to.roc2 <- predictions2[, 2]
pred.rocr2 <- prediction(pred.to.roc2, as.factor(testset$Creditability))
perf.rocr2 <- performance(pred.rocr2, measure = "auc", x.measure = "cutoff")
perf.tpr.rocr2 <- performance(pred.rocr2, "tpr", "fpr")
```

# 比較 ROC

```
plot(perf.tpr.rocr1,main='ROC Curve', col=1)  
legend(0.7, 0.2, c('rpart', 'randomforest'), 1:2)  
plot(perf.tpr.rocr2, col=2, add=TRUE)
```



Random Forest 預測能力明顯較好



# 迴歸問題 (Regression Analysis)

---

# 簡單線性迴歸

---

線性迴歸是研究**單一依變項**(dependent variable)與一個或以上**自變項**(independent variable)之間的關係

線性迴歸有兩個主要用處：

**預測**指的是用已觀察的變數來預測依變項

**因果分析**則是將自變項當作是依變項發生的原因

# 線性迴歸

## 數學模型

$$y = \beta_1 x + \beta_0 + \epsilon$$

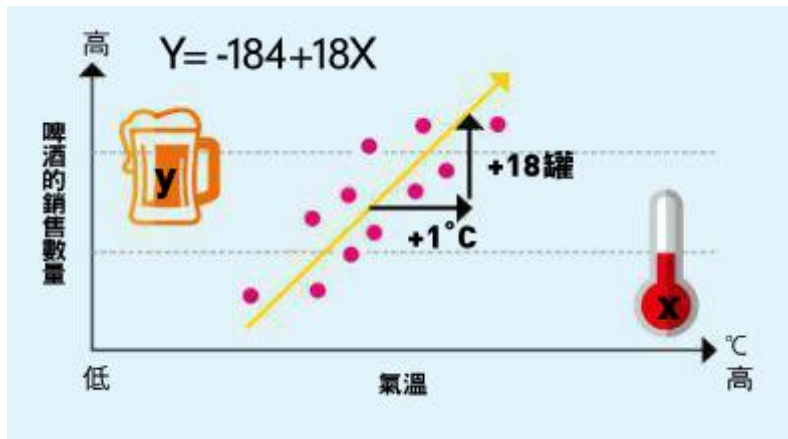
y 是依變數

x 是自變數

$\beta_i$  是迴歸係數

$\beta_0$  是截距

$\epsilon$  是誤差

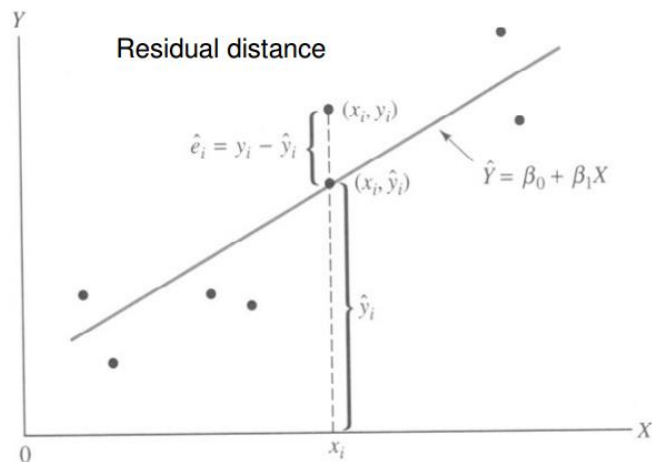


# 最小平方估計法 - OLS

找出殘差平方和最小的一條線

$$\text{殘差 } e_i = (y_i - \hat{y}_i)$$

$$\text{殘差平方和 } SSE = \sum_{i=1}^n (y_i - \hat{y}_i)^2$$





# 多元迴歸分析 (Multiple Regression)

---

當要探索多個自變量與一個依變量之間的關係時

e.g.  $\text{Sales} = 100 + 25\text{print} - 100\text{TV} + 67\text{Radio}$

基礎假設

- 依變量為隨機變數

- 變數之間有統計關聯

- 自變數與依變數有線性關係

- 自變數之間的共線性(Co-linearity 最小)

# 數學模型

---

■  $y_i = \beta_1 x_{1i} + \beta_2 x_{2i} + \cdots \beta_k x_{ki} + \beta_0 + \epsilon$

$x_1 \cdots x_k$  為自變量

$y$  是依變數

$\beta_i$  是迴歸係數

$\beta_0$  是截距

$\epsilon$  是誤差

目標:

最小化殘差平方和

$$SSR = \sum_{i=1}^n e_i^2 = \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

# 房價預測

---

- 建立變數

```
house_prices$brick_d<-ifelse(house_prices$Brick=="Yes",1,0)
house_prices$east<-ifelse(house_prices$Neighborhood=="East",1,0)
house_prices$north<-ifelse(house_prices$Neighborhood=="North",1,0)
```

- 建立訓練與測試資料集

```
set.seed(110)
sub <- sample(nrow(house_prices), floor(nrow(house_prices) * 0.6))
training_data <- house_prices[sub,]
validation_data <- house_prices[-sub,]
```

# 建立多元迴歸模型

---

- 建立多元迴歸模型

```
lm.fit1 <- lm(Price ~ SqFt+Bathrooms+Bedrooms+Offers+  
              north+east+brick_d, data=training_data)  
summary(lm.fit1)
```

- 減少不顯著的變數

```
lm.fit1.step <- stepAIC(lm.fit1)  
summary(lm.fit1.step)
```

# 預測值

---

- 訓練資料

```
training_data$predict.price <- predict(lm.fit1)
```

```
training_data$error <- residuals(lm.fit1)
```

- 測試資料

```
validation_data$predict.price <- predict(lm.fit1,newdata=validation_data)
```

```
validation_data$error <- validation_data$predict.price -
```

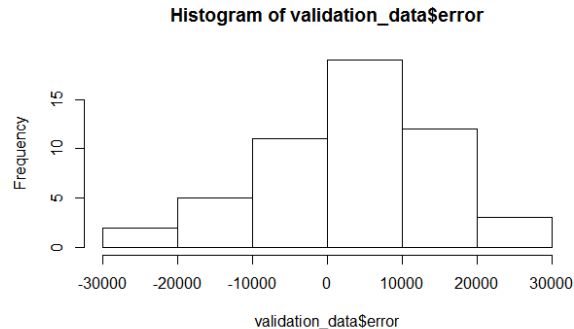
```
validation_data$Price
```

# 驗證結果

- 檢視殘值

```
hist(training_data$error)
```

```
hist(validation_data$error)
```



- 檢視R Square

```
a<-cor(training_data$Price,training_data$predict.price)
```

```
b<-cor(validation_data$Price,validation_data$predict.price)
```

```
a*a
```

```
b*b
```



# THANK YOU

---

EMAIL: [david@largitdata.com](mailto:david@largitdata.com)

網站: [www.largitdata.com](http://www.largitdata.com)

電話: 0929094381