

# AI基礎訓練中級班

## 三、指標衡量方法 (包含正確率分析及機率統計分析)

制定部門：總管理處技訓中心  
編定日期：2021年6月28日編印  
版次：R2

本著作非經著作權人同意，不得轉載、翻印或轉售。

著作權人：台灣塑膠工業股份有限公司  
南亞塑膠工業股份有限公司  
台灣化學纖維股份有限公司  
台塑石化股份有限公司

## AI中級班課程項目：

一、PYTHON相關操作及實務

二、資料前置處理

三、指標衡量方法

四、資料視覺化分析(進階)

五、資料預處理(含深度學習網路建模)

# 課程目的

完成本課程後，您將能夠：

1. 對於“正確率分析”及“機率統計分析”有基本的概念
2. 對於正確率、標準差、均方根誤差、變異數、變異係數、主成分分析、偏最小二乘法…等有基本的認識。

- (一)統計量數及分析資料工具（基本知識及應用知識）
- (二)正確率分析（基本知識）
- (三)單因子變異數分析（應用知識）
- (四)迴歸分析（應用知識）
- (五)主成分分析（應用知識）

## (一)統計量數的概念及分析資料工具

1. 變數資料之類型 .....	14
2. 集中趨勢和相對量數 .....	15
3. 平均數 Mean .....	16~17
4. 百分位數 Percentiles .....	18
5. 百分位數與四分位數程式範例 .....	19
6. 差異量數或離散量數 .....	20~21
7. 全距(range)程式範例 .....	22
8. 樣本的變異數與標差 .....	23
9. 樣本的變異數與標準差程式範例 .....	24

## (一)統計量數的概念及分析資料工具

10. 變異係數·····	25
11. 變異係數程式範例·····	26
12. 偏態的測定數·····	27
13. 偏態的測定數程式範例·····	28
14. 常態分布·····	29
15. 常態分布程式範例·····	30
16. 常態分佈線下的區域·····	31
17. 標準常態分配:Z分布·····	32
18. 標準分數·····	33
19. 標準分數程式範例·····	34
20. 練習·····	35

## (二)正確率分析的概念

1. 正確率(Accuracy)高一定好嗎? .....37
2. 分類演算法的評價指標.....38~39
3. Relevant & retrieved .....40~41
4. TP / FP / TN / FN.....42
5. 分類演算法的評價指標.....43~47
6. 練習 I.....48~49
7. 練習 II.....50~51



## (三)單因子變異數分析

1. 變異數分析的概念.....	53
2. 變異數分析的數學模型.....	54
3. 變異數分析原理說明.....	55~60
4. 變異數分析範例1.....	61~62
4-1. 變異數分析範例1分析結果.....	63
5. 變異數分析範例2.....	64
5-1. 單因子變異數分析結果.....	65
6. 資料前處理.....	66
7. ANOVA分析.....	67~68
8. 分析各組間之差異情形.....	69
9. ANOVA分析—事後測試.....	70~71

## (四) 複迴歸分析

1. 迴歸原理	73
2. 簡單迴歸與多元迴歸	74
3. 線性迴歸原理	75
4. 迴歸分析的基本概念	76
5. 迴歸分析的基本統計概念	77
6. 相關分析的基本概念	78
7. 不同的線性相關情形圖示	79
8. 相關係數的強度大小與意義	80
9. 迴歸模式之判定係數	81
10. 迴歸模型範例	82
11. 載入Python的套件與範例的資料	83

## (四) 複迴歸分析

12. 建立範例的多元迴歸分析模型.....	84
13. 顯示範例中迴歸模型的判定係數.....	85
14. 範例的迴歸分析模型.....	86
15. 迴歸分析模型的簡化 .....	87~88
16. 迴歸分析模型的簡化 .....	89
17. 曲線(非線性)迴歸 : 迴歸分析模型的精進.....	90
18. 範例轉換之後的 XY 散佈圖.....	91
19. 迴歸分析模型的精簡(簡化與轉換之後).....	92
20. 精簡後的迴歸分析模型.....	93
21. 精簡迴歸分析模型: 縮小自變項X1的數值.....	94

## (五)主成分分析

1. PCA 的動機.....	96
2. 簡介PCA .....	97~98
3. PCA 影片講解 .....	99~101
4. 資料集說明.....	102
5. Python in PCA.....	103
6. PCA對特徵降維.....	104~105
7. 查看PCA降維結果.....	106
8. 降維後的數據轉換成原始數據.....	107
9. PCA降維的優缺點.....	108

# (一)統計量數的概念 及分析資料工具

該ppt的內容來源來自以下教科書

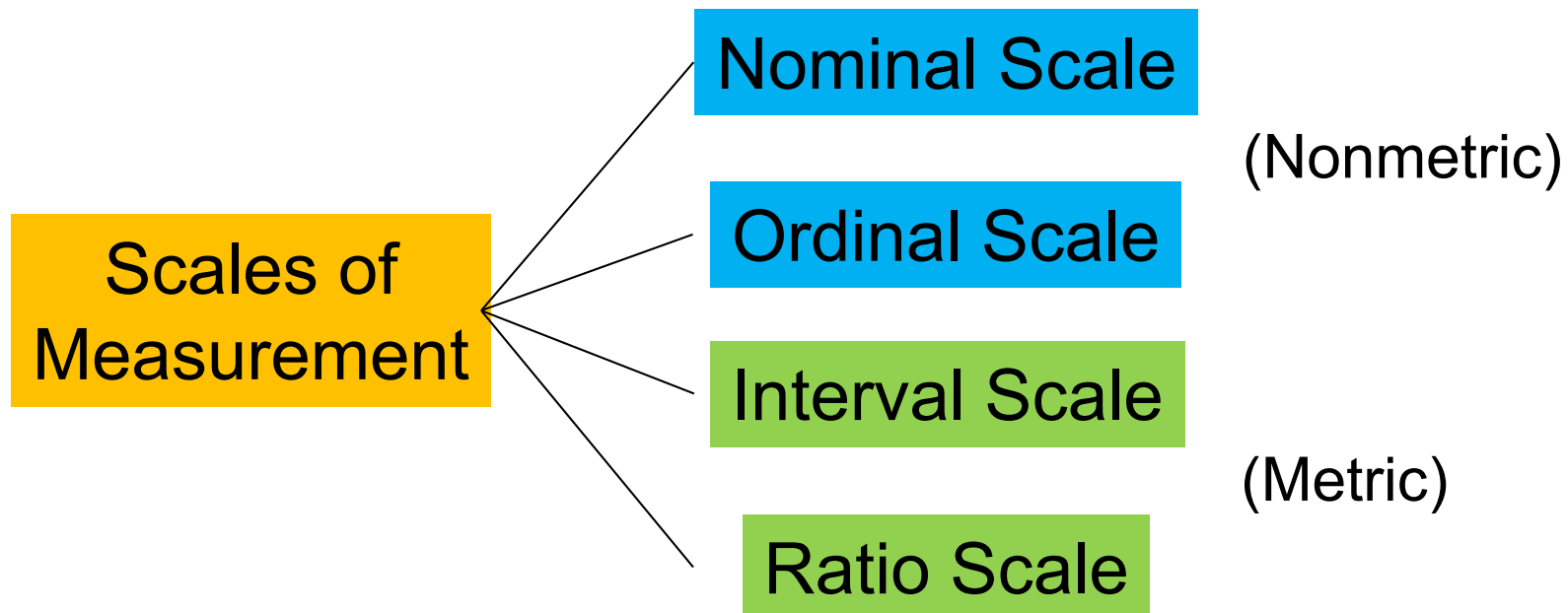
Neil A. Weiss. Introductory Statistics 10th ed., Pearson, Addison Wesley, 2017.

俞洪亮、蔡義清、莊懿妃，2018，商管研究資料分析：SPSS的應用，修訂三版，台北：華泰文化

# 1. 變數資料之類型

變數資料之類型細分為：

- (1) 名目資料(nominal data)，如：婚姻狀況
- (2) 順序資料(ordinal data)，如：礦物的硬度
- (3) 區間資料(interval data)，如：溫度
- (4) 比例資料(ratio data)，如：長度、金錢



## 2. 集中趨勢和相對量數 (central tendency and percentile value)

### (1) 眾數 Mode

- 樣本中出現次數最多的數值

```
>>>import statistics as st  
>>>nums=[1, 2, 3, 5, 6, 6, 6, 4, 7, 8]  
  
#眾數  
>>>st.mode(nums)  
6
```

### 3. 平均數 Mean

#### THE MEAN $\bar{x}$

To find the **mean** of a set of observations, add their values and divide by the number of observations. If the  $n$  observations are  $x_1, x_2, \dots, x_n$ , their mean is

$$\bar{x} = \frac{x_1 + x_2 + \dots + x_n}{n}$$

or in more compact notation,

$$\bar{x} = \frac{1}{n} \sum x_i$$

假設資料內容如下：

14.0, 15.0, 17.0, 16.0, 15.0

$$\bar{X} = \frac{\Sigma X}{n} = \frac{14.0 + \dots + 15.0}{5} = \frac{77}{5} = 15.4$$

```
>>>import statistics as st  
>>>nums=[1, 2, 3, 5, 6, 6, 6, 4, 7, 8]
```

```
#平均數  
>>>st.mean(nums)  
4.8
```



### 3. 中位數 Median

(1) 將樣本從小到大做排序，如果樣本是奇數個中位數即為最中間的值；如果是偶數個，中位數是中間兩個數值的平均

(2) 假設資料內容如下：

1, 2, 3, 5, 6, 6, 6, 4, 7, 8

則中位數為5和6的平均5.5

```
>>>import statistics as st
>>>nums=[1, 2, 3, 5, 6, 6, 6, 4, 7, 8]

# 中位數
>>>st.median(nums)
5.5
```

## 4. 百分位數 Percentiles

- (1) 百分位數，如果將一組數據從小到大排序，並計算相應的累計百分位，則某 $P$ 分位所對應數據的值就稱為第 $P$ 百分位數。
- (2) 至少 $P\%$ 的數據位於第 $P$ 個百分點以下，最多 $(100 - P)\%$ 的數據位於第 $P$ 個百分點以上
- (3) 排序數據從小到大後，計算 $P$ 百分位位置

$$i = \frac{P}{100}(n) \quad n \text{表示資料筆數}$$

- 若 $i$ 是整數，第 $i$ 筆資料與第 $i+1$ 筆資料的平均是 $P$ 百分位數
- 若 $i$ 不是整數，第 $\lceil i \rceil$ 筆資料的平均是 $P$ 百分位數

## 5. 百分位數與四分位數程式範例

(1) 百分比位數  
Percentiles

(2) 四分位數      Quartiles

```
>>>import numpy as np  
>>>nums=np. array([1, 2, 3, 5, 6, 6, 6, 4, 7, 8]  
)
```

#百分比位數 (75%)

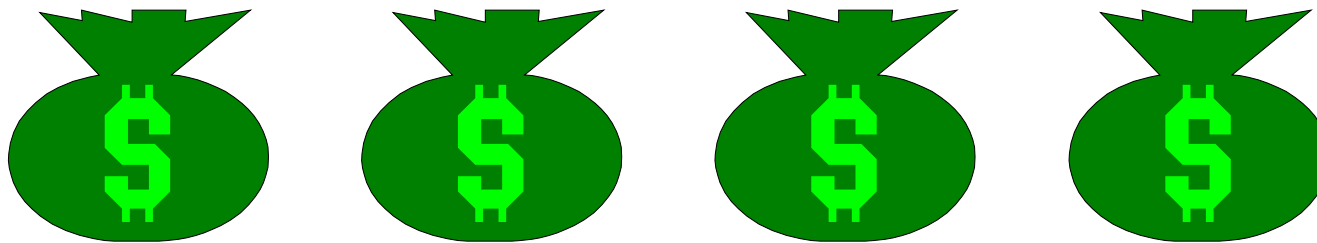
```
>>>np. percentile(nums, 75)  
6
```

#四分比位數 (25%)

```
>>>np. percentile(nums, 25)  
3.25
```

## 6. 差異量數或離散量數 (dispersion)

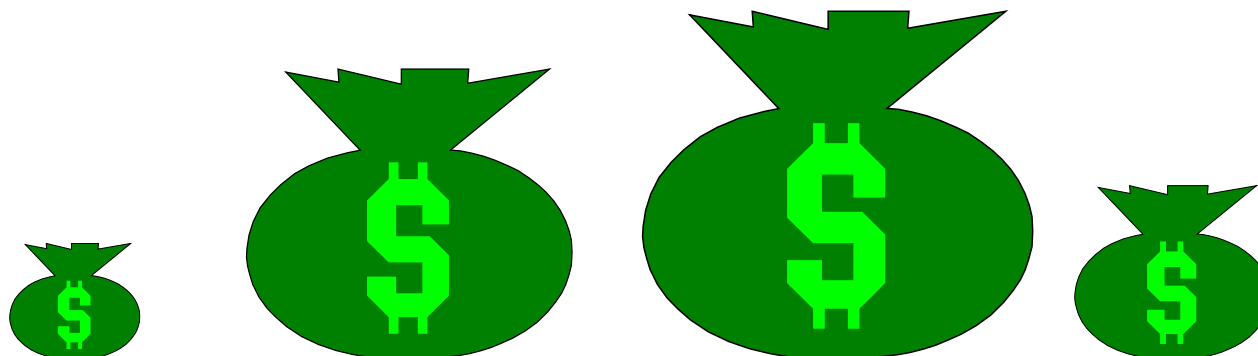
No Variability in Cash Flow



Mean



Variability in Cash Flow



Mean



## 6. 差異量數或離散量數 (dispersion)

衡量資料中各觀測值之差異或離散程度。

(1) 全距(range)。

(2) 樣本的變異數與標準差

(sample variance and standard deviation)。

(3) 變異係數(coefficient of variation, CV)。

(4) 偏態的測定數(skewness)

## 7. 全距(range)程式範例

資料中最大值與最小值的差距

```
>>>import numpy as np  
>>>nums=np. array([1, 2, 3, 5, 6, 6, 6, 4, 7, 8])
```

#最大值

```
>>>np. max(nums)
```

6

#最小值

```
>>>np. min(nums)
```

3.25

#全距

```
>>>np. max(nums) - np. min(nums)
```

7

## 8. 樣本的變異數與標準差 (sample variance and standard deviation)

- (1) 標準差：一種表示分散程度的統計觀念
- (2) 樣本的標準差：母體標準差是通過隨機抽取一定量的樣本( $n < N$ )並計算樣本標準差估計的。
- (3) 假設從一大組數值當中取出樣本數值組合為  $X = \{x_1, x_2, x_3, \dots, x_n\}$ ，則樣本變異數與標準差定義如下：

- 樣本變異數公式：
$$s^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2$$

- 樣本標準差公式：
$$s = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2}$$

## 9. 樣本的變異數與標準差程式範例

```
>>>import statistics as st  
>>>nums =  
    [1, 2, 3, 5, 6, 6, 6, 4, 7, 8]
```

#樣本變異數

```
>>> st. variance(nums)  
5.066666666666666
```

#樣本標準差

```
>>> st. stdev(nums)  
2.250925735484551
```

```
>>>import statistics as st  
>>>nums = [1, 2, 3, 5, 6, 6, 6, 4, 7, 8]
```

#母體變異數

```
>>> st. variance(nums)  
4.5600000000000005
```

#母體標準差

```
>>> st. pstdev(nums)  
2.1354156504062622
```



## 10. 變異係數 (coefficient of variation,CV)

變異係數，是機率分布離散程度的一個正規(Normalized)量度，其定義為標準差與平均值之比。

公式：

$$CV = \frac{\text{標準差}}{\text{平均值}} \times 100\%$$
$$= \frac{s}{\bar{x}} \times 100\% \left( \text{或 } \frac{\sigma}{\mu} \times 100\% \right)$$

優點：變異係數是一個比例尺度，因此在比較兩組因尺度不同或均值不同的數據時，應該用變異係數來作為較的參考。

缺陷：當平均值接近於0的時候，微小的擾動也會對變異係數產生巨大影響，因此造成精確度不足。變異係數無法發展出類似於均值的置信區間的工具。

## 11. 變異係數程式範例

```
>>>import statistics as st  
>>>nums = [1, 2, 3, 5, 6, 6, 6, 4, 7, 8]
```

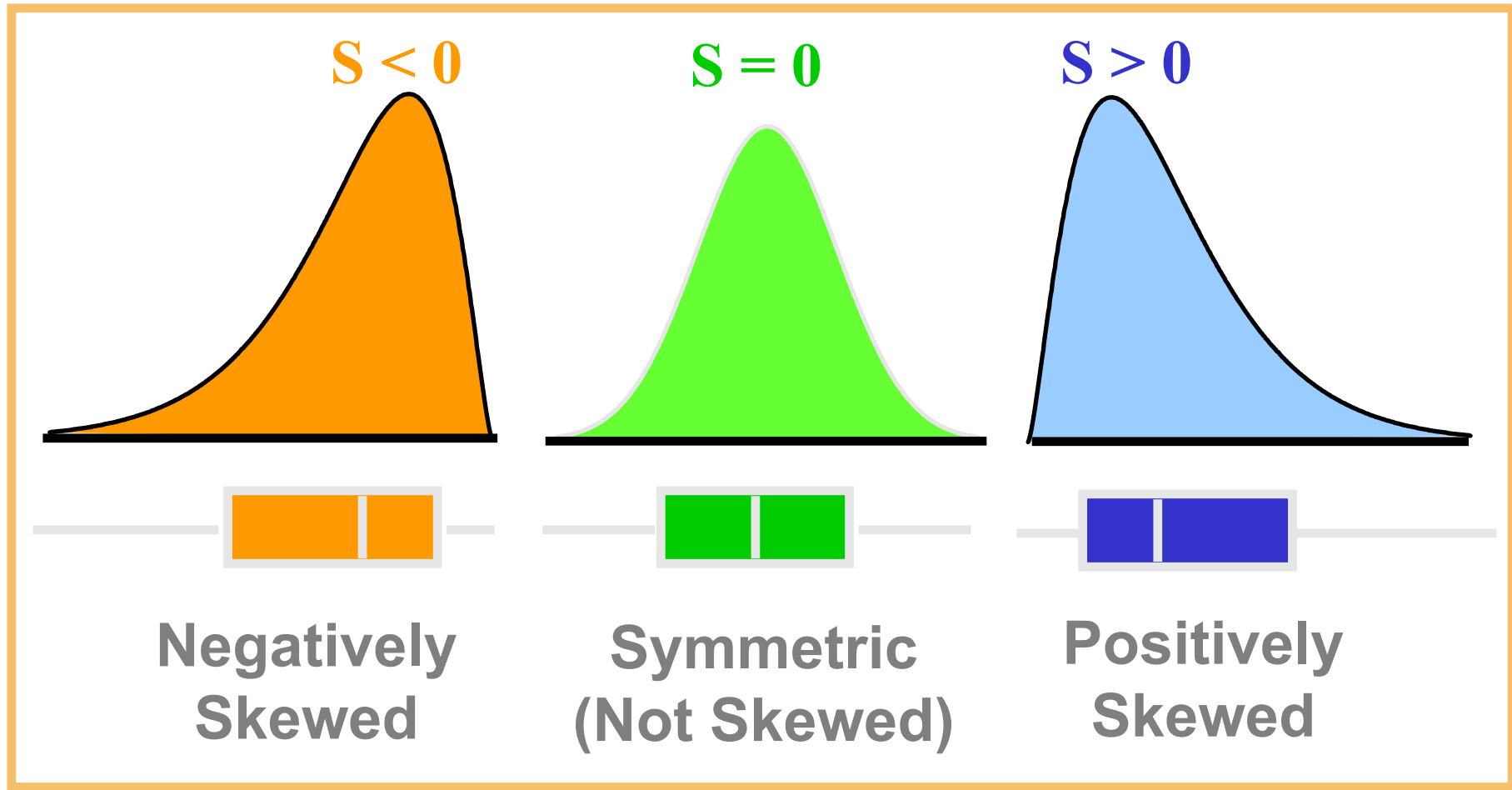
#樣本變異係數

```
>>> st.stdev(nums) * 100 / st.mean(nums)  
46.89428615592815
```

## 12. 偏態的測定數

### Measures of skewness

偏度（Skewness），亦稱歪度，在機率論和統計學中衡量實數隨機變數機率分布的不對稱性

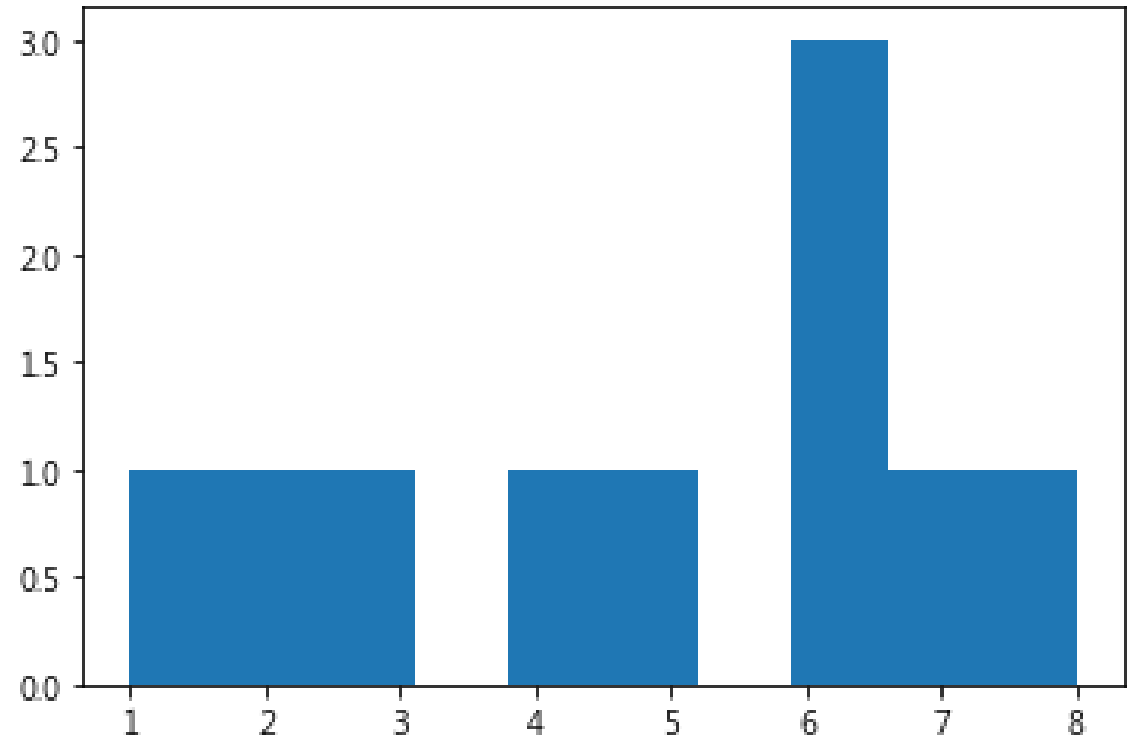


## 13. 偏態的測定數程式範例

```
>>>import scipy.stats as sts  
>>>nums =  
[1, 2, 3, 5, 6, 6, 6, 4, 7, 8]
```

#偏度

```
>>> sts.skew(nums)  
-0.35491672859937834 #左偏
```



```
>>>import matplotlib.pyplot as plt  
>>>nums = [1, 2, 3, 5, 6, 6, 6, 4, 7, 8]
```

#直方圖

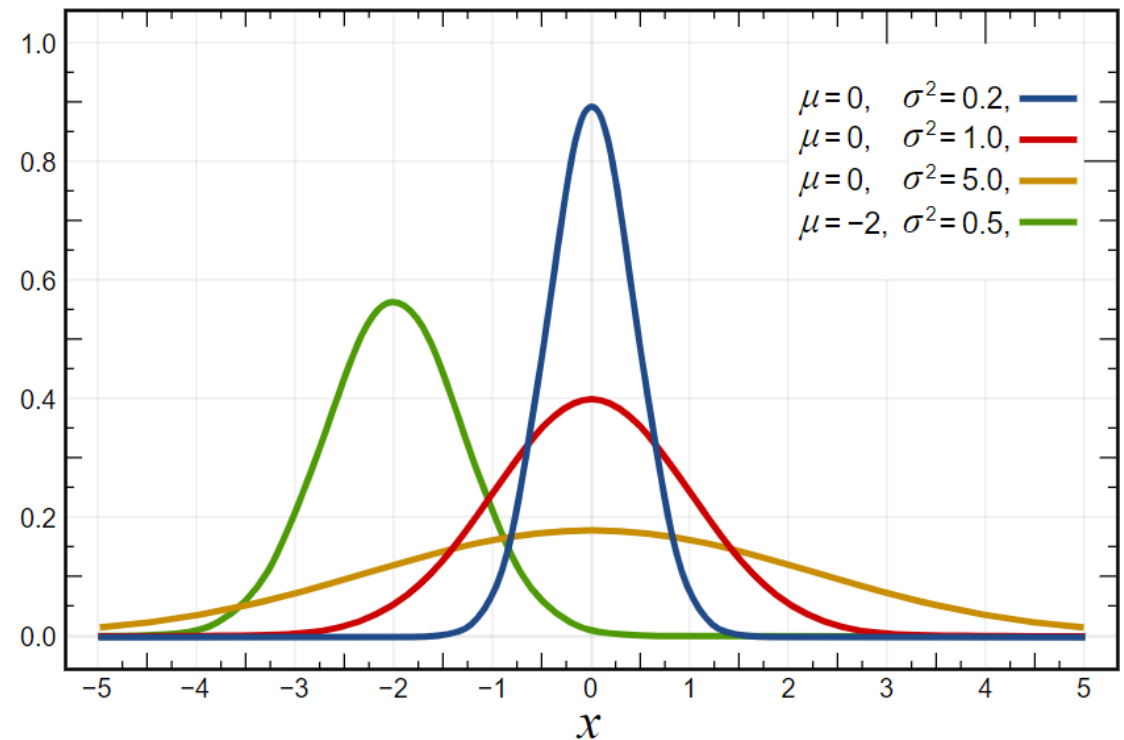
```
>>> plt.hist(nums)
```

## 14. 常態分布 (normal distribution)

又名高斯分布（Gaussian distribution），是一個非常常見的連續機率分布。常態分布在統計學上十分重要，經常用在自然和社會科學來代表一個不明的隨機變量。

若隨機變量  $x$  服從一個位置參數為  $\mu$ 、尺度參數為  $\sigma$  的常態分布，記為：

$$x \sim N(\mu, \sigma^2)$$



## 15. 常態分布程式範例

使用numpy及matplotlib函式庫劃出常態分布圖

```
x = numpy.random.normal(mu, sigma, size)
```

```
# mu 期望值
```

```
# sigma 標準差
```

```
# size 基於normal distribution生成的數量
```

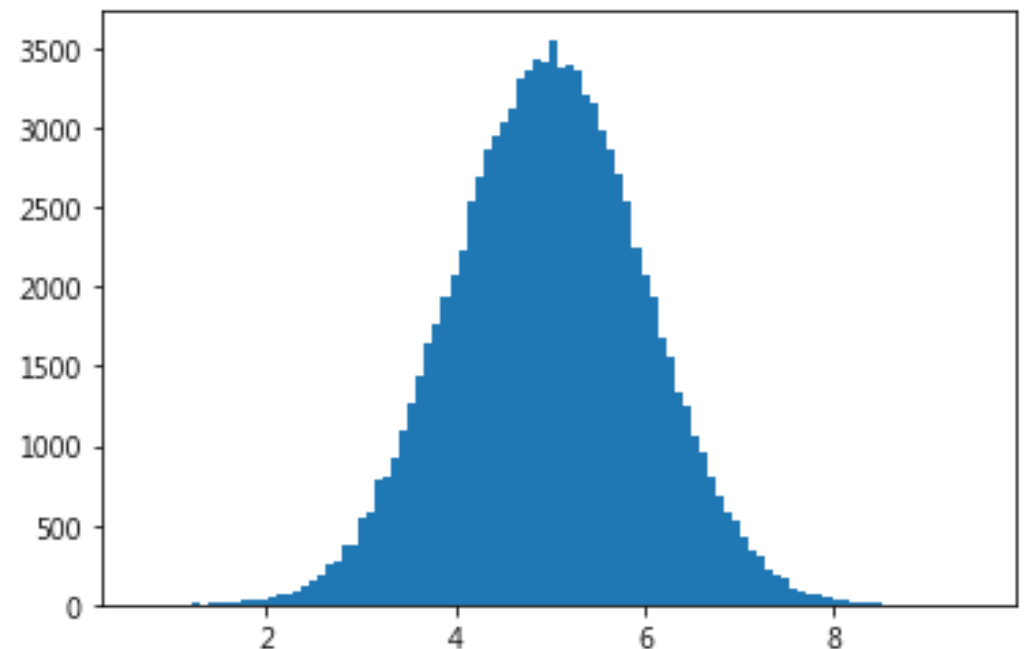
```
>>>import numpy as np
```

```
>>>import matplotlib.pyplot as plt
```

```
x = np.random.normal(5.0, 1.0, 100000)
```

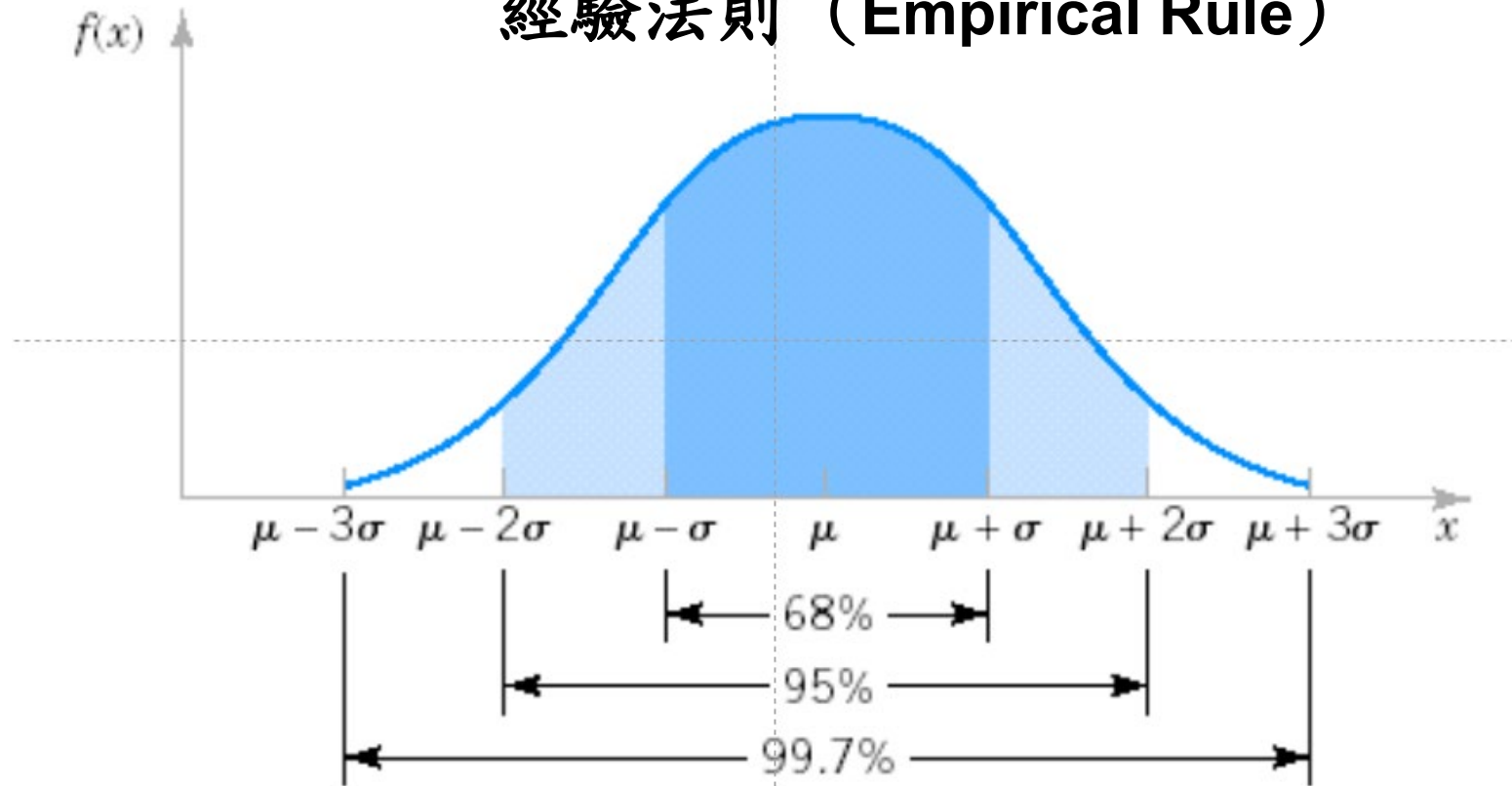
```
plt.hist(x, 100)
```

```
plt.show()
```



## 16. 常態分佈線下的區域

### 經驗法則 (Empirical Rule)



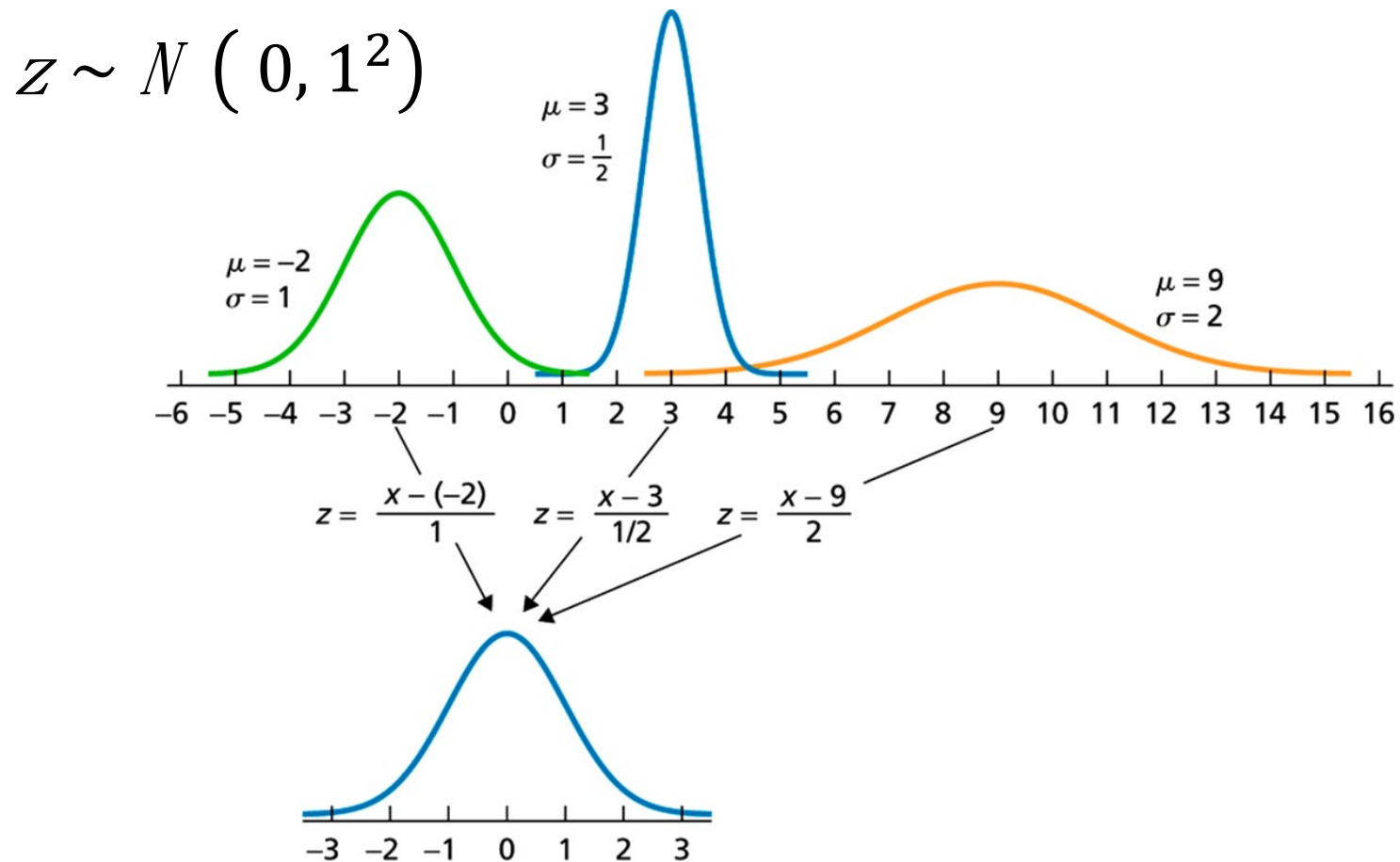
若資料為鐘形分配，則大約有68%的資料會落在  $\pm s$  的範圍內，大約有95%的資料會落在  $\pm 2s$  的範圍內則，大約有**99%的資料會落在  $\pm 3s$  的範圍內**。

考慮一組數據具有近似於常態分布的機率分布。若其假設正確，則約68%數值分布在距離平均值1個標準差之內的範圍，約95%數值分布在距離平均值2個標準差之內的範圍，以及約99.7%數值分布在距離平均值3個標準差之內的範圍

## 17. 標準常態分配: Z 分布 (Standard Normal Distribution)

將任一常態分布經過Z轉換後所得之分佈(仍為常態分布)。

即取平均數為 0 與標準差為 1 之常態分佈。





## 18. 標準分數(Standard Score)

標準分數 (Standard Score，又稱z-score，中文稱為Z-分數或標準化值) 在統計學中是一種無因次值 (Dimensionless value)，是藉由從單一 (原始) 分數中減去母體的平均值，再依照母體 (母集合) 的標準差分割成不同的差距，按照Z值公式，各個樣本在經過轉換後，通常在正、負五到六之間不等。

如  $x \sim N(\mu, \sigma^2)$  標準分數可藉由以下公式求出：

$$Z = \frac{x - \mu}{\sigma}$$

Z值的量代表著原始分數和母體平均值之間的距離，是以標準差為單位計算。在原始分數低於平均值時Z則為負數，反之則為正數。換句話說，Z值是從感興趣的點到均值之間有多少個標準差。

## 19.標準分數程式範例

```
>>> import numpy as np

>>> nums = np.array([1, 2, 3, 5, 6, 6, 6, 4, 7, 8])
>>> mu = np.mean(nums)
>>> std = np.std(nums)
>>> z_score = (nums - mu) / std

-1.77951
-1.31122
-0.842927
0.0936586
0.561951
0.561951
0.561951
-0.374634
1.03024
1.49854
```

## 20.練習

(1)問題：假設有一個資料集內容如下：

`scores = [31, 24, 22, 25, 14, 25, 13, 12, 26, 23, 32, 34, 43, 41, 21, 23, 26, 26, 34, 42]`

(2)請分別計算此資料集的下列資訊：

(1)眾數

(2)平均

(3)中位數

(4)最大值

(5)最小值

(6)全距

(7)樣本變異數與標準

(8)樣本變異係數

(9)偏度

(10)標準分數

解答：ch2\_1a.py

## (二)正確率分析的概念

正確率分析在很大程度上取決於所使用的方法，但是採用哪種方法取決於自變數和依變數所需的資料衡量尺度。在以下各章節中，我們將介紹各種不同的數據類型的方法所會用的各種評價指標。

分析資料工具	自變數	依變數
分類演算法	類別/數值資料	類別資料
變異數分析	類別資料	數值資料
迴歸分析	數值資料	數值資料

正確率、準確率和靈敏度常被搞混，正確率、準確率和靈敏度表示『不相同』的統計意義

# 1. 正確率(Accuracy)高一定好嗎？

- (1) 正確率確實是一個很好很直觀的評價指標，但是有時候正確率高並不能代表一個演算法就必然是好的。

例如地震的預測，類別只有兩個(發生或不發生)。假設一個地震預測系統，每一次都預測不會發生地震，那麼它可能達到99%的正確率，但真的地震來臨時，這個系統去無法示警，導致帶來的損失是巨大的。

- (2) 為什麼99%的正確率的分類器卻不是我們想要的，因為這裡資料分佈不均衡，某一種類別的資料太少，即使忽略這種分類依然可以達到很高的正確率，卻忽視了我們關注的東西。

## 2. 分類演算法的評價指標

- (1) 分類演算法有很多，這裏用簡單的二元分類模型展示。
- (2) 二元分類模型就是輸出結果只有兩種類別的模型，  
例如：（陽性／陰性）（有病／沒病）（垃圾郵件／非垃圾郵件）。
- (3) 就二元分類模型而言，評估分類結果的指標很多，這些指標皆源自混淆矩陣（Confusion Matrix），在二元分類模型中這是  $2 \times 2$  的表格。
- (4) 當訊號偵測（或變數測量）的結果是一個連續值時，類與類的邊界必須用一個臨界值（threshold）來界定。

## 2. 分類演算法的評價指標

舉例來說，用血壓值來檢測一個人是否有**高血壓**，測出的血壓值是連續的實數（從0~200都有可能），以收縮壓140／舒張壓90為閾值，閾值以上便診斷為有高血壓，閾值未滿者診斷為無高血壓。

二元分類模型的個案預測有四種結局：

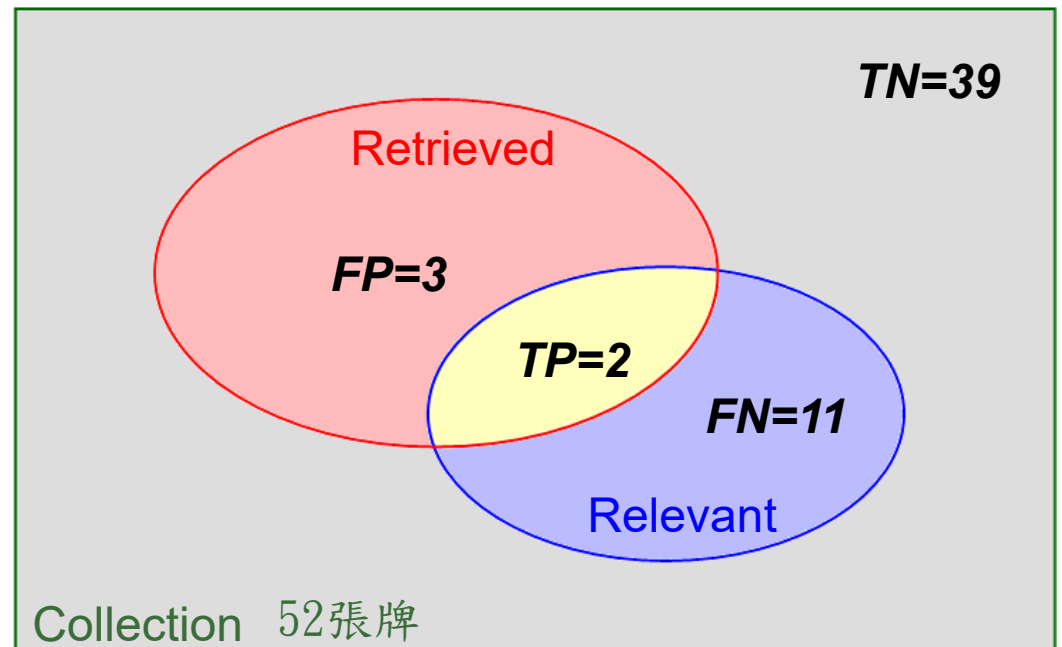
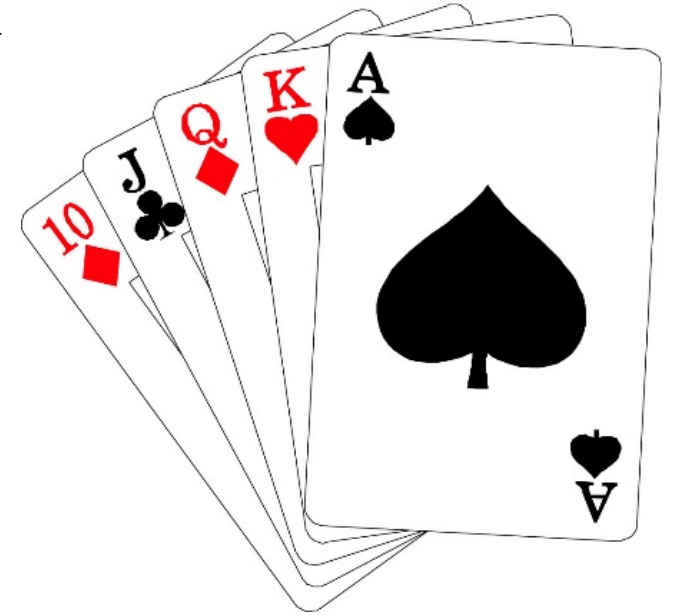
- (1) **真陽性**（True Positive，TP）：診斷為有，實際上也有高血壓
- (2) **偽陽性**（False Positive，FP）：診斷為有，實際卻沒有高血壓
- (3) **真陰性**（True Negative，TN）：診斷為沒有，實際上也沒有高血壓
- (4) **偽陰性**（False Negative，FN）：診斷為沒有，實際卻有高血壓





### 3.Relevant(實際類別) & retrieved (預測類別) (Real) (Predict)

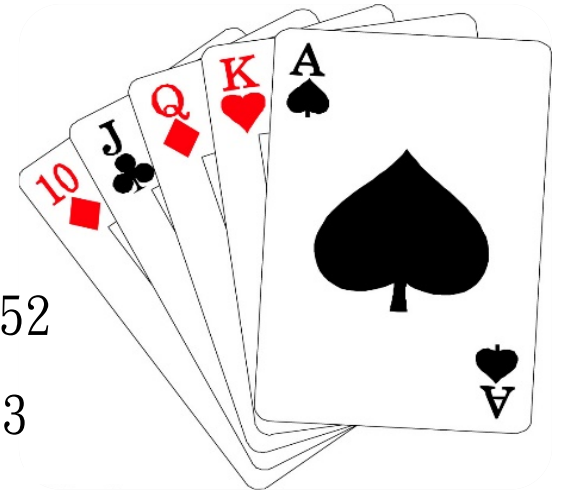
1. 一副撲克牌，洗完牌後隨意抽出5張
2. 得到2張方塊 ◆
3. 試算方塊 ◆ 的各評價指標
  - (1)一副撲克牌共52張
  - (2)一副撲克牌有13張方塊 ◆
  - (3)抽出5張
  - (4)得到2張方塊 ◆





### 3.Relevant(實際類別) & retrieved (預測類別)

(Real) (Predict)



1. 一副撲克牌，洗完牌後隨意抽出5張

2. 得到2張方塊 ◆

3. 試算方塊 ◆ 的各評價指標

(1)一副撲克牌共52張

(2)一副撲克牌有13張方塊 ◆

(3)抽出5張

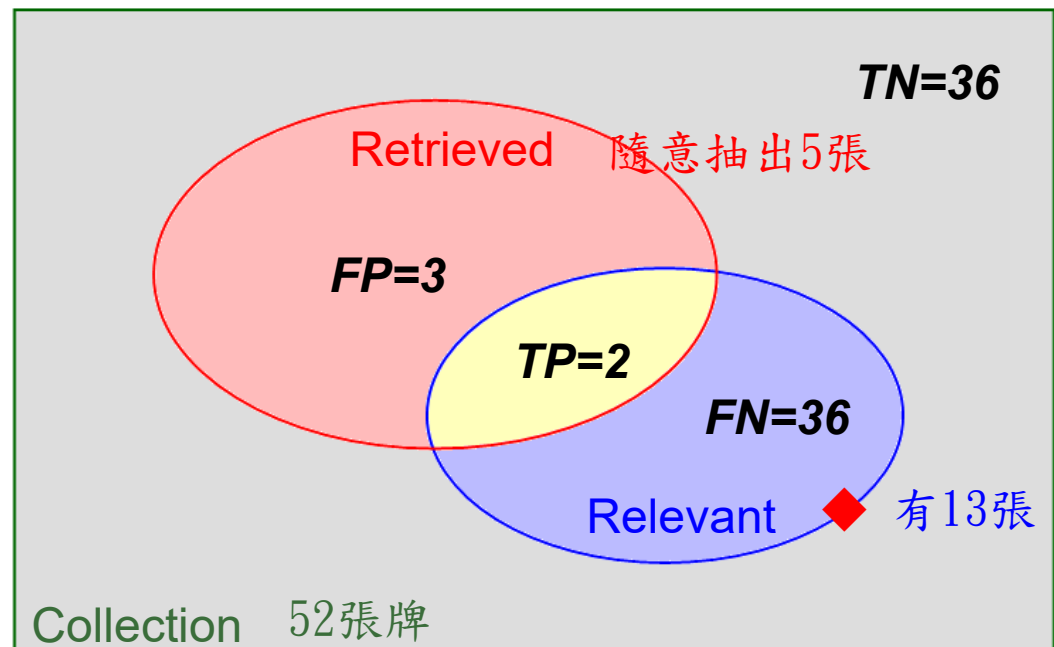
(4)得到2張方塊 ◆

collection = 52

relevant = 13

retrieved = 5

TP = 2





## 4.TP / FP / TN / FN

### 1. True Positive (TP) : 真的正確

(1) 得到2張方塊 ♦

### 2. False Positive (FP) : 錯的正確

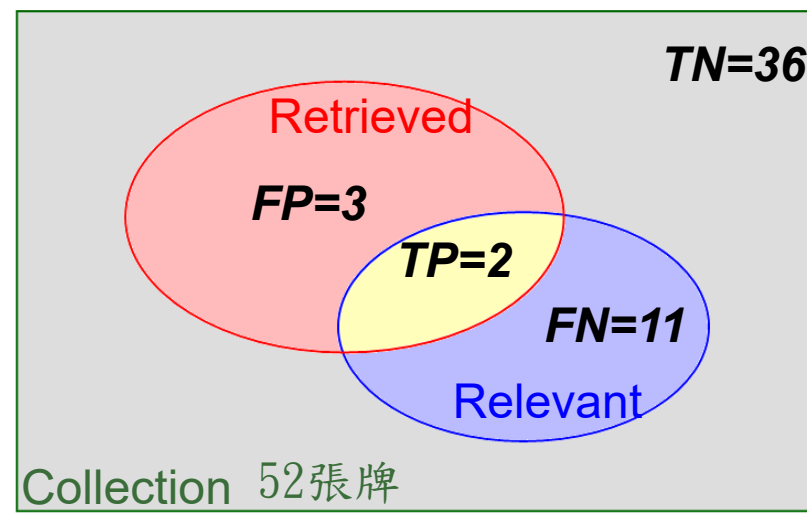
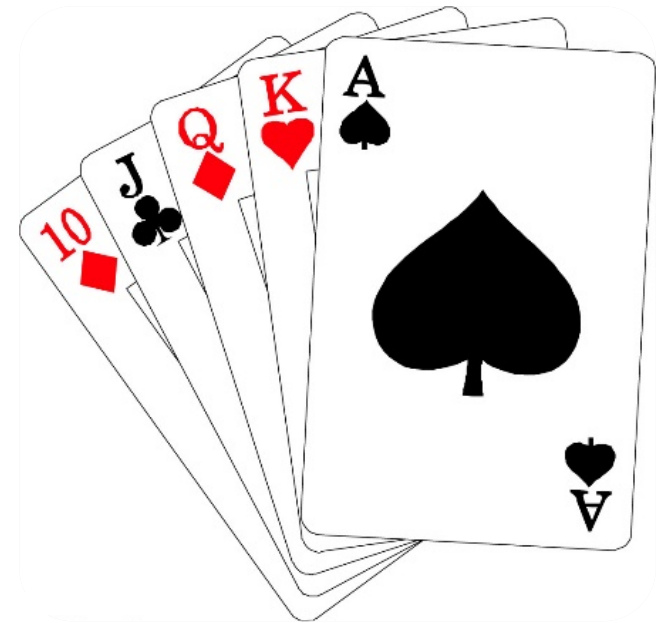
(2) 抽出的牌不是方塊 (♠ ♥ ♣) =  $5 - 2 = 3$

### 3. False Negative (FN) : 錯的否定

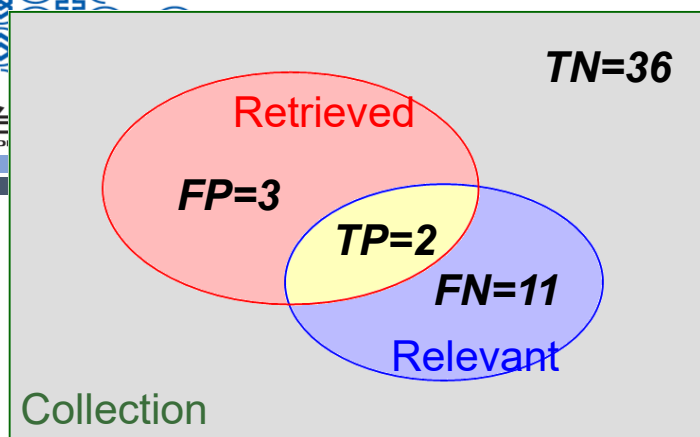
(3) 沒有被抽到的方塊 ♦ =  $13 - 2 = 11$

### 4. True Negative (TN) : 真的否定

(4) 其他花色(♠ ♥ ♣)且沒有被抽出的牌 =  $52 - 2 - 3 - 11 = 36$



## 5. 分類演算法的評價指標



		預測類別		總計
		YES	NO	
實際類別	YES	TP=2	FN=11	P(實際為YES)=13
	NO	FP=3	TN=36	N(實際為NO)=39
總計		P' (被分為為 YES)=5	N' (被分為為 NO)=37	總合S (= P+N or P'+N')=52

### 1. 正確率 (accuracy)

**accuracy** = (TP + TN) / S，正確率是最常見的評價指標。很容易理解，這個就是被分對的樣本數除以所有的樣本數。

### 2. 錯誤率 (error rate)

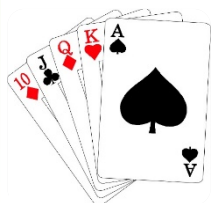
**error rate** = (FP + FN) / S 描述被分錯的樣本數除以所有的樣本數。

錯誤率則與正確率互斥，所以 **accuracy = 1 - error rate**。

$$\text{正確率 (accuracy)} = (2+36)/52 = 0.7308$$

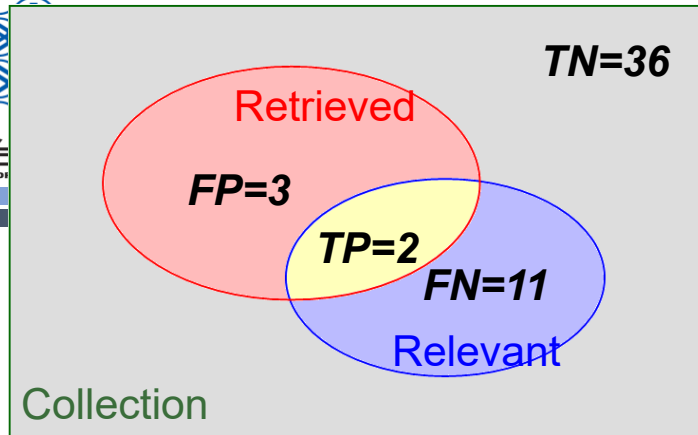
$$\text{錯誤率 (error rate)} = (3+11)/52 = 0.2692$$

$$= 1 - 0.7308$$



TP=2	FN=11
FP=3	TN=36

## 5. 分類演算法的評價指標



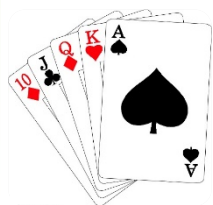
		預測類別		總計
		YES	NO	
實際類別	YES	TP=2	FN=11	P(實際為YES)=13
	NO	FP=3	TN=36	N(實際為NO)=39
總計		P' (被分為為 YES)=5	N' (被分為為 NO)=37	總合S (= P+N or P'+N')=52

3. 靈敏度 (Sensitivity) or (真陽性率, true positive rate)

**sensitivity** = (TP / P), 表示的是樣本中實際符合某特定條件的，被正確診斷為符合那個特定條件結果的比率。

4. 專一性 or 特異度 (Specificity) or (真陰性率, true negative rate)

**specificity** = (TN / N), 表示的是樣本中實際不符合某特定條件的，被正確診斷為不符合那個特定條件的比率。



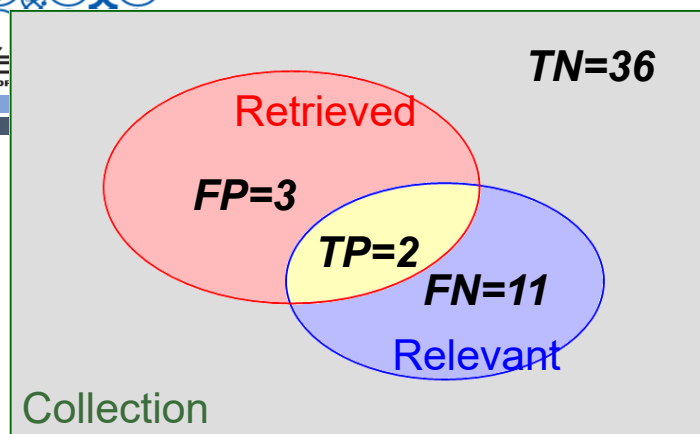
TP=2	FN=11
FP=3	TN=36

靈敏度 (Sensitivity) =  $2 / (2 + 11) = 0.1538$

專一性 (Specificity) =  $36 / (3 + 36) = 0.9231$



## 5. 分類演算法的評價指標

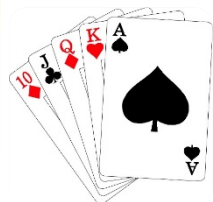


		預測類別		總計
		YES	NO	
實際類別	YES	TP=2	FN=11	P(實際為YES)=13
	NO	FP=3	TN=36	N(實際為NO)=39
總計		P' (被分為為 YES)=5	N' (被分為為 NO)=37	總合S (= P+N or P'+N')=52

5. 假陽性率 (False Positive Rate) (第一類錯誤)

**false positive rate** = (FP / N)，表示的是樣本中實際不符合某特定條件，但根據診斷被識別為符合那個特定條件的比率。

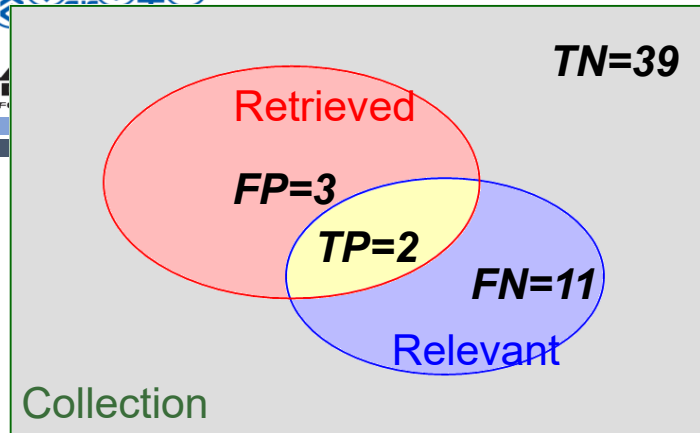
所以也稱誤診率 =  $1 - \text{specificity}$  (TN / N)



TP=2	FN=11
FP=3	TN=36

$$\begin{aligned}
 \text{假陽性率 (false positive rate)} &= 3 / (3 + 36) \\
 &= 0.0769 \\
 &= 1 - 36 / (3 + 36)
 \end{aligned}$$

## 5. 分類演算法的評價指標



		預測類別		總計
		YES	NO	
實際類別	YES	TP=2	FN=11	P(實際為YES)=13
	NO	FP=3	TN=36	N(實際為NO)=30
總計		P' (被分為為 YES)=5	N' (被分為為 NO)=47	總合S (= P+N or P'+N')=52

6. 假陰性率 (False Negative Rate) (第二類錯誤)

**false negative rate** =  $(FN / P)$ ，表示的是樣本中實際符合某特定條件，但根據診斷被識別為不符合那個特定條件的比率。

所以也稱漏診率 =  $1 - \text{sensitivity}(TP / P)$

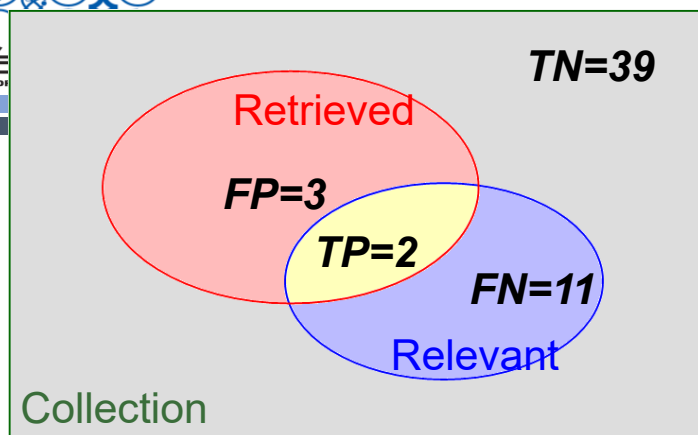


TP=2	FN=11
FP=3	TN=36

$$\begin{aligned}
 \text{假陰性率 (false negative rate)} &= 11 / (2 + 11) \\
 &= 0.8462 \\
 &= 1 - 2 / (2 + 11)
 \end{aligned}$$



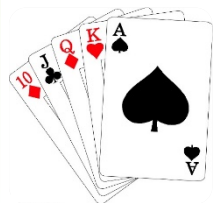
## 5. 分類演算法的評價指標



		預測類別		總計
		YES	NO	
實際類別	YES	TP=2	FN=11	P(實際為YES)=13
	NO	FP=3	TN=36	N(實際為NO)=39
總計		P' (被分為為 YES)=5	N' (被分為為 NO)=47	總合S (= P+N or P'+N')=52

## 7. 精確率(Precision)

$\text{precision} = \text{TP} / (\text{TP} + \text{FP})$  是精確性的度量，表示被分為陽性(正例)的樣本中實際為陽性(正例)的比例。





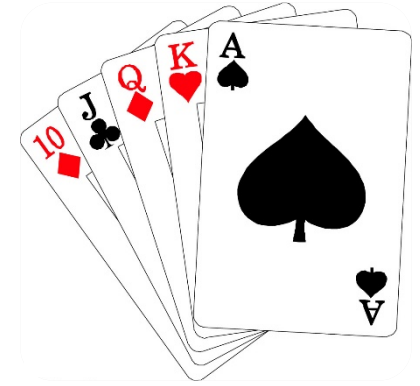
TP=2	FN=11
FP=3	TN=36

$$\text{精確率}(\text{precision}) = 2 / (2 + 3) = 0.4$$



## 6. 練習 I

1. 一副撲克牌，洗完牌後隨意抽出5張
2. 得到2張方塊 
3. 試算方塊  的各評價指標



# 52張鋪克牌(整體總量)

collection = 52

# 抽出5張(預測類別)

retrieved = 5

# 共13張方塊(實際類別)

relevant = 13

# 真的正確(2張方塊)

tp = 2

# 錯的正確

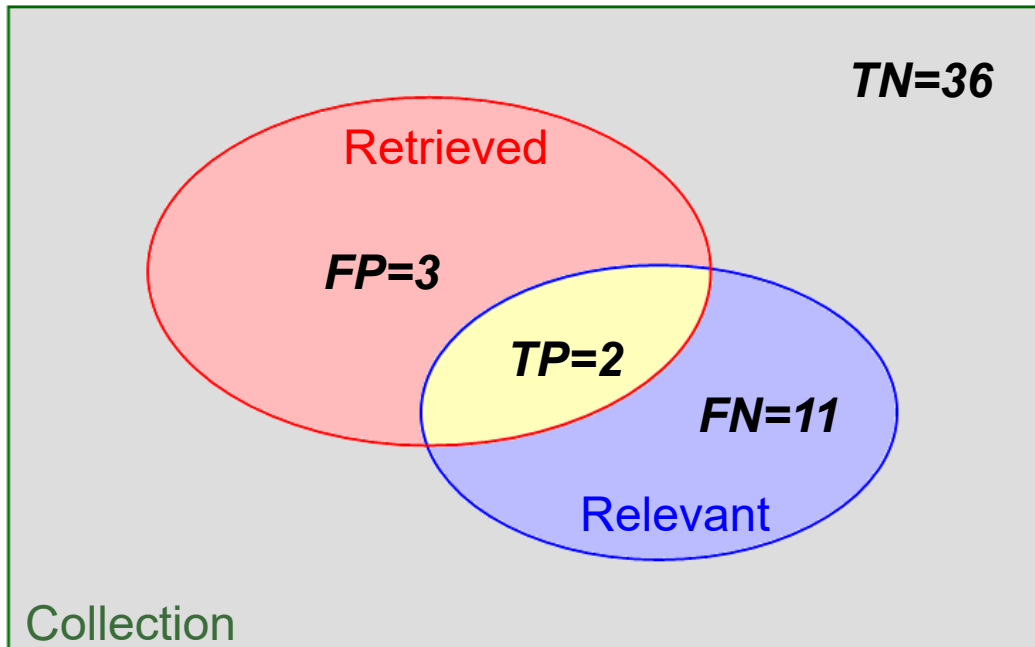
fp = retrieved - tp

# 真的否定

fn = relevant - tp

# 錯的否定

tn = collection - (tp + fp + fn)







# 正確率

```
accuracy = (tp + tn) / collection
```



```
accuracy = (accuracy * 100)
```

```
print("正確率：" + str(round(accuracy,2)) + "%")
```

# 錯誤率

```
error_rate = (fp + fn) / collection
```

```
error_rate = (error_rate * 100)
```

```
print("錯誤率：" + str(round(error_rate,2)) + "%")
```

# 靈敏度(recall)

```
sensitivity = tp / (tp + fn)
```

```
sensitivity = (sensitivity * 100)
```

```
print("靈敏度(recall)：" + str(round(sensitivity,2)) + "%")
```

# 專一性

```
specificity = tn / (fp + tn)
```

```
specificity = (specificity * 100)
```

```
print("專一性：" + str(round(specificity,2)) + "%")
```

# 假陽性

```
fp_rate = fp / (fp + tn)
```

```
fp_rate = (fp_rate * 100)
```

```
print("假陽性：" + str(round(fp_rate,2)) + "%")
```

# 假陰性

```
fn_rate = fn / (tp + fn)
```

```
fn_rate = (fn_rate * 100)
```

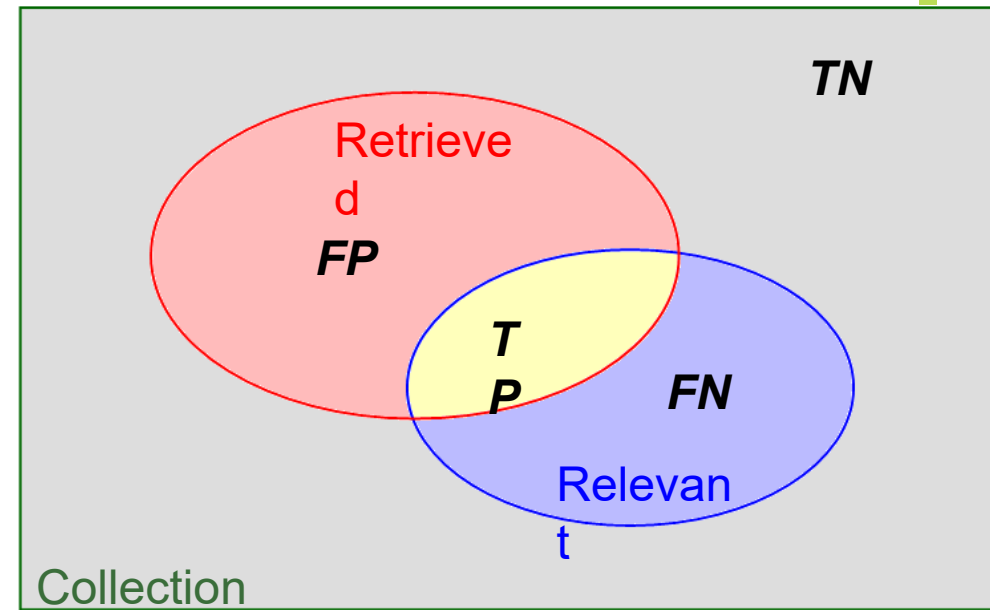
```
print("假陰性：" + str(round(fn_rate,2)) + "%")
```

# 精確率

```
precision = tp / (tp + fp)
```

```
precision = (precision * 100)
```

```
print("精確率：" + str(round(precision,2)) + "%")
```



## 執行結果

正確率：73.08%

錯誤率：26.92%

靈敏度(recall)：15.38%

專一性：92.31%

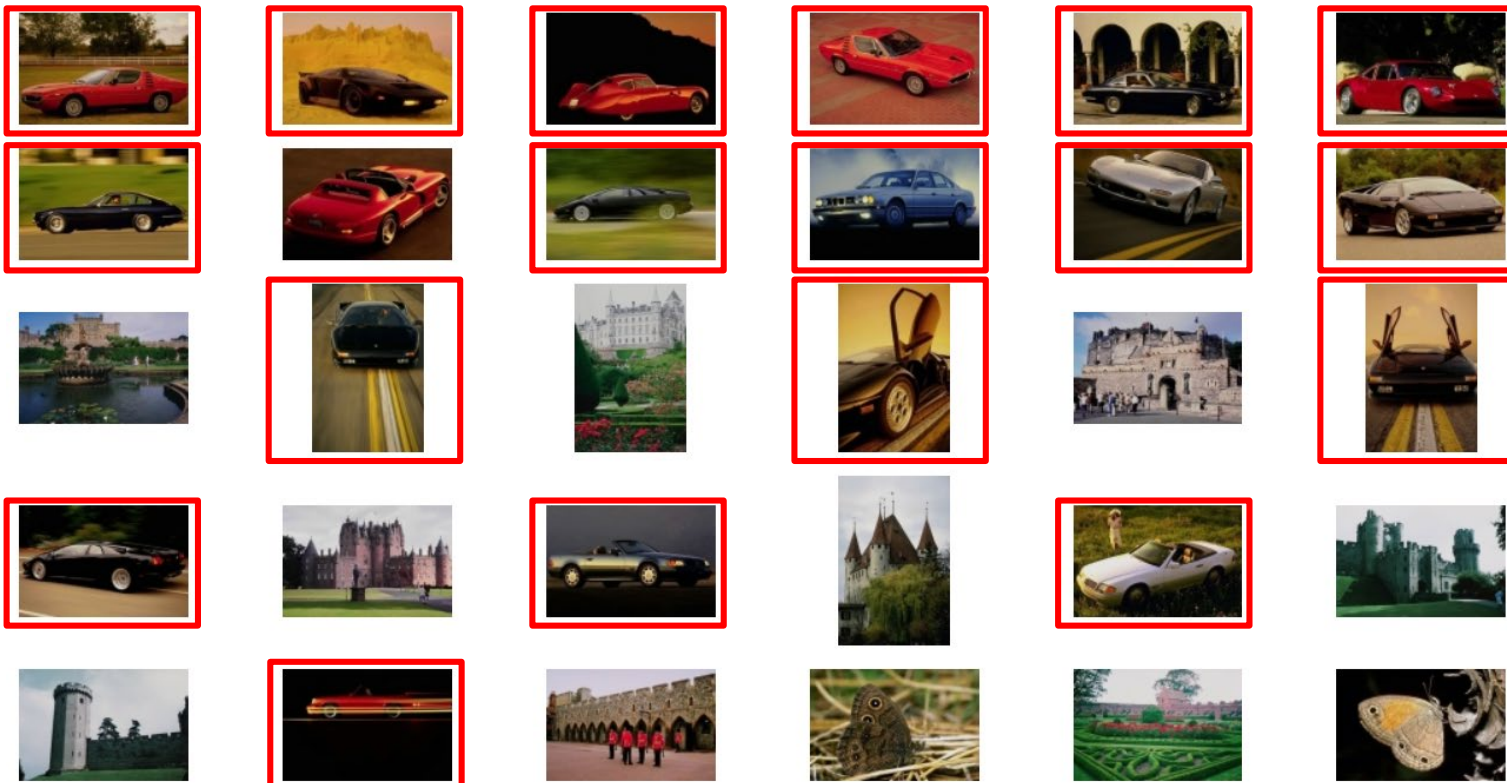
假陽性：7.69%

假陰性：84.62%

精確率：40.0%

## 7. 練習 II

1. 收集**100種**物體(如車子、椅子、房子…等)的圖片**各50張**，供搜尋資料庫使用
2. 一**搜尋系統搜尋**關於“車”的圖片，並**得到30張圖片**，而其中有**19張照片**中**真的有包含車子**
3. 試算此搜尋結果的各評價指標



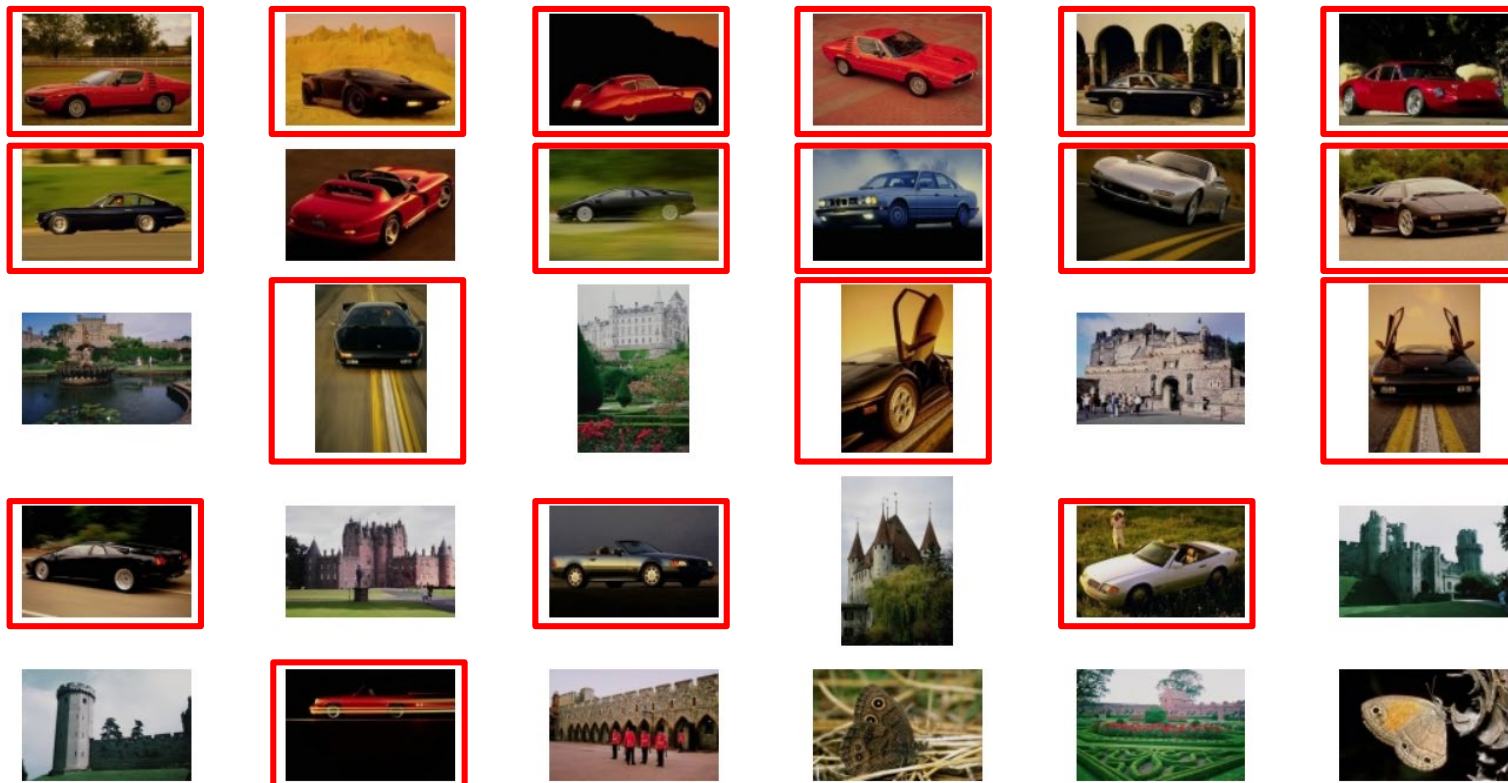
## 7. 練習 II

collection = 100x50  
relevant = 50

1. 收集100種物體(如車子、椅子、房子…等)的圖片各50張，供搜尋資料庫使用
2. 一搜尋系統搜尋關於“車”的圖片，並得到30張圖片，而其中有19張照片中真的有包含車子
3. 試算此搜尋結果的各評價指標

retrieved = 30

TP = 19



# (三)單因子變異數分析

## 1-ANOVA

### (Analysis of Variance)

# 1. 變異數分析的概念

檢定三個以上的獨立母體之平均值是否相等時，可採用變異數分析 (analysis of variance; ANOVA)。

$$H_0 : \mu_1 = \mu_2 = \cdots = \mu_k$$

$H_1$  : 並非所有的  $\mu_i$  皆相等

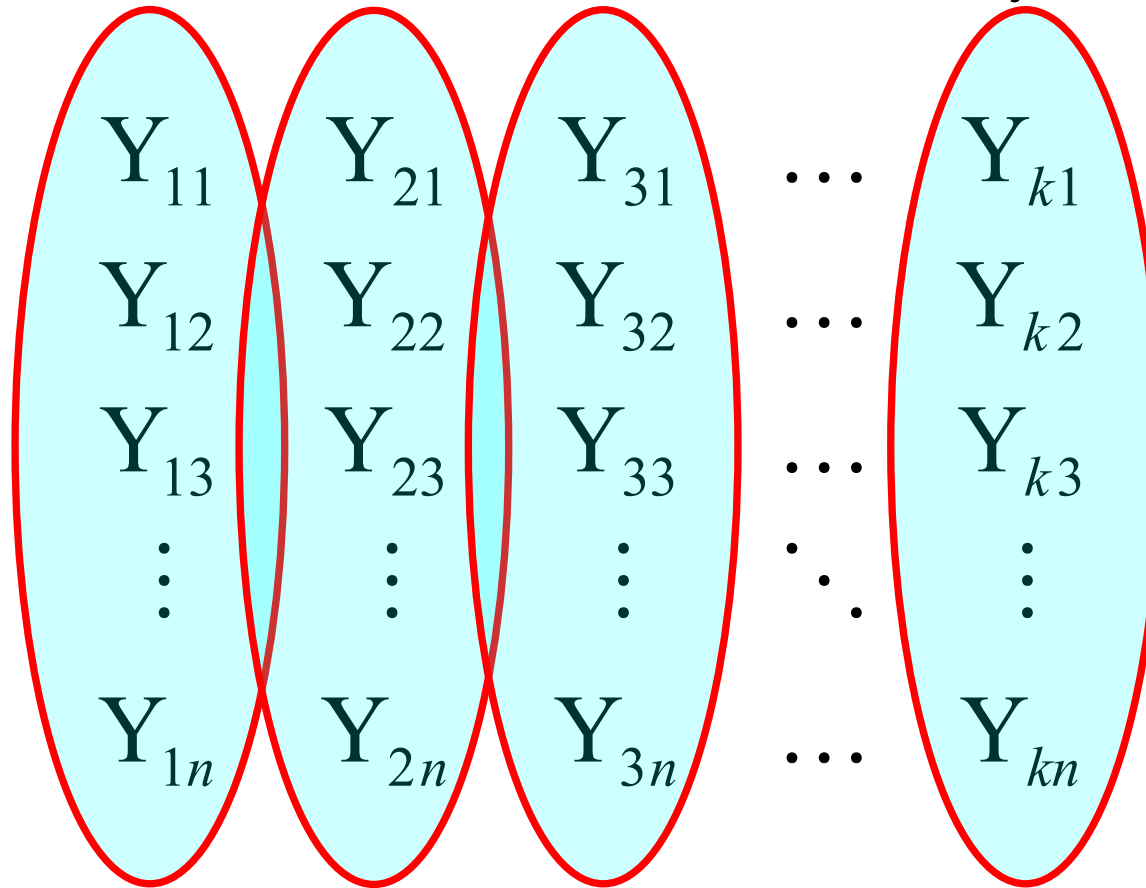
進行 ANOVA 分析時，均必須符合以下之假定：

- (1) 各母體呈常態分配。
- (2) 變異數同質：各母體的變異數  $\sigma^2$  都相等。
- (3) 所有樣本都是隨機抽樣，而且彼此獨立，可以進行累積與加減。
- (4) 對極端值應有足夠的敏感性。



## 2. 變異數分析的數學模型

假設我們有一組樣本數據集,  $Y_{ij}$



$$\bar{Y}_{total} = \sum_i \sum_j Y_{ij} / \sum_i n_i$$

$$\bar{Y}_i = \sum_j Y_{ij} / n_i$$

for each  $i = 1, 2, \dots, k$

此外，我們假設數據集可以按其列分類，我們想了解如果每個觀察  $Y_{ij}$  是否可以計算為  $Y_{ij} = \mu_i + \varepsilon_{ij}$ 。

### 3. 變異數分析原理說明

總變異可分為兩部分，即組間變異與組內變異

每個觀察值與總平均差異的來源

(1)來自分組或方法別所造成的差異（組間變異）

(2)來自觀察值本身的個別差異（組內變異）

檢定統計量： $F$ -值來進行

$F$ 檢定值越大→隱含各組存在差異

(1)組間均方  $>$  組內均方

(2)組間變異量 $>$ 組內變異量

### 3. 變異數分析原理說明

$$TSS = \sum_i \sum_j (Y_{ij} - \bar{Y}_{total})^2$$

**TSS 總變異**  
(Total Sum of Squares)

**BSS 組間變異**  
(Treatment Sum of Squares)

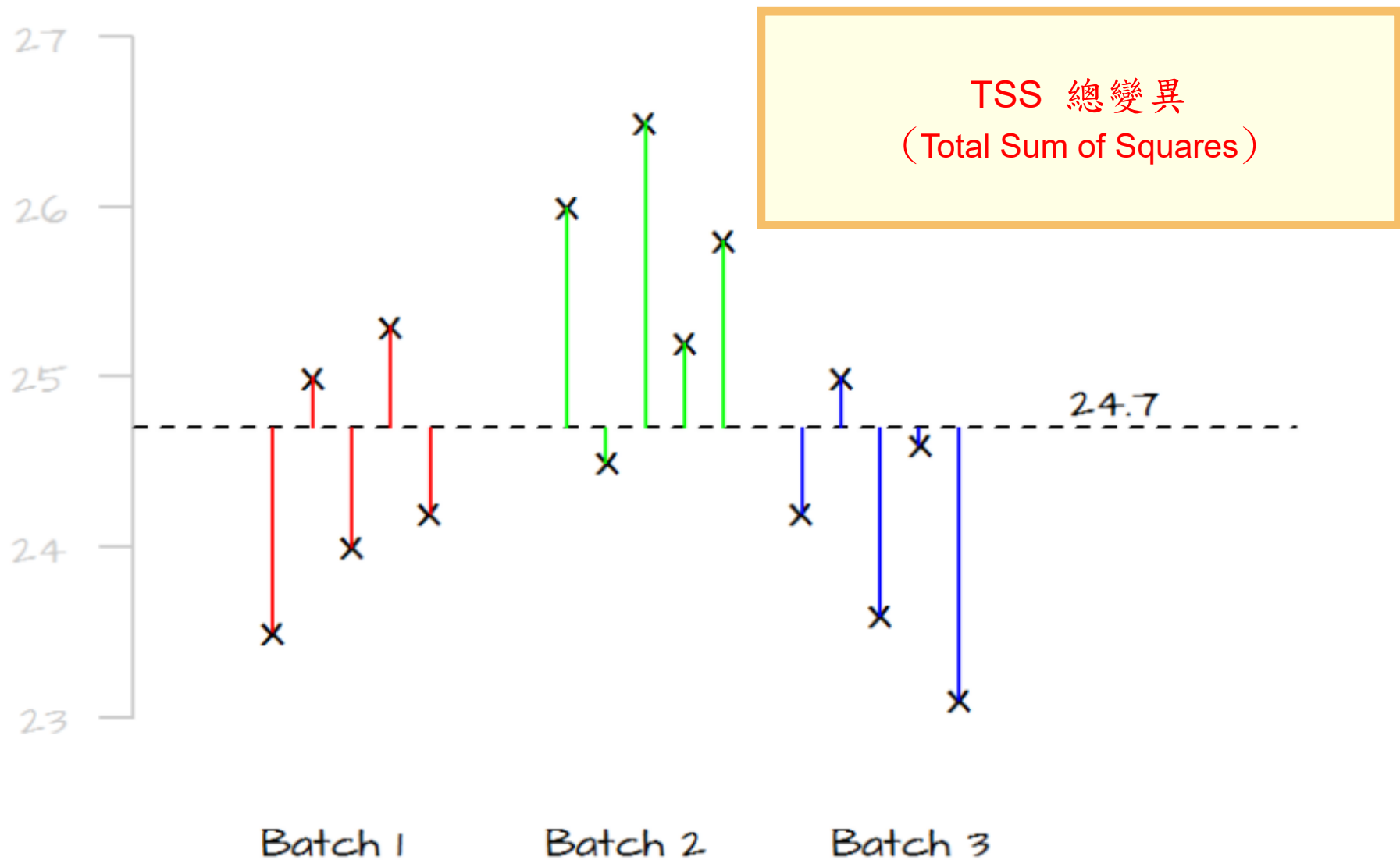
$$BSS = \sum_i n_i (\bar{Y}_i - \bar{Y}_{total})^2$$

**WSS 組內變異**  
(Error Sum of Squares)

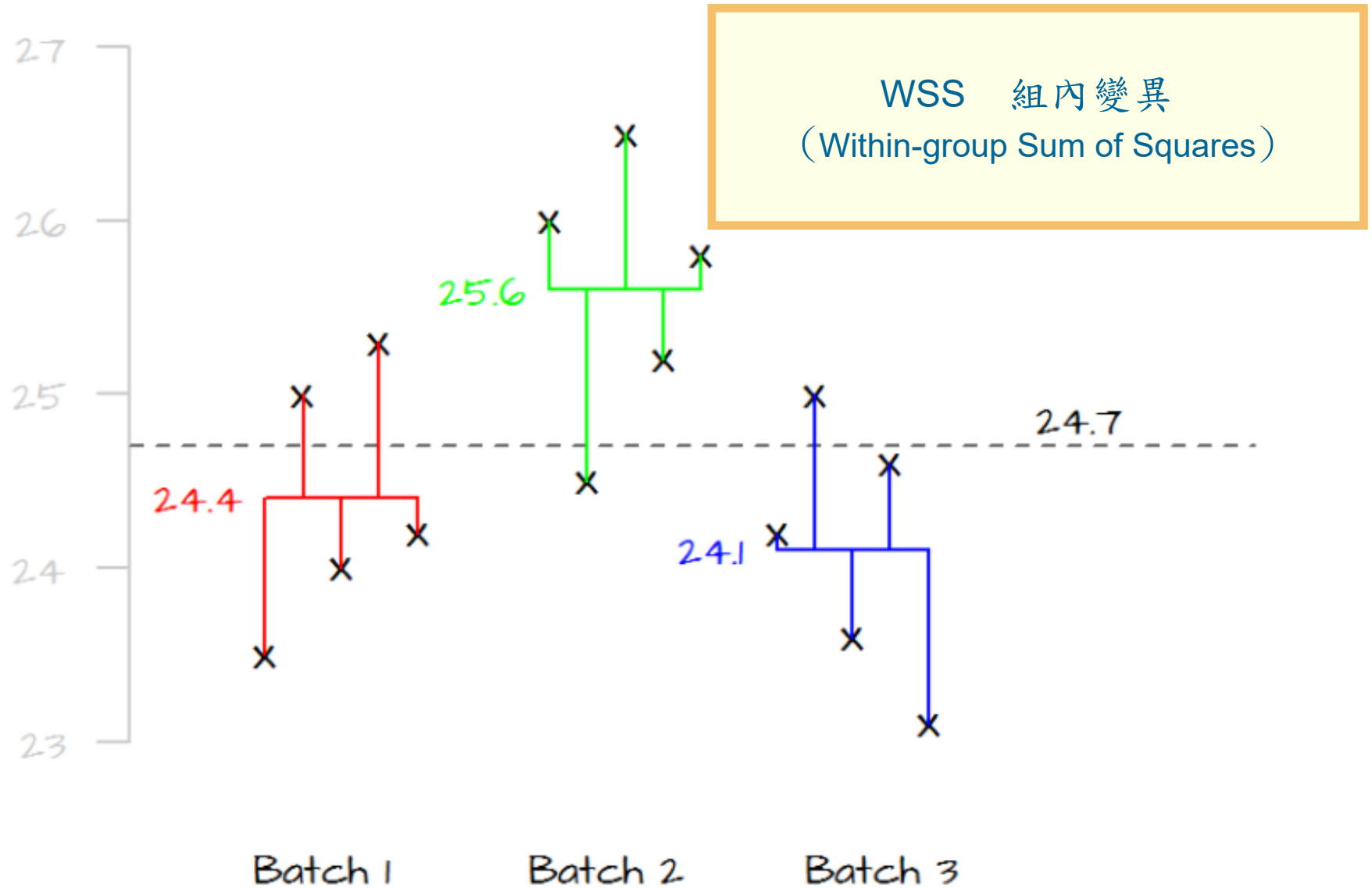
$$WSS = \sum_i \sum_j (Y_{ij} - \bar{Y}_i)^2$$



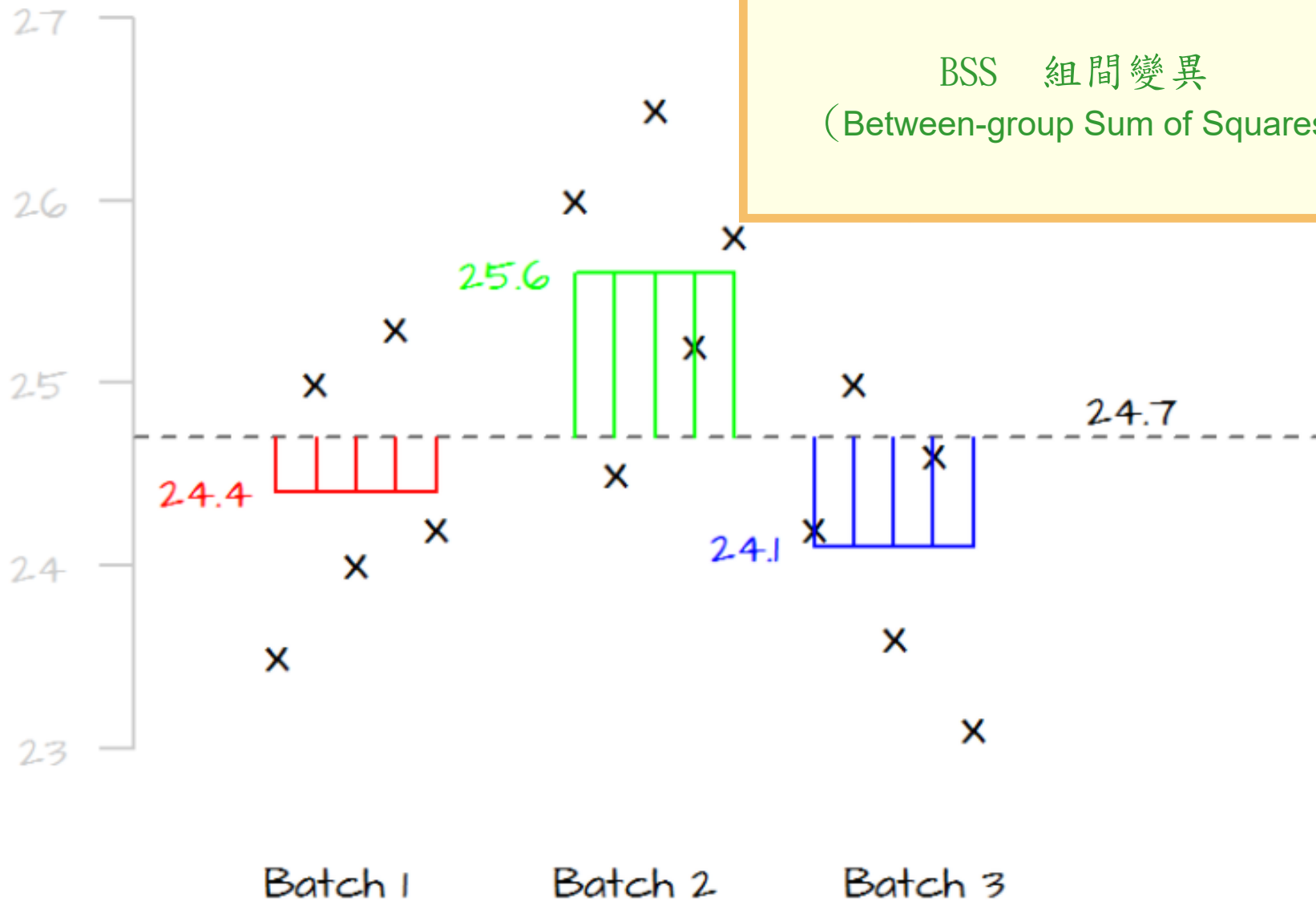
### 3. 變異數分析原理說明



### 3. 變異數分析原理說明



### 3. 變異數分析原理說明



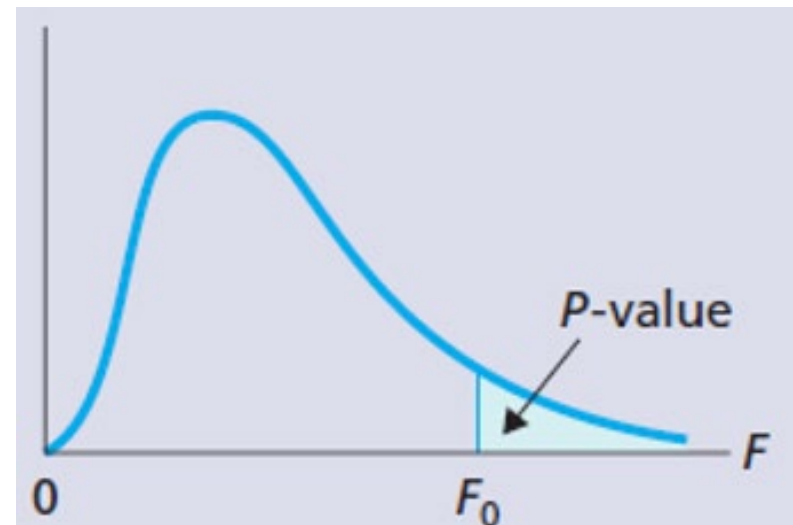
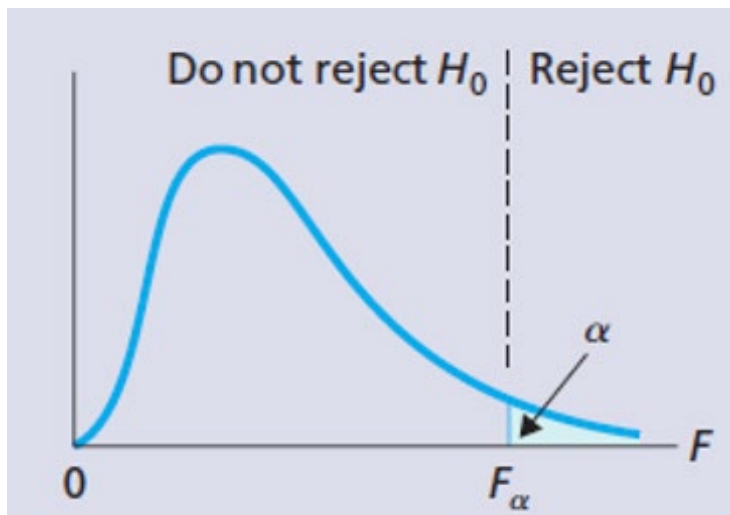
### 3. 變異數分析原理說明

變異數分析摘要表

	離均差平方和(SS)	自由度(DF)	均方和(MS)	F (檢定)	顯著性
組間	BSS (組間變異)	$DF_B = (\# \text{組別} - 1) = K - 1$	$MS_B$	$F_0 = MS_B / MS_W$	p-值
組內	WSS (組內變異)	$DF_W = (N - 1) - (K - 1) = N - K$	$MS_W$		
全體	TSS (總變異)	$DF_T = (\# \text{樣本數} - 1) = (N - 1)$			

如果  $F_0 \geq F_\alpha$  (p-值很小)，即組間變異顯著，代表所檢定的組別中，最少有一組之平均數是與其他組有顯著差異的，因此拒絕  $H_0$

如果  $F_0 < F_\alpha$  (p-值很不夠小)，即組間變異不顯著 (在  $\alpha$  水準下)，無法拒絕  $H_0$



## 4. 變異數分析範例1

變異數分析可檢驗不同生產線、生產方法或多群作業員在生產績效上是否有差異

### (1) 範例

A. 假設有3條生產線

B. 廠商想檢測各生產線的產量是否不同

### (2) 統計概念

A. 比較3條生產線的產量平均數是否存在顯著差異

B. 設定顯著水準( $\alpha$ )=0.05

## 4. 變異數分析範例1

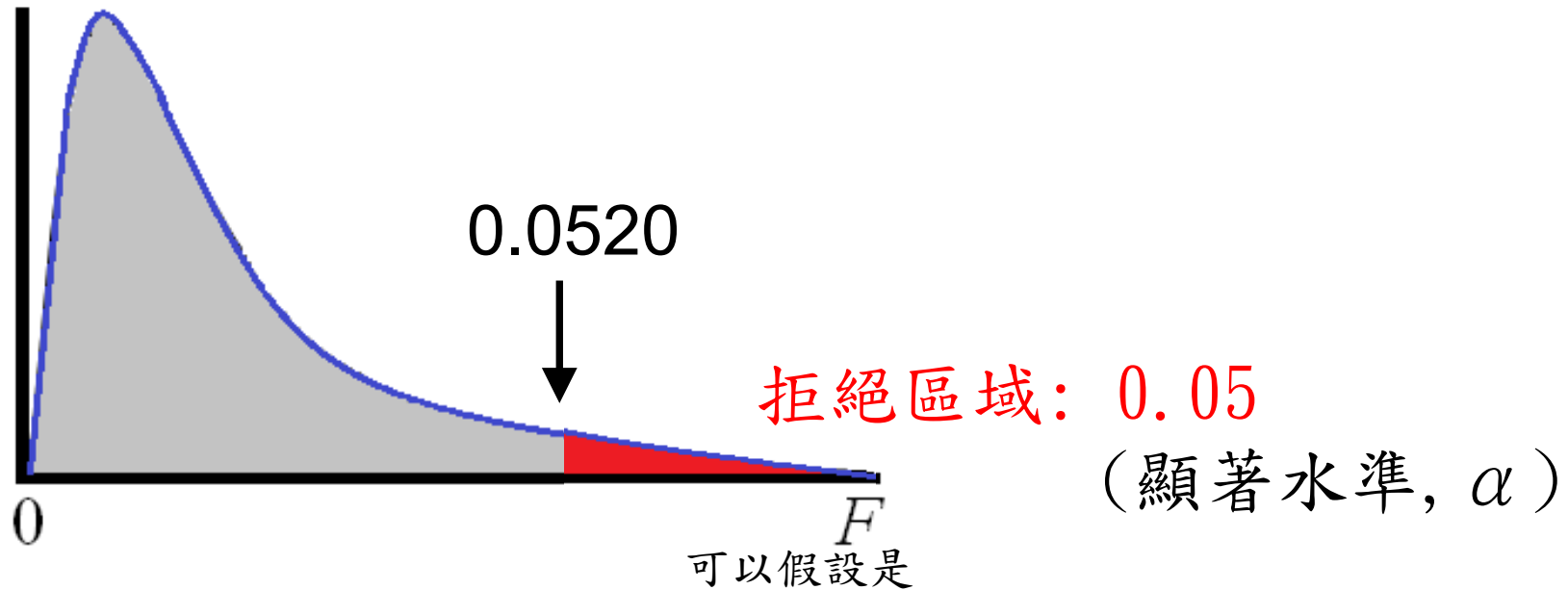
統計假設：

$H_0: \mu_1 = \mu_2 = \mu_3$  (3條生產線的平均產量皆相等)

$H_1$ ：至少有一生產線的平均產量與其他生產線的平均產量不同

摘要				
組	個數	總和	平均	變異數
Method 1	125	6902.224	55.21779	187.6627
Method 2	90	4683.064	52.03404	99.92591
Method 3	100	5189.13	51.8913	98.33051

## 4-1. 變異數分析範例1分析結果



ANOVA						
變源	SS	自由度	MS	F	P-值	臨界值
組間	801.70	2	400.8504	2.9850	0.0520	3.0247
組內	41898.31	312	134.2894			
總和	42700.01	314				

不在拒絕區域，證據不顯著，因此接受  $H_0$ ；換句話說可以假設3條生產線的平均產量皆相等

## 5. 變異數分析範例2

此範例資料可於anova.csv檔案取得

統計假設：

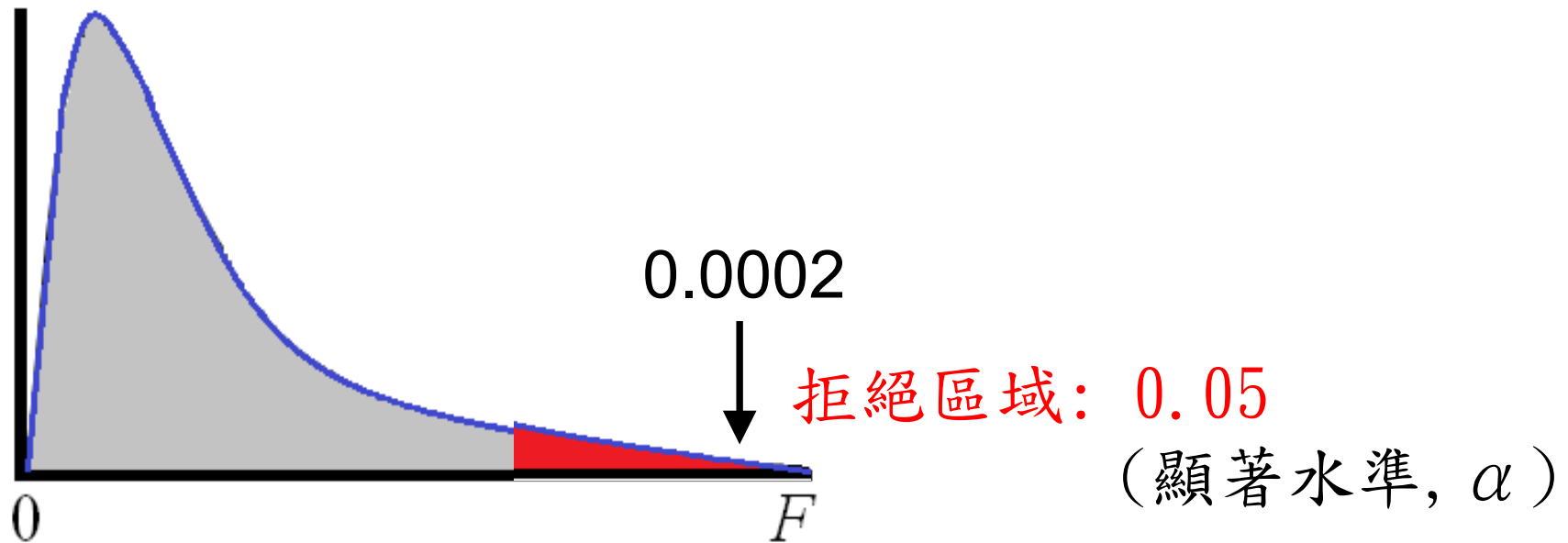
$H_0: \mu_1 = \mu_2 = \mu_3$  (3條生產線的平均產量皆相等)

$H_1$ ：至少有一生產線的平均產量與其他生產線的平均產量不同

摘要				
組	個數	總和	平均	變異數
Method 1	125	7148.224	57.18579	113.3948
Method 2	90	4779.064	53.10071	72.73423
Method 3	100	5189.13	51.8913	98.33051



## 5-1. 變異數分析範例2分析結果



ANOVA						
變源	SS	自由度	MS	F	P-值	臨界值
組間	1750.15	2	875.0774	9.0199	0.0002	3.0247
組內	30269.02	312	97.0161			
總和	32019.17	314				

在拒絕區域內，證據顯著，因此拒絕 $H_0$ ；換句話說不能假設3條生產線的平均產量皆相等

## 6. 資料前處理

```
import pandas as pd
```

```
# 載入anova.csv檔案
```

```
df = pd.read_csv("anova.csv", sep= ",")
```

```
# 將df變形，使dataframe適合statsmodelse套件所需格式
```

```
df_melt = pd.melt(frame=df, value_vars=[ 'Method1',  
    ' Method2', ' Method3' ])
```

```
# 去除空值（當各組資料個數不相等使用）
```

```
df_melt = df_melt[df_melt['value'].notna()]
```

## 7. ANOVA分析

```
# 替 dataframe 添加欄位名稱
```

```
df_melt.columns = [ ' treatments' , ' value' ]
```

```
import statsmodels. api as sm
```

```
from statsmodels. formula. api import ols
```

```
# Ordinary Least Squares (OLS) 模型估計
```

```
model = ols(' value ~ C(treatments)' ,  
            data=df_melt). fit()
```

```
# 產生 anova 表
```

```
anova_table = sm. stats. anova_lm(model, typ=2)
```

## 7. ANOVA分析

# 列印 ANOVA 分析表

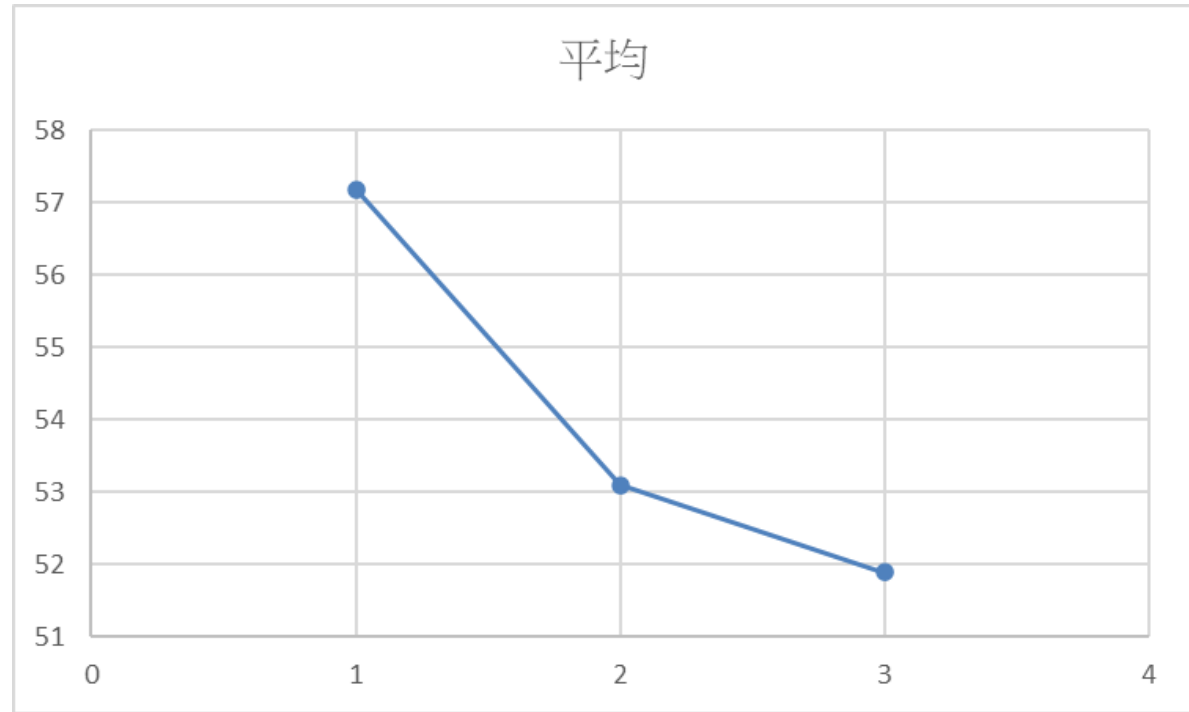
anova\_table

	sum_sq	df	F	PR(>F)
<b>C(treatments)</b>	1750.384551	2.0	9.021581	0.000155
<b>Residual</b>	30267.421168	312.0	NaN	NaN

1. 判斷方法： $p\text{-value} = 0.0002 < 0.05$ ，故拒絕  $H_0$
2. 分析詮釋：我們有足夠證據足以顯示至少有一條生產線的平均產量與其他生產線的平均產量有所不同
3. 結論：生產線的平均產量有顯著差異(但我們不知道哪一條與哪一條不同)

## 8. 分析各組間之差異情形

平均值圖形：



我們可能需要回答以下問題

生產線1的平均產量和生產線2的平均產量不同？

生產線1的平均產量和生產線3的平均產量不同？

生產線2的平均產量和生產線3的平均產量不同？

## 9. ANOVA分析—事後測試

```
!pip install bioinfokit
```

```
from bioinfokit. analys import stat
```

```
# 用Tukey法事後檢定兩兩之間是否有顯著性差異
```

```
res=stat()
```

```
res. tukey_hsd(
```

```
    df=df_melt,
```

```
    res_var=' value' ,
```

```
    xfac_var=' treatments' ,
```

```
    anova_model=' value~C(treatments)'
```

```
)
```

```
# 列印兩兩比較分析表
```

```
res. tukey_summary
```

$\alpha = 0.05$

	group1	group2	Diff	Lower	Upper	q-value	p-value
0	Method1	Method2	4.085573	0.878877	7.292270	4.243406	0.008172
1	Method1	Method3	5.294740	2.182657	8.406823	5.666475	0.001000
2	Method2	Method3	1.209167	-2.161148	4.579481	1.194910	0.660817

## 9. ANOVA分析—事後測試

	group1	group2	Diff	Lower	Upper	q-value	p-value
0	Method1	Method2	4.085573	0.878877	7.292270	4.243406	0.008172
1	Method1	Method3	5.294740	2.182657	8.406823	5.666475	0.001000
2	Method2	Method3	1.209167	-2.161148	4.579481	1.194910	0.660817

在5%的顯著水準下，

拒絕 $H_0$ 的假設：我們有足夠證據足以顯示Method1的平均產量和Method2與Method3的平均產量都不相同

接受 $H_0$ 的假設：證據不顯著，Method2的平均產量和Method3的平均產量相同

單因子變異數分析是一種統計假設檢定的分法用來檢定：(1)兩組資料的變異數是否相同 (2)兩組資料的平均值是否相同 (3)至少三組資料以上的變異數是否相同 (4)至少三組資料以上的平均值是否相同

## (四) 複迴歸分析 (Multiple Regression Analysis)

該ppt的內容來源來自以下教科書

Neil A. Weiss. Introductory Statistics 10th ed., Pearson, Addison Wesley, 2017.

俞洪亮、蔡義清、莊懿妃，2018，商管研究資料分析：SPSS的應用，修訂三版，台北：華泰文化

邱皓政，量化研究法（二）統計原理與分析技術，2007，雙葉書廊



# 1. 迴歸原理

迴歸分析是運用一個或多個變項來預測另一個變項的統計技術

- 被預測的變項稱為依變項或應變項( $Y$ )
- 預測變項可稱為自變項或獨立變項( $X$ )

迴歸分析的原理是找出最適切的數學方程式來表示自變項和依變項之間的關係( $Y=f(X)$ )，所找出的方程式 $f$  稱為迴歸方程式

- 線性迴歸 (linear regression)：假定自變項和依變項間的函數關係為線性
- 非線性迴歸 (nonlinear regression)：假定自變項和依變項間的函數關係為非線性

## 2. 簡單迴歸與多元迴歸

### Simple and Multiple regression

#### (1) 基本定義

A. 各變項均為連續性變項，或是可虛擬為連續性變項者

B. 簡單迴歸 (simple regression)：只根據一個自變項來預測依變項的迴歸分析(可用單一自變項去解釋依變項)

C. 多元迴歸或複迴歸 (multiple regression)：以多個自變項來預測依變項的迴歸分析(需同時用多個自變項才能準確解釋依變項)

#### (2) 線性迴歸方程式的型式

A. 簡單迴歸： $\hat{Y} = b_1x_1 + b_0$

B. 多元迴歸： $\hat{Y} = b_1x_1 + b_2x_2 + b_3x_3 + \dots + b_nx_n + b_0$

### 3. 線性迴歸原理

X	Y
15	15
19	19
20	20
21	21

#### (1) 線性迴歸原理

A. 將連續變項的線性關係以一最具代表性的直線來表示，此線性方程式表示為  $\hat{Y} = b_0 + b_1X$ ， $b_1$  為斜率， $b_0$  為截距

#### (2) 最小平方法與迴歸方程式

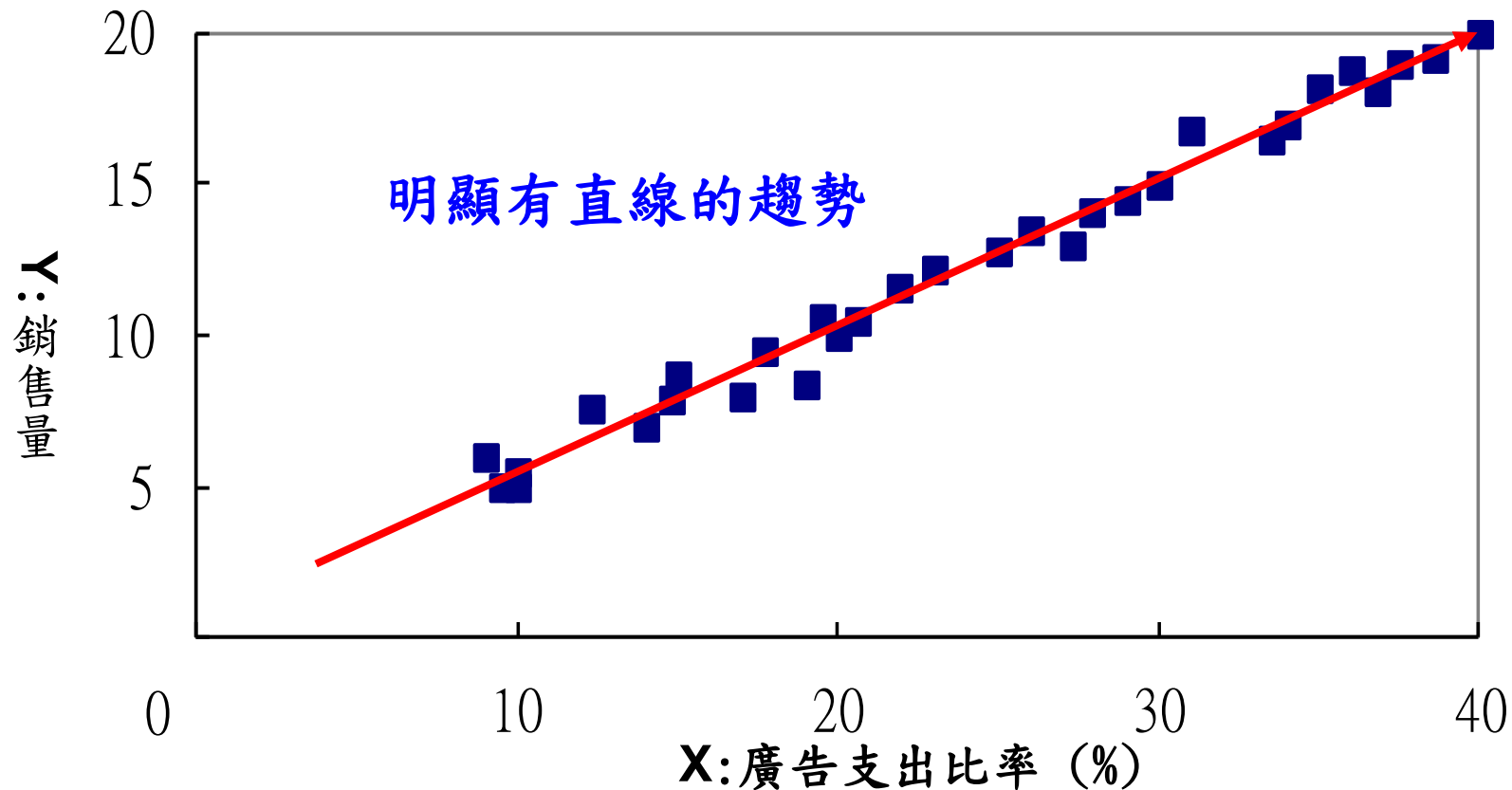
A. 配對觀察值 (X, Y)，將 X 值代入方程式，得到的數值為 Y 變項的預測值，以  $\hat{Y}$  表示

B. 差值  $\varepsilon$ ：Y 的實際值與預測值之差 ( $Y - \hat{Y} = \varepsilon$ ) 稱為殘差 (residual)，表示利用迴歸方程式無法準確預測的誤差

C. 最小平方法：求取殘差(誤差)的平方和最小化的一種估計迴歸線的方法， $\text{Min} \sum (Y - \hat{Y})^2$

## 4. 迴歸分析的基本概念

在許多研究問題中，變數與變數之間有時會呈現**線性相關**。  
例如**廣告支出金額**與**銷售量**之間的關係，如圖：



廣告支出金額(X)與銷售量(Y)的散佈圖

當X增加時，Y也會跟著增加，即是代表X與Y之間有很高的正相關，通常我們用相關係數  $r$  來表示兩個變數間之線性相依程度。

## 5. 迴歸分析的基本統計概念

(1)一般來說，我們利用迴歸分析是想瞭解：

A.能否找出一個線性方程式，用來說明一組自變數( $X_i$ )與依變數( $Y$ )的關係。

B.瞭解這個方程式的預測能力如何？

- 即自變數與依變數的關係強度有多大。

C.探討整體關係是否達到顯著水準？

D.是否只採用某些自變數即具有足夠的預測力。

(2)線性關係 (linear relationship)

A.指兩個變項的關係呈現直線般的共同變化

B.數據的分佈可以被一條最具代表性的直線來表達的關聯情形。

## 6. 相關分析的基本概念

當X增加時，Y也會跟著增加，即是代表X與Y之間有很高的相關(正相關)，通常我們用皮爾森(Pearson)相關係數 $r$ 來表示兩個變數間之線性相依程度，計算公式如下：

X	Y
15	15
19	19
20	20
21	21
22	22
23	23
24	24
25	25

$$r = \frac{\sum (X - \bar{X})(Y - \bar{Y})}{\sqrt{\sum (X - \bar{X})^2 (Y - \bar{Y})^2}}$$

$\bar{X}$  是X的平均值， $\bar{Y}$  是Y的平均值

相關係數為一標準化之數字，其值不受變項特性的影響，其數值是介於-1至+1之間。

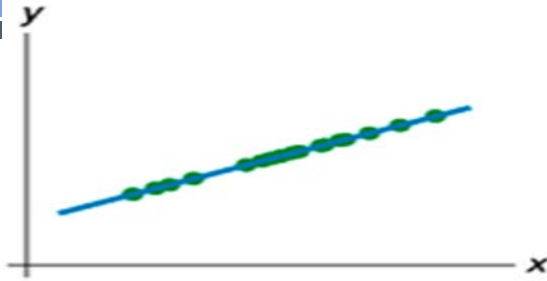
(1)  $r \rightarrow +1$ : 表示這個關係趨向正相關的關係

(2)  $r \rightarrow -1$ : 表示這個關係趨向負相關的關係

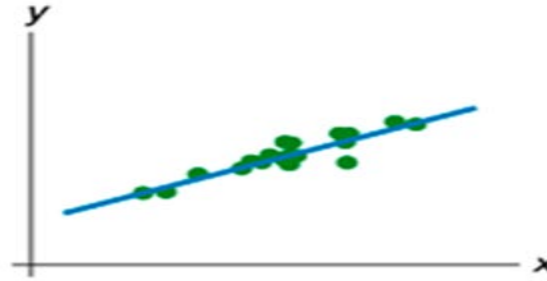
各種關聯程度

相關係數是用於衡量兩個變數X和y之間的線性相依程度，被定義為兩個變數之間的共變異數和標準差的商，其值介於-1與1之間

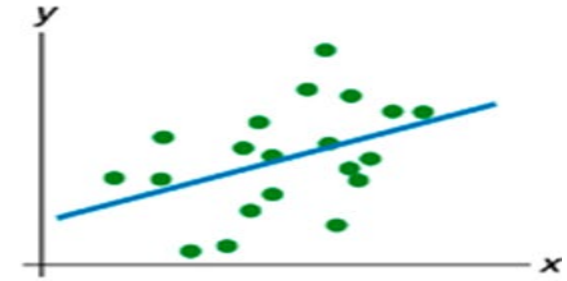
## 7. 不同的線性相關情形圖示



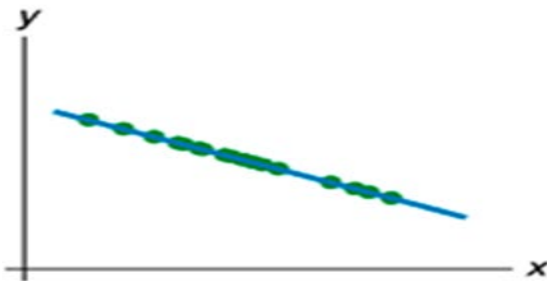
(a) 完全正相關  $r = 1$



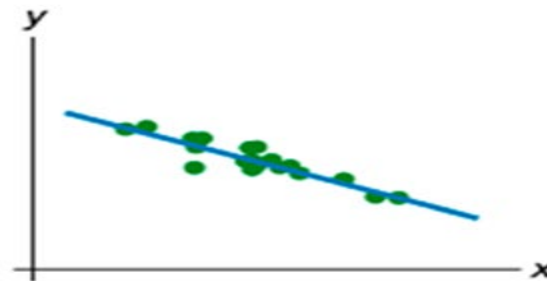
(b) 強的正相關  $r = 0.9$



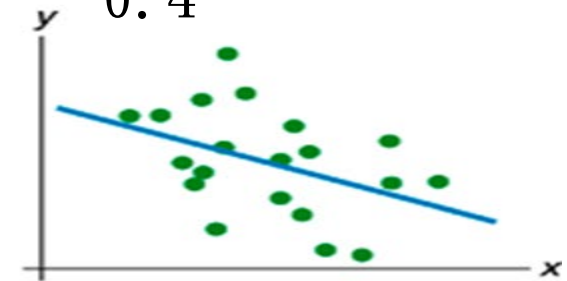
(c) 弱的正相關  $r = 0.4$



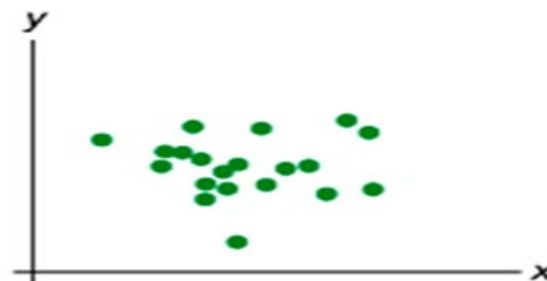
(d) 完全負相關  $r = -1$



(e) 強的負相關  $r = -0.9$



弱的負相關  $r = -0.4$



(g) 零線性相關  $r = 0$

XY的關係可用  
XY散佈圖來觀察

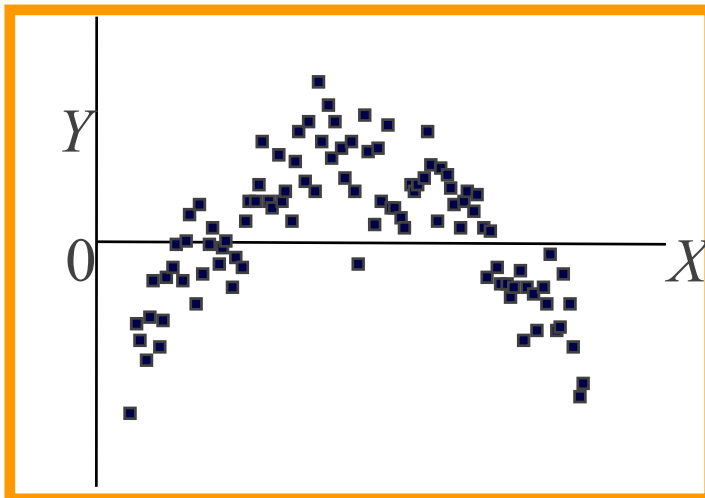
X	Y
15	15
19	19
20	20
21	21
22	22
23	23
24	24
25	25



## 8. 相關係數的強度大小與意義

相關係數範圍(絕對值)	變項關聯程度
1.00	完全相關
.70 至 .99	高度相關
.40 至 .69	中度相關
.10 至 .39	低度相關
.10 以下	微弱或無相關

商管研究資料分析：SPSS的應用

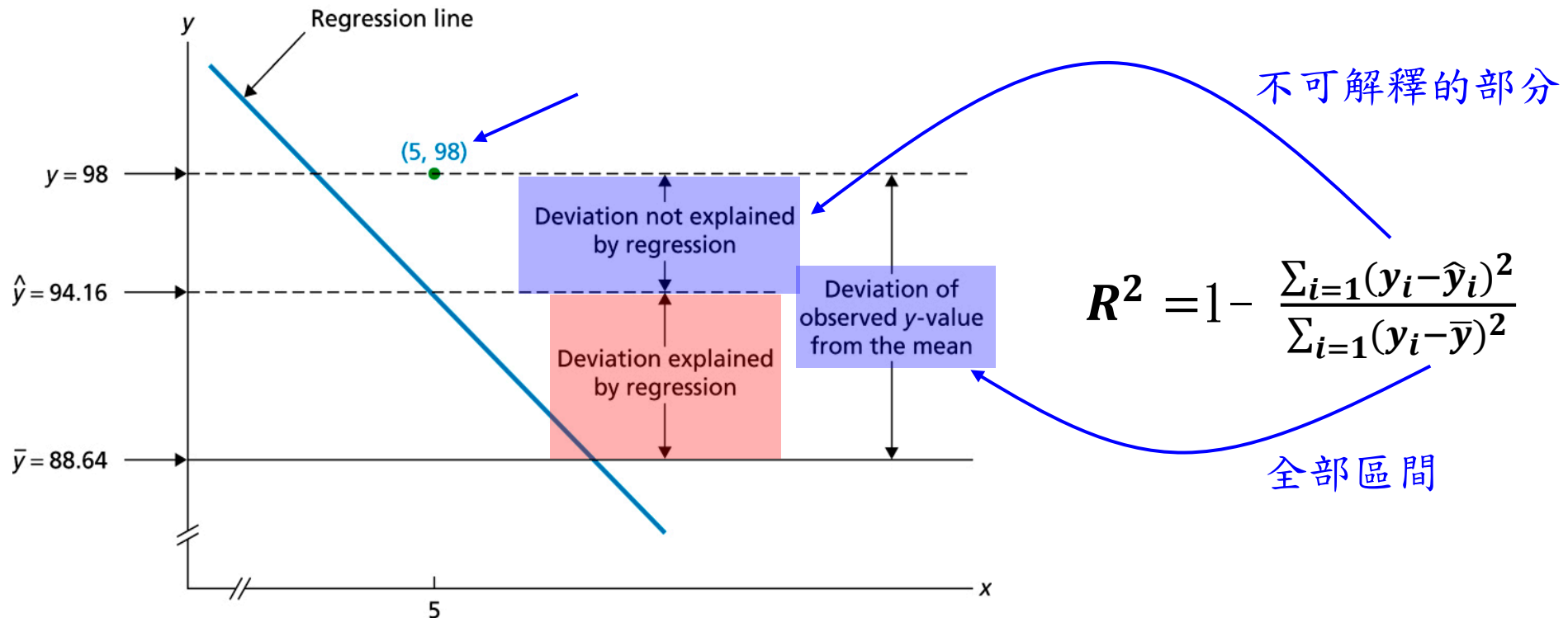


如果X和Y具有非線性關係，則相關係數可能表示X和Y是沒有關係。



## 9. 迴歸模式之判定係數

- (1) 根據現有的資料建立一個迴歸模式時，必須檢定此模式與資料的**符合程度**。檢定適合度最常用的量數是 $R^2$ 或  $r^2$  (R-square)，或稱**判定係數**(coefficient of determination)。
- (2) 表示Y被X解釋的百分比，是一種機率的概念。
- (3) **判定係數** ( $R^2$ ) 是依變數 (Y) 的變量之總變化的比例 (%)，可由自變數 (X) 的之變化解釋。它的範圍從0%到100%。



## 10. 迴歸模型範例

1. 假如在某製造過程中，有一些因素可能會影響製程產量，如

- (1) 某原料的使用量
- (2) 可調整的程序參數
- (3) 某處理過程的加工時間長短

2. 對於製造商而言，找出哪些因素會影響製程產量以及它們如何影響生產可能至關重要。下列是製程產量與影響因素：

- (1) 產量 (Outcome,  $Y$ )
- (2) 原料A的使用量 (IngredientA,  $X_1$ )
- (3) 程序參數1 (Criterion1,  $X_2$ )
- (4) 程序參數2 (Criterion2,  $X_3$ )
- (5) 程序參數3 (Criterion3,  $X_4$ )
- (6) 加工時間 (Time,  $X_5$ )

Outcome	IngredientA	Criterion1	Criterion2	Criterion3	Time
730	409.5	522	20	134.1	16.7
549	443.3	347	22	81.9	21.2
500	387.5	325	25	70.4	19
727	366.4	1158	26	119.1	21.6
831	532	888	15	131.5	24.6
722	360.8	388	19	106.9	20.7
881	651.6	540	11	126.4	21.1
784	591.6	414	17	105.8	20
739	558.6	786	23	100.4	21.8
596	484.4	608	18	83.6	22.7
963	521.7	691	16	95.6	23.8
429	393.2	223	20	70.2	16.2
447	266.7	242	22	71.9	17.3
612	536.2	523	17	105	25.1

試使用收集的資料進行迴歸分析？

## 11. 載入Python的套件與範例的資料

1. 先載入所用到的Python的程式套件
2. 使用pandas套件的`read_csv()`函數來載入範例資料進Python，並以 `Dataframe` 來作處理
3. 用`head()`函數呈現前5列資料及屬性名稱

```
import statsmodels.api as sm
import pandas as pd
import matplotlib.pyplot as plt

df=pd.read_csv('C:/Temp/Data4MultipleRegression.csv')
df.head()
```

→ 匯入統計套件，並簡稱sm  
→ 匯入pandas套件，並簡稱pd  
→ 匯入畫圖套件，並簡稱plt  
  
→ 載入範例資料檔，並簡稱df  
→ 呈現前5列資料及屬性名稱

Out[39]:

	Outcome	IngredientA	Criterion1	Criterion2	Criterion3	Time
0	730	409.5	522	20	134.1	16.7
1	549	443.3	347	22	81.9	21.2
2	500	387.5	325	25	70.4	19.0
3	727	366.4	1158	26	119.1	21.6
4	831	532.0	888	15	131.5	24.6

## 12. 建立範例的多元迴歸分析模型

1. 設定依變項
2. 設定自變項
3. 加入常數項
4. 建立多元迴歸模型

### Python 程式碼

```
Y = df[ "Outcome" ] # Assign output variable

# Multiple regression
X = df[ ["IngredientA", "Criterion1", "Criterion2", "Criterion3", "Time" ] ]
X = sm.add_constant(X) # let's add an intercept (beta_0) to our model
model = sm.OLS(Y, X).fit() # sm.OLS(output, input), OLS : Ordinary Least Squares
```

## 13. 顯示範例中迴歸模型的判定係數

列印模型結果

```
# Print out model statistics  
model.summary()
```

```
OLS Regression Results  
=====
```

Dep. Variable:	Outcome	R-squared:	0.645
Model:	OLS	Adj. R-squared:	0.604
Method:	Least Squares	F-statistic:	15.96
Date:	Fri, 26 Mar 2021	Prob (F-statistic):	5.90e-09
Time:	17:26:44	Log-Likelihood:	-296.28
No. Observations:	50	AIC:	604.6
Df Residuals:	44	BIC:	616.0
Df Model:	5		
Covariance Type:	nonrobust		

```
=====
```

## 14. 範例的迴歸分析模型

$$\hat{Y} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_4 + \beta_5 X_5$$

	coef	std err	t	P> t	[0.025	0.975]
const	121.1668	144.439	0.839	0.406	-169.932	412.265
IngredientA	0.4217	0.187	2.255	0.029	0.045	0.799
Criterion1	0.2254	0.098	2.292	0.027	0.027	0.424
Criterion2	-1.7188	3.886	-0.442	0.660	-9.550	6.113
Criterion3	1.0679	0.957	1.116	0.270	-0.860	2.996
Time	7.4027	7.382	1.003	0.321	-7.474	22.279

產量 = 121.1668 + 0.4217(原料A的使用量) + 0.2254(程序參數1)  
 - 1.7188(程序參數2) + 1.0679(程序參數3)  
 + 7.4027(加工時間) + 殘差 = 預測值 + 殘差(預測誤差)

下列因素對 產量 (Y) 有顯著影響 (P-value < 0.05) :

1. 原料A的使用量 ( $X_1$ )
2. 程序參數1 ( $X_2$ )

迴歸分析是用一個線性方程式，來說明一組自變數與依變數的關係

## 15. 迴歸分析模型的簡化 (進行屬性挑選，移除自變項 $X_3, X_4, X_5$ )

屬性挑選：移除下列對產量 ( $Y$ ) 沒有顯著影響 ( $P\text{-value} \geq 0.05$ ) 的因素，再進行一次迴歸分析

1. 程序參數2 ( $X_3$ )
2. 程序參數3 ( $X_4$ )
3. 加工時間 ( $X_5$ )

### Python 程式碼

```
X = df[["IngredientA", "Criterion1"]]  
X = sm.add_constant(X) ## let's add an intercept (beta_0) to our model  
model = sm.OLS(Y, X).fit() # sm.OLS(output, input), OLS stands for Ordinary Least  
Squares  
model.summary()
```

- (1) 設定 IngredientA, Criterion1 為自變項
- (2) 加入常數項
- (3) 建立多元迴歸模型

# 15. 迴歸分析模型的簡化 (進行屬性挑選，移除自變項 $X_3, X_4, X_5$ )

列印模型結果

```
# Print out model statistics
model.summary()
```

簡化前之判定係數

OLS Regression Results

=====			
Dep. Variable:	Outcome	R-squared:	0.622
Model:	OLS	Adj. R-squared:	0.606
Method:	Least Squares	F-statistic:	38.65
Date:	Wed, 07 Apr 2021	Prob (F-statistic):	1.19e-10
Time:	12:53:13	Log-Likelihood:	-297.83
No. Observations:	50	AIC:	601.7
Df Residuals:	47	BIC:	607.4
Df Model:	2		
Covariance Type:	nonrobust		
=====			



## 16. 迴歸分析模型的簡化 (刪除 $X_3, X_4, X_5$ )

	coef	std err	t	P> t	[0.025	0.975]
const	210.7795	51.084	4.126	0.000	108.011	313.548
IngredientA	0.6255	0.115	5.435	0.000	0.394	0.857
Criterion1	0.3097	0.069	4.500	0.000	0.171	0.448

產量

$$= 210.7795 + 0.6255(\text{原料A的使用量}) + 0.3097(\text{程序參數1})$$

+ 殘差

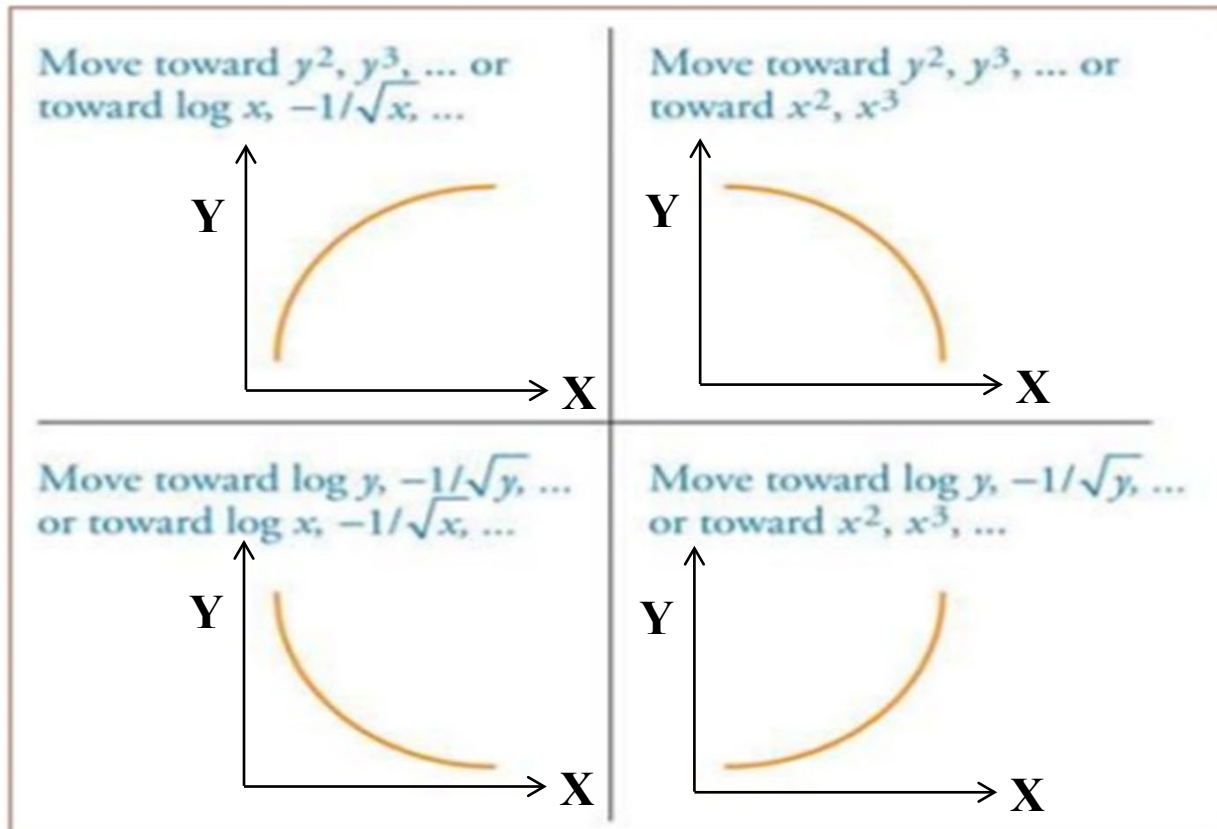
$$= \text{預測值} + \text{殘差(預測誤差)}$$

# 17. 曲線(非線性)迴歸：迴歸分析模型的精進

## Curvilinear Regression

使用XY的散佈圖來估計曲線模型

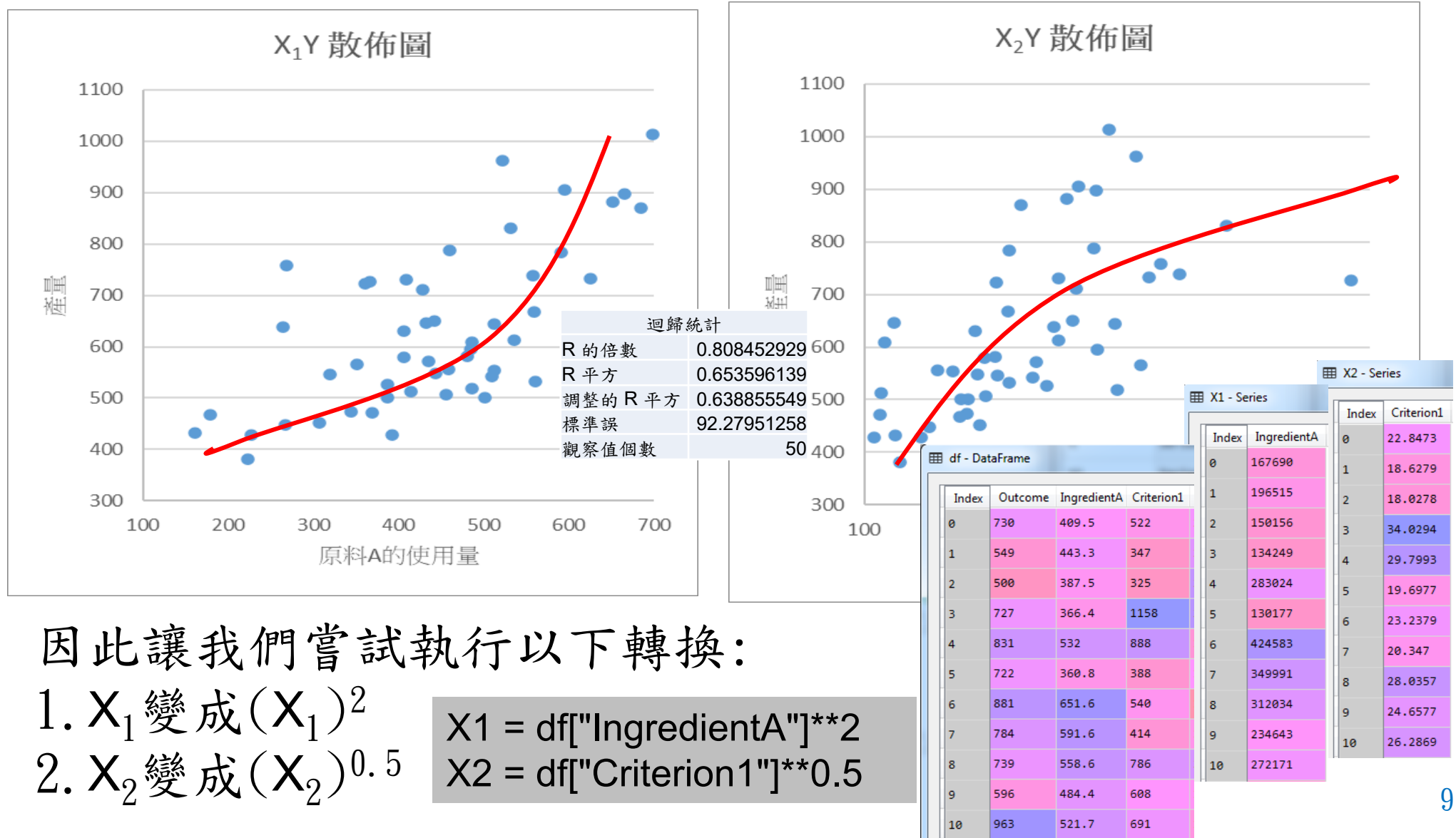
1. 尋找獨立變項與依變項關係的最佳函數關係
2. 不同的函數關係：線性、對數、倒數、二次、三次、冪次、複合、S曲線、Logistic、成長，以及指數函數



**Turkey's Four  
Quadrants Approach**

## 18. 範例轉換之後的 XY 散佈圖

產量(依變項 $Y$ )和原料A的使用量(自變項 $X_1$ )或程序參數1(自變項 $X_2$ )之間的 XY 散佈圖



因此讓我們嘗試執行以下轉換：

1.  $X_1$  變成  $(X_1)^2$

2.  $X_2$  變成  $(X_2)^{0.5}$

```
X1 = df["IngredientA"]**2
X2 = df["Criterion1"]**0.5
```

# 19. 迴歸分析模型的精簡(簡化與轉換之後)

## Python 程式碼

```
X1 = df["IngredientA"]**2
X2 = df["Criterion1"]**0.5
X = pd.concat([X1, X2], axis=1)
X = sm.add_constant(X)
model = sm.OLS(Y, X).fit()
model.summary()
```

→ 橫向串接X1和X2

X1 - Series		X2 - Series		X - DataFrame			
Index	IngredientA	Index	Criterion1	Index	const	IngredientA	Criterion1
0	167690	0	22.8473	0	1	167690	22.8473
1	196515	1	18.6279	1	1	196515	18.6279
2	150156	2	18.0278	2	1	150156	18.0278
3	134249	3	34.0294	3	1	134249	34.0294
4	283024	4	29.7993	4	1	283024	29.7993
5	130177	5	19.6977	5	1	130177	19.6977
6	424583	6	23.2379	6	1	424583	23.2379
7	349991	7	20.347	7	1	349991	20.347
8	312034	8	28.0357	8	1	312034	28.0357
9	234643	9	24.6577	9	1	234643	24.6577
10	272171	10	26.2869	10	1	272171	26.2869

精簡前之係數

簡化前之係數

## OLS Regression Results

```
=====
Dep. Variable:          Outcome    R-squared:          0.654
Model:                  OLS        Adj. R-squared:      0.639
Method:                 Least Squares    F-statistic:        44.34
Date:                  Wed, 07 Apr 2021    Prob (F-statistic):  1.51e-11
Time:                  14:51:17    Log-Likelihood:     -295.64
No. Observations:      50    AIC:                597.3
Df Residuals:          47    BIC:                603.0
Df Model:               2
Covariance Type:       nonrobust
=====
```

## 20. 精簡後的迴歸分析模型

注意：自變項 $X_1$ 的迴歸係數過小，容易有進位誤差，可能造成預測值誤差變大

	coef	std err	t	P> t	[0.025	0.975]
const	207.5454	55.051	3.770	0.000	96.797	318.294
IngredientA	0.0007	0.000	5.817	0.000	0.000	0.001
Criterion1	12.8053	2.802	4.571	0.000	7.169	18.441
Omnibus:		0.751	Durbin-Watson:			1.934
Prob(Omnibus):		0.687	Jarque-Bera (JB):			0.854
Skew:		0.219	Prob(JB):			0.653
Kurtosis:		2.533	Cond. No.			1.01e+06

產量

$$= 207.5454 + 0.0007(\text{原料A的使用量})^2 + 12.8053(\text{程序參數1})^{0.5}$$

+ 殘差

$$= \text{預測值} + \text{殘差}$$

## 21. 精簡迴歸分析模型：縮小自變項 $X_1$ 的數值

### Python 程式碼

```
X1 = df["IngredientA"]**2 / 1000
X2 = df["Criterion1"]**0.5
X = pd.concat([X1, X2], axis=1)
X = sm.add_constant(X)
model = sm.OLS(Y, X).fit()
model.summary()
```

→ 縮小自變項  $X_1$  1000 倍，以放大其迴歸係數，避免因迴歸係數過小而產生進位誤差，造成預測值誤差變大

舊迴歸係數

	coef	std err	t	P> t	[0.025	0.975]
const	207.5454	55.051	3.770	0.000	96.797	318.294
IngredientA	0.7340	0.126	5.817	0.000	0.480	0.988
Criterion1	12.8053	2.802	4.571	0.000	7.169	18.441

產量

$$= 207.5454 + 0.7340(\text{原料A的使用量})^2 / 1000 + 12.8053(\text{程序參數1})^{0.5} + \text{殘差}$$

# (五)主成分分析 (Principal Component Analysis)

該ppt的內容來源來自以下教科書

Multivariate Data Analysis – A global perspective, 7/e, Prentice Hall International, J.F. Hair, W.C. Black, B.J. Babin and R.E. Anderson.

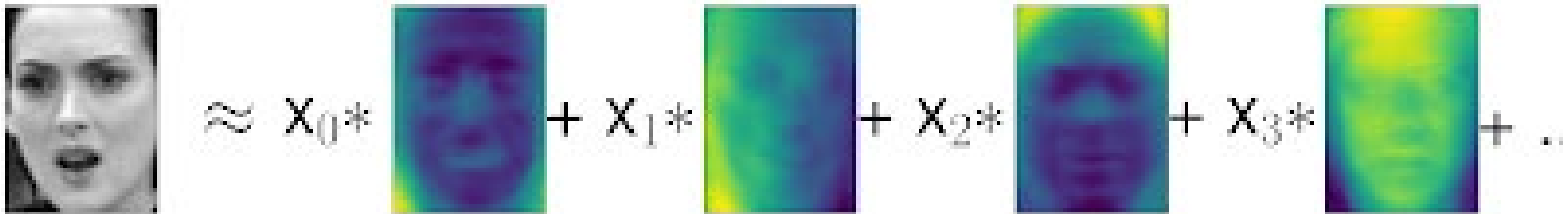
# 1.PCA 的動機

1. 一般來說，特徵數愈多，我們愈容易得到好的預測效果。  
。然而，當特徵過多時，也容易對運算造成龐大的負荷。
2. 要來聊聊如何在盡量保留特徵貢獻度下，透過降低特徵的維度以減少運算成本。
3. 降維算法有很多，比如PCA（Principal Component Analysis）、ICA、SOM、MDS、ISOMAP、LLE等。
4. PCA是一種無監督降維算法，它是最常用的降維算法之一，可以很好的解決因變量太多而複雜性，計算量增大的弊端。



## 2. 簡介PCA (1/2)

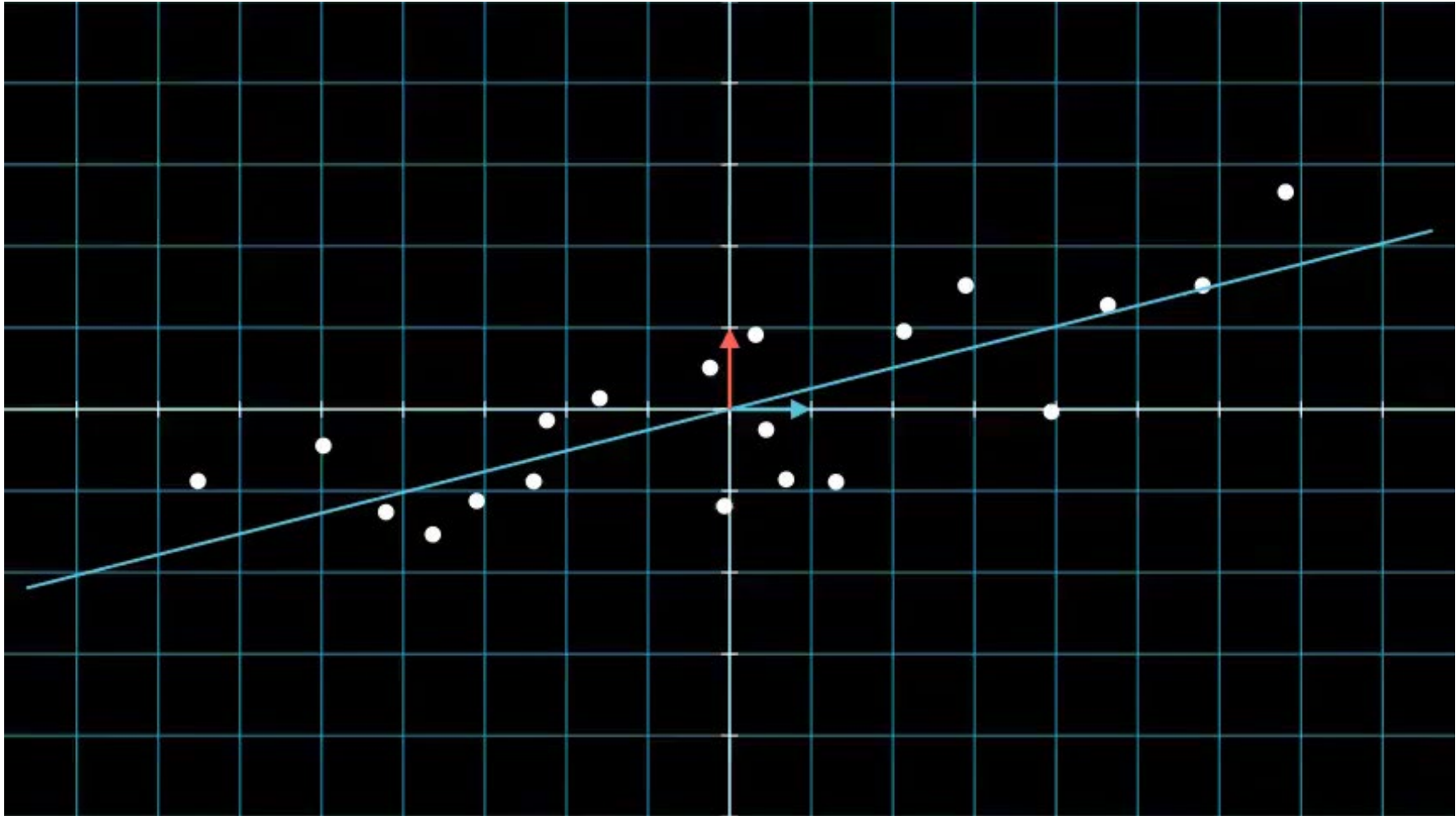
1. Principal components analysis (簡稱 PCA) 是一種簡化數據集的技術。
2. 依需求刪去重要性低的特徵，利用重要性高的特徵去表達整組資料。
3. 在降低數據維度的過程中我們希望能盡量保存貢獻最大的成份，經由這些貢獻最大的成份疊加我們也可以得到具有原數據特徵的數據。
4. PCA 視覺化高維度資料：



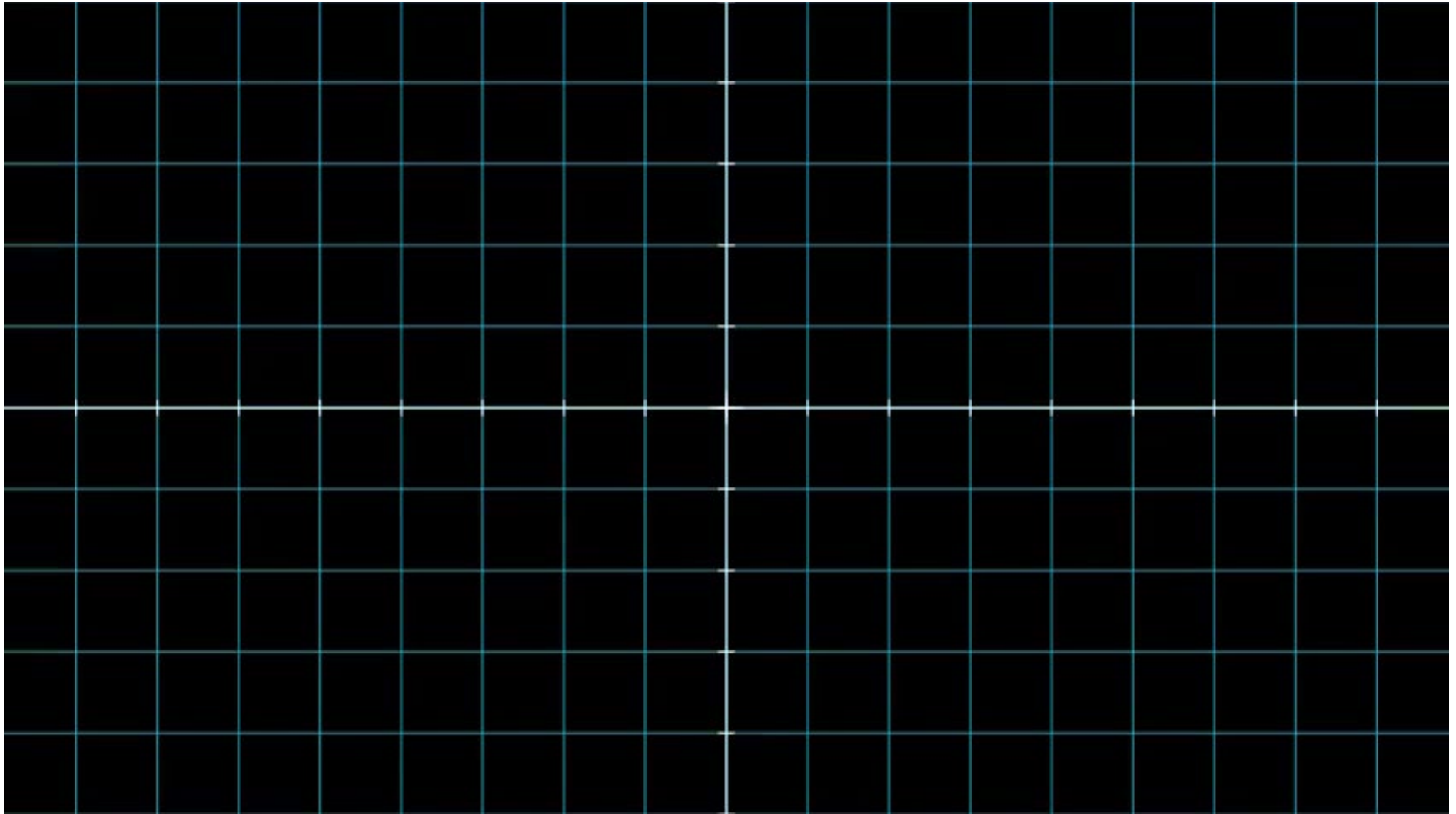
## 2. 簡介PCA (2/2)

1. 假設我們手上有 100 張圖好了，要先找出大家共有的特徵(綠色輪廓)，用最少的資料去紀錄這 100 張圖，將這些特徵的重要性排序，這些特徵就是指 **Principal components**。
2. 然後依需求刪去重要性低的特徵，利用重要性高的特徵去表達整組資料。
3. 所以 **Principal components** 它會隨著你整組 data 的不同而有所改變，同張圖在不同資料組中會被以不同的方式表示。

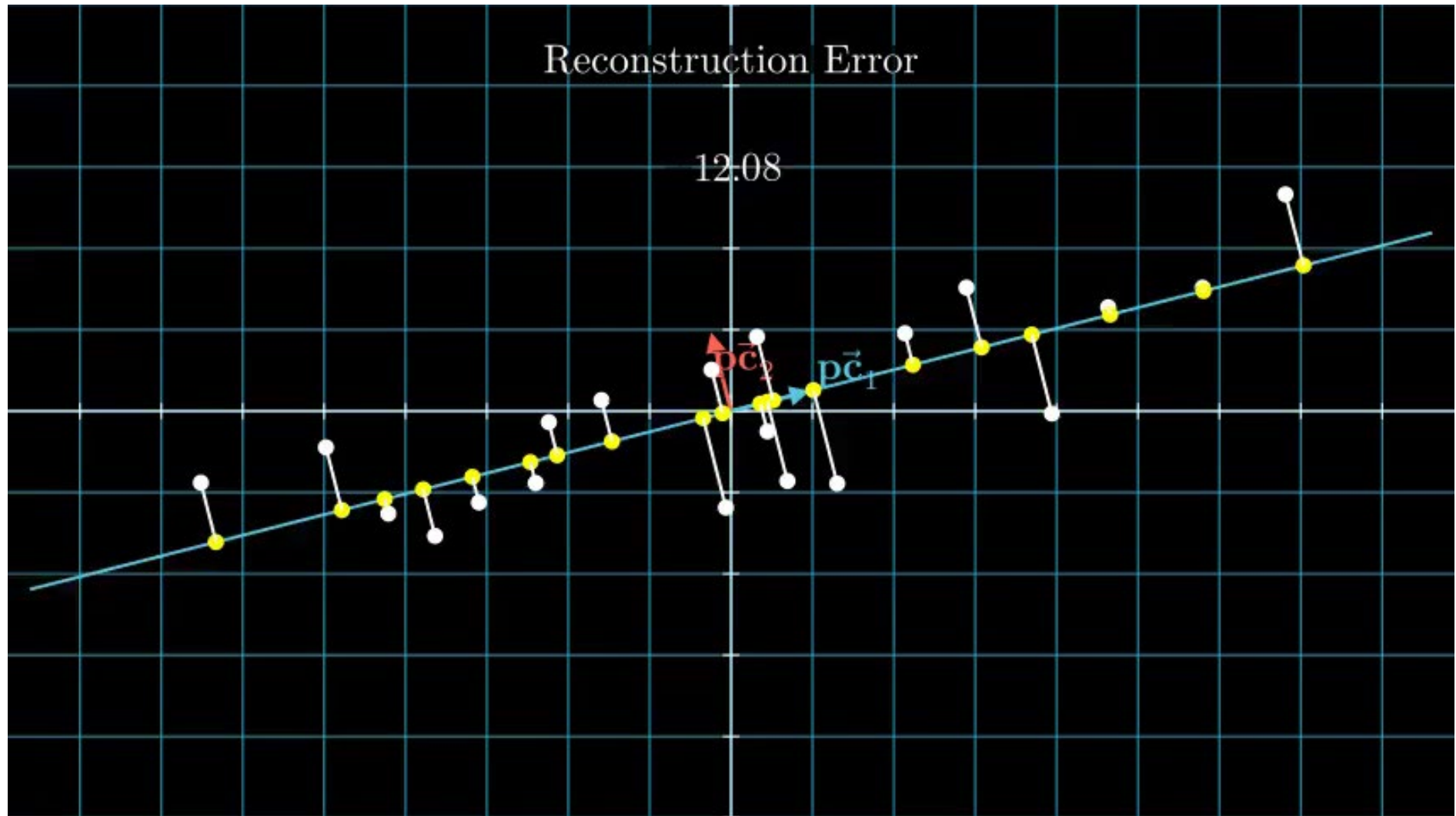
### 3. PCA 影片講解 (1/3)



### 3. PCA 影片講解 (2/3)



### 3. PCA 影片講解 (3/3)



## 4. 資料集說明

- 採用大家所熟悉的鳶尾花資料集(IRIS)，IRIS數據集是常見的分類試驗數據集，也成為鳶尾花數據集，是一類多重變量分析的數據集。
- 數據集包含150個數據集，分為三類，每類50個數據，每個數據包含4個特徵。
- 可以通過花萼長度，花萼寬度，花瓣長度，花瓣寬度(sepal length, sepal width, petal length, petal width)等4個特徵預測鳶尾花卉屬於(Setosa, Versicolour, Virginica)三個種類中的哪一類。

## 5. Python in PCA

### (1) 引入模組

```
from sklearn.decomposition import PCA  
from sklearn import datasets  
import pandas as pd  
import numpy as np
```

### (2) 載入資料

Online:

```
iris = datasets.load_iris()  
X = iris.data
```

Offline:

```
# reading from the file  
df = pd.read_csv("iris.csv", header=0)  
#df = pd.read_csv("iris.csv")  
array = df.values  
X = array[:,0:4]  
print(X)
```

## 6. PCA對特徵降維

(1) 多維降成2維(舉X資料集，是鳶尾花花瓣及花萼的長度，n\_components是dataset最終的維度)

```
pca=PCA(n_components=2)  
pca=PCA(n_components='mle')
```

也可以給定'mle'讓演算法用最大概似法幫我們決定合適的components數量

(2) PCA旨在找到讓特徵映射後資料變異量最大的投影向量

```
pca.fit(X).transform(X)  
NewX = pca.fit_transform(X)
```

→NewX就是降維後的數據



## 6. PCA對特徵降維

```
[[5.1 3.5 1.4 0.2]
 [4.9 3.0 1.4 0.2]
 [4.7 3.2 1.3 0.2]
 [4.6 3.1 1.5 0.2]
 [5.0 3.6 1.4 0.2]
 [5.4 3.9 1.7 0.4]
 [4.6 3.4 1.4 0.3]
 [5.0 3.4 1.5 0.2]
 [4.4 2.9 1.4 0.2]
 [4.9 3.1 1.5 0.1]
 [5.4 3.7 1.5 0.2]
 [4.8 3.4 1.6 0.2]
 [4.8 3.0 1.4 0.1]]
```



```
array([[ -2.68412563,  0.31939725],
       [ -2.71414169, -0.17700123],
       [ -2.88899057, -0.14494943],
       [ -2.74534286, -0.31829898],
       [ -2.72871654,  0.32675451],
       [ -2.28085963,  0.74133045],
       [ -2.82053775, -0.08946138],
       [ -2.62614497,  0.16338496],
       [ -2.88638273, -0.57831175],
       [ -2.67275558, -0.11377425],
       [ -2.50694709,  0.6450689 ],
       [ -2.61275523,  0.01472994],
       [ -2.78610927, -0.235112  ]],
```

## 7. 查看PCA降維結果

(1)看特徵數量:(注意，這是經過映射後的新特徵，而非原本特徵的刪除)

```
pca.n_components → 2
```

(2)看新特徵的解釋能力

```
pca.explained_variance_ratio →
```

```
array([0.92461872, 0.05306648])
```

它代表降維後的各主成分的方差值占總方差值的比例，這個比例越大，則越是重要的主成分。

(3)看新特徵的解釋能力

```
np.cumsum(pca.explained_variance_ratio_) →
```

```
array([0.92461872, 0.97768521])
```

把解釋能力累加起來，看看總共保留了多少特徵貢獻度

## 8. 降維後的數據轉換成原始數據

(1) `inverse_transform()`將降維後的數據轉換成原始數據

```
X = pca.inverse_transform(NewX)
```

## 9. PCA降維的優缺點

(1)總括來說，PCA最大的優點就是可以盡可能在訊息損失極小化的情況下，透過降維的方式降低數據量，以降低之後建模的運算成本，並且避免維度災難等問題。

PCA的優點是減少維度、歸納變數、進行探索性的研究，能找出資料中潛在的共同特徵

(2)然而，在映射的過程中，透過特徵的融合，最後要解釋模型的時候會比較困擾，因為特徵已經是轉換過後的，而非原本純粹的特徵了。

PCA是一種統計分析、簡化數據集的方法，它的優點是降低數據的複雜性，識別最重要的多個特徵；缺點是可能損失有用信息。