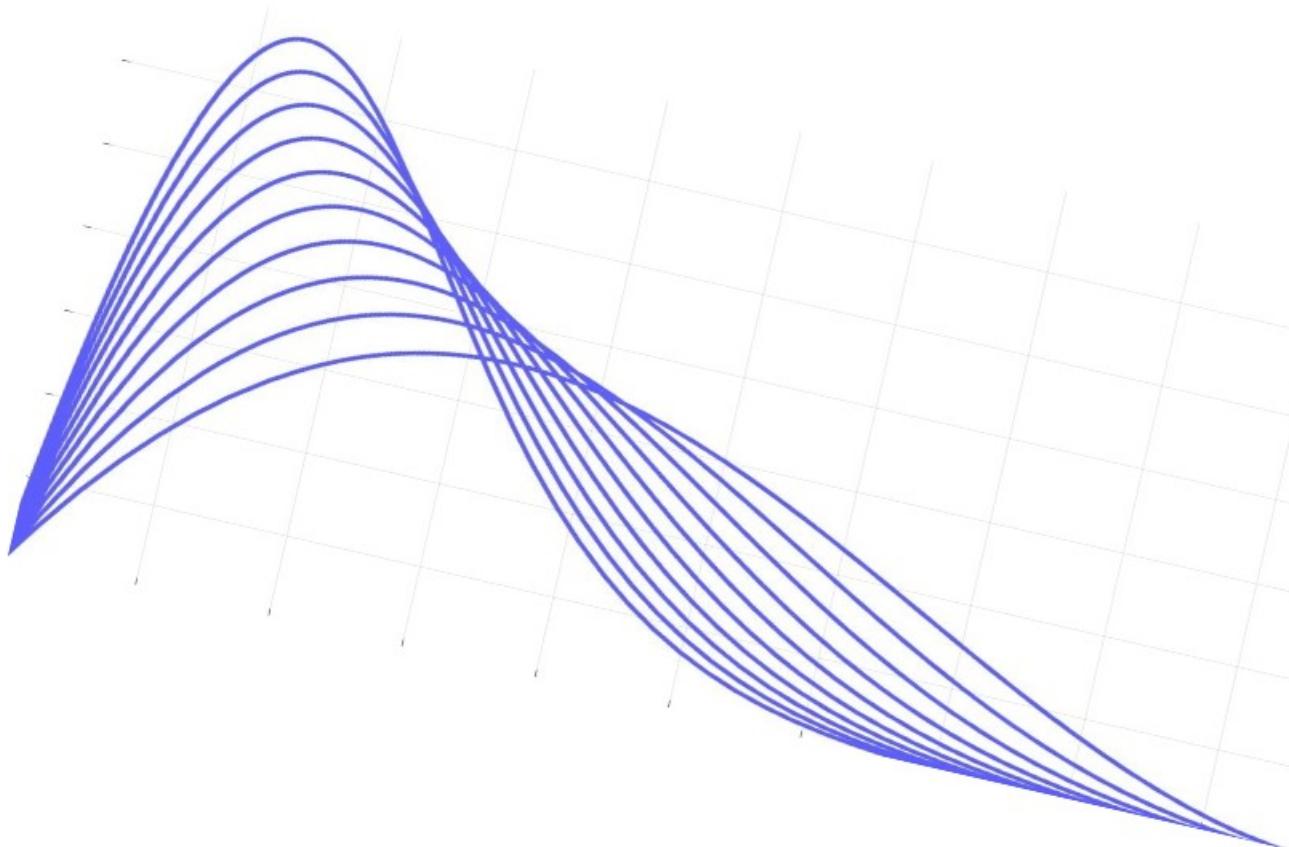


X_ELATE_E 統計與程式

江柏學

710933118

September 21, 2020



LATE_E & MATLAB

2 X_EL^AT_EX 統計與程式

目錄

序	xiii
1 L^AT_EX 操作手冊之統計與數學	1
1.1 集合 Set thoery	2
1.2 函數 Function	5
1.3 統計方法之應用 Application of statistical methods	9
1.3.1 簡單隨機抽樣 Simple random sampling	10
1.3.2 分層隨機抽樣 Stratified random sampling	12
1.3.3 集群抽樣 Cluster sampling	15
1.3.4 系統抽樣 Systematic sampling	17
1.4 結論 Conclusion	20
2 MATLAB 實作函數圖形	21
2.1 基礎函數圖形介紹 - 以 12 種函數為例	22
2.1.1 $y = f(x) = \sin(x) + \cos(x)$	22
2.1.2 $y = f(x) = \frac{1-e^{-2x}}{1+e^{-2x}}$	23
2.1.3 $y = f(x) = \sqrt[3]{\frac{4-x^3}{1+x^2}}$	25
2.1.4 $y = f(x) = \frac{1}{x-1}$	26
2.1.5 $y = f(x) = \frac{1}{2\sqrt{2\pi}} e^{\frac{-(x-1)^2}{8}}$	28
2.1.6 $y = f(x) = \sqrt[3]{x^2}$	30

2.1.7	$y = f(x) = 2x^3 - x^4$	31
2.1.8	$y = f(x) = x\sqrt{4 - x^2}$	34
2.1.9	$y = f(x) = \frac{\ln x}{x^3}$	35
2.1.10	$y = f(x) = 3, 1 \leq x \leq 5$	37
2.1.11	$x^2 + y^2 = 1$	38
2.1.12	正方形	40
2.2	特殊函數圖形呈現	41
2.2.1	$f(y) = \frac{1}{\beta}e^{-\frac{y}{\beta}}, 0 \leq y \leq \infty$	41
2.2.2	$f(y) = \left[\frac{\gamma(\alpha+\beta)}{\gamma(\alpha)\gamma(\beta)} \right] y^{\alpha-1}(1-y)^{\beta-1}, 0 \leq y \leq 1$	42
2.2.3	X^2 分配	43
2.3	其他參數與圖表	44
2.3.1	Marker 使用	44
2.3.2	linestyle 使用	46
2.3.3	Bar 使用	48
2.3.4	Histogram 使用	50
2.3.5	Scatter 使用	51
2.3.6	Pie 使用	52
2.4	結論 Conclusion	54
3	MATLAB 實作之統計分配	55
3.1	分配函數	55
3.1.1	連續型函數	55
3.1.2	離散型函數	75
3.2	亂數產生與相關圖形	82
3.3	抽樣分配	89
3.4	結論	96
4	監督式學習之迴歸分類	97

4.1	迴歸模型	97
4.1.1	簡單線性迴歸	98
4.1.2	加廣型迴歸	103
4.2	模擬資料	105
4.3	結論	115
5	監督式學習之判別式與 KNN	117
5.1	Discriminant	117
5.1.1	LDA(Linear Discriminant Analysis)	119
5.1.2	QDA(Quadratic Discriminant Analysis)	123
5.2	K-Nearest Neighbors(KNN)	126
5.3	模型比較	129
5.4	結論	135
6	監督式學習之類神經網路	137
6.1	類神經網路 (ANN)	137
6.2	ANN 實務應用——機器手臂	140
6.3	ANN 實務應用——分類器	144
6.4	結論	147
7	MATLAB 分類方法總結	149
7.1	資料集建立	149
7.2	模型測試	153
7.3	結論	157

圖目錄

圖 1.1 聯集	3
圖 1.2 交集	3
圖 1.3 差集	4
圖 1.4 乘集	5
圖 1.5 函數	6
圖 1.6 映像	7
圖 1.7 像原	7
圖 1.8 反函數	9
圖 2.1 $y = f(x) = \sin(x) + \cos(x)$	22
圖 2.2 $y = f(x) = \frac{1-e^{-2x}}{1+e^{-2x}}$	24
圖 2.3 $y = f(x) = \sqrt[3]{\frac{4-x^3}{1+x^2}}$	25
圖 2.4 部分放大 $y = f(x) = \sqrt[3]{\frac{4-x^3}{1+x^2}}$	26
圖 2.5 $y = f(x) = \frac{1}{x-1}$	27
圖 2.6 $y = f(x) = \frac{1}{x-1}$	27
圖 2.7 $y = f(x) = \frac{1}{2\sqrt{2\pi}} e^{\frac{-(x-1)^2}{8}}$	29
圖 2.8 $y = f(x) = \sqrt[3]{x^2}$	30
圖 2.9 $y = f(x) = 2x^3 - x^4$	32
圖 2.10 函數 $y = f(x) = 2x^3 - x^4$ 之左側放大	32
圖 2.11 函數 $y = f(x) = 2x^3 - x^4$ 之中間放大	33

圖 2.12 函數 $y = f(x) = 2x^3 - x^4$ 之右側放大	33
圖 2.13 $y = f(x) = x\sqrt{4 - x^2}$	34
圖 2.14 $y = f(x) = x\sqrt{4 - x^2}$	35
圖 2.15 $y = f(x) = \frac{\ln x}{x^3}$	36
圖 2.16 $y = f(x) = \frac{\ln x}{x^3}$	36
圖 2.17 $y = f(x) = \frac{\ln x}{x^3}$	37
圖 2.18 $y = f(x) = 3, 1 \leq x \leq 5$	38
圖 2.19 $x^2 + y^2 = 1$	39
圖 2.20 正方形	40
圖 2.21 $f(y) = \frac{1}{\beta}e^{-\frac{y}{\beta}}, 0 \leq y \leq \infty$	41
圖 2.22 Beta 分配	43
圖 2.23 X^2 分配	44
圖 2.24 Marker 展示”-gs” 和”-o”	45
圖 2.25 Marker 展示”*” 和”h”	46
圖 2.26 Marker 展示”+” 和”<”	46
圖 2.27 linestyle 展示	47
圖 2.28 一般常見的 Bar 展示	48
圖 2.29 多條 Bar 重疊展示	49
圖 2.30 並排 Bar 展示	50
圖 2.31 Histogram 展示	51
圖 2.32 Scatter 展示	52
圖 2.33 Pie 分割展示	53
圖 2.34 一般常見的 Pie 展示	53
圖 2.35 特定缺口之 Pie 展示	54
圖 3.1 常態分配 $\mu = 0, \sigma = 1$	56
圖 3.2 多種常態分配	57

圖 3.3 T 分配 (自由度由 0.1 至 1)	59
圖 3.4 T 分配趨近標準常態分配)	60
圖 3.5 卡方分配，自由度為 5	61
圖 3.6 卡方分配，自由度 (v) 的變化	62
圖 3.7 F 分配，自由度為 (5,6)	64
圖 3.8 F 分配，自由度 (1)> 自由度 (2)	65
圖 3.9 F 分配，自由度 (1)= 自由度 (2)	66
圖 3.10 F 分配，自由度 (1)< 自由度 (2)	66
圖 3.11 F 分配	67
圖 3.12 β 分配 ($\alpha > \beta$)	69
圖 3.13 β 分配 ($\alpha = \beta$)	69
圖 3.14 β 分配 ($\alpha < \beta$)	70
圖 3.15 β 分配	71
圖 3.16 Gamma 分配 ($\alpha > \beta$)	72
圖 3.17 Gamma 分配 ($\alpha = \beta$)	73
圖 3.18 Gamma 分配 ($\alpha < \beta$)	74
圖 3.19 Gamma 分配	74
圖 3.20 二項分配之直方圖	76
圖 3.21 二項分配之莖葉圖	77
圖 3.22 二項分配之階梯圖	78
圖 3.23 Poisson 分配之莖葉圖	80
圖 3.24 Poisson 分配之階梯圖	80
圖 3.25 Poisson 分配之多種形態	81
圖 3.26 樣本數大小之差異	83
圖 3.27 樣本數大小與 qqplot 比較	84
圖 3.28 常態分配	85
圖 3.29 卡方分配，bins 大小差異	86

圖 3.30 卡方分配與理論值之配適	86
圖 3.31 卡方分配_ECDF 圖	88
圖 3.32 卡方分配_qqplot	89
圖 3.33 CLT 樣本大小差異	90
圖 3.34 抽樣分配趨近常態分配	91
圖 3.35 Z^2 之轉換直方圖	92
圖 3.36 Z^2 之轉換 ECDF 圖	93
圖 3.37 卡方分配可加性	94
圖 3.38 卡方分配可加性與理論線之配適	95
圖 4.1 範例資料集	98
圖 4.2 REG 範例資料大致分佈	99
圖 4.3 Design Matrix	100
圖 4.4 分類器	101
圖 4.5 錯誤資料	102
圖 4.6 Augmented Model Design Matrix(前十筆)	104
圖 4.7 Augmented Model	105
圖 4.8 模擬資料之散佈圖	106
圖 4.9 模擬不同資料集	107
圖 4.10 模擬資料大致分布	108
圖 4.11 模擬資料之簡單迴歸分類器	109
圖 4.12 模擬資料之加廣型迴歸分類器	110
圖 4.13 兩分類器表現差異	110
圖 4.14 圖 4.9 (d) 資料散佈情形	111
圖 4.15 圖 4.9 (d) 資料集之簡單迴歸分類器	111
圖 4.16 圖 4.9 (d) 資料集之加廣型迴歸分類器	113
圖 4.17 三群簡單迴歸分類器	113

圖 4.18 三群加廣型分類器	115
圖 5.1 資料大致分布	119
圖 5.2 平均值坐落位置	121
圖 5.3 新資料 (0,3) 位置	122
圖 5.4 LDA 分類線	123
圖 5.5 QDA Model	125
圖 5.6 QDA Model Predict	125
圖 5.7 KNN Predict	126
圖 5.8 KNN 分類線	127
圖 5.9 KNN Meshgrid	128
圖 5.10 KNN 立體圖	129
圖 5.11 實驗資料集	130
圖 5.12 四種分類模型	132
圖 6.1 類神經網路之概念圖	138
圖 6.2 權重調整輸入值	139
圖 6.3 機器手臂運動範圍 (灰色區域)	141
圖 6.4 機器手臂之訓練資料集 (N=100)	141
圖 6.5 機器手臂之訓練資料集 (N=400)	142
圖 6.6 目標變數之真實值資料	143
圖 6.7 模擬資料集之散佈圖	144
圖 6.8 confusion matrix	145
圖 6.9 confusion matrix (三群)	146
圖 7.1 常態分配且變異數相同之資料集	150
圖 7.2 常態分配但變異數不同之資料集	151
圖 7.3 非常態分配之資料集	152
圖 7.4 所有模型正確率以資料集一為例	154

圖 7.5 所有模型正確率以資料集二為例 155

圖 7.6 所有模型正確率以資料集三為例 156

表目錄

表 1.1 例一之統計量	11
表 1.2 每周收看電視的時數	13
表 1.3 來自表 1.2 之資料彙整	14
表 1.4 人均所得	16
表 1.5 樣本群集居民資訊統計量	17
表 1.6 製造業樣本的員工和薪資資料	19
表 4.1 $\hat{\beta}$ vector	104
表 5.1 兩種類別之訓練正確率	132
表 5.2 兩種類別之測試正確率	132
表 5.3 三種類別之訓練正確率	134
表 5.4 三種類別之測試正確率	134
表 5.5 綜合兩群及三群之測試正確率	135
表 6.1 不同隱藏層之 performance(N=100)	143
表 6.2 不同隱藏層之 performance(N=400)	143
表 6.3 不同隱藏層之錯誤率以兩群為例	145
表 6.4 不同隱藏層之錯誤率以三群為例	146
表 7.1 所有分類器比較 _ 以資料集一為例	153
表 7.2 所有分類器比較 _ 以資料集一為例	154

表 7.3 所有分類器比較_以資料集一為例 155

表 7.4 所有分類器之正確率 157

序

對於統計而言，數學是理論的根本，預測與分析則是能善加利用的工具，然而，在這之間許多人已經受理論所馴服，對於工具僅有表面的操作，卻無法熟悉其中的內涵，而往往導致於此之因素皆來自艱困的數學理論，因此本文利用程式剖析數學，利用程式闡發統計，將原本艱澀的數學以及抽象的理論以圖形呈現，並結合程式中的演算法將簡化數學，讓讀者不再對於理論有空泛的想法，並也不再畏懼數學攏長的計算過程。

而其中，MATLAB 則是本文所示範的程式軟體，因為其簡便的繪圖能力及內建機器學習的應用程式，讓我們能更便利著手繪圖以及了解機器學習，而 MATLAB 在程式語法中，架構與大部分語法相似，而繪圖能力也有其語法操作和使用者介面能圖形化操作，因此對於剛入門的使用者來說，學習上並不吃力，並且在資源上除了許多應用程式能供下載外，也有搜尋器可讓不熟悉的使用者進行查詢，結合上述諸多好處，MATLAB 可算是最適合本文介紹的軟體之一。

透過 MATLAB，我們將實作統計中許多的理論，並以圖形輔佐觀察，亦透過圖形了解分配在不同參數下的呈現樣貌，並且利用程式，我們能快速探討統計中最主要的議題之一，預測，利用程式實現諸多預測方式，並探討不同預測方式的成效優劣，也以圖形繪製其資料散

佈與分類線，將過去所熟悉的議題不再紙上談兵，而是透過電腦直接進行資料分析。

最終，我們除了記錄結果以及分析各資料優劣外，更以 L_AT_EX 呈現所有研究過程，除了圖形，表格，更著重在數學的呈現，透過 L_AT_EX 之學習，也能熟悉在排版配置以及文章包羅萬象的變化，結合 L_AT_EX 與 MATLAB，將統計與數學充分的呈現，即是本文最終目標！

第 1 章

LAT_EX 操作手冊之統計與數學

統計學是在資料分析的基礎上，研究測定、收集、整理、歸納和分析反映資料資料，以便給出正確訊息的科學。這一門學科自 17 世紀中葉產生並逐步發展起來，它廣泛地應用在各門學科，從自然科學、社會科學到人文學科，甚至被用於工商業及政府的情報決策。隨著巨量資料時代來臨，統計的面貌也逐漸改變，與資訊、計算等領域密切結合，是資料科學中的重要主軸之一。

譬如自一組資料中，可以摘要並且描述這份資料的集中和離散情形，這個用法稱作為 **描述統計學**。另外，觀察者以資料的形態，建立出一個用以解釋其隨機性和不確定性的數學模型，以之來推論研究中的步驟及母體，這種用法被稱做 **推論統計學**。這兩種用法都可以被稱作為 **應用統計學**。**數理統計學**則是討論背後的理論基礎的學科。¹

因此，對於資料的分析，描述，預測等等都是統計學中重要的學問，其中**集合與函數**的觀點亦為統計基本理論，因此本文除了將介紹統計在實務上面的應用外，亦會介紹基礎集合與函數觀念，從這幾項重要的學門，帶出最原始的統計風貌，並且了解如何以統計解決各領域方面的事務。

¹資料來自 <https://zh.wikipedia.org/wiki/%E7%BB%9F%E8%AE%A1%E5%AD%A6>

1.1 集合 Set thoery

我們都知道幾乎所有的近代數學理論，都建築在集合論 (*set thoery*) 之上。對於微積分統計學而言，當然也不例外。但若要以嚴格的公設法來討論它，則需要很多時間。為方便起見我們仍以數學家 Cantor 的直觀集合論為基礎。

1. 集合 (**set**) 與元素 (**element**): 元素 x 屬於 A 集合，記為 $x \in A$ ，否則記為 $x \notin A$ 。若集合 A 滿足性質 $P(x)$ 之元素所組成，則記為 $A = \{x \mid P(x)\}$ 。
2. 子集 (**subset**): 設 A, B 為集合，則

$$A \subseteq B \Leftrightarrow (x \in A \Rightarrow x \in B)$$

.

3. 相等 (**equal**): 設 A, B 為集合，則

$$A = B \Leftrightarrow (x \in A \Leftrightarrow x \in B) \Leftrightarrow (A \subseteq B \wedge B \subseteq A)$$

4. 空集合 (**empty set**):

$$\emptyset = \{x \mid x \neq x\}$$

5. 聯集 (**union**): 設 $A, B, A_1, \dots, A_n, \dots$ 皆為集合，則

- $A \cup B = \{x \mid x \in A \vee x \in B\}$
- $\cup_{j=1}^n A_j = \{x \mid \exists j \in \{1, 2, \dots, n\}, \text{使得 } x \in A_j\}$
- $\cup_{j=1}^{\infty} A_j = \{x \mid \exists j \in \mathbb{N}, \text{使得 } x \in A_j\}$

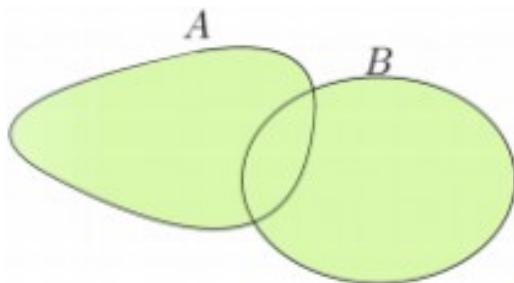


圖 1.1: 聯集

6. 交集 (*intersection*):

- $A \cap B = \{x \mid x \in A \wedge x \in B\}$
- $\cap_{j=1}^n A_j = \{x \mid \forall j \in \{1, \dots, n\}, x \in A_j\}$
- $\cap_{j=1}^{\infty} A_j = \{x \mid \forall j \in \mathbb{N}, x \in A_j\}$

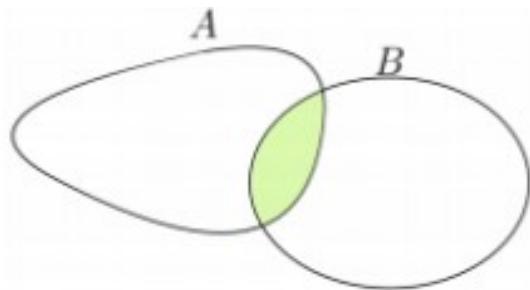


圖 1.2: 交集

7. 差集 (*difference*): 設 A, B 為集合, 則

$$A \setminus B = \{x \mid x \in A \wedge x \notin B\}$$

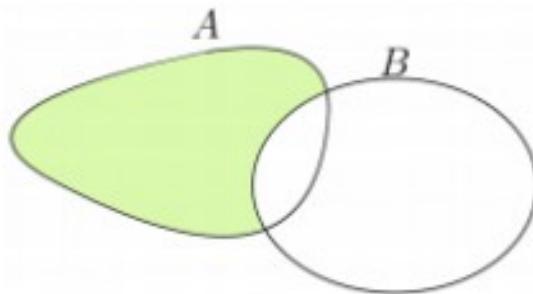


圖 1.3: 差集

8. 幕集 (*power set*):

$$P(A) = \{B \mid B \subseteq A\}$$

9. 序對 (*(ordered pair)*): 設 x, y 為二元素, 令

$$(x, y) = \{\{x\}, \{x, y\}\}$$

由上述定義可以證得: $((x, y) = (a, b) \Leftrightarrow x = a, y = b)$.

10. 積集合 (*(Cartesian product)*): 設 A, B 為集合, 則

$$A \times B = \{(x, y) \mid x \in A \wedge y \in B\}$$

- 例如 $A = \{1, 4\}, B = \{2, 3\}$, 則 $A \times B = \{(1, 2), (1, 3), (4, 2), (4, 3)\}$
- 例如 $A = [1, 4], B = [2, 3]$, 則

$$A \times B = \{(x, y) \mid 1 \leq x \leq 4, 2 \leq y \leq 3\}$$

其乃由平面上 $(1, 2), (4, 2), (4, 3), (1, 3)$ 四點所圍之區域, 如下圖所示:

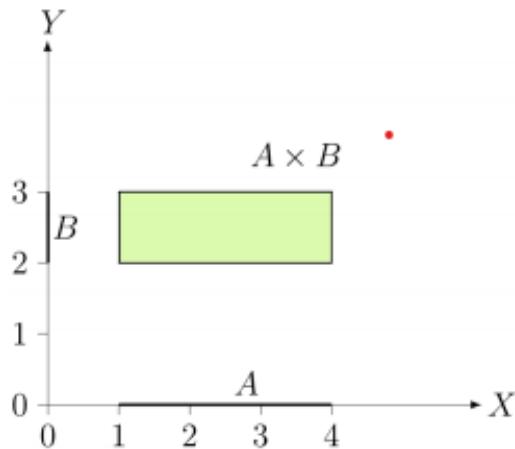


圖 1.4: 乘集

1.2 函數 Function

1. 函數 (*function*): 設 A, B 為二非空集合,

$$f \text{ 為自 } A \text{ 至 } B \text{ 之一函數} \Leftrightarrow \begin{cases} f \subset A \times B \\ \forall x \in A \exists! y \in B \text{ 使得 } (x, y) \in f \end{cases}$$

若 $(x, y) \in f$, 我們常以 $y = f(x)$ 表之. (註: 關於函數之定義亦可界定為: 『 A 中每一元素 x , 必存在 B 中唯一元素 y 與之對應.』但對應二字並非邏輯符號或已界定之名詞, 為了數學之完美, 故以上述方式界定).

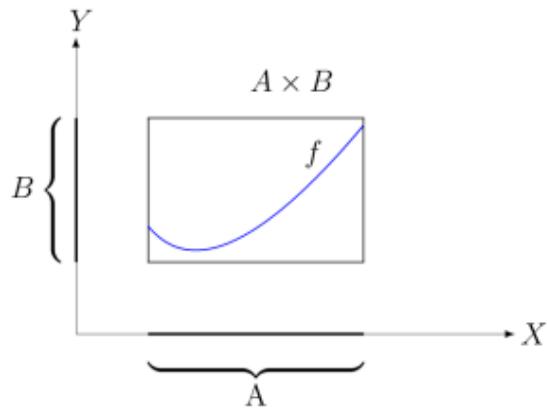


圖 1.5: 函數

函數之記法

♣ $f \rightarrow B : f(x) = \dots ;$

♣ $f \rightarrow B : x \rightarrow \dots .$

本文中，通常以上述方式表示函數，但至目前為止，仍有許多人偏愛以傳統較簡單方式表示，如

♣ $y = x^2;$

♣ $y = \frac{\sin x}{x} \forall x \neq 0;$

♣ $z = x^2 - y^2;$

2. 定義域 (**domain**) 與值域 (**range**): 上述定義中，集合 A 稱為 f 之定義域，常寫為 Df ；而 A 之元素 x 稱為自變數或變數 (**variable**)，此時， $f(x)$ 稱為 x 之函數值 (**value**)，而所有函數值所成之集合稱為 f 之值域，常寫為 Rf ，(某些學者將集合 B 稱為對應域 (**co-domain**)).
3. 映像 (**image**) 與像原 (**inverse image**): 設 $f : A \rightarrow B$ 為一函數，

$S \subset A$, 則

$$f(S) = \{f(x) \mid x \in S\}$$

稱為 S 之 f 映像.(因此, f 之值域 Rf 乃其定義域之 f 映像, 即 $Rf = f(Df)$).

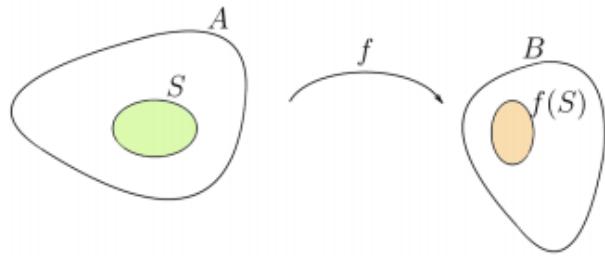


圖 1.6: 映像

若 $T \subset B$, 則

$$f^{-1}(T) = \{x \in A \mid f(x) \in T\}$$

稱為 T 之 f 像原. 顯然 B 及 Rf 之像原皆為定義域 A .

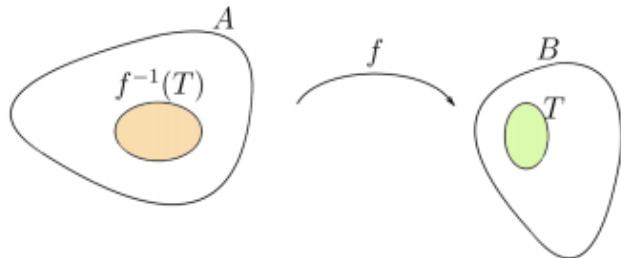


圖 1.7: 像原

4. 嵌射 (**injective or 1-1**) : 其本意為不相同之變數其值亦不同, 更精確的說:

$$f : A \rightarrow B \text{ 為嵌射} \Leftrightarrow (\forall x_1, x_2 \in A)(x_1 \neq x_2 \Rightarrow f(x_1) \neq f(x_2)).$$

5. 蓋射 (*surjective or onto*) : 對應域 B 中之每一元素皆存在 A 中之元素與之對應, 更精確的說:

$$f : A \rightarrow B \text{為蓋射} \Leftrightarrow [\forall y \in B, \exists x \in A \text{使得} y = f(x)]$$

6. 對射 (*bijection or 1-1 onto*): 嵌射且蓋射之意.
7. 函數相等 (*equality of two functions*): 我們稱二函數 f 與 g 相等 (記為 $f = g$), 若其滿足以下二條件:

(a) $D_f = D_g$

(b) $\forall x \in D_f, f(x) = g(x)$

所謂二函數不相等係指上述定義之反面; 更明白地說:

$$f \neq g \Leftrightarrow [Df \neq Dg \text{或} \exists x \in Df \text{使得} f(x) \neq g(x)].$$

8. 反函數 (*inverse function*): 設 f 為一函數, 令 $f^{-1} = \{(y, x) \mid (x, y) \in f\}$, 若 f^{-1} 為一函數則稱 f 為可逆 (*invertible*), 並稱 f^{-1} 為 f 之反函數.

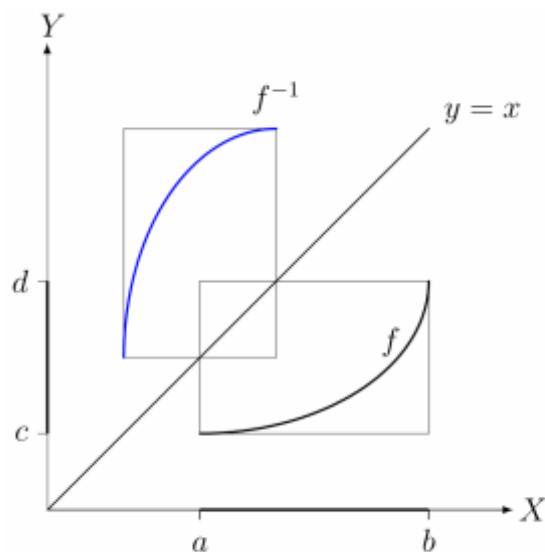


圖 1.8: 反函數

由反函數之定義可知, f^{-1} 與 f 之圖形對稱於直線 $y = x$.

1.3 統計方法之應用 Application of statistical methods

在統計方法的應用之中，從資料之蒐集開始即有自己的一套理論存在，例如抽樣調查，而在分析當中，更是可以簡單區分為有母數 (*parametric statistics*) 與無母數 (*nonparametric statistics*) 之方法運用，而本文之統計實務運用上，將著重於抽樣調查設計的簡單理論概述，比舉例說明，以清楚讓讀者了解。

抽樣調查的目的是藉由母體中所選取樣本的資訊來推論母體，通常是以估計母體平均數 (如每一家戶的平均收入) 或母體比例 (如支持某一特定議題之投票者的比例) 的形式來進行，加上誤差界線範圍內的參數等，對於那些喜歡方法學甚於理論的人，我們會盡量用直覺性論述去

證明估計量的使用。

每一個從母體中選取的觀察值，都包含與母體參數相關的定量資訊。因為獲得資訊需要花錢，所以實驗者必須決定他應該購買多少資訊。資訊太少無法讓實驗者獲得好估計值，資訊過多則會導致金錢浪費。從樣本中獲得的資訊量，取決於抽樣項目數，與資料的變異程度。後者可以透過選取樣本的方法，也就是抽樣調查設計 (*the design of the sample survey*) 來控制。在我們選取的每個元素都有精確測量值的情況下，調查設計與樣本大小會決定樣本中與母體參數有關的資訊量。而以下將會介紹幾種基本的抽樣設計。

1.3.1 簡單隨機抽樣 Simple random sampling

:

Definition 1.3.1. 如果從大小為 N 的母體中選取大小為 n 的一組樣本，使得每一組大小為 n 的樣本都有相同機會被選取，這種抽樣程序被稱為簡單隨機抽樣 (Simple random sampling)。由此得到的樣本，被稱簡單隨機樣本 (Simple random sample)。

其中，我們採用樣本平均數 \bar{y} 來估計 μ 。

$$\bar{y} = \frac{\sum_{i=1}^n y_i}{n}$$

對於母體總和 τ 的不偏估計式如下：

$$\bar{\tau} = \sum_{i=1}^n \frac{y_i}{\pi_i} = \sum_{i=1}^n \frac{y_i}{n/N} = N\bar{y}$$

當然，單獨的 \bar{y} 數值不太能讓我們知道有關母體平均數的資訊，因此我們也要訂出一個估計誤差界線，要完成這項工作，我們需要估計量

的變異數，隊一組來自母體大小為 N 的簡單隨機樣本而言：

$$V(\bar{y}) = \frac{\sigma^2}{n} \left(\frac{N-n}{N-1} \right)$$

考慮樣本變異數

$$s^2 = \frac{1}{n-1} \sum_{i=1}^n (y_i - \bar{y})^2$$

我們可以證明：

$$E(s^2) = \frac{N}{N-1} \sigma^2$$

所以 $V(\bar{y})$ 可以由樣本用下列估計式做不偏地估計

$$\hat{V}(\bar{y}) = \left(1 - \frac{n}{N}\right) \frac{s^2}{n}$$

而估計誤差界限：

$$2\sqrt{\hat{V}(\bar{y})} = 2\sqrt{\left(1 - \frac{n}{N}\right) \frac{s^2}{n}}$$

Example 1. 從母體 $\{1, 2, 3, 4\}$ 中選取大小 $n = 2$ 的樣本，下表顯示六組大小 $n = 2$ 的可能樣本以及相關的樣本統計。

表 1.1: 例一之統計量

樣本	樣本機率	\bar{y}	s^2	$\hat{V}(\bar{y})$
$\{1, 2\}$	$\frac{1}{6}$	6	1.5	0.5
$\{1, 3\}$	$\frac{1}{6}$	8	2.0	0.500
$\{1, 4\}$	$\frac{1}{6}$	10	2.5	4.5
$\{2, 3\}$	$\frac{1}{6}$	10	2.5	0.5
$\{2, 4\}$	$\frac{1}{6}$	12	3.0	2.0
$\{3, 4\}$	$\frac{1}{6}$	14	3.5	0.5

如果從母體中隨機抽出一個觀察值 y ，那麼 y 可以是四個可能數值中任意一個，且機率相同，因此：

$$\mu = E(y) = \sum yp(y) = 1\left(\frac{1}{4}\right) + 2\left(\frac{1}{4}\right) + \cdots + 4\left(\frac{1}{4}\right) = 2.5$$

且

$$\begin{aligned}\sigma^2 = V(y) &= E(y - \mu)^2 = \sum (y - \bar{y})^2 p(y) \\ &= (1 - 2.5)^2 \left(\frac{1}{4}\right) + (2 - 2.5)^2 \left(\frac{1}{4}\right) + \cdots + (4 - 2.5)^2 \left(\frac{1}{4}\right) \\ &= \frac{5}{4}\end{aligned}$$

因為各個樣本平均可能出現的機率為 $\frac{1}{6}$ ，所以我們可以計算 $E(\bar{y})$ 與 $V(\bar{y})$ 。從期望值的定義：

$$E(\bar{y}) = \sum \bar{y} p(\bar{y}) = 1.5 \left(\frac{1}{6}\right) + 2.0 \left(\frac{1}{6}\right) + \cdots + 3.5 \left(\frac{1}{6}\right) = 2.50 = \mu$$

且

$$\begin{aligned}V(\bar{y}) &= E(\bar{y} - \mu)^2 \\ &= \sum (\bar{y} - \mu)^2 p(\bar{y}) \\ &= (1.5 - 2.5)^2 \left(\frac{1}{6}\right) + (2.0 - 2.5)^2 \left(\frac{1}{6}\right) + \cdots + (3.5 - 2.5)^2 \left(\frac{1}{6}\right) = \frac{5}{12}\end{aligned}$$

1.3.2 分層隨機抽樣 Stratified random sampling

:

Definition 1.3.2. 分層隨機樣本 (**Stratified random sample**) 是將母體元素分成不重疊群體，稱為層 **strata**，然後再從每一層選取一組簡單隨機樣本來構成樣本。

其中，令 \bar{y}_i 表示從第 i 層中選取之簡單隨機樣本的樣本平均數， n_i 表示第 i 層的樣本數， μ_i 表示第 i 層的母體平均數，以及 τ_i 表示第 i 層的母體總和，那麼母體總和 τ 就等於 $\tau_1 + \cdots + \tau_n$ 。我們再每一層都有一組簡單隨機樣本，而以下用 \bar{y}_{st} 來表示 μ 的估計量，其中的下標 **st** 意指

使用了分層隨機抽樣。

母體平均數 μ 的估計量：

$$\bar{y}_{st} = \frac{1}{N} [N_1 \bar{y}_1 + N_2 \bar{y}_2 + \cdots + N_L \bar{y}_L] = \frac{1}{N} \sum_{i=1}^L N_i \bar{y}_i$$

\bar{y}_{st} 的估計變異數：

$$\begin{aligned}\hat{V}(\bar{y}_{st}) &= \frac{1}{N^2} [N_1^2 \hat{V}(\bar{y}_1) + N_2^2 \hat{V}(\bar{y}_2) + \cdots + N_L^2 \hat{V}(\bar{y}_L)] \\ &= \frac{1}{N^2} \left[N_1^2 \left(1 - \frac{n_1}{N_1} \right) \left(\frac{s_1^2}{n_1} \right) + \cdots + N_L^2 \left(1 - \frac{n_L}{N_L} \right) \left(\frac{s_L^2}{n_L} \right) \right] \\ &= \frac{1}{N^2} \sum_{i=1}^L N_i^2 \left(1 - \frac{n_i}{N_i} \right) \left(\frac{s_i^2}{n_i} \right)\end{aligned}$$

Example 2. 假設廣告公司有足夠的時間與金錢訪問 $n = 40$ 個家戶，並決定從城鎮 A 選出一組大小為 $n_1 = 20$ 的隨機樣本，城鎮 B 選出一組 $n_2 = 8$ 的隨機樣本，以及鄉村地區選出一組大小為 $n_3 = 12$ 的隨機樣本。選取簡單隨機樣本，並且進行訪問，下表呈現每周收看電視時數的測量結果。

表 1.2: 每周收看電視的時數

城鎮 A	城鎮 B	鄉村
35	27	8
43	15	14
36	4	12
39	41	15
28	49	30
28	25	32
29	10	21

續接下頁

承接上頁

城鎮 A	城鎮 B	鄉村
25	30	20
38	—	34
27	—	7
26	—	11
32	—	24
29	—	—
40	—	—
35	—	—
41	—	—
37	—	—
31	—	—
45	—	—
34	—	—

由表 1.2 資料彙整後，可得以下統計量：

表 1.3: 來自表 1.2 之資料彙整

	N	n	平均數	中位數	標準差
城鎮 A	155	20	33.90	34.50	5.95
城鎮 B	62	8	25.12	26.00	15.25
鄉村	93	12	19.00	17.50	9.36

其中

$$\begin{aligned}\bar{y}_{st} &= \frac{1}{N}[N_1\bar{y}_1 + N_2\bar{y}_2 + \cdots + N_L\bar{y}_L] \\ &= \frac{1}{310}[155(33.90) + 62(25.12) + 93(19.00)] \\ &= 27.7\end{aligned}\tag{1.1}$$

此為該郡所有家戶每周收看電視之平均時數的最佳估計值，而且，

$$\begin{aligned}
 \hat{V}(\bar{y}_s t) &= \frac{1}{N^2} \sum_{i=1}^L N_i^2 \left(1 - \frac{n_i}{N_i} \right) \left(\frac{s_i^2}{n_i} \right) \\
 &= \frac{1}{310^2} \left[\frac{(155^2)(0.871)(5.95)^2}{20} + \frac{(62^2)(0.871)(15.25)^2}{8} + \frac{(93^2)(0.871)(9.36)^2}{12} \right] \\
 &= 1.97
 \end{aligned} \tag{1.2}$$

有著近似 2-SD 估計誤差界限的母體平均數的估計值為

$$\bar{y}_s t \pm 2\sqrt{\hat{V}(\bar{y}_s t)} \text{ 或 } 27.675 \pm 2\sqrt{1.97} \text{ 或 } 27.7 \pm 2.8$$

因此，我們估計在該郡中，家戶收看電視的每周平均時數為 27.7 小時。

在機率近似乎等於 0.95 之下，估計誤差應該小於 2.8 小時。

1.3.3 集群抽樣 Cluster sampling

:

Definition 1.3.3. 集群樣本 (**Cluster sample**) 是每一個抽樣單位都是一組或一集群元素的機率樣本。

如果獲得列出所有母體元素的底冊非常昂貴，或如果獲得觀察值的費用隨母體元素之間的距離增加而提高，集群抽樣就簡單隨機抽樣或是分層隨機抽樣花費少。

假設: N = 母體中的集群數

n = 簡單隨機樣本中被選取的集群數

m_i = 集群 i 中的元素個數， $i = 1, 2, \dots, N$

$\bar{m} = \frac{1}{n} \sum_{i=1}^n m_i$ = 樣本的平均集群大小

$$M = \sum_{i=1}^N m_i = \text{母體元素個數}$$

$$\bar{M} = \frac{M}{N} = \text{母體的平均集群大小}$$

y_i = 第 i 個集群中所有觀測值的總和

母體平均數 μ 的估計量是樣本平均數 \bar{y} ，假定如下

$$\bar{y} = \frac{\sum_{i=1}^n y_i}{\sum_{i=1}^n m_i}$$

且 \bar{y} 的估計變異數:

$$\hat{V}(\bar{y}) = \left(1 - \frac{n}{N}\right) \frac{s_r^2}{n\bar{M}^2}$$

其中

$$s_r^2 = \frac{\sum_{i=1}^n (y_i - \bar{y}m_i)^2}{n-1}$$

Example 3. 假設社會學家決定將某城鎮之地圖區塊是為一群集，且地圖有 415 個區塊，亦即此城鎮有 415 個集群，而今日有足夠金錢與時間抽樣 $n=25$ 個集群，並訪問各集群中的每一個家戶，蒐集之資料如下表所示: 在獲取樣本集群資訊後，即可獲得以下統計量: 母體平均數 μ

表 1.4: 人均所得

集群	居民數 m_i	每群總所得 y_i	集群	居民數 m_i	每群總所得 y_i
1	8	96,000	14	10	49,000
2	12	121,000	15	10	53,000
3	4	42,000	16	10	50,000
4	5	65,000	17	10	32,000
5	6	52,000	18	10	22,000
6	6	40,000	19	10	45,000
7	7	75,000	20	10	37,000
8	5	65,000	21	10	51,000
9	8	45,000	22	10	30,000
10	3	50,000	23	10	39,000
11	2	85,000	24	10	47,000
12	6	43,000	25	10	41,000
13	5	54,000	—	—	—

表 1.5: 樣本群集居民資訊統計量

	N	平均數	中位數	標準差
居民	25	6.040	6.000	2.371
所得	25	53,160	49.000	21,784
$y_i - \bar{y}m_i$	25	0	993	25,189

的最佳估計值計算如下：

$$\bar{y} = \frac{\sum_{i=1}^n y_i}{\sum_{i=1}^n m_i} = \frac{\$1,329,000}{151} = \frac{\$53,160}{6.04} = \$8801$$

因為 M 未知，所以 \bar{M} 必須用 \bar{m} 來估計，其中

$$\bar{m} = \frac{\sum_{i=1}^n m_i}{n} = \frac{151}{25} = 6.04$$

且已知 $N = 415$ ，所以

$$\hat{V}(\bar{y}) = \left(1 - \frac{n}{N}\right) \frac{s_r^2}{n\bar{M}^2} = \left[1 - \frac{25}{415}\right] \frac{(25,189)^2}{25(6.04)^2} = 653,785$$

因此，附加估計誤差界限之 μ 的估計值為

$$\bar{y} \pm 2\sqrt{\hat{V}(\bar{y})} = 8801 \pm 2\sqrt{653,785} = 8801 \pm 1617$$

人均所得的最佳估計值是 $\$8801$ ，且在機率接近 95% 之下，估計誤差應該小於 $\$1617$ ，這個估計誤差界限相當大，可藉由抽樣更多個集群來加以降低。

1.3.4 系統抽樣 Systematic sampling

:

Definition 1.3.4. 從底冊中最初的 k 個元素中隨機選取一個元素，從那之後每 k 個元素隨機選取一個元素，這樣得到的樣本稱為具隨機起點的 k 取 1 系統樣本 (**1-in-k systematic sample**)。

因下列理由，系統抽樣提供較簡單隨機抽樣有用的另一種選擇：

1. 系統抽樣在現場比較容易執行。也因此，與簡單隨機樣本或分層隨機樣本相比，較不會受制於現場調查工作者的選擇偏誤，楊騏是當沒有好的底冊可使用時。
2. 在母體元素的排列具特定的模式時，同樣的單位花費下，系統抽樣能提供的資訊比簡單隨機抽樣更多

我們可以利用從系統樣本平均數 \bar{y} 估計母體平均數 μ 。這個結果顯示如下：母體平均數 μ 的估計量

$$\hat{\mu} = \bar{y}_{sy} = \frac{\sum_{i=1}^n y_i}{n}$$

其中下標 sy 表明我們使用的是系統抽樣
 \bar{y}_{sy} 的估計變異數：

$$\hat{V}(\bar{y}_{sy}) = \left(1 - \frac{n}{N}\right) \frac{s^2}{n}$$

Example 4. 聯邦政府利用蒐集像員工人數和薪資等變數的年度資料，追蹤國內產業表現各項指標，顯示如下表：

表 1.6: 製造業樣本的員工和薪資資料

樣本	SIC	描述	2000 年員工人數(千)			2001 年員工人數(千)			2002 年員工人數(千)		
			2000 年製造廠產品	2000 年員工人數(千)	2001 年員工人數(千)	2001 年員工人數(千)	2002 年員工人數(千)				
1	204	穀物製造廠產品	122.4	122.2	122.2	122.2	122.2	122.2	122.2	122.2	34.9
2	212	雪茄菸	2.9	2.9	3.2	3.2	3.2	3.2	3.2	3.2	26.9
3	225	編織場	120.1	98.6	98.6	98.6	98.6	98.6	98.6	98.6	26.0
4	233	女性，小姐和青少年外衣	169.9	137.3	137.3	137.3	137.3	137.3	137.3	137.3	23.0
5	241	筏木業	78.2	73.6	73.6	73.6	73.6	73.6	73.6	73.6	29.8
6	252	辦公室家具	80.7	69.2	69.2	69.2	69.2	69.2	69.2	69.2	32.5
7	265	硬紙箱和硬紙盒	219.4	207.2	207.2	207.2	207.2	207.2	207.2	207.2	33.8
8	276	各式商業類型	42.0	36.5	36.5	36.5	36.5	36.5	36.5	36.5	33.5
9	284	肥皂，洗潔劑等打掃用品	156.0	149.2	149.2	149.2	149.2	149.2	149.2	149.2	37.8
10	299	石油與煤的各種產品	13.2	14.1	14.1	14.1	14.1	14.1	14.1	14.1	41.9
11	313	軋子和鞋子的切割材料和工具	1.1	0.8	0.8	0.8	0.8	0.8	0.8	0.8	26.1
12	322	玻璃與玻璃器皿，壓製或吹製	67.6	60.0	60.0	60.0	60.0	60.0	60.0	60.0	32.9
13	329	磨料，石綿以及其他類	74.0	67.1	67.1	67.1	67.1	67.1	67.1	67.1	34.4
14	339	各種主要金屬製品	26.8	25.4	25.4	25.4	25.4	25.4	25.4	25.4	35.7
15	347	塗層，雕版和相關服務	149.6	128.5	128.5	128.5	128.5	128.5	128.5	128.5	29.5
16	355	特殊產業機器	170.9	146.4	146.4	146.4	146.4	146.4	146.4	146.4	42.1
17	363	家庭電器	106.3	104.8	104.8	104.8	104.8	104.8	104.8	104.8	30.6
18	372	飛機和零件	466.6	450.5	450.5	450.5	450.5	450.5	450.5	450.5	49.5
19	382	實驗室儀器以及分析控制儀器	311.4	282.4	282.4	282.4	282.4	282.4	282.4	282.4	46.1
20	394	玩偶，玩具，遊戲運動用品	101.0	90.7	90.7	90.7	90.7	90.7	90.7	90.7	31.2
			n	平均數	中位數	標準誤					
			2001 年員工人數	20	113.4	94.6	105.6				
			員工減少人數	20	10.61	7.25	10.29				

從上表中的統計摘要並利用簡單隨機抽樣的標準公式，隊平均員工人數的分析進行如下：

$$\bar{y}_{sy} = 113.4$$
$$\hat{V}(\bar{y}_{sy}) = \left(1 - \frac{20}{140}\right) \left(\frac{1}{20}\right) (105.6)^2$$
$$2\sqrt{\hat{V}(\bar{y}_{sy})} = 2\sqrt{\left(1 - \frac{20}{140}\right) \left(\frac{1}{20}\right)} (105.6) = 43.72$$

因此，每個產業的估計平均員工人數約為 11.34 萬人，加減大約 4.4 萬人。關於員工減少人數的類似計算，產生了 1.061 萬人的估計平均數，估計誤差限度大約是 0.426 萬人。製造業在一年內減少的員工數值算是相當的大，但是因為樣本很小且員工資料大變異量，所以估計誤差限度也很大。

1.4 結論 Conclusion

統計在生活中的不可或缺，是無法雄辯的事實之一，但現代人往往無法正確且有效的運用統計方面的知識，因此本文藉由基礎集合理論，當作概述統計的開端，雖無法讓讀者進一步參透機率論的偉大，卻也利用集合讓一般非統計專業人士了解如何最原始的定義資料歸屬。

再者，本文只透過函數來表達統計內涵的最基本理論，函數只是統計的外衣，分佈與深層的推論才是最真實的統計，因此，隨著基礎集合論後，銜接函數，了解數學的理論，最後才介紹統計的用途，除了在抽樣設計外，ANOVA，迴歸分析都是統計在實務上最顯而易見的例子，也證明生活中統計的無所不在與其強大之處，而本文亦利用 X_EL_AT_EX 呈現所表達的諸多內容，雖說難易度上較一般軟體艱困些許，但方便程度以及其自動排版的能力，決不亞於市面上常見軟體，結合 X_EL_AT_EX 介紹數學與統計，即是本文最終的目的！

第 2 章

MATLAB 實作函數圖形

MATLAB 是 MATrix LABoratory（矩陣實驗室）的縮寫，是一款由美國 The MathWorks 公司出品的商業數學軟體。MATLAB 是一種用於演算法開發、資料視覺化、資料分析以及數值計算的進階技術計算語言和互動式環境。除了矩陣運算、繪製函數/資料圖像等常用功能外，MATLAB 還可以用來建立使用者介面及與呼叫其它語言（包括 C、C++、Java、Python 和 FORTRAN）編寫的程式。

儘管 MATLAB 主要用於數值運算，但利用為數眾多的附加工具箱（*Toolbox*）它也適合不同領域的應用，例如控制系統設計與分析、圖像處理、訊號處理與通訊、金融建模和分析等。另外還有一個配套軟體包 **Simulink**，提供一個視覺化開發環境，常用於系統類比、動態/嵌入式系統開發等方面。¹

而本文將介紹如何實作 MATLAB 程式，並且著重在數學之函數圖形上的展現，除了將探討每個數學式子所呈現的圖形外，更藉由 **X_EL_AT_EX** 編制，結合兩種方便並廣為人知的語言，介紹數學最原始的樣貌，而以下將先探討基礎函數圖形，接著深入分析複雜之函數圖形以及利用 MATLAB 中不同繪圖方式，展現較多元的圖片風貌。

¹ 資料源自：維基百科 (<https://zh.wikipedia.org/wiki/MATLAB>)

另外，由於經費限制，硬體設備無法更進，知識涵養也仍在努力充實，EPS 圖無法有效顯示，因此以下圖片皆以 JPG 圖檔為例，還請讀者多多包容，感謝。

2.1 基礎函數圖形介紹 - 以 12 種函數為例

2.1.1 $y = f(x) = \sin(x) + \cos(x)$

MATLAB 語法:

```
f = @(x) sin(x)+cos(x);  
fplot(f,[-10,10],'LineWidth',3);  
ylim([-2,2]);  
grid;  
title('f(x) = sin(x) + cos(x)');
```

函數圖形呈現如下:

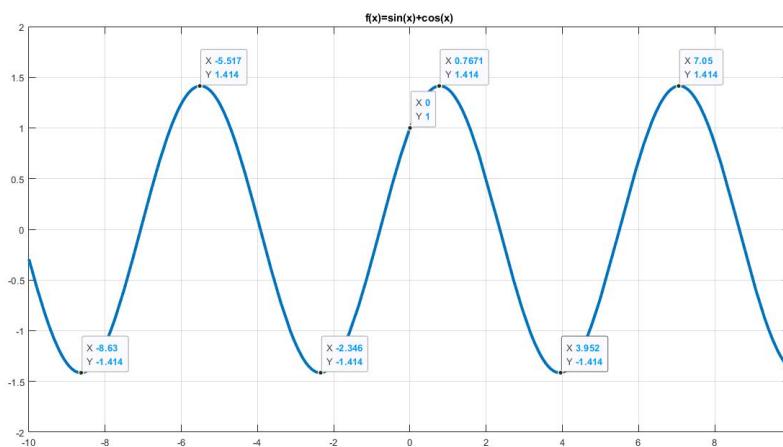


圖 2.1: $y = f(x) = \sin(x) + \cos(x)$

可以從圖 2.1 看出 $\sin(x) + \cos(x)$ 在 $y = -1.5$ 與 $y = 1.5$ 之間徘徊，

形成波型圖，且圖中也可顯而易見此通過 $(0, 1)$ 且至高點與低點在 $y = 1.414$ 與 $y = -1.414$ ，而此標示僅需點擊 MATLAB 產生之圖片即可，而程式碼中也透漏可以從”LineWidth” 調整線的粗細，此題則以 3 單位為例，最後以”grid” 加上格線，title 產生標題，則形成基本函數圖型。

2.1.2 $y = f(x) = \frac{1-e^{-2x}}{1+e^{-2x}}$

MATLAB 語法:

```
f = @(x) (1-exp(-2*x))*(1+exp(-2*x)).^-1;
plot2 = fplot(f);
xlim([-3,3]);grid;
title('f(x)=(1-e\gamma(-2x))/(1+e\gamma(-2x))')
set(plot2,'linewidth',4);set(plot2,'color','red');
text(0.1,0,'when x = 0 -> y = 0');
text(1.8,0.92,'lim(x->\infty) , y->1');
text(-2.5,-0.9,'lim(x->-\infty) , y->-1');
set(plot2,'linestyle',':')
```

函數圖形呈現如下：

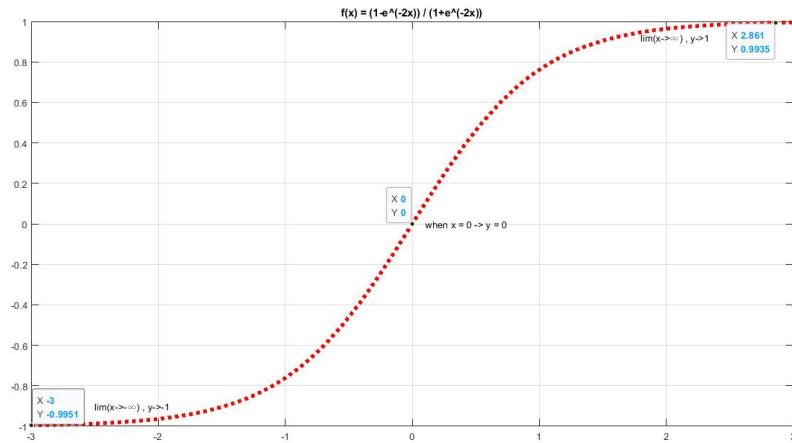


圖 2.2: $y = f(x) = \frac{1-e^{-2x}}{1+e^{-2x}}$

由圖 2.2 可見，此圖不但通過原點，並且當 x 趨近於負無限大時， y 將收斂到 -1 ，而當 x 趨近於無限大時， y 則趨近於 1 ，而從 MATLAB 角度來觀察，則可發現此圖增加圖片敘述，以”text” 呈現，而線也藉由”linestyle”，由實線變成虛線，最後以”color” 改變色彩，則可形成較圖 2.1更精美的圖型，而由於 MATLAB 中變數較常以向量形式表現，因此在除法上不易實現，本例特別由基本除法，改成 -1 次方。

$$2.1.3 \quad y = f(x) = \sqrt[3]{\frac{4-x^3}{1+x^2}}$$

MATLAB 語法:

```
f = @(x) ((4-x^3)*(1+x^2).^( -1))^(1/3);
myFigure = figure; ylim([0,3]);
plot3 = fplot(f,[-20,3]);
set(plot3,'color','# ff b5 ff');
title('f(x) = ((4-x^3)/(1+x^2))^(1/3)');
set(plot3, 'Marker', 'd');
magnify(myFigure); grid; set(plot3,'linewidth',2);
```

函數圖形呈現如下:

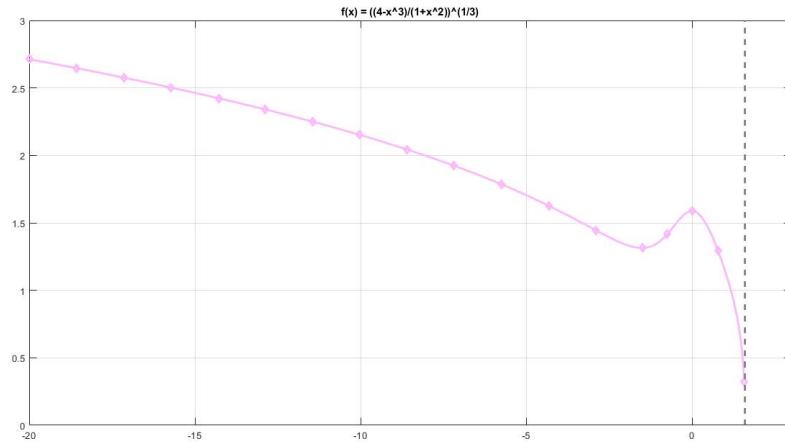


圖 2.3: $y = f(x) = \sqrt[3]{\frac{4-x^3}{1+x^2}}$

圖 2.3 中新增了”Marker”，讓圖形內每個點更能顯而易見的看出，並且，我們可以看出在 $x = 0$ 時，圖形上有明顯的起伏變化，因此利用額

外下載之函數”magnify”，讓使用者能透過游標的移動，有著放大鏡的效果，如此可以更簡易的觀察有興趣之區域，如下圖所示：

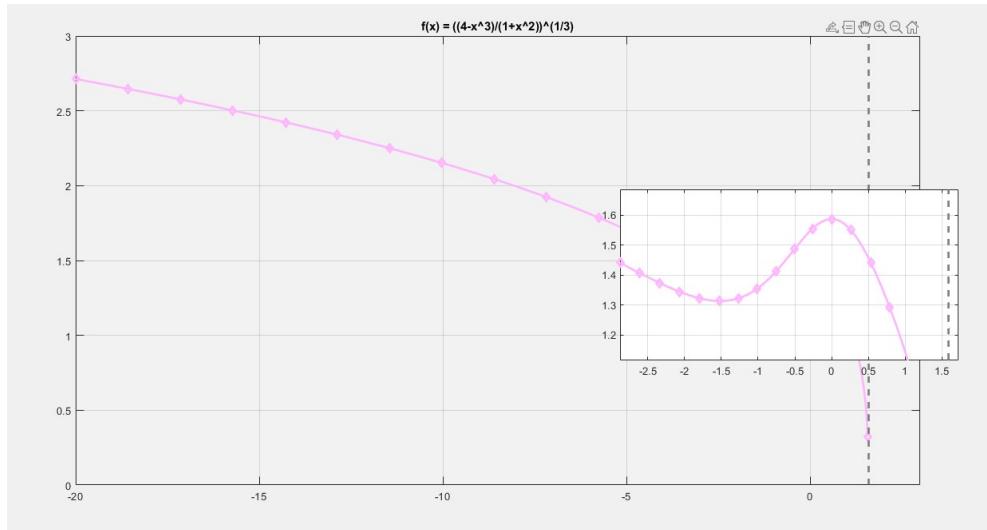


圖 2.4: 部分放大 $y = f(x) = \sqrt[3]{\frac{4-x^3}{1+x^2}}$

2.1.4 $y = f(x) = \frac{1}{x-1}$

MATLAB 語法:

```
f= @(x) 1/(x-1);
plot4 = fplot(f);
xlim([-1,3]);
ylim([-50,50]);
set(plot4, 'linestyle', ':');
set(plot4,'linewidth',3);
grid;
title('f(x)=1/(1-x)');
```

函數圖形呈現如下：

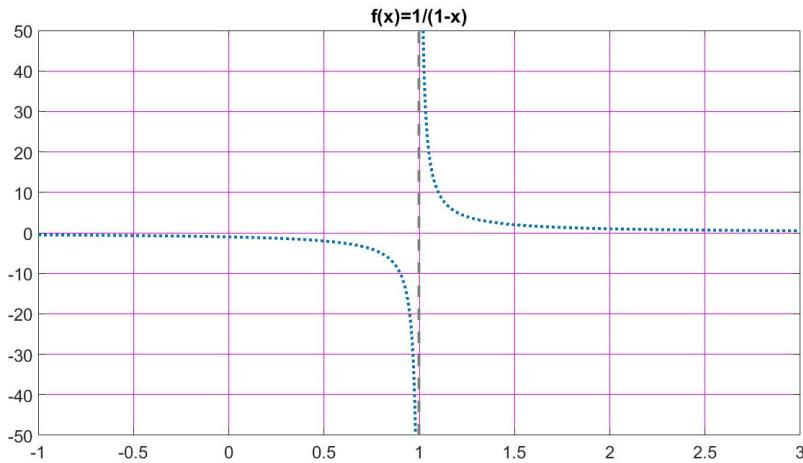
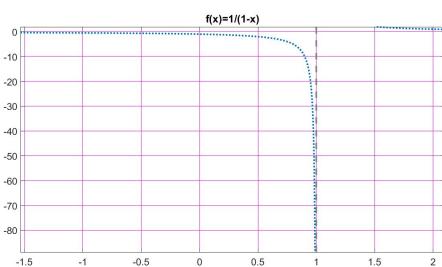
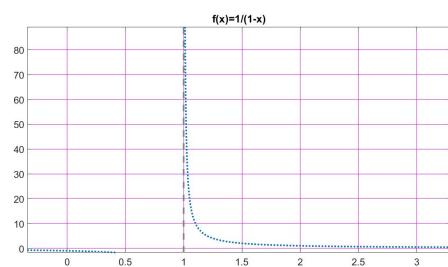


圖 2.5: $y = f(x) = \frac{1}{x-1}$

圖 2.5 以原點對稱，並且無限趨近於 $x = 0$ 與 $y = 1$ ，而此例隨無透過原始程式碼作圖形上的更動，但卻能由 MATLAB 內建繪圖程式，對圖片進行修改，以此為例，修改了”grid”之色彩，以及其透明度，而以下兩圖則是為了能夠更明顯看出漸進效果而增添：



(a) 圖形左側分佈



(b) 圖形右側分佈

圖 2.6: $y = f(x) = \frac{1}{x-1}$

$$\mathbf{2.1.5} \quad y = f(x) = \frac{1}{2\sqrt{2\pi}} e^{\frac{-(x-1)^2}{8}}$$

MATLAB 語法:

```
f = @(x) (1/(2*(sqrt(2*pi))))*exp((-x-1)^2/(8));
plot5 = fplot(f);
xlim([-6,8]);
ylim([0,0.3]);
set(plot5,'LineWidth',2);
set(plot5, 'MarkerSize', 10);
set(plot5,'Color','blue');
grid;
line([1 1],[-100 0.2],'LineStyle',':','LineWidth',1,'Color','red')
line([-3 -3],[-100 100],'LineStyle',':','LineWidth', 1,'Color','red')
line([5 5],[-100 100],'LineStyle',':','LineWidth',1,'Color','red')
text(1,0.21,'mean = 1');
text(-4.3,0.03,'mean - 2sigma');
text(5.1,0.028,'mean + 2sigma');
title('f(x)=(1/2*(2\pi)\^(1/2))*e\^(-(x-1)\^2/8)');
```

函數圖形呈現如下：

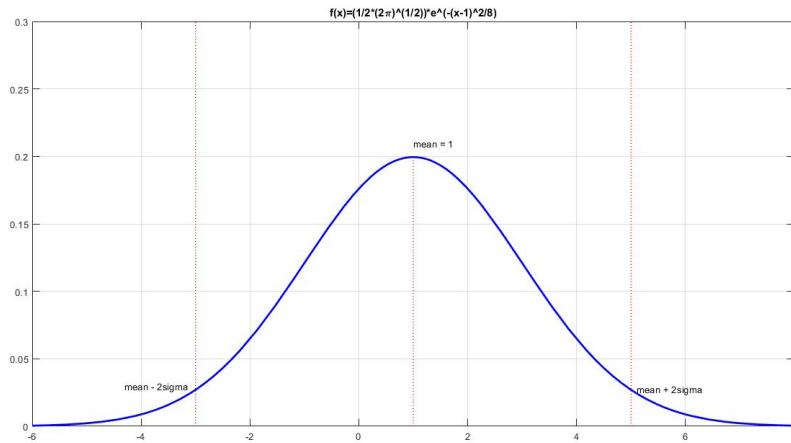


圖 2.7: $y = f(x) = \frac{1}{2\sqrt{2\pi}}e^{\frac{-(x-1)^2}{8}}$

圖 2.7 在統計上呈現常態分佈，且平均數為 1，變異數為 4，因此才以 $x = 1$ 呈現左右對稱之情形，而此例中更是增加三條虛線，分別是 $\mu - 2\sigma$, μ , $\mu + 2\sigma$ ，在 MATLAB 語法上則是透過”line”來實現垂直線的做法，其中本例第一個參數是 x 之範圍，第二個參數是 y 範圍，其後等同”plot”可改變屬性，其中若無法直覺觀察出本例是常態分配的話，可用 x 與 y 界限調整，讓”xlim”調至令圖形對稱位置，可較輕鬆理解圖形分佈。

2.1.6 $y = f(x) = \sqrt[3]{x^2}$

MATLAB 語法:

```
f = @(x) x^(2/3)
plot6 = fplot(f);
xlim([-10,10]);
ylim([-10,10]);
set(plot6,'LineWidth',3);
set(plot6, 'MarkerSize', 8);
set(plot6,'Color','red');
set(plot6,'MarkerEdgeColor','b');
set(plot6,'MarkerFaceColor','y');
set(plot6, 'Marker', 'd');
title("f(x)=x^(2/3)");
grid
```

函數圖形呈現如下:

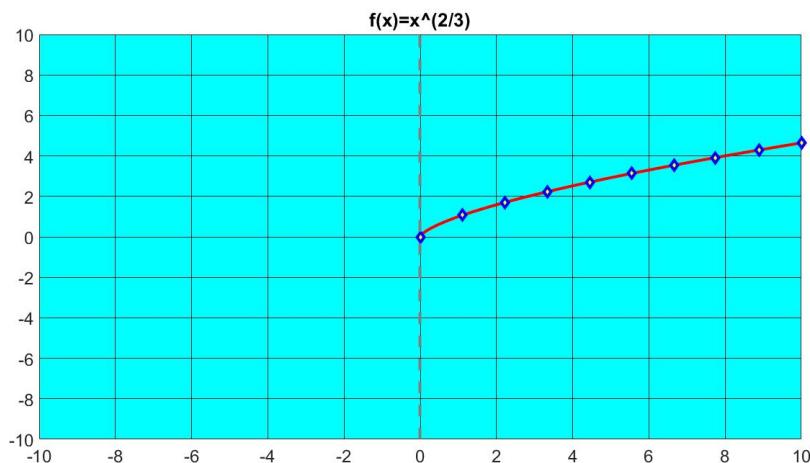


圖 2.8: $y = f(x) = \sqrt[3]{x^2}$

此例在函數呈現上較單純，由於 $f(x)$ 恒正，因此圖形僅在第一象限呈現，而值得一提的是，本例特別將”marker”做多方面修改，第一是改變樣式，在圖 2.3 中有提到，而這裡更進一步修改其大小”MarkerSize”，邊線顏色”MarkerEdgeColor”，以及其內部填滿顏色”MarkerFaceColor”，可透過不同英文字母，所更動其中各種樣式，最後，背景則透過 MATLAB 內建繪圖工具完成。

2.1.7 $y = f(x) = 2x^3 - x^4$

MATLAB 語法:

```
f = @(x) 2*x.^3-x.^4;
plot7 = fplot(f);
ylim([-1000,100]);
xlim([-5,6]);
set(plot7,'linewidth',2);
title('f(x)=2x\3-x\4');
line([-10 10],[0 0],'color','red');
grid
```

函數圖形呈現如下：

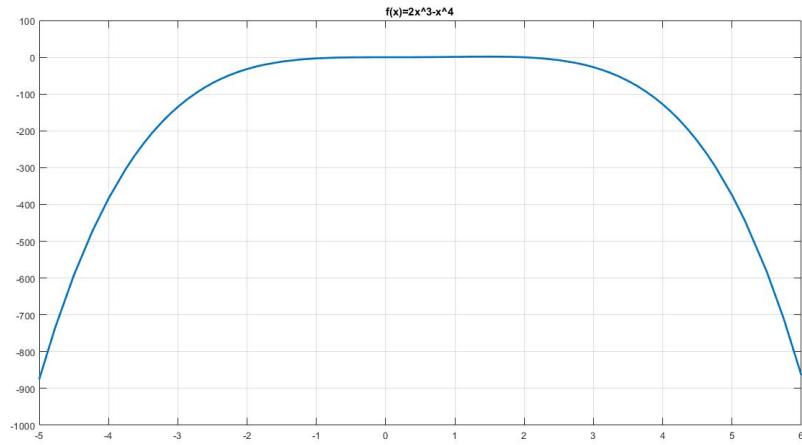
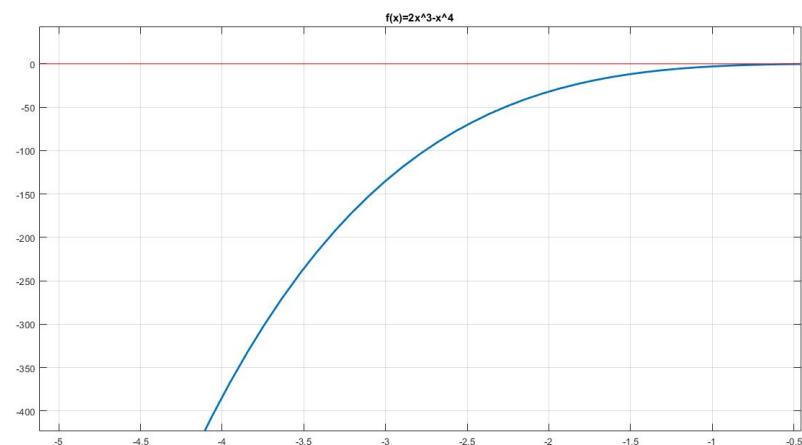
圖 2.9: $y = f(x) = 2x^3 - x^4$

圖 2.9 透過”ylim”的調整縮小後，可以明顯觀察出是一類似拋物曲線，且開口向下，但由此圖無法看出是否與 0 有切線，亦或是割線產生，因此藉由以下三種放大可進一步觀察；

圖 2.10: 函數 $y = f(x) = 2x^3 - x^4$ 之左側放大

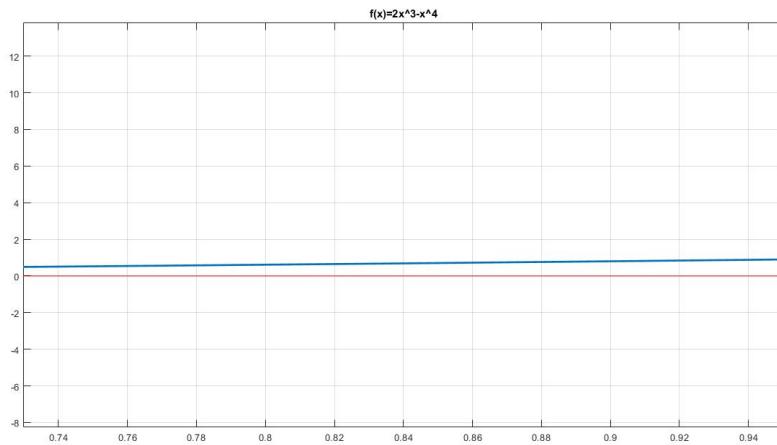


圖 2.11: 函數 $y = f(x) = 2x^3 - x^4$ 之中間放大

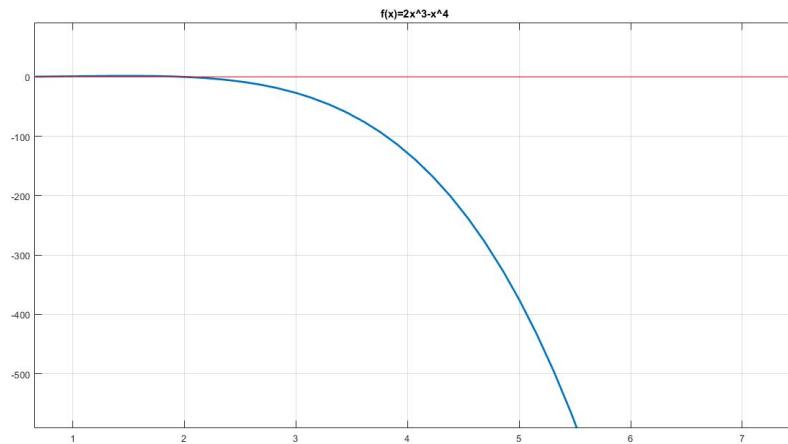


圖 2.12: 函數 $y = f(x) = 2x^3 - x^4$ 之右側放大

其中在圖 2.11(中圖) 中，在放大若干倍可見函數以落在紅線 ($y = 0$) 之上，因此更可確定此函數與 x 軸焦於兩點。

2.1.8 $y = f(x) = x\sqrt{4 - x^2}$

MATLAB 語法:

```
f = @(x) x*sqrt(4-x.^2);
plot8 = fplot(f);
ylim([-2.5,2.5]);
xlim([-3,3]);
set(plot8,'linewidth',2);
title("f(x)=x*sqrt(4-x^2)");
line([-10 10],[2 2],'linestyle',':', 'color','red');
line([-10 10],[-2 -2],'linestyle',':', 'color','red');
grid
```

函數圖形呈現如下:

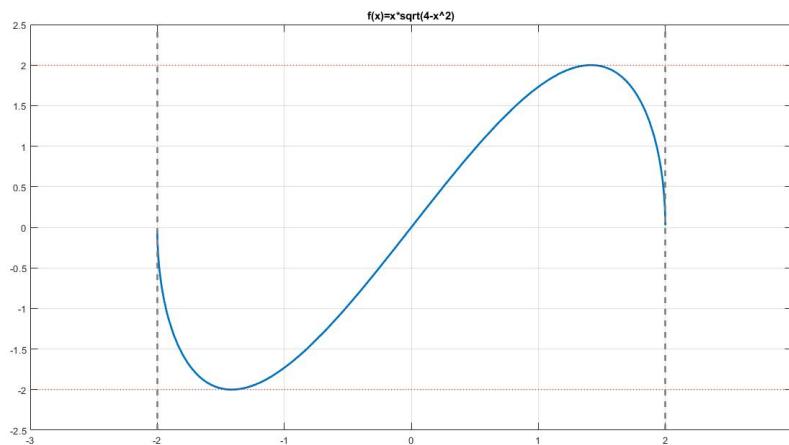
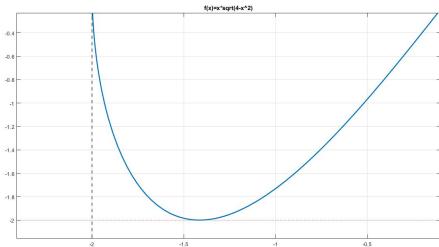


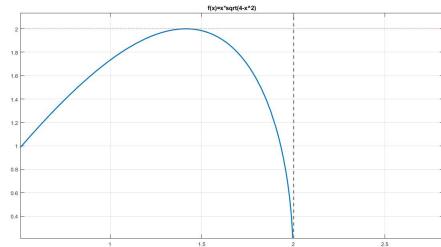
圖 2.13: $y = f(x) = x\sqrt{4 - x^2}$

此例是一個類似 S 形的圖形，其中 x, y 皆介於 2 至 -2 之間，因此刻意加上”line”指令做出虛線，更方便辨識其定義域，而此例也特別放

大其漸進 $x = 2$ 與 $y = 2$ 的部分，以及 $x = -2$ 和 $y = -2$ ，如下圖所示：



(a) 圖形左側分佈



(b) 圖形右側分佈

圖 2.14: $y = f(x) = x\sqrt{4 - x^2}$

2.1.9 $y = f(x) = \frac{\ln x}{x^3}$

MATLAB 語法:

```
f = @(x) log(x)*x.^(-3);
plot9 = fplot(f);
line([-10 10],[0 0],'linestyle',':', 'color','red');
grid
title("f(x)=ln(x)/x^3");
set(plot9,'linewidth',2);
xlim([-2 10]);
ylim([-10 2]);
```

函數圖形呈現如下：

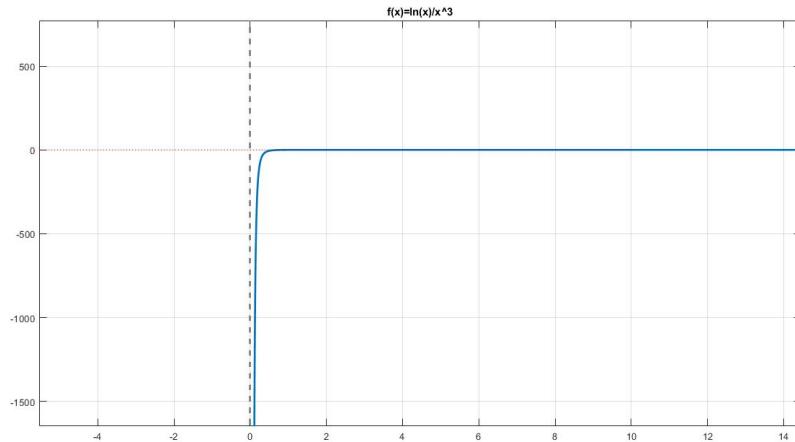


圖 2.15: $y = f(x) = \frac{\ln x}{x^3}$

此例一樣須注意向量中，除法不易執行，因此一樣使用 -3 次方完成除法的部分，而此圖看似與 x 軸有漸進線，卻無法確定是否存在交點，因此刻意放大觀察，如下圖：

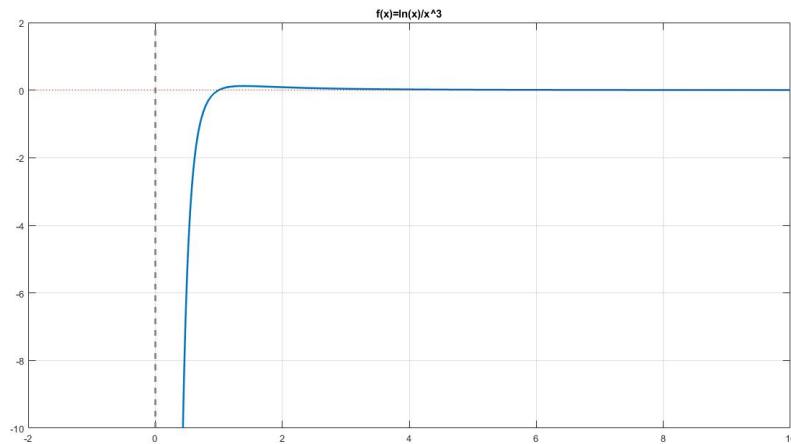


圖 2.16: $y = f(x) = \frac{\ln x}{x^3}$

確定有與 x 軸有交點，並非漸進線，更進一步觀察：

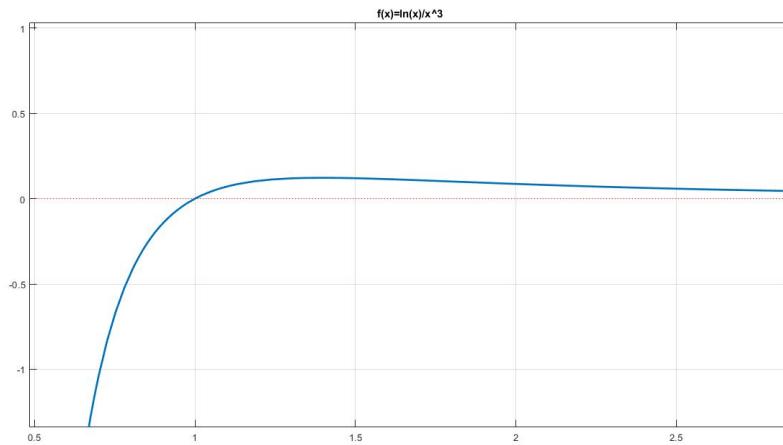


圖 2.17: $y = f(x) = \frac{\ln x}{x^3}$

約在 $x = 1$ 時存在交點。

2.1.10 $y = f(x) = 3, 1 \leq x \leq 5$

MATLAB 語法:

```
line([1 5],[3 3],'color','red','linewidth',3);
title('y=f(x)=3');
xlim([0.5 5.5])
grid
```

函數圖形呈現如下：

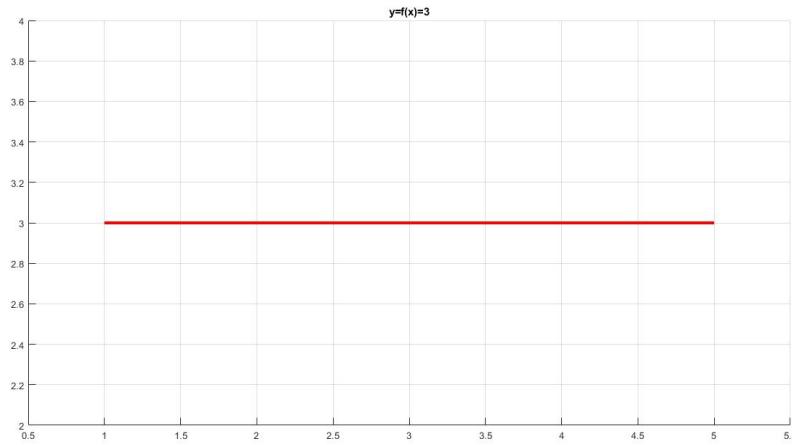


圖 2.18: $y = f(x) = 3, 1 \leq x \leq 5$

此例較為單純，僅須注意 x 的定義域僅限於 1 至 5，因此設定'line'時， x 範圍以”[1 5]”表現。

2.1.11 $x^2 + y^2 = 1$

MATLAB 語法:

```
f=@(x,y) x.^2+y.^2-1;  
plot11=fimplicit(f, [-1.5 1.5 -1.5 1.5]);  
set(plot11,'marker','s');  
grid  
title('x\^2+y\^2=1')
```

函數圖形呈現如下：

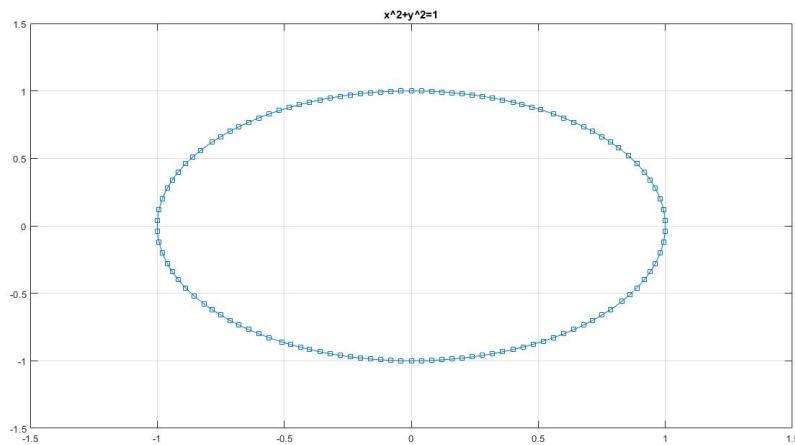


圖 2.19: $x^2 + y^2 = 1$

本例子需要用到”`fimplicit`”輔佐”，有別於先前僅是透過”`fplot`”可完成，由於此例是方程式，並非多項式，因此需將等號左右整理成類似 $ax + by + c = 0$ 之形式，將左式帶入先前變數 `f`，再藉由”`fimplicit`” 將此多項式形式轉成方程式存入變數”`plot11`”，而其中後面”[-1.5 1.5 -1.5 1.5]” 則是設定 x, y 界限，亦可用”`set`” 設定，最後作法和先前一致即可完成方程式圖形。

2.1.12 正方形

MATLAB 語法:

```
title(' 正方形');
grid;
xlim([0 3])
ylim([0 3])
line([1 2],[1 1],'color','red','linewidth',3);
line([1 2],[2 2],'color','red','linewidth',3);
line([1 1],[1 2],'color','red','linewidth',3);
line([2 2],[1 2],'color','red','linewidth',3);
```

函數圖形呈現如下:

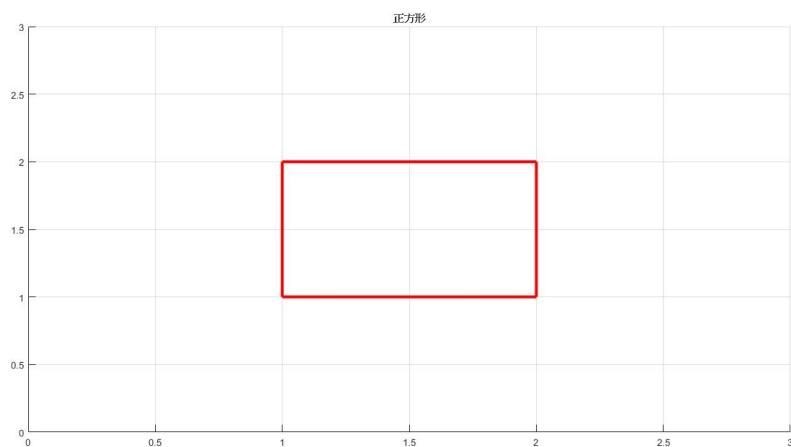


圖 2.20: 正方形

先前多次以”line”做出水平，垂直線等，而正方形亦可透過水平與垂直線形成，僅需控制 x 與 y 的範圍即可達成！

2.2 特殊函數圖形呈現

2.2.1 $f(y) = \frac{1}{\beta}e^{\frac{-y}{\beta}}, 0 \leq y \leq \infty$

MATLAB 語法:

```
x=[0 :0.1:5];
beta=1;
y=(beta^(-1))*exp((-x)*(beta^(-1)));
plotO1 = plot(x,y);
set(plotO1,'linewidth',3);
ylim([0 1])
title("Exponential Probability Distribution (\beta = 1)")
xlabel("x")
ylabel('Probability Density Function')
grid;
set(gca,'fontsize',14);
```

函數圖形呈現如下：

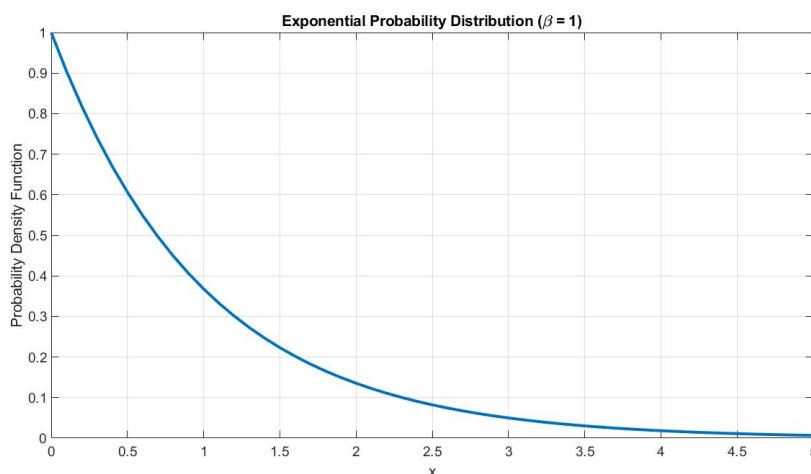


圖 2.21: $f(y) = \frac{1}{\beta}e^{\frac{-y}{\beta}}, 0 \leq y \leq \infty$

$$\mathbf{2.2.2} \quad f(y) = \left[\frac{\gamma(\alpha+\beta)}{\gamma(\alpha)\gamma(\beta)} \right] y^{\alpha-1}(1-y)^{\beta-1}, 0 \leq y \leq 1$$

MATLAB 語法:

```
alpha = 1;beta =2;x = [0:0.01:1];
temp=gamma((alpha+beta)/(gamma(alpha)*gamma(beta)));
y1 = temp.*(x.^(alpha-1)).*(1-x).^(beta-1);
alpha = 1;beta = 3;
y2 = temp.*(x.^(alpha-1)).*(1-x).^(beta-1);
alpha = 2;beta = 2;
y3 = temp.*(x.^(alpha-1)).*(1-x).^(beta-1);
alpha = 2;beta = 3;
y4 = temp.*(x.^(alpha-1)).*(1-x).^(beta-1);
alpha = 3;beta = 1;
y5 = temp.*(x.^(alpha-1)).*(1-x).^(beta-1);
alpha = 6;beta = 1;
y6 = temp.*(x.^(alpha-1)).*(1-x).^(beta-1);
plot(x,y1,x,y2,x,y3,x,y4,x,y5,x,y6,'linewidth',2)
xlim([0,1]);ylim([0,2]);grid
legend("\alpha =1 \beta =2", "\alpha =1 \beta =3", "\alpha =2\beta =2", "\alpha =2 \beta =3", "\alpha =3 \beta =1", "\alpha =6 \beta =1")
title("Beta Dist."); xlabel("x")
ylabel('Probability Density Function')
set(gca,'fontsize',14);
```

函數圖形呈現如下：

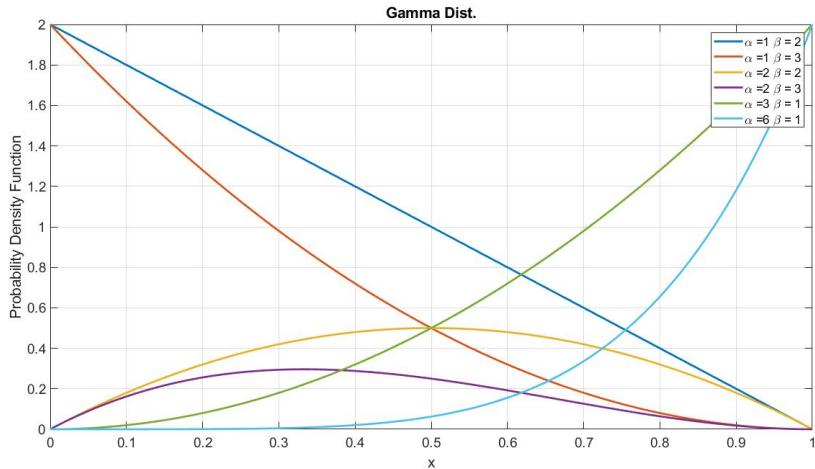


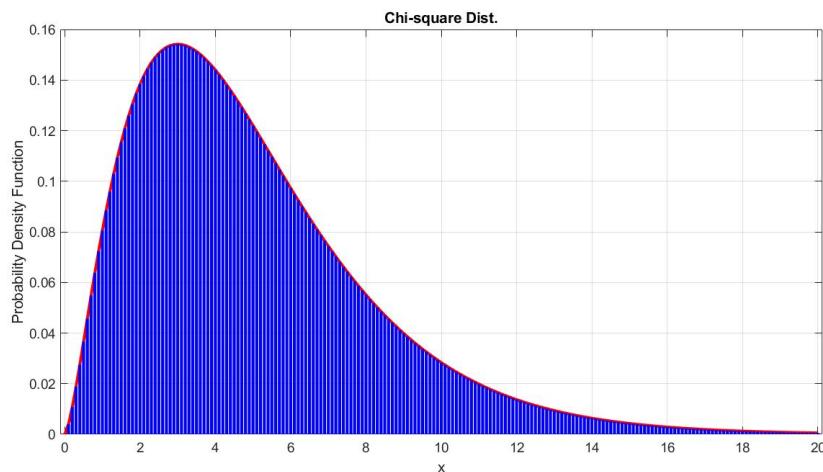
圖 2.22: Beta 分配

2.2.3 X^2 分配

MATLAB 語法:

```
v=5;alpha = v/2;beta = 2;x = [0:0.1:20];
temp=1/((gamma(alpha))^(beta));
y = temp.*((x.^alpha-1)).*exp(-x./beta);
plot(x,y,'linewidth',3,'color','red');hold on;
title("Chi-square Dist.");
xlabel("x");grid;
ylabel('Probability Density Function')
set(gca,'fontsize',14);
myBar=bar(x,y);hold off;
color _ background=['c' 'm' 'y' 'k' 'r' 'g' 'b'];
set(myBar,'FaceColor',color_ background(7));
```

函數圖形呈現如下：

圖 2.23: X^2 分配

其中利用了”bar”讓函數底下面積也能夠一併顯示。而”colorbackground”則先將所有顏色存入後，未來方便使用，直接從此 array 內一一叫出即可。

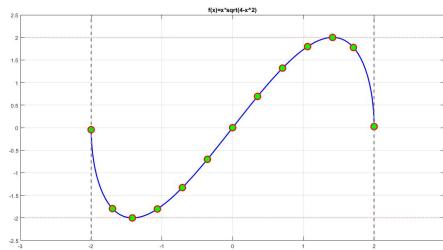
2.3 其他參數與圖表

2.3.1 Marker 使用

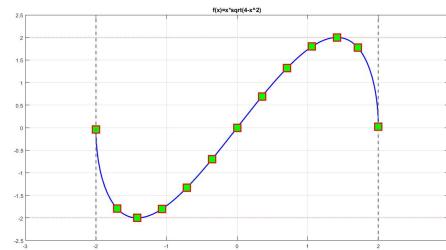
MATLAB 語法:

```
f = @(x) x*sqrt(4-x.^2);
plot8 = fplot(f);
ylim([-2.5,2.5]);
xlim([-3,3]);
set(plot8,'linewidth',2);
title("f(x)=x*sqrt(4-x^2)");
line([-10 10],[2 2],'linestyle',':', 'color','red');
line([-10 10],[-2 -2],'linestyle',':', 'color','red');
set(plot8, 'Marker', '<');
set(plot8, 'MarkerSize', 18);
set(plot8,'color','b');
set(plot8,'MarkerEdgeColor','g');
set(plot8,'MarkerFaceColor',[1 .6 .6]);
grid
```

此例”Marker” 設定為”<”，也就是三角形的圖式，以下總共六種範例以供參考，皆式調整”Marker” 參數可能，其中包含”+”，”*”，”o”，”p”等等。

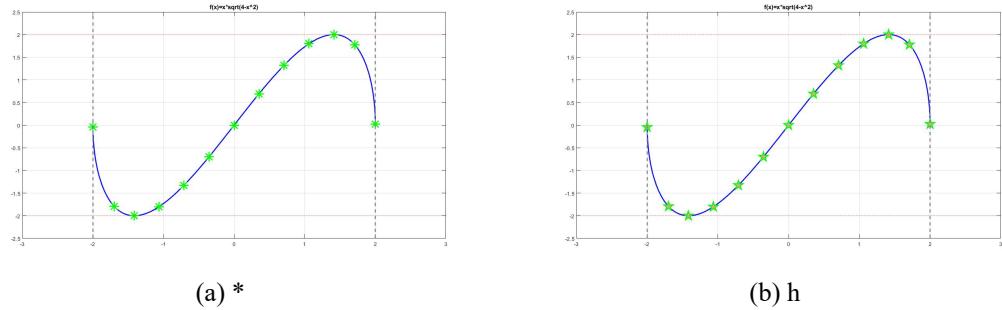


(a) -gs



(b) -o

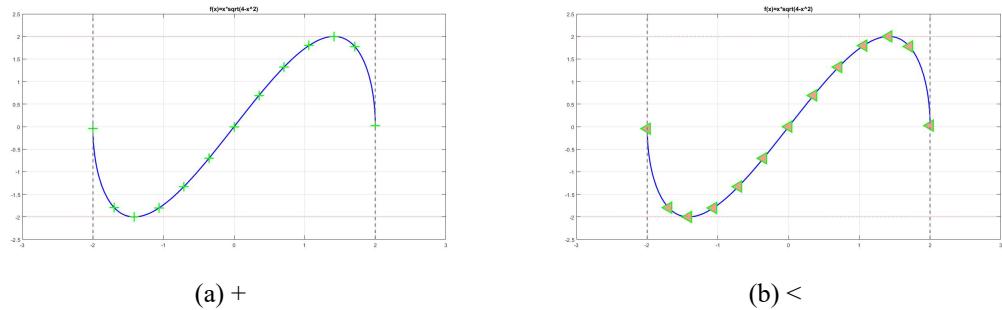
圖 2.24: Marker 展示”-gs” 和”-o”



(a) *

(b) h

圖 2.25: Marker 展示”*”和”h”



(a) +

(b) <

圖 2.26: Marker 展示”+”和”<”

2.3.2 linestyle 使用

MATLAB 語法:

```

x = 0:pi/100:2*pi;
y1 = sin(x);
y2 = sin(x-0.25);
y3 = sin(x-0.5);
y4 = sin(x-0.75);
y5 = sin(x-1);
y6 = sin(x-1.25);
figure
plot(x,y1,x,y2,'-',x,y3,:',x,y4,'b-o',x,y5,'c*',x,y6,'-.')
title('LineStyle Display')
set(gca,'fontsize',16);
grid

```

圖形呈現如下:

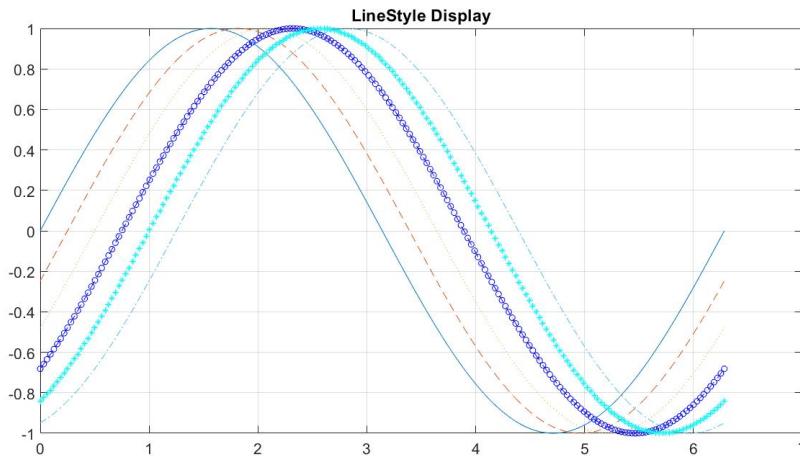


圖 2.27: linestyle 展示

透過"linestyle" 可以設定不同種類的線，其中包含"-","c*","-.等

等。

2.3.3 Bar 使用

MATLAB 語法:

```
x = 1900:10:2000;  
y = [100 91 50 123.5 131 20 179 203 200 249 100];  
bar(x,y,'facecolor','black')
```

圖形呈現如下：

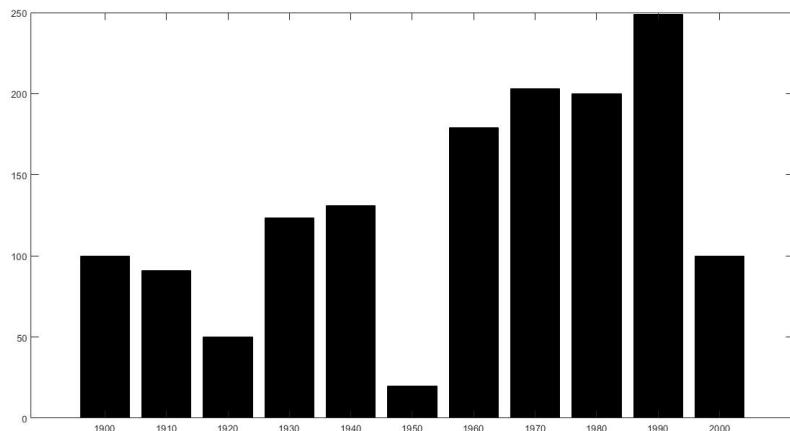


圖 2.28: 一般常見的 Bar 展示

MATLAB 語法:

```
y = [2 2 3; 2 5 6; 2 8 9; 2 11 12];  
bar(y,'stacked')
```

圖形呈現如下：

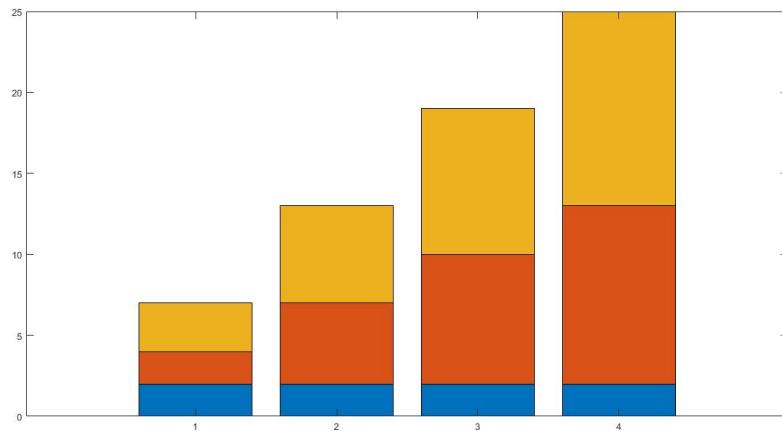


圖 2.29: 多條 Bar 重疊展示

MATLAB 語法:

```
x = [1 2 3];  
vals = [20 15 6; 11 23 26];  
b = bar(x,vals);  
set(b(1),'FaceColor','y');  
set(b(2),'FaceColor','g');  
grid;  
title("My Bar");  
set(gca,'fontsize',16);
```

圖形呈現如下：

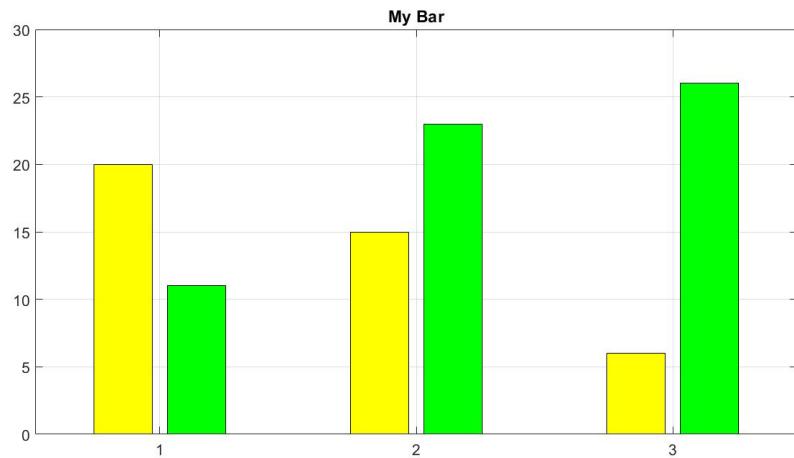


圖 2.30: 並排 Bar 展示

2.3.4 Histogram 使用

MATLAB 語法:

```
x = randn(1000,5);  
nbins = 7;  
hist(x,nbins);
```

圖形呈現如下：

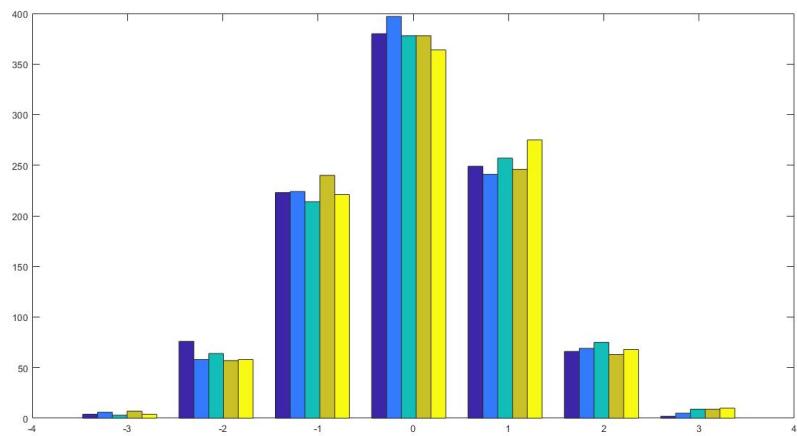


圖 2.31: Histogram 展示

2.3.5 Scatter 使用

MATLAB 語法:

```
x = linspace(0,3*pi,200);  
y = cos(x) + rand(1,200);  
sz = 50;  
c = linspace(1,10,length(x));  
scatter(x,y,sz,c,'filled')
```

圖形呈現如下：

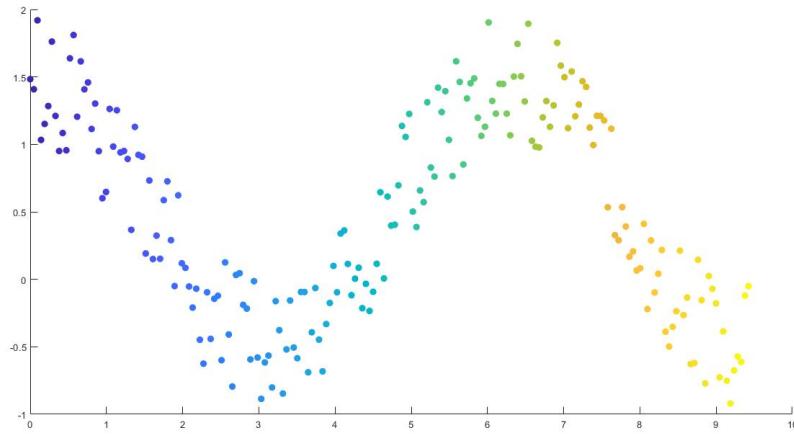


圖 2.32: Scatter 展示

其中可以利用 `scatter` 中第三個參數調整點的大小，第四個參數調整顏色，第五個參數調整是否填滿。

2.3.6 Pie 使用

MATLAB 語法:

```
X = [2 2 0.3 5 1];  
explode = [0 0 1 1 0];  
pie(X,explode)
```

圖形呈現如下：

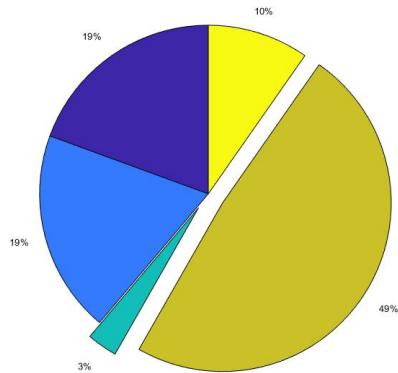


圖 2.33: Pie 分割展示

MATLAB 語法:

```
X = [2 2 0.3 5 1];  
pie(X)
```

圖形呈現如下：

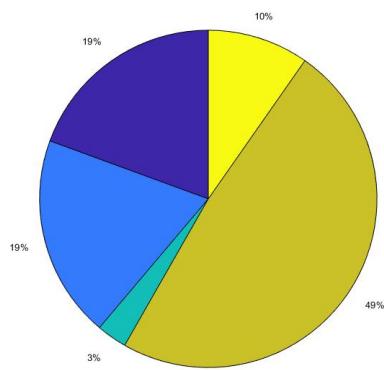


圖 2.34: 一般常見的 Pie 展示

MATLAB 語法:

```
X = [0.3 0.4 0.1];  
pie(X)
```

圖形呈現如下:

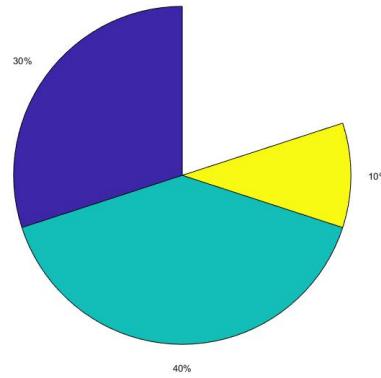


圖 2.35: 特定缺口之 Pie 展示

2.4 結論 Conclusion

透過 MATLAB 可以實現各種不同數學函數的圖表產生，並且有別於其他語言，MATLAB 更可以透過產生後的圖表，自行利用內建圖形化界面設定圖內參數等等，不需要一一透過指令即可達成，而指令上來說，與大部分程式大同小異，若有程式基礎，大概可以猜出大部分內容，並且此軟體也支援許多深度學習，統計，機器學習方面的 APP，若需要亦可從介面中下載，可說是非常方便，在使用方便度與實用性來說不亞於 python 或 R，而此次也展現其在繪圖上功能的完善之處，與其多樣性，可說是對數學或函數圖形上最友善的程式語言之一。

第 3 章

MATLAB 實作之統計分配

MATLAB 在處理各種數學函數上，有不盡其數的語法能使用，甚至在繪圖上，皆不亞於其他程式語言，而今日也在此以 MATLAB 為主要示範語法，來著手其在 [函數展現](#)，[繪圖討論](#)，[亂數產生](#)等等方面之應用，除了討論基本語法的變化，繪圖美觀的琢磨，更討論函數圖形在統計上的意義，介紹在不同參數下，各個分配變化多端的樣貌！

3.1 分配函數

在統計學上，充滿各種不同類型的分配，每個分配都有各自獨樹一幟的絕美風貌，更是各自帶著專屬於自己的參數，而每個參數的組成，又能形成各種各樣的不同分配形狀，這也是統計的迷人所在，而以下也將討論各種不同的分配，如何利用 MATLAB 呈現，而不同的參數，又是如何形成新的圖形樣貌：

3.1.1 連續型函數

本文所討論的，又可稱作連續型隨機變數，在統計上，連續的分配涵蓋各式各樣層面，有在生活中常見的 [常態分配](#)，檢定中佔據重要地位的 [卡方分配](#)，亦或是變化多端的 [貝塔分配](#)，都屬於連續型的一種，而

以下也配合幾種範例，展示 MATLAB 在統計函數上的使用，以及各式分配的樣貌：

1. 常態分配：

常態分布有兩個參數，分別為位置參數 (*location*) μ 、尺度參數 (*scale*) σ 。當 $\mu = 0$ 且 $\sigma = 1$ 時，就稱為標準常態分布。

令 X 為一連續隨機變數，若 X 符合常態分布，其機率密度分布函數 (P.D.F) 為：

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \quad -\infty < x < \infty$$

常態分布的機率密度函數圖形：

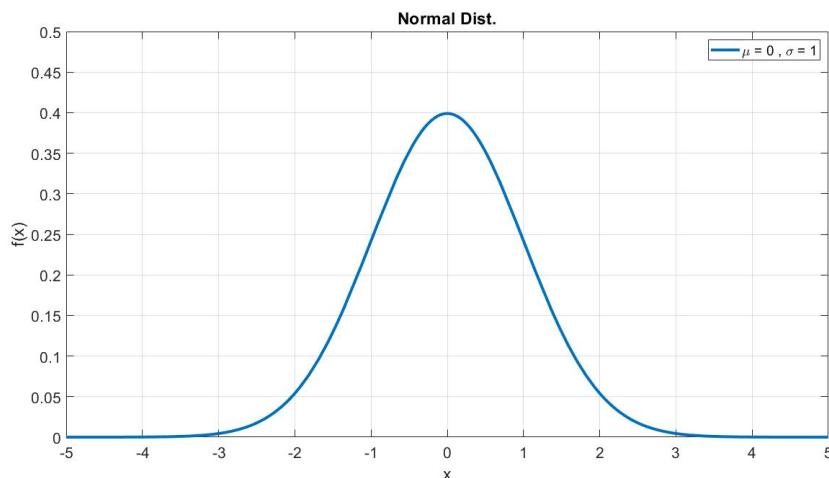


圖 3.1: 常態分配 $\mu = 0, \sigma = 1$

圖 3.1 是當 $\mu = 0$ 且 $\sigma = 1$ 時的常態分佈圖形，可看出其鐘型分布的樣子，且至高點即是平均數 μ ，而在 3 個標準差中，包含了絕大部分的資料，而如此函數呈現，在 MATLAB 上僅需短短幾行程式碼：

MATLAB 語法:

```
f=@(x) normpdf(x,0,1);
fplot(f,'LineWidth',3);
grid;
legend('mu = 0 , \sigma = 1');
title("Normal Dist.");
xlabel("x");
ylabel("f(x)");
ylim([0 0.5]);
set(gca,'fontsize',16);
```

除此之外，我們熟知變異數越大時，常態分配圖形呈現越為矮胖，而當變異數越小時，常態分配則是越為高瘦，而此參數變化，我們也能以 MATLAB 呈現，如圖 3.2：

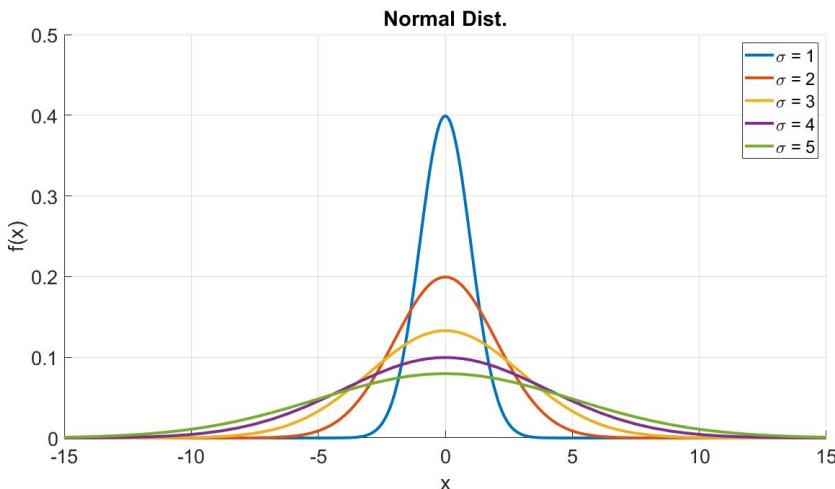


圖 3.2: 多種常態分配

在 MATLAB 應用上，加入**for** 迴圈的使用，讓多種不同參數的程

式碼，更加簡潔：

MATLAB 語法:

```
figure,hold on;
for i = 1:5
    f=@(x) normpdf(x,0,i);
    fplot(f,[-15 15],'LineWidth',3);
end
legend('sigma = 1','sigma = 2','sigma = 3','sigma =4','sigma
= 5');
title("Normal Dist.");
xlabel("x");
ylabel("f(x)");
ylim([0 0.5]);
set(gca,'fontsize',20);
grid;
hold off;
```

2. T 分配:

在機率論和統計學中，司徒頓 **t**-分布（*Student's t-distribution*）可簡稱為**t** 分布，用於根據小樣本來估計呈常態分布且變異數未知的總體的平均值。如果總體變異數已知（例如在樣本數量足夠多時），則應該用常態分布來估計總體均值。

在上述例子中提及，常態分配在 $\mu = 0, \sigma = 1$ 的情況下，又可稱作 **標準常態分配**，而當 T 分配中，**自由度**在 30 以上時，圖形呈現非常趨近標準常態分配，如此函數關係，我們可以簡單的

從 MATLAB 中觀察出來，但在這之前，我們得先對於 T 分配的函數圖形有所了解，並且熟悉參數(自由度)改變時，T 分配會如何變化，而圖 3.3 是當自由度由 0.1 至 1 時的變化：

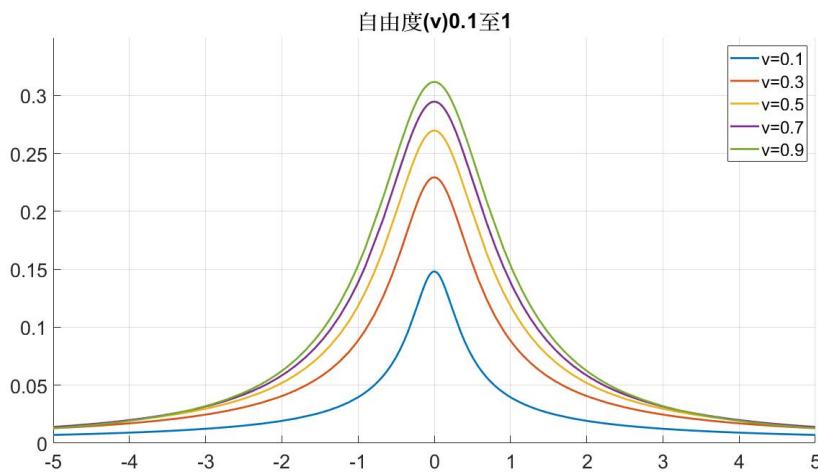


圖 3.3: T 分配 (自由度由 0.1 至 1)

而我們一樣由迴圈可以簡單完成不同參數的 T 分配，其中此分配的 (**KEY WORD**) 為 **tpdf**，而其在 MATLAB 上程式碼如下：

MATLAB 語法:

```
figure, hold on;grid;
title("自由度 (v)0.1 至 1");sInterval=[-5 5];
nu=[0.1:0.2:1]; n=length(nu);
for i=1:n
    f=@(x) tpdf(x,nu(i));
    fplot(f,sInterval,'LineWidth',2);
end
legend('v=0.1','v=0.3','v=0.5','v=0.7','v=0.9');
set(gca,'fontsize',20);hold off
```

`nu` 在此例為自由度的參數名稱，因此，由圖 3.3 我們熟悉了 T 分配，當自由度越大時，分配資料越往中心點趨近，而我們更有興趣在自由度為 30 或自由度更大時，T 分配和標準常態分配在圖形上的差異，而以 MATLAB 做圖形上驗證，如圖 3.4：

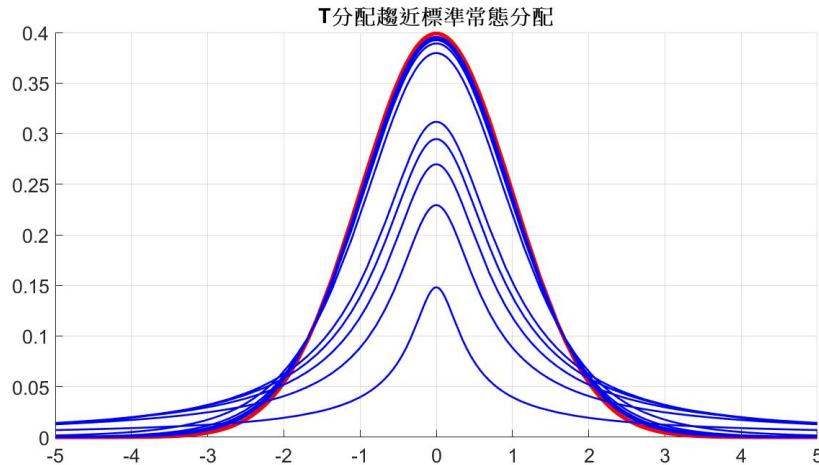


圖 3.4: T 分配趨近標準常態分配)

圖 3.4 中，紅色線段為標準常態分配，而藍色線段為 T 分配中，自由度由 0.1 至 30 的變化，可看出當自由度越大時，分配圖形越接近標準常態分配的圖形，也驗證統計學內所描述的內容。

3. 卡方分配:

卡方分布 (*chi-square distribution*)，或寫作 χ^2 分布是機率論與統計學中常用的一種機率分布。 k 個獨立的標準常態分布變數的平方和服從自由度為 k 的卡方分布。卡方分布是一種特殊的 **伽瑪分布**，是統計推斷中應用最為廣泛的機率分布之一，例如**假設檢驗**和**信賴區間**的計算。

令 X 為一連續隨機變數，若 X 符合卡方分布，且在自由度為 k 的情形下，其機率密度分布函數 (P.D.F) 為：

$$f_k(x) = \frac{\frac{1}{2}^{\frac{k}{2}}}{\gamma(\frac{k}{2})} x^{\frac{k}{2}-1} e^{-\frac{x}{2}}$$

卡方分布的機率密度函數圖形：

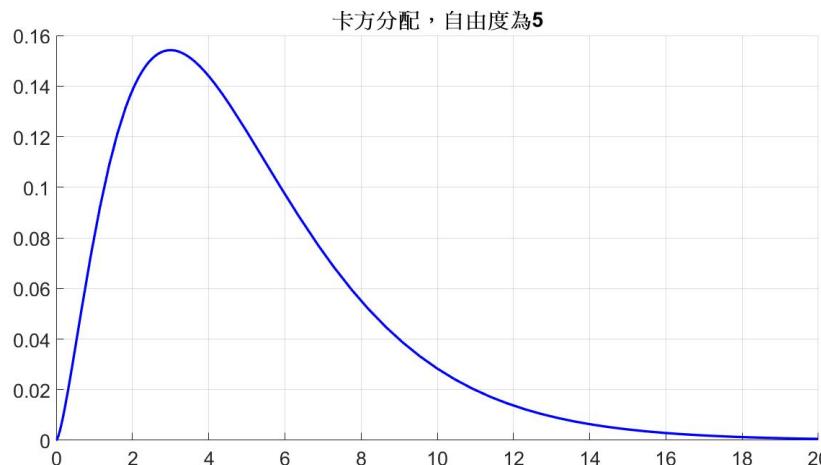


圖 3.5: 卡方分配，自由度為 5

圖 3.5 可看出，卡方分配在正常的情況下，為一右偏機率密度函數，而繪製此圖，利用 MATLAB 中的 **chi2pdf** 即可完成，如下所示：

MATLAB 語法:

```
figure, hold on;
title("卡方分配，自由度為 5")
xInterval = [0 20];grid;
f=@(x) chi2pdf(x,5);
fplot(f,xInterval,'LineWidth',2.5,'color','blue');
set(gca,'fontsize',20);hold off
```

其中，我們知道卡方是有標準常態分配的平方轉換而成的分布，在數學上恆正，因此我們只關心在 X 軸中，大於 0 的部分，而此例中，參數(自由度)設為 5，為一特殊例子，我們更有興趣在自由度不同時，此分配會有何表現，因此同樣利用 MATLAB，實驗結果如圖 3.6：

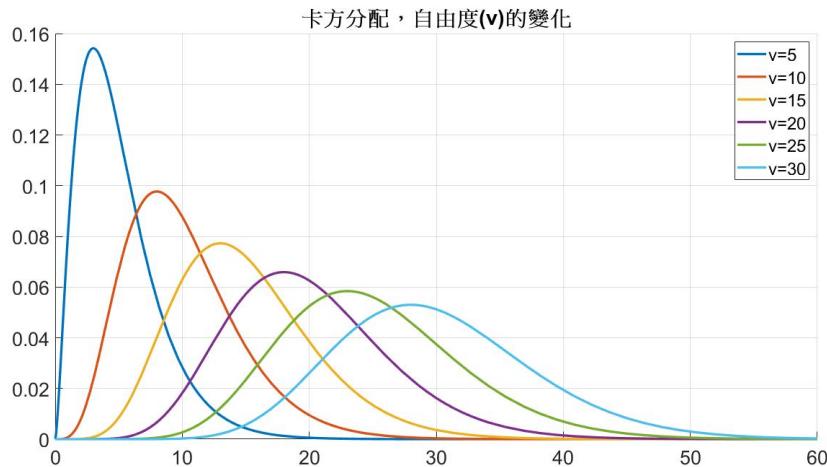


圖 3.6: 卡方分配，自由度 (v) 的變化

其中，程式語法如下：

MATLAB 語法:

```

figure, hold on;
title("卡方分配，自由度 (v) 的變化")
nu=[5:5:30]; n=length(nu);
xInterval = [0 max(nu)*2];
for i=1:n
    f=@(x) chi2pdf(x,nu(i));
    fplot(f,xInterval,'LineWidth',2.5);
end
legend('v=5','v=10','v=15','v=20','v=25','v=30');
grid;
set(gca,'fontsize',20);
hold off

```

由圖 3.6 可看出，卡方分配的自由度越大時，整個分配逐漸向右移動，由右偏的機率密度函數，隨著自由度增加，逐漸趨近左偏的機率密度函數。

4. F 分配:

F 分布是美國統計學家為了彰顯英國統計學家費雪對統計的貢獻，以費雪名字開頭的字母，當作這類型分布的名稱。以 F 分布為基礎，所衍生出的檢定方法，如 [變方分析](#) 中的 **F 檢定** 及兩族群 [變方相等性檢定](#) 等，都是各領域的學者經常使用的統計檢定。

而此分布也是由上述中的卡方分布衍伸而來，在兩卡方分布中，各自除去自己的自由度後再相除，即得到 F 分配，因此 F 分配和卡方相同皆恆正，而此特殊分布圖形如圖 3.7：

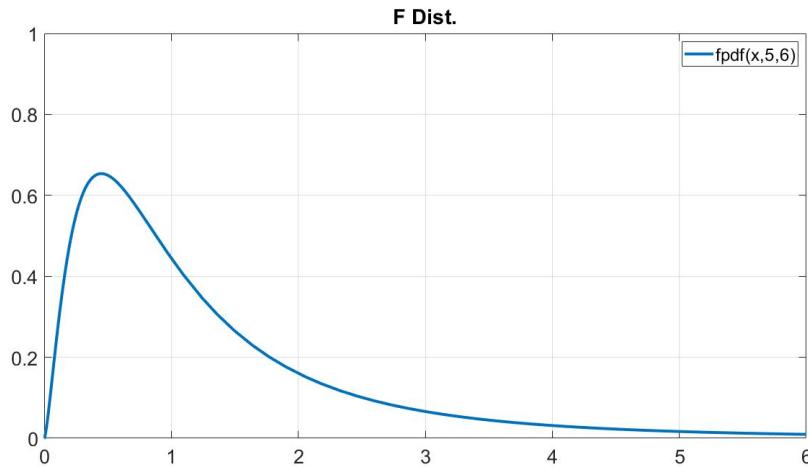


圖 3.7: F 分配，自由度為 (5,6)

其中，程式語法如下：

MATLAB 語法:

```
f = @(x) fpdf(x,5,6);
fplot(f,[0,6],'LineWidth',3);
ylim([0 1]);
grid;
title("F Dist.");
set(gca,'fontsize',20);
legend;
```

我們知道，F 分配藉由兩項自由度所控制，圖 3.7 設定為 (5, 6) 為一特殊例子，我們更有興趣的是，當自由度有不同變化時，F 分配會有何表現，因此，透過 MATLAB 實驗，分成以下 3 種討論形式：

- (a) 自由度 (1) > 自由度 (2)

我們假設第一個自由度大於第二個自由度，並觀察當第一個自由度越大時，函數圖形會有何種表現，如圖 3.8：

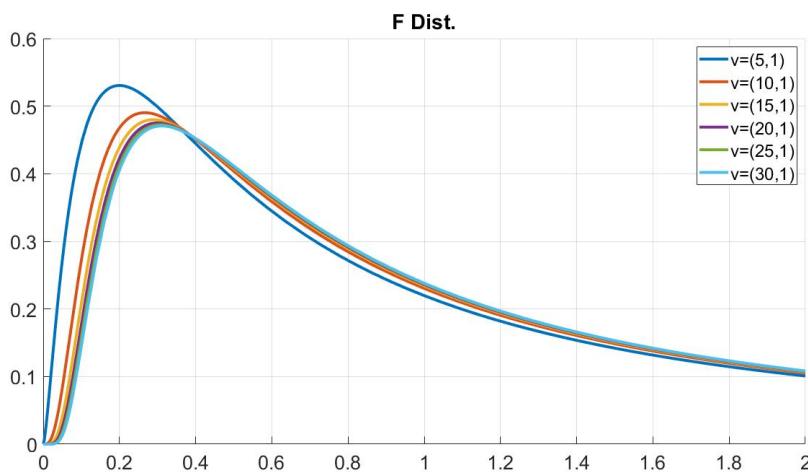


圖 3.8: F 分配，自由度 (1)> 自由度 (2)

由圖 3.8 可觀察出，當第一項自由度越大時，分配會逐漸向座標軸的右下所移動。

(b) 自由度 (1) = 自由度 (2)

我們假設第一個自由度等於第二個自由度，並另自由度由 5 至 30，觀察兩自由度相等，且同時變大時，函數圖形會有何種表現，如圖 3.9：

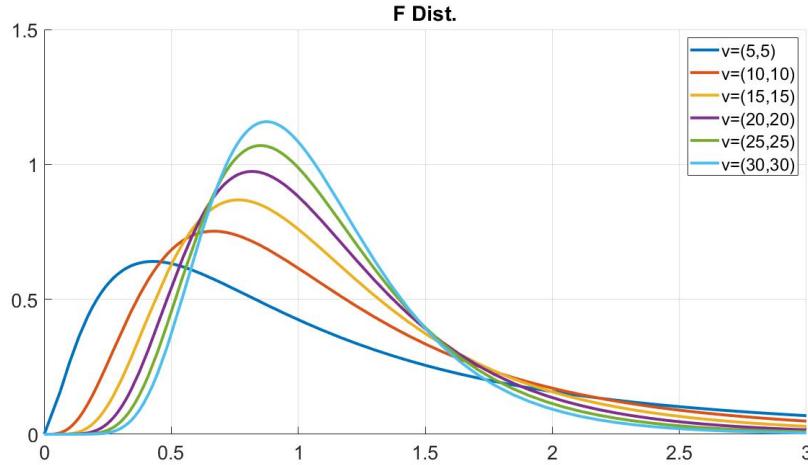


圖 3.9: F 分配，自由度 (1)= 自由度 (2)

圖 3.9可以明顯看出，函數圖形明顯趨於集中，且圖形分布逐漸向座標軸右邊移動。

(c) 自由度 (1) < 自由度 (2)

同理，我們假設第一個自由度小於第二個自由度，觀察函數圖形如圖 3.10：

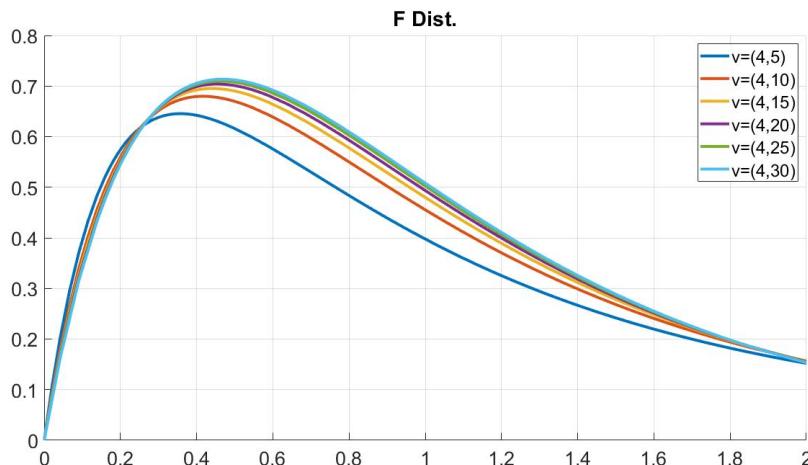


圖 3.10: F 分配，自由度 (1)< 自由度 (2)

圖 3.10 中，隨著第二個自由度增大，函數呈現向座標軸右上移動。

綜合以上三種圖形，可發現 F 分配大致分布如圖 3.11 呈現之樣貌：

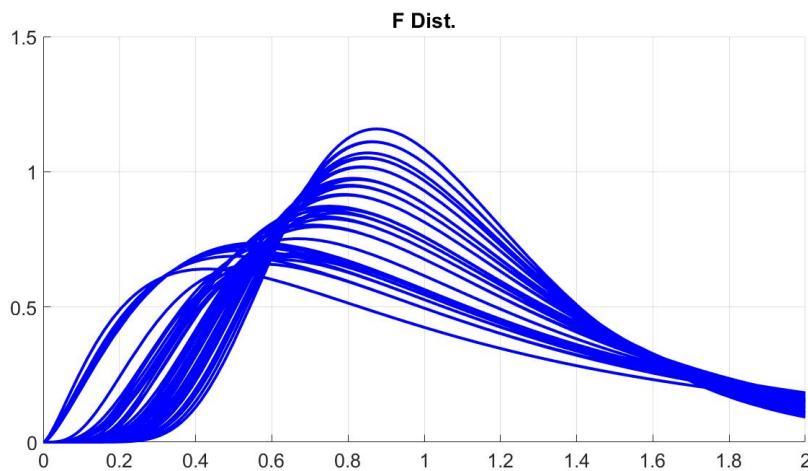


圖 3.11: F 分配

其中，程式語法如下：

MATLAB 語法:

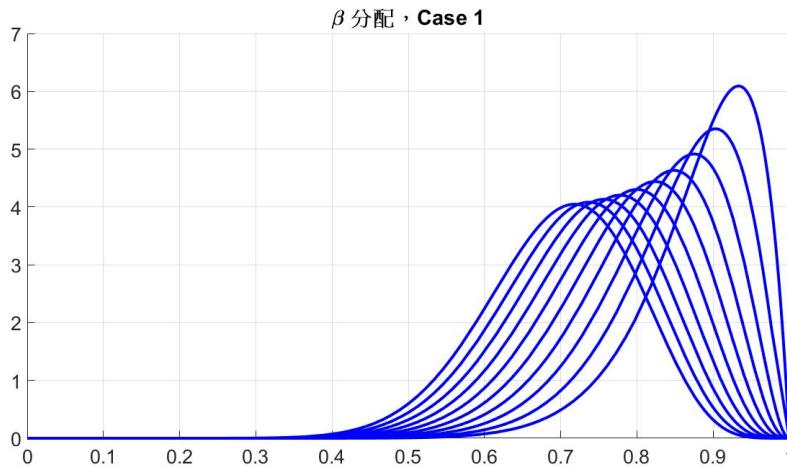
```
figure,hold on;
v1 = [5:5:30];v2 = [5:5:30];
for i=1:length(v1)
    for j=1:length(v2)
        f = @(x) fpdf(x,v1(i),v2(j));
        fplot(f,[0,2],'LineWidth',3,'color','b');
        pause(0.3);
    end
end
grid;ylim([0 1.5]);title("F Dist.");
set(gca,'fontsize',20);hold off;
```

5. 貝塔分配:

貝塔分配 (β 分配)，在函數圖形上，有各式各樣的風貌，包括左偏，右偏，甚至均勻分配，其中改變其分配形狀的主要依據，就是來自參數 α 以及 β ，我們甚至也可以知道，當 α 和 β 的大小關係改變，會造成圖形如何的變化，因此我們和上述一樣區分成 3 種討論形式：

(a) $\alpha > \beta$

我們假設 $\alpha > \beta$ ，並觀察當 β 越來越接近 α 時，函數圖形會有何種表現，如圖 3.12：

圖 3.12: β 分配 ($\alpha > \beta$)

由圖 3.12 可觀察出，當 β 越趨近 α 但同時不超過 α 時，分配中左偏傾向，會越來越不明顯，高峰逐漸向原點移動。

(b) $\alpha = \beta$

我們假設 $\alpha = \beta$ ，並另兩者同時由 1 至 9 遞增，觀察兩參數相等，且同時變大時，函數圖形會有何種表現，如圖 3.13：

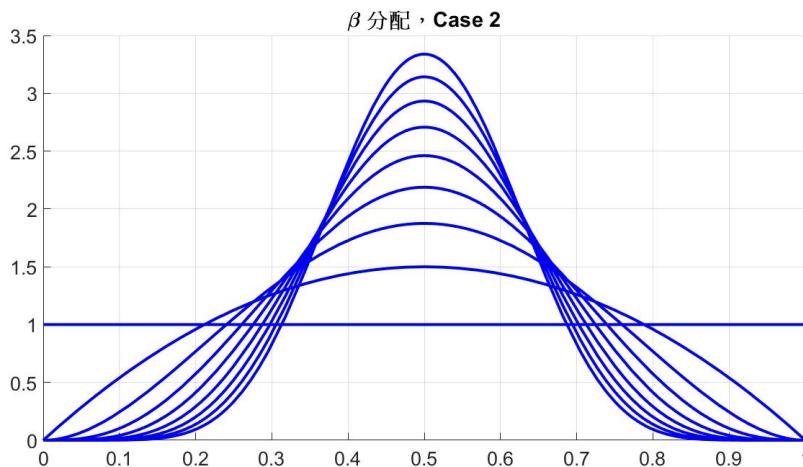
圖 3.13: β 分配 ($\alpha = \beta$)

圖 3.13可以明顯看出，當參數同時為 1 時，圖形形成均勻分配，而參數同時遞增時，函數圖形明顯向中心點集中，並且對稱。

(c) $\alpha < \beta$

同理，我們假設 $\alpha < \beta$ ，並且同時遞增 β 讓兩參數差距逐漸拉大，觀察函數圖形如圖 3.14：

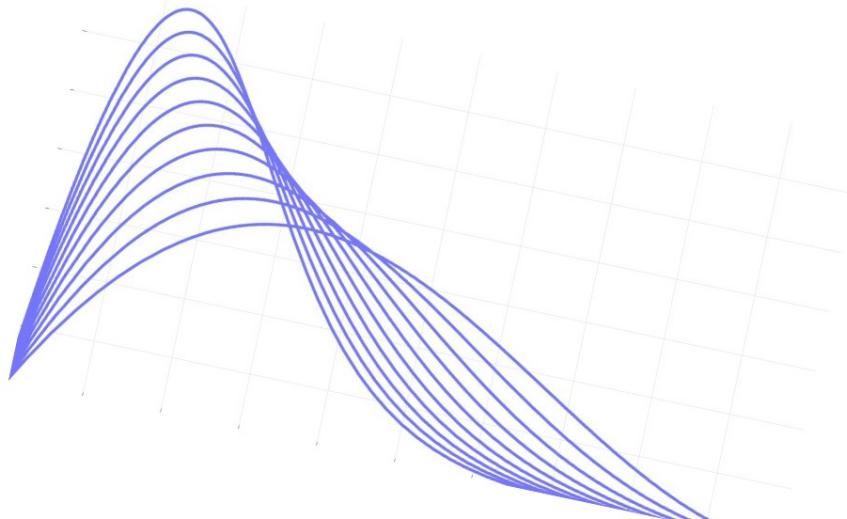
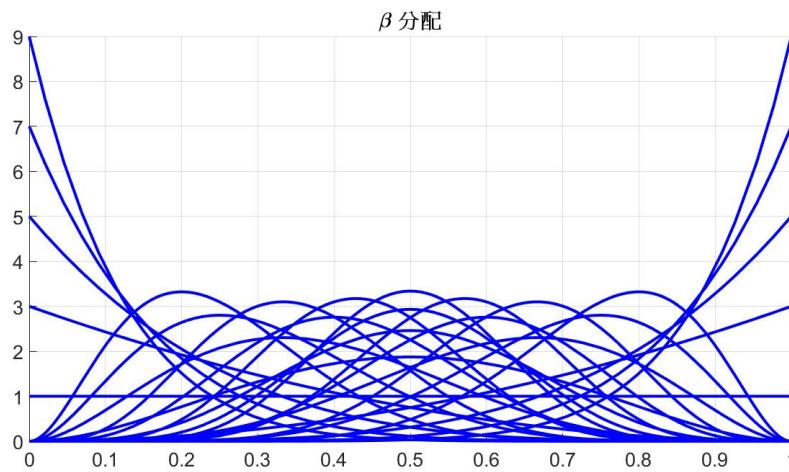


圖 3.14: β 分配 ($\alpha < \beta$)

圖 3.14中，隨著 β 增大，函數逐漸右偏，高峰逐漸向原點移動，偏態越來越明顯。綜合以上三種圖形，我們可以觀察出， β 分配中，當參數 β 愈大，且離 α 差距愈大，函數形狀愈為右偏，高峰愈往原點集中，如圖 3.14，反之，當 α 愈大，且離 β 差距愈大時，函數愈為左偏，此時高峰愈遠離原點，如圖 3.12，而當此二參數相等時，函數則是向中間集中，呈現對稱狀，如圖 3.13，而我們同時將三種圖彙整後，可得圖 3.15：

圖 3.15: β 分配

其中，程式語法如下：

MATLAB 語法:

```
figure, hold on;
alpha=1:2:9; beta=1:2:9;xInterval = [0 1];
for i=1:length(alpha)
    for u=1:length(beta)
        f=@(x) betapdf(x,alpha(i),beta(u));
        fplot(f,xInterval,'LineWidth', 3,'Color','b');
    end
end
hold off;
title("\beta 分配");ylim([0 4]);
grid;set(gca,'fontsize',20);
```

6. 伽瑪分配:

伽瑪分配，又稱 Gamma 分配，也是統計上極為常見的分配之一，包含本文提到的卡方分配，或是指數分配，都是由 Gamma 分配的參數改變而得，而 Gamma 分配有兩種參數改變其分配形狀：

- (a) α ，**形狀參數 (Shape parameter)**，影響 P.D.F 圖形之陡峭，程度。
- (b) β ，**尺度參數 (Scale parameter)**影響 P.D.F 圖形之散佈，程度。

由於 Gamma 分配一樣有兩個參數改變，我們同樣分開討論三種可能發生的情況：

- (a) $\alpha > \beta$

同樣我們假設 $\alpha > \beta$ ，並觀察當 β 越來越接近 α 時，函數圖形會有何種表現，如圖 3.16：

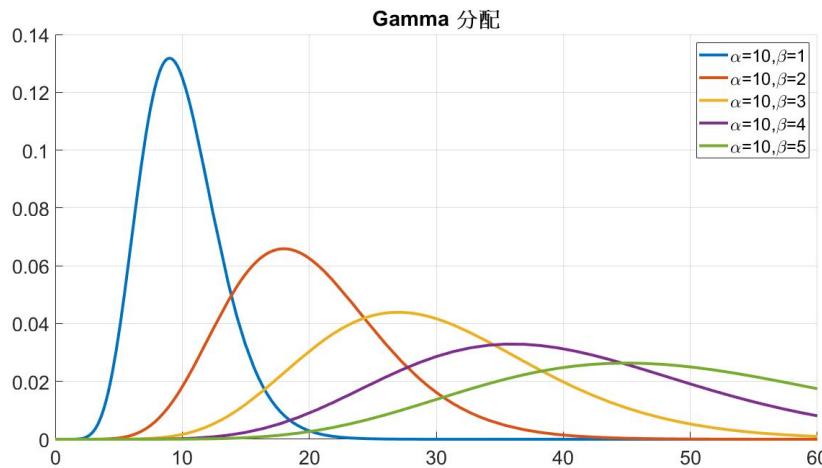


圖 3.16: Gamma 分配 ($\alpha > \beta$)

由圖 3.16 可觀察出，當控制陡峭程度的形狀參數 α 愈大於尺度參數 β 時，圖形和理論相同，愈為陡峭，而當 β 愈接近 α

時，圖形則欲趨近平緩。

(b) $\alpha = \beta$

接著我們測試當 $\alpha = \beta$ 時，並令兩者同時由 2 至 6 變化，觀察函數圖形會有何種表現，如圖 3.17：

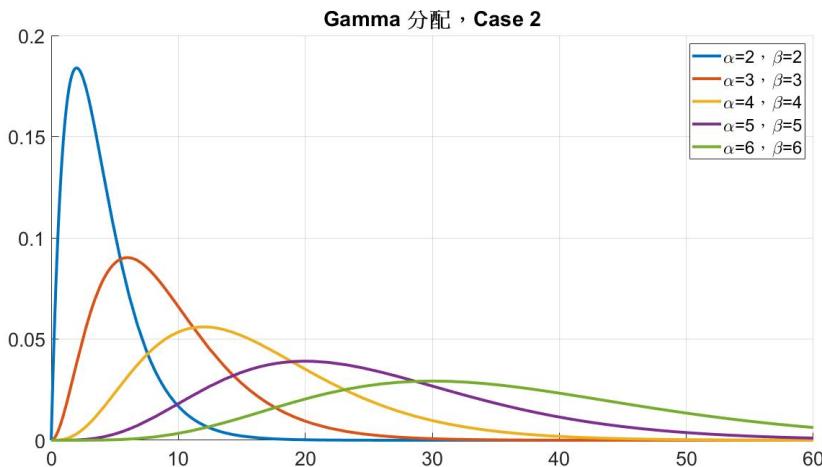
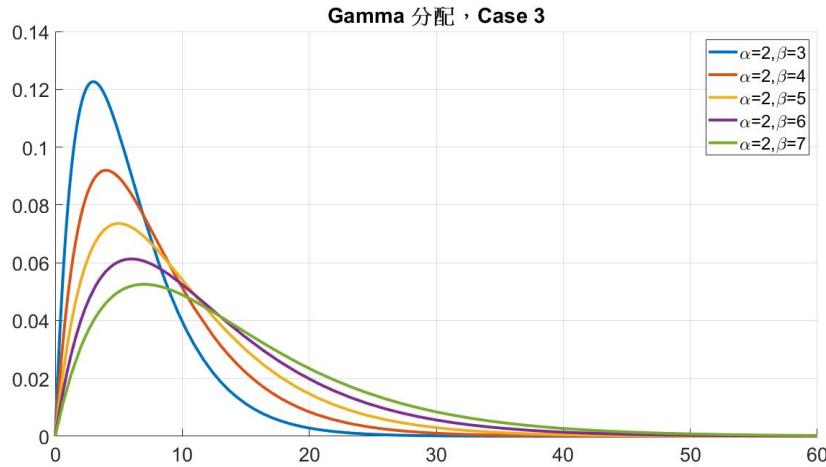


圖 3.17: Gamma 分配 ($\alpha = \beta$)

由圖 3.17 可觀察出，雖然兩參數相等，但當兩者同時增大時，函數一樣會由陡峭趨近平緩，而右偏傾向也愈來愈不明顯。

(c) $\alpha < \beta$

最後我們測試當 $\alpha < \beta$ 時，並讓 β 逐漸增大，遠離 α ，觀察函數圖形會有何種表現，如圖 3.18：

圖 3.18: Gamma 分配 ($\alpha < \beta$)

由圖 3.18 可觀察出，圖形變化上與前兩者差異不大，同樣利用 β 參數改變，亦可變化圖形至平緩。

由三種可能情況，我們同樣可以歸納出最終 Gamma 分配所有形狀的組合，大致呈現如圖 3.19：

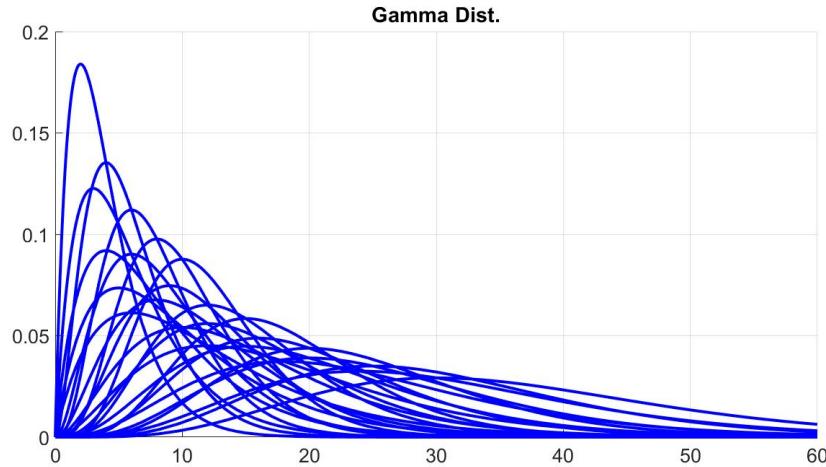


圖 3.19: Gamma 分配

有別於貝塔分配，Gamma 分配無論參數大小變化如何，都不容

易讓函數圖形由右偏至左偏，而在此一樣整理出 MATLAB 語法提供參考：

MATLAB 語法:

```
figure,hold on;
v1 = [2:6];
v2 = [2:6];
for i=1:length(v1)
    for j=1:length(v2)
        f = @(x) gampdf(x,v1(i),v2(j));
        fplot(f,[0,60],’LineWidth’,3,’color’,’b’);
    end
end
grid;
title(”Gamma Dist.”);
set(gca,’fontsize’,20);
hold off;
```

3.1.2 離散型函數

同理，在離散型函數在本文中亦可稱作離散型隨機變數，而日常生活中一樣有許多離散型隨機變數的例子，最簡單的擲銅板就是在統計上典型的例子之一，而為了展現 MATLAB 在離散型函數圖形上的發揮，我們一樣利用些例子加以說明：

1. 二項分配：

在機率論和統計學中，**二項分布 (Binomial distribution)** 是 n 個獨立的是/非試驗中成功的次數的離散機率分布，其中每次試驗的

成功機率為 p 。這樣的單次成功/失敗試驗又稱為 **伯努利試驗**。實際上，當 $n = 1$ 時，二項分布就是伯努利分布，而在此我們利用簡單的直方圖來探討二項分配在圖形上呈現的樣貌：

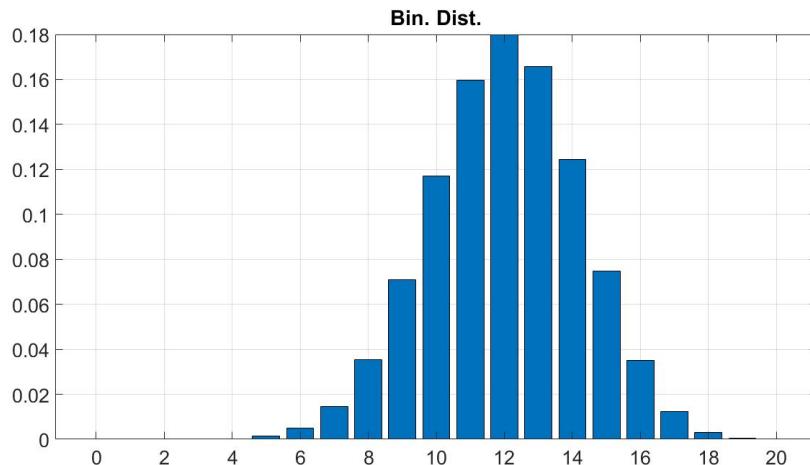


圖 3.20: 二項分配之直方圖

上述提到，二項分配是觀察 n 次試驗中，在成功機率為 p 之情形下的試驗結果，因此我們得知，二項分配的圖形藉由兩個參數 n 、 p 來改變其分配樣貌，而由圖 3.20 可看出，在我們令 n 為 20 且 p 為 0.6 的情形下，成功次數發生在 12 次的機率最高，而其中，12，即是二項分配在 $n = 20, p = 0.6$ 中理論上的平均值 (期望值)，藉由 MATLAB 實作圖形後，我們也能得確定實際與理論上並無差別。

然而，在離散型分配上，我們有時候也關心其**莖葉圖** (**Stem plot**)，如圖 3.21：

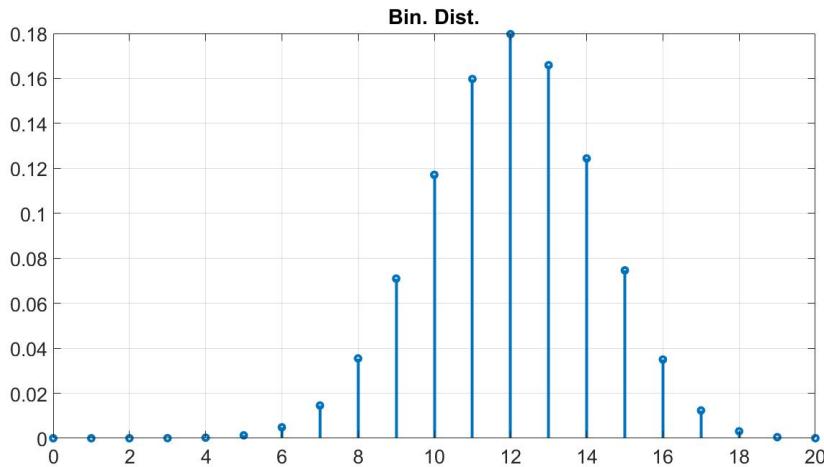


圖 3.21: 二項分配之莖葉圖

其 MATLAB 語法透過 **stem** 展現，而二項分配則是利用 **binopdf** 的函數來實踐，程式碼如下：

MATLAB 語法:

```
N=20;p=0.6;
x=0:N;
y=binopdf(x,N,p);
fig=stem(x,y);
fig.LineWidth=3;
grid;
title("Bin. Dist.");
set(gca,'fontsize',20);
```

最後，當我們觀察離散型分配函數時，次數相對成為重要的事情之一，而累積次數在特定時候又是我們希望知道的事情，因此我們可以透過 **stairs** 呈現其階梯圖，利用累積機率密度函數觀察離

散分配會有何種表現，如圖 3.22：

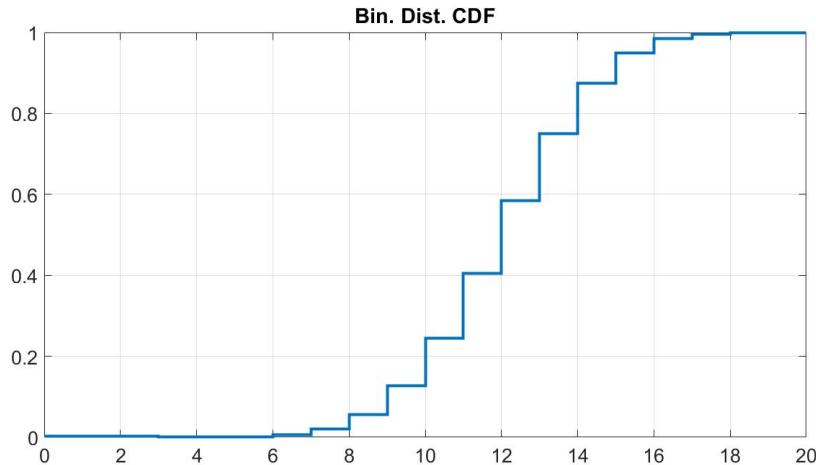


圖 3.22: 二項分配之階梯圖

透過圖 3.22 可知，理論上會有將近 8 成的實驗，成功次數都在 13 次以下，而成功次數超過 15 次的實驗，不超過 1 成，而呈現累積機率密度函數，與莖葉圖的語法如下：

MATLAB 語法:

```
N=20;p=0.6;  
x=0:N;  
y=binocdf(x,N,p);  
fig=stairs(x,y);  
fig.LineWidth=3;  
grid;  
title("Bin. Dist. CDF");  
set(gca,'fontsize',20);
```

將原本的 **binopdf** 改成 **binocdf** 即可得到累積機率密度函數，而透過莖葉圖與階梯圖，我們更容易觀察出在離散型函數分配上，簡單的敘述統計量與其分配特性等等。

2. Poisson 分配：

最後，我們在分配函數中，討論 Poisson 分配。常常在電視上聽到某某交通法規實施或嚴格取締交通違規行動後，道路上每月發上車禍的次數明顯減少了。但是所謂的「明顯減少」是怎麼判斷的呢，每個月都會偶爾發生一些意外事故，事件數量到底要有多大的改變才算是明顯的變化？

或者家裡附近的警察每兩個小時固定會出來巡邏一次，那麼家裡門前在兩個小時之內都沒有任何警察經過的機率是多少呢？這些問題都可以仰賴 Poisson 分配來解決。假設某區域單位時間之內平均事件發生次數為 λ ，那麼在這區域中事件發生的次數 X 就符合 Poisson 分配。還有許多日常生活中週遭的現象也符合 Poisson 分配，例如：每小時進入學校大門口的人數、隔壁麵店每小時的客人數量、每次紅綠燈之間的車流量等等。¹

以下是 Poisson 分布的數學式：

Poisson 分布只有一個參數，單位時間平均事件發生次數 λ 。令 X 為一離散隨機變數，若 X 符合 Poisson 分布，其機率密度分布函數為 (P.D.F) 為：

$$f(x) = \frac{e^{-\lambda} \lambda^x}{x!}, \quad x = 0, 1, 2, \dots$$

而其函數圖形如圖 3.23：

¹資料來源：<http://www.agron.ntu.edu.tw/biostat/Poisson.html>

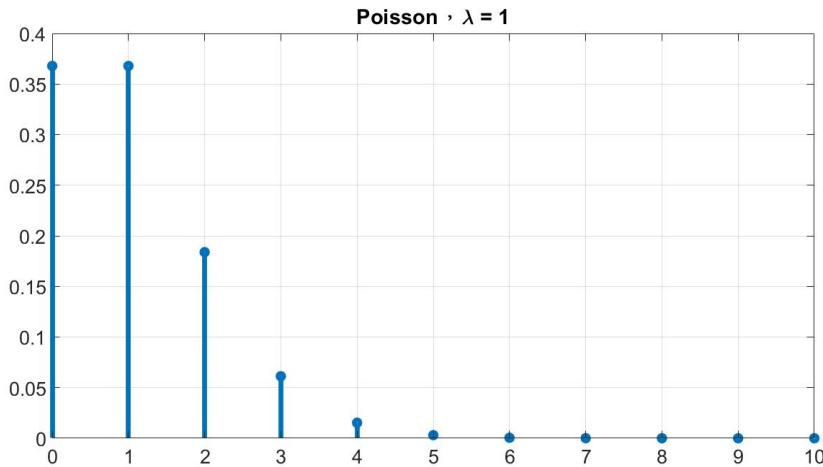


圖 3.23: Poisson 分配之莖葉圖

上述提到已知警察平均每兩個小時巡邏一次。那麼家裡門前在兩個小時內出現警察次數 (x) 的機率圖形如圖 3.23，可簡單觀察出，2 小時內出現 0 次警察的機率約為 0.37 左右，那在此題延伸，若是想知道出現 1 次以上，2 次以上等等呢？我們可以藉由累積機率密度函數輔佐解決此問題，如圖 3.24：

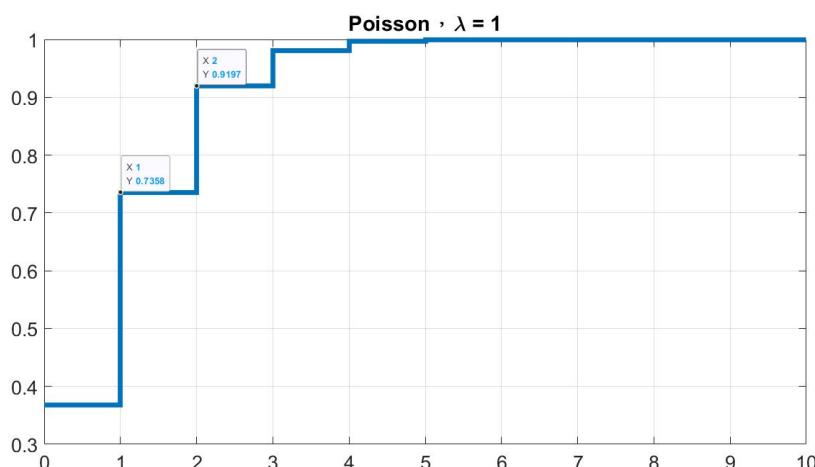


圖 3.24: Poisson 分配之階梯圖

由圖 3.24即可得知，警察出現超過 1 次的機率已經不到 3 成，而大於 2 次的機率甚至不到 1 成，因此透過累積機率密度圖，亦可解決其他繁瑣問題。

其中語法如下：

MATLAB 語法:

```
x=[0:10];
y=poisscdf(x,1);
h=stairs(x,y);
h.LineWidth=5;
grid;
title("Poisson , \lambda = 1");
set(gca,'fontsize',20);
```

最後，整理出幾個在參數 λ 不同的情形下，Poisson 的各種形狀表現，如圖 3.25：

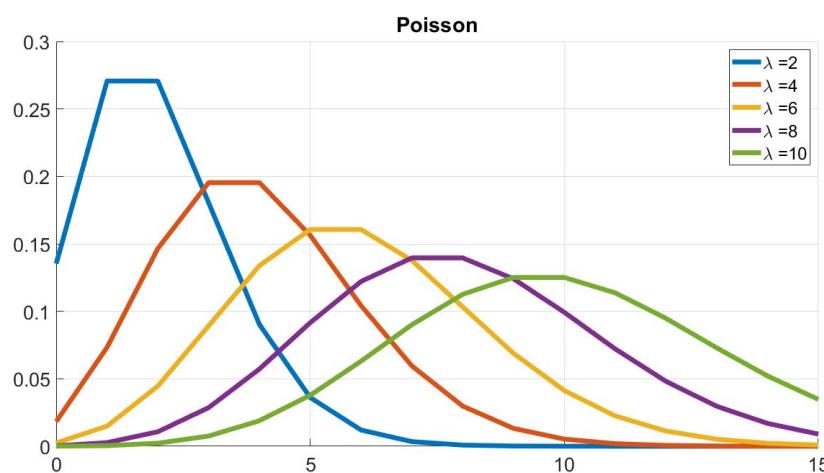


圖 3.25: Poisson 分配之多種形態

圖 3.25 中， λ 愈大，圖形愈趨近平緩，由於 λ 為平均數，亦為變異數，因此隨著 λ 愈大高峰點也愈大，圖形也愈矮胖，而其語法如下：

MATLAB 語法:

```
x=[0:15];lam=[2:2:10];
figure, hold on;
for i=1:length(lam)
    y=poisspdf(x, lam(i));
    h=plot(x,y);
    h.LineWidth=5;
end
grid;title("Poisson");
set(gca,'fontsize',20);
legend(string("\lambda ="+lam));
hold off;
```

3.2 亂數產生與相關圖形

亂數是一連串獨立數字的組成，在此章的亂數，卻也是有規律的一串數字，我們將討論到如何從特定分配中抽一連串亂數，並做圖形分析，敘述統計量之討論，母體推論與驗證，樣本數大小的差異等等，而利用 MATLAB 解決諸如此類的問題，而以下將舉幾個例子做討論：

1. 常態分配之亂數產生：

上節我們已經了解常態分配的圖形，以及其參數變化後，函數形狀會如何改變，而這節我們一樣透過最常見的常態分配，隨機產生亂數，並進一步探討，而在產生亂數之前，我們釐清一個問

題：

「亂數要產生多少個才夠？」，而這個問題也直接牽扯到樣本數大小，我們從常態分配中產生亂數，要產生多少個亂數，這些樣本才會足夠像母體呢？以下我們也利用程式解決此問題：

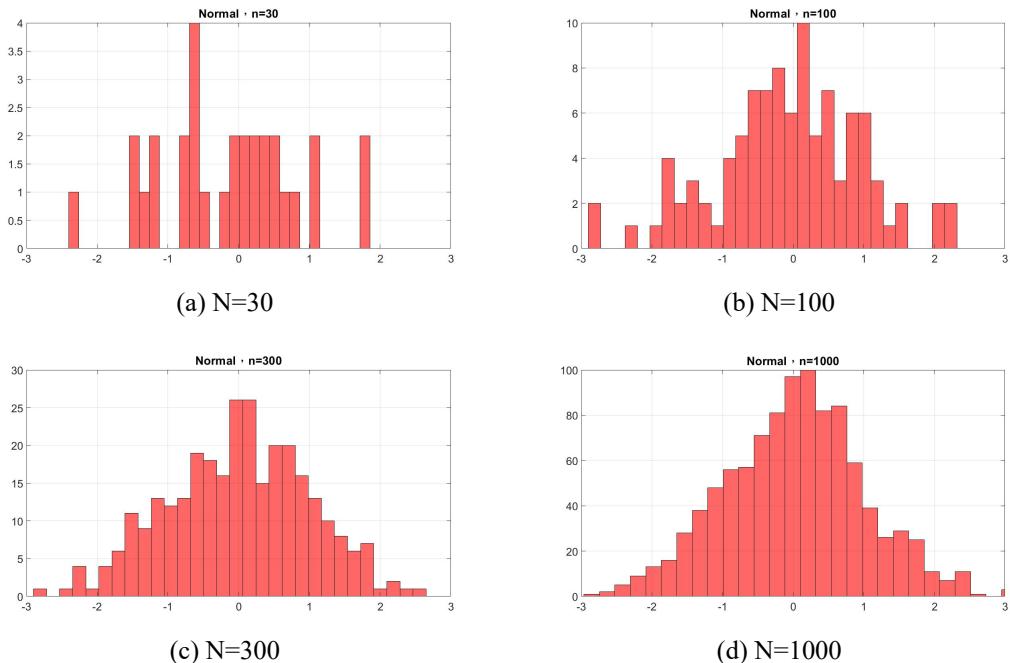


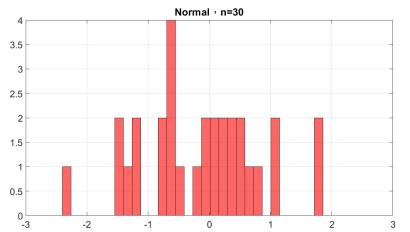
圖 3.26: 樣本數大小之差異

其中僅需改變 N 的大小，就能產生不同樣貌的常態分配樣本圖形，程式碼如下：

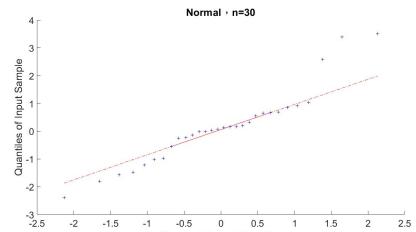
MATLAB 語法:

```
n=30;x=normrnd(0,1,1,n);
h=histogram(x);
h.NumBins=30;h.FaceColor='red';
xlim([-3,3]);grid;set(gca,'fontsize',20);
title(string("Normal , n=" + n));
```

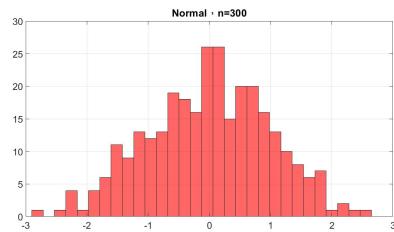
由圖 3.26 可看出，當樣本數愈大時，圖形才會愈明顯像常態分佈，而若是以 qqplot 圖來觀察的話，可以更明顯看出其中差異，如圖 3.27：



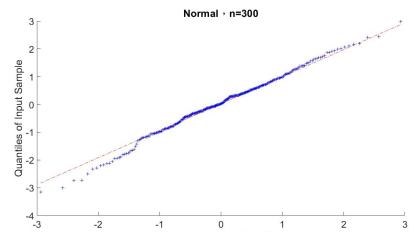
(a) N=30



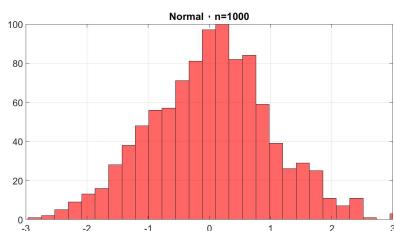
(b) qqplot , N=30



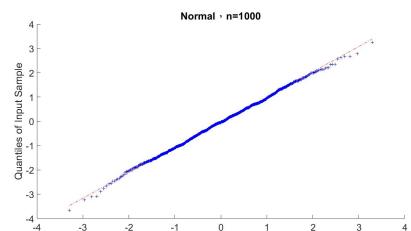
(c) N=300



(d) qqplot , N=300



(e) N=1000



(f) qqplot , N=1000

圖 3.27: 樣本數大小與 qqplot 比較

圖 3.27我們已經能觀察出，樣本數超過 300 時已經夠像常態分配的形狀，當樣本數 1000 時，qqplot 幾乎吻合常態分配該有的樣貌。

最後我們能夠透過 boxplot 來觀察我們產生亂數後的一些敘述統計量，亦能看出分配大至偏態形狀等等，如圖 3.28；

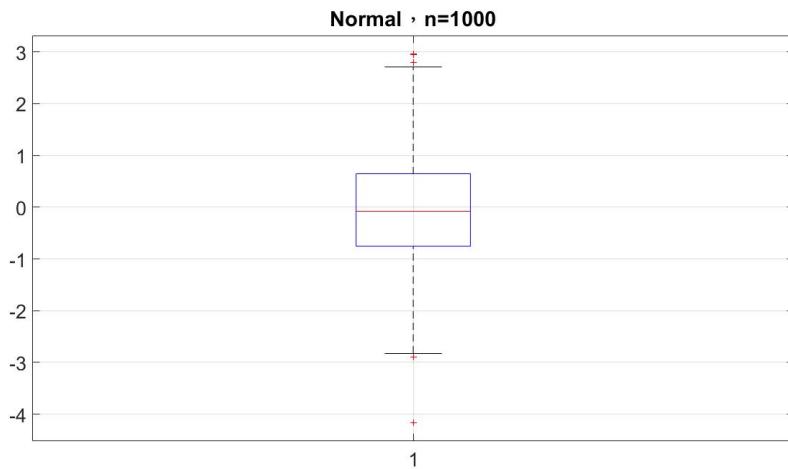


圖 3.28: 常態分配

由此也能看出，圖形大致呈現對稱狀，中位數落在 0 的位置左右，而也有幾個 outlier 在圖中，其程式碼如下：

MATLAB 語法:

```
n=1000;
x=normrnd(0,1,1,n);
boxplot(x);
grid;
title(string("Normal , n="+n));
set(gca,'fontsize',20);
```

2. 卡方分配之亂數產生：

由上例可知，樣本數愈大，所產生的樣本分配愈趨近於母體，而卡方分配一樣如此，然而，在程式繪圖中，仍有些參數設定須注意，例如 **NumBins**，若切割太少，則圖形顯示上不足以具代表性，因此切割數也不能過低：如圖 3.29：

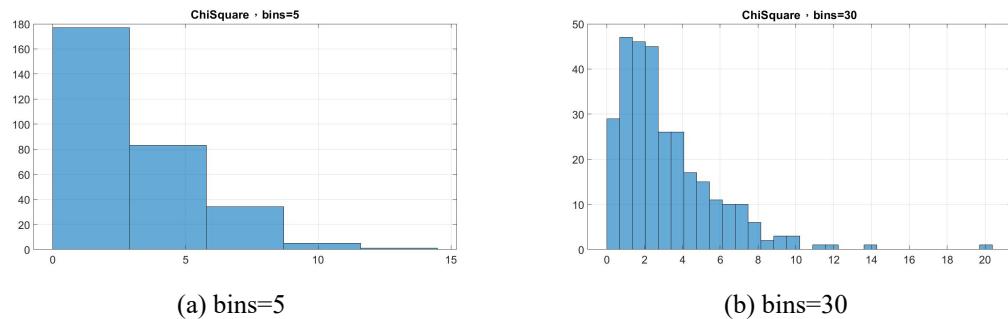


圖 3.29: 卡方分配，bins 大小差異

而當我們最後決定好 bins 以及樣本數後，即可繪製卡方圖形，並且以理論的卡方分配線來配適此樣本分配之直方圖，看兩者之間是否吻合，如圖 3.30：

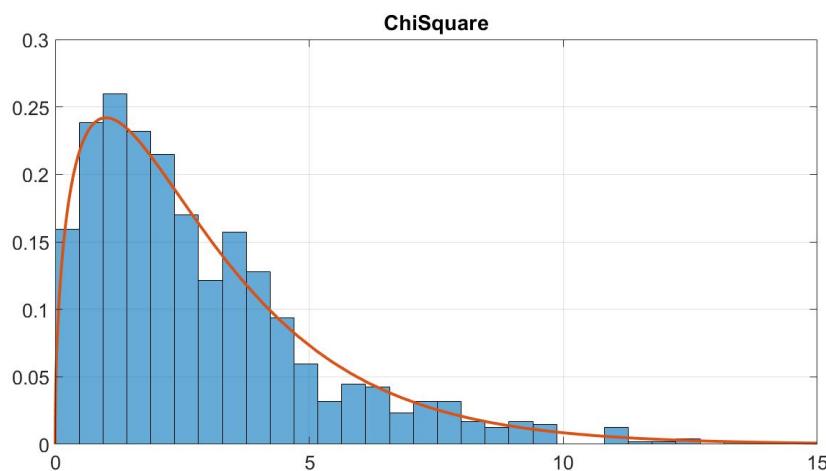


圖 3.30: 卡方分配與理論值之配適

圖 3.30看出，理論的卡方分配 (紅線) 與我們隨機抽樣之直方圖，大致吻合，如此可以簡單驗證此樣本確實是來自卡方分配。而透過理論線段重疊樣本圖形之程式碼如下：

MATLAB 語法:

```
n=1000;  
x=chi2rnd(3,1,n);  
h=histogram(x,'Normalization','pdf');  
h.NumBins=30;  
title("ChiSquare");  
set(gca,'fontsize',20);  
hold on;  
f = @(x1) chi2pdf(x1,3);  
fplot(f,'LineWidth',3);  
hold off;  
xlim([0 15]);grid;
```

或是透過 ECDF 圖形，也能看出在累積機率密度函數中，理論值與我們抽樣值是否吻合，如圖 3.31：

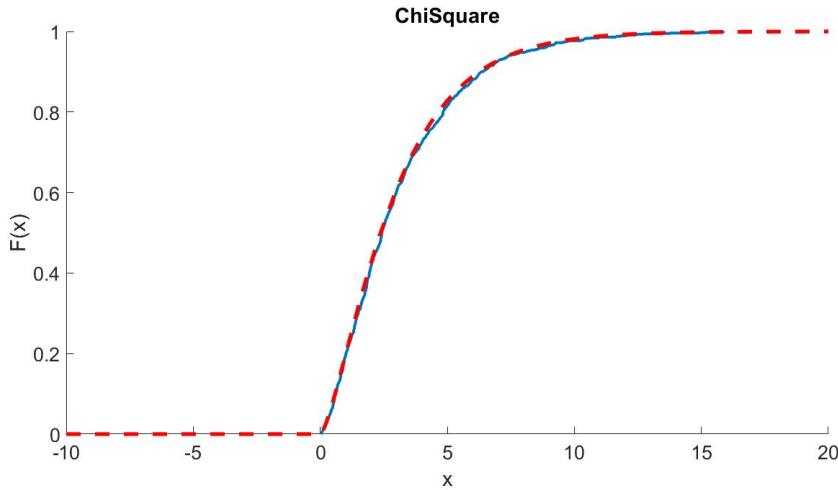


圖 3.31: 卡方分配 _ ECDF 圖

程式語法如下：

MATLAB 語法:

```
figure ,hold on;
n=1000;
x=chi2rnd(3,1,n);
h=cdfplot(x)'
h.LineWidth=3;
title("ChiSquare");
set(gca,'fontsize',20);
f = @(x1) chi2cdf(x1,3);
fplot(f,'LineStyle','-', 'Color','r','LineWidth',4);
hold off;grid;
```

圖 3.31 中，紅線為理論值，藍線為樣本，可見大致吻合，也可確定此樣本是來自卡方分配。最後，我們已經大致確定此樣本真的是來自卡方分配後，以 `qqplot` 來觀察圖形會形成怎樣形狀，驗證

其不會符合常態分配之形狀，如圖 3.32：

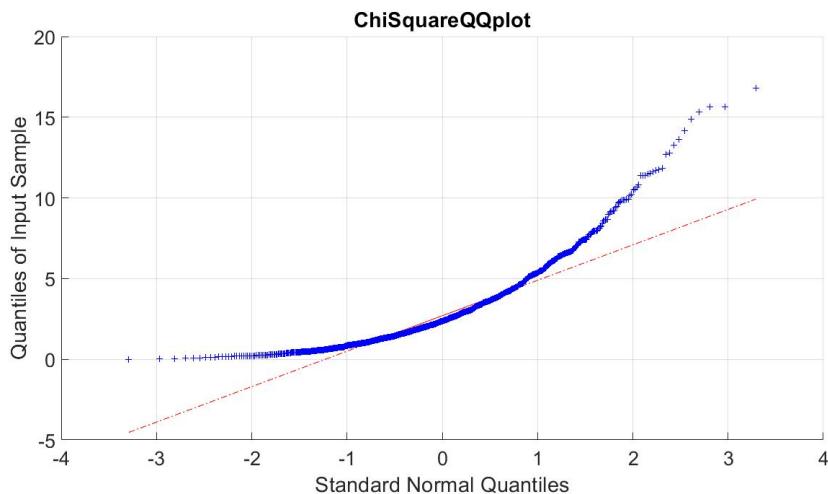


圖 3.32: 卡方分配 _ qqplot

圖 3.32明顯看出，樣本資料大致不落在 45 度線上，可見此分配確實不是常態分配。

3.3 抽樣分配

在上章節中提到透過某特定母體抽取亂數，做出多方面探討，然而，在日常生活中，有些情況卻無法直接利用母體取得後之樣本，直接做進一步分析，中間反而需經過不同種類的函數變化，而衍生出新的樣本，而此時我們更關心這樣新的樣本，會形成怎樣的分配，而此分配又可稱作抽樣分配，在此我們除了討論抽樣分配外，也透過程式演練，驗證過去所學的理論，與如今實際操演的結果，是否雷同，因此以下也舉幾個例子研究此問題：

1. 中央極限定理

中央極限定理是機率論中的一組定理。中央極限定理說明，在適當的條件下，大量相互獨立隨機變數的平均數經適當標準化後依

分布收斂於常態分布。這組定理是數理統計和誤差分析的理論基礎，指出了大量隨機變數之抽樣分配近似服從常態分布的條件。為了實踐中央極限定理，我們假設從二項分配中抽取 n 組樣本，

計算其平均數，並且重複實驗 1000 次 ($N = 1000$)，觀察在不同數量之樣本底下，抽樣分配最後收斂的結果，如圖 3.33：

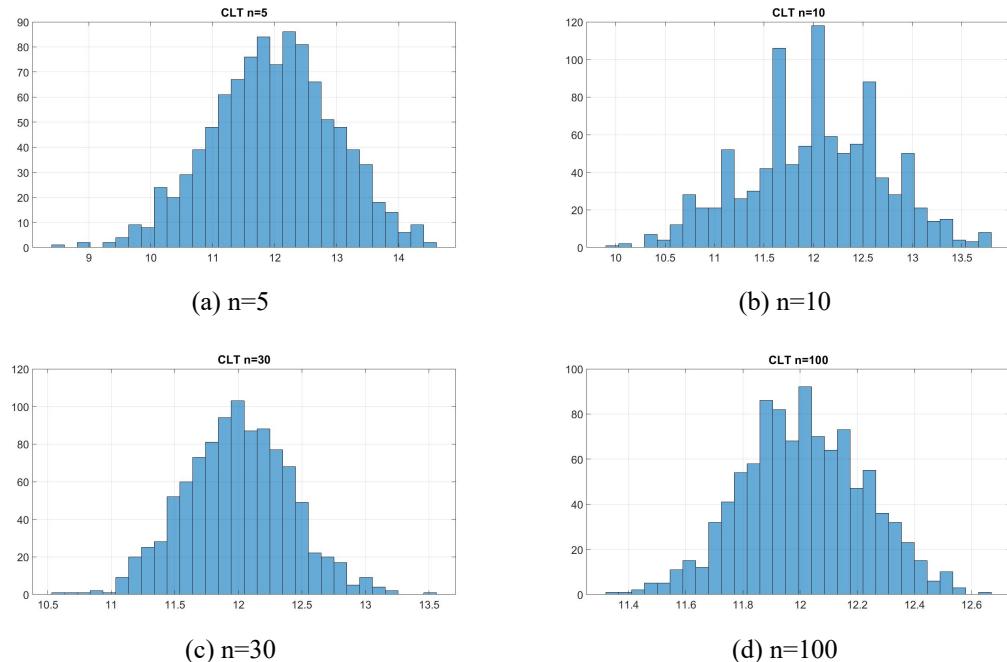


圖 3.33: CLT 樣本大小差異

由圖 3.33 可看出，在 $n = 5$ 時圖形還有左偏傾向，但到了 $n = 30$ 圖形大致已形成對稱的鐘形分布，而仔細觀察 x 座標，可以發現當 n 愈大，樣本愈集中，到了 $n = 100$ 時，樣本大致已分布在 11.5 與 12.5 之間，可見樣本數愈大，變異程度愈小。

最後我們知道透過理論的常態分配曲線，來看是否此隨機樣本在經過平均的函數調整下，形成的新的分配，真的服從常態分配：

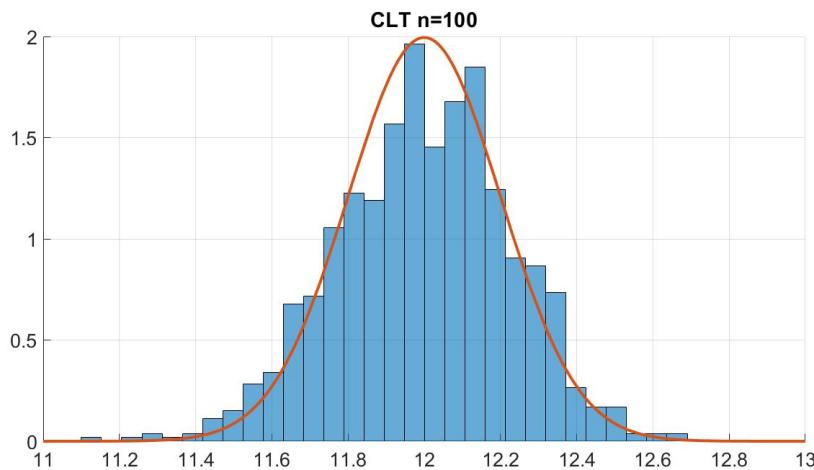


圖 3.34: 抽樣分配趨近常態分配

透過圖 3.34，看出理論曲線，和我們的抽樣分配近乎吻合，可以推測此抽樣分配來自常態分配，而也和過去所了解的中央極限定理相符，以下是透過 MATLAB 實作之程式碼：

MATLAB 語法:

```
figure,hold on;
n=100;N=1000;
x=binornd(20,0.6,n,N);
histogram(mean(x),'NumBins',30,'Normalization','pdf');
set(gca,'fontsize',20);
f = @(x1) normpdf(x1,12,0.2);
h=fplot(f);
h.LineWidth=3;
title("CLT n="+string(n));xlim([11 13]);grid;hold off;
```

2. 卡方分配的由來：

卡方分配，是由標準常態分配經過函數轉換而來，這在數理統計中時常被提及，而這項真理如何從理論實作，以下將以 MATLAB 用圖形實踐理論。同樣假設我們隨機從標準常態分配中抽樣 n 筆資料，並對其進行平方轉換，得出新的資料集，在觀測新的資料集的分配是否服從卡方分配：

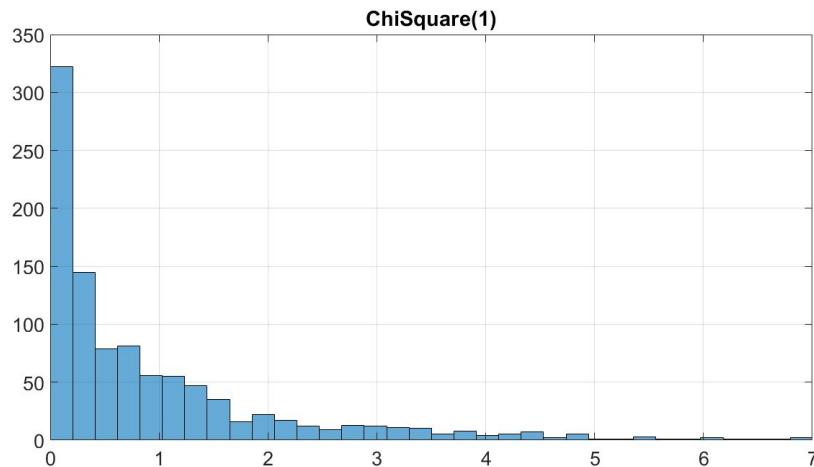


圖 3.35: Z^2 之轉換直方圖

圖 3.35形狀類似理論上的卡方分配，且自由度為 1，而我們這次透過 ECDF 圖來驗證我們得出的樣本，是否真的和理論上一樣服從卡方 1：

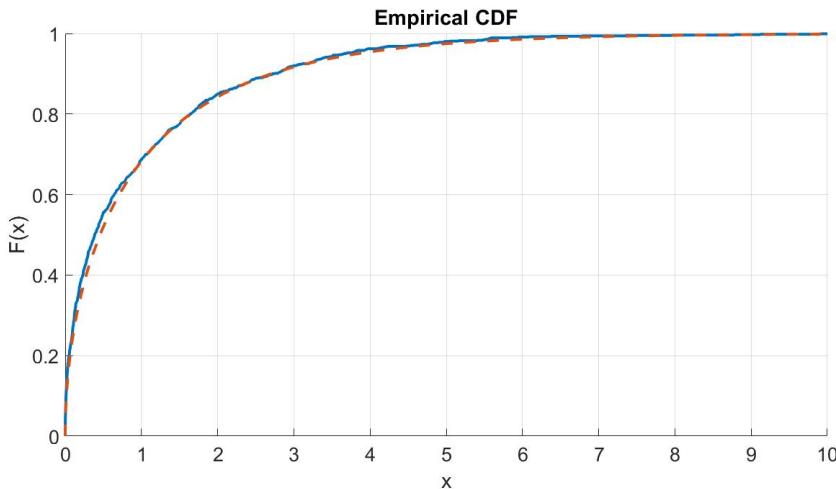
圖 3.36: Z^2 之轉換 ECDF 圖

圖 3.36 中，理論值為紅線，和樣本大致吻合，因此推測此抽樣分配最後形成卡方分配，且自由度為 1。而其中，程式碼如下：

MATLAB 語法:

```
figure,hold on;
n=1000;
x=normrnd(0,1,1,n);
newX=x.^2;
grid;xlim([0 10]);
set(gca,'fontsize',20);
h=cdfplot(newX);
h.LineWidth=3;
f = @(x1) chi2cdf(x1,1);
fplot(f,'LineWidth',3,'LineStyle','--');
hold off;
```

3. 分配可加性

最後，我們討論到分配的可加性，分配可加性亦即兩特定分配相加後，依然服從該分配，例如先前提及的卡方 1 加上卡方 1，最後依然會形成卡方分配，且自由度為 2，而如此數學理論，我們依然能利用 MATLAB 實作其抽樣分配的樣貌：

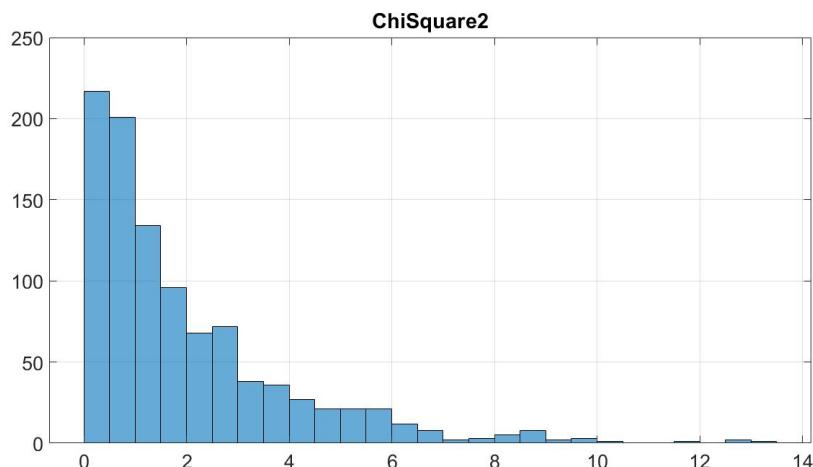


圖 3.37: 卡方分配可加性

我們透過隨機產生亂數，由卡方 1 中產生 2 組 1000 比亂數，接著對它進行相加的動作，結果顯現如圖 3.37，可以看出圖 3.37 分配形狀接近卡方分配，但我們仍須一條理論線來輔佐我們判斷，如圖 3.38；

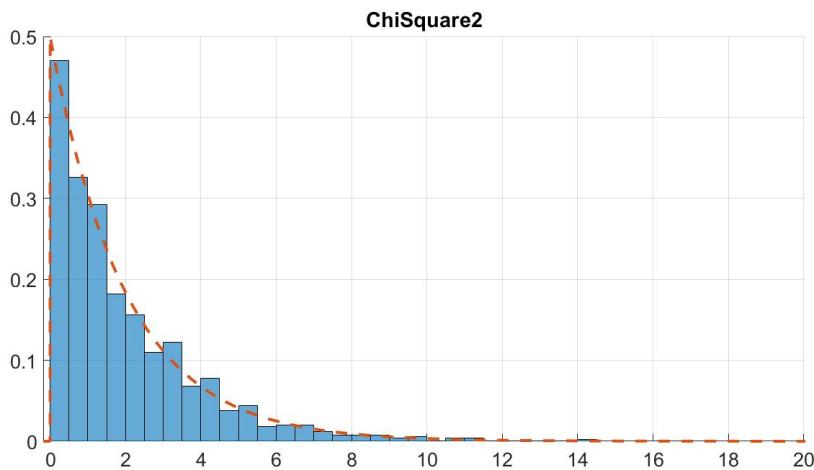


圖 3.38: 卡方分配可加性與理論線之配適

同樣透過理論值來輔佐，我們看出實際與理論極為接近，因此驗證卡方分配具有可加性，而 MATLAB 實作語法如下：

MATLAB 語法:

```
figure,hold on;
n=1000;
x1=chi2rnd(1,1,n);
x2=chi2rnd(1,1,n);
newX=x1+x2;
histogram(newX,'Normalization','pdf');
grid;xlim=[0:10];title("ChiSquare2");
set(gca,'fontsize',20);
f = @(x3) chi2pdf(x3,2);
fplot(f,'LineWidth',3,'LineStyle','-');
hold off;
```

3.4 結論

透過 MATLAB，我們了解統計分配上的各種圖形呈現，不需要手動輸入繁瑣複雜的機率密度函數，僅須記得短短幾行程式碼的語法，接著以各種圖形包括盒形圖，莖葉圖，直方圖等等呈現函數最原始的樣貌，並且 MATLAB 中亦可以從某種特定分配中，產生一連串自訂樣本數的亂數，除了對於現今樣本數少有所益處外，更能增加學習方式，而最後透過 ECDF 圖或是其他理論圖驗證此樣本來自的母體分配，對應到最初數理統計學的多種理論，搭配實用的程式語言，交錯學習中更能以圖形記憶，增加基礎觀念。

第 4 章

監督式學習之迴歸分類

監督式學習 (**Supervised learning**) 是電腦從標籤化 (labeled) 的資訊中分析模式後做出預測的學習方式。標記過的資料就好比標準答案，而本文所探討的方式，正屬於監督式學習。假設我們擁有一組資料集，我們想知道若是存在新的一筆資料，它是否也屬於此資料集，或是對此資料集進行分類，有助於判斷新的資料歸屬的類別，舉例來說，若我們能得到大量醫院資料，並找出幾項誘發高血壓的因素 (x)，是否我們就能歸類有何種 x 傾向的人，較可能罹患高血壓，而面對此問題，最重要的一點，就是找出合適的分類器，而這也是本文所要探討的，從統計的觀點建立合適的分類器。

4.1 迴歸模型

回到上述的問題中，假設我們的資料集中，有一群屬性資料 (x) 以及最後目標變數 (y)，要如何得知擁有何種 x 會形成何種 y ，而在面對此問題時，我們先簡單將資料呈現於平面圖形上：

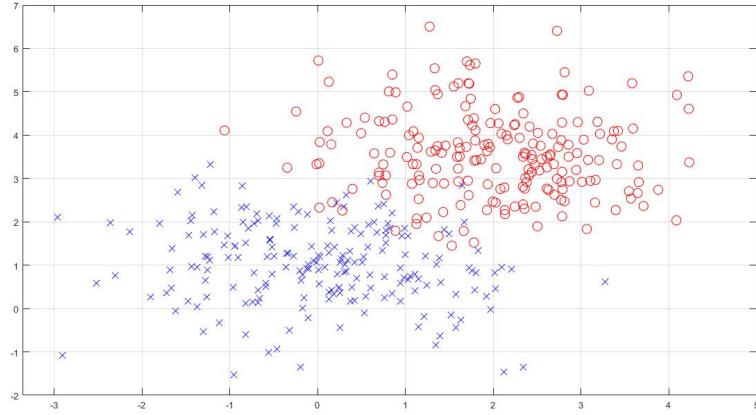


圖 4.1: 範例資料集

圖 4.1 我們假設 x 軸為變數 x_1 , y 軸為變數 x_2 , 而在此我們以顏色還有圖形簡單區分 y 變向，可以明顯的看出，圖形呈現兩種不同的類別，並彷彿能以一條線區分開此二群，但要以何種線段區分呢，本文將一一介紹。

4.1.1 簡單線性迴歸

我們首先假設資料集能以 $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2, \forall y \in \{0, 1\}$ 配適，而在此模型下，當 $y < 0.5$ 我們就區分為 0 群，當 $y > 0.5$ 就區分成 1 群，如此便可得一分類模型，而在此模型底下我們僅需估計出 β_0 、 β_1 、 β_2 即可，並且令 $y = 0.5$ 時，便是此分類器區分之依據，也是圖 4.1 中，我們所需的分類線段，而在此我們利用 MATLAB 實作幫助研究，透過模擬資料集，一步一步探討此問題：

Step 1：取得並了解資料集

在此我們透過 MATLAB 取得既有的資料集，並對此資料集先進行簡單的觀察，看有無問題，以及觀察變數大概落的位置，

或是先觀察哪些是類別資料哪些是連續資料，如圖 4.2

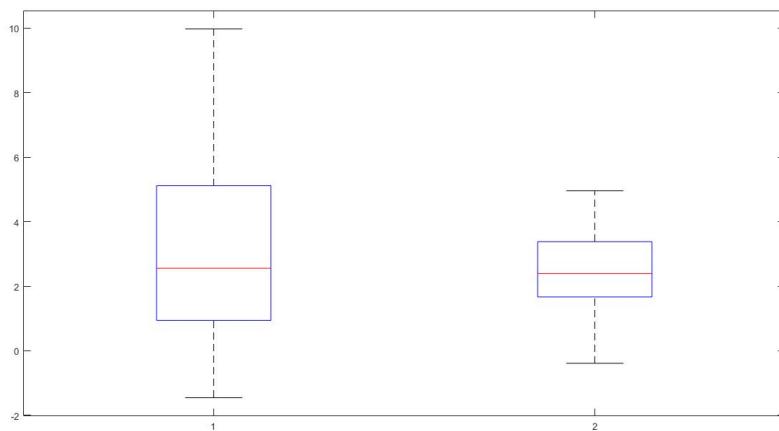


圖 4.2: REG 範例資料大致分佈

由圖 4.2 可看出兩變數 (x_1, x_2) 平均數差不多，但 x_1 變異程度較大，且可能為右偏分布，而 x_2 則看似對稱，變異程度也較小，兩變數皆無離群值，看起來無較不妥之處。

Step 2 : 定義 Design matrix 以及估計 Coefficien vector

在此我們先定義 **Design matrix** 也就是矩陣 X ，令先前資料集中 x_1 及 x_2 為此設計矩陣之兩變數後，矩陣型態如圖 4.3：

x = 200x3			
	1	2	3
1	1.0000	5.7185	0.4656
2	1.0000	0.3126	3.3040
3	1.0000	7.8441	1.5223
4	1.0000	5.0310	2.0959
5	1.0000	2.0761	4.9329
6	1.0000	1.5803	2.3814
7	1.0000	4.7541	3.2159
8	1.0000	0.1621	3.9525

圖 4.3: Design Matrix

接著有了設計矩陣後，我們便可透過統計的觀念估計 β ，如式 4.1

$$\hat{\beta} = (x^T x)^{-1} x^T y \quad (4.1)$$

如此就可以取得完整的線性模型方程式，而每一點也能夠計算出其配適值 (**fitted value**)，亦可計算出殘差等等資訊。

Step 3：建立模型與分類器

最後，由上步的估計值，便可建立出完整迴歸模型，如式 4.2

$$\hat{y} = 0.7419 - 0.1477x_1 + 0.0835x_2 \quad (4.2)$$

而有了模型後，我們透過 $y = 0.5$ 建立出分類器，而此分類器即是方程式 4.3

$$x_2 = -2.897 + 1.769x_1 \quad (4.3)$$

最後，便可繪製一條分類線，而此線依據 y 值將資料集區分成兩區塊，如圖 4.4：

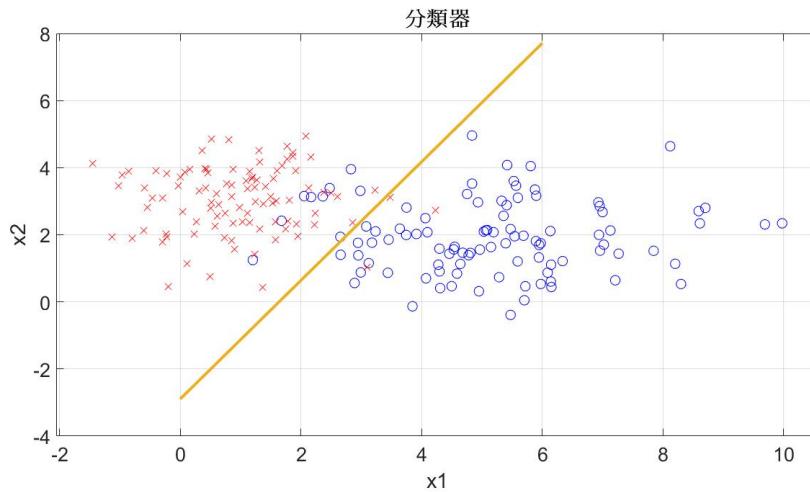


圖 4.4: 分類器

如此我們便透過迴歸的概念完成一條簡易的分類器，將原始資料區分成兩類，並且建立模型，因此倘若有新資料，我們亦能推算出新資料的類別，而其中所有程式碼如下所示：

MATLAB 語法:

```
D = load('la_1.txt');
x1=D(:,1); x2=D(:,2); y=D(:,3);
gscatter(x1,x2,y,'br','ox',10);
N=size(D,1);x=[ones(N,1),x1,x2];
b=(x'*x)\(x'*y)
f=@(x) (-b(2)/b(3))*x+(0.5-b(1))/b(3);
hold on;
fplot(f,[0 6],'LineWidth',3);
hold off;grid;
title(' 分類器');legend('hide');
set(gca,'fontsize',20);
```

我們完成迴歸模型之分類器後，更想知道此分類器判斷的到底對不對，若是有錯誤存在，那麼錯誤率大概是多少，因此我們實際透過資料集中的 y 和我們所得到的配適值 \hat{y} 相減，並取絕對值加總後，即得到錯誤的筆數，其中我們令 $\hat{y} < 0.5$ 為 0， $\hat{y} > 0.5$ 為 1，以此資料集為例，錯誤筆數有即有 12 筆，而總筆數為 200 筆，因此錯誤率為 0.06，而我們也標明錯誤判斷之資料如圖 4.5：

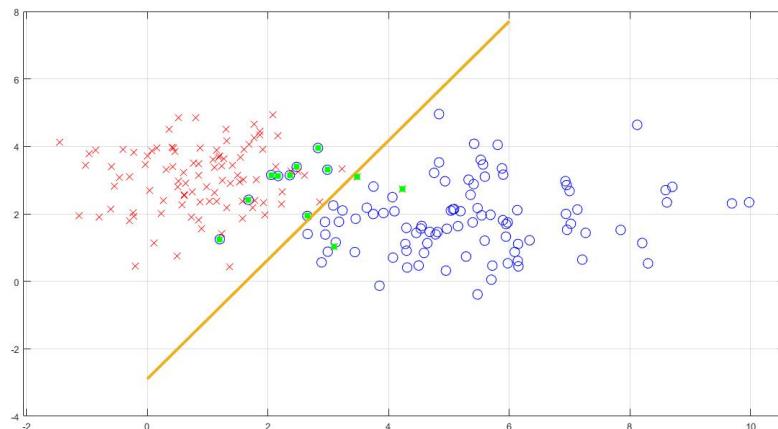


圖 4.5: 錯誤資料

圖 4.5 中，以綠色點標示的資料，即為錯誤判斷的資料，而以下為建模後並取得配適值 (\hat{y}) 後的程式碼：

MATLAB 語法:

```

temp=[x,y,yHat];
errorData=temp(temp(:,3)~=temp(:,4),1:2)
gscatter(x(:,1),x(:,2),y,'br','ox',10);
f=@(x) (-b(2)/b(3))*x+(0.5-b(1))/b(3);
hold on;
fplot(f,[0 6],'LineWidth',3)
plot(errorData(:,1),errorData(:,2),'+', 'Color', 'g', "LineWidth",5);
hold off;grid;legend('hide');

```

4.1.2 加廣型迴歸

然而，在日常生活中，並非所有資料都能用簡單線性模型套用，而線性模型也不一定是最好的選項，因此衍生出其他模型種類，而在此也將介紹其中一種，加廣型迴歸，有別於線性模型，加廣型有更豐富的可能性，產生的圖形也並非直線，而在繪圖之前，我們先了解其在迴歸上的內涵。

由於加廣型的概念是認為最初假設的函數，可能包含二次項，可能具有共線性，因此我們將最初的假設加以修改，如式 4.4：

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2 + \beta_4 x_1^2 + \beta_5 x_2^2 \quad (4.4)$$

式 4.4 除了基本兩變數之外，更加入其平方項以及相乘項，而此方式效果呈現如何，將以 MATLAB 作展示，和簡單線型迴歸的做法相同，我們透過既有的資料集，進行程式實作：

Step 1：定義 Design matrix

首先我們已經確定了資料集如上節所示，但由於考量變數由 2

項增至 5 項，因此我們的設計矩陣同時也更正，如圖 4.6：

```
DesignMatrix = 10x6
1.0000    5.7185    0.4656    2.6624    32.7013    0.2168
1.0000    0.3126    3.3040    1.0328    0.0977    10.9165
1.0000    7.8441    1.5223    11.9414    61.5301    2.3175
1.0000    5.0310    2.0959    10.5447    25.3112    4.3929
1.0000    2.0761    4.9329    10.2410    4.3100    24.3337
1.0000    1.5803    2.3814    3.7633    2.4973    5.6712
1.0000    4.7541    3.2159    15.2886    22.6017    10.3417
1.0000    0.1621    3.9525    0.6409    0.0263    15.6220
1.0000    2.6582    1.4037    3.7313    7.0663    1.9703
1.0000    0.7281    1.9146    1.3939    0.5301    3.6655
```

圖 4.6: Augmented Model Design Matrix(前十筆)

圖 4.6 中明顯相較於圖 4.3 增加許多變項，因此迴歸模型也會較為複雜。

Step 2：估計 Coefficien vector

在此和簡單線性迴歸一樣，需估計 β ，而慶幸的是，此步驟在程式執行上和先前相同，可以參考式 4.1，而在此以 MATLAB 估計出的參數如表 4.1：

表 4.1: $\hat{\beta}$ vector

β_0	β_1	β_2	β_3	β_4	β_5
0.8051	-0.2287	0.0943	-0.0043	0.0125	-0.0016

Step 3：建立模型與分類器

在我們估計出參數後，建立模型如式 4.5 所示：

$$\hat{y} = 0.8051 - 0.2287x_1 + 0.0943x_2 - 0.0043x_1x_2 + 0.0125x_1^2 - 0.0016x_2^2 \quad (4.5)$$

同理，一樣透過 $y = 0.5$ 來建立分類器，並且利用 MATLAB 繪製分類線，如圖 4.7：

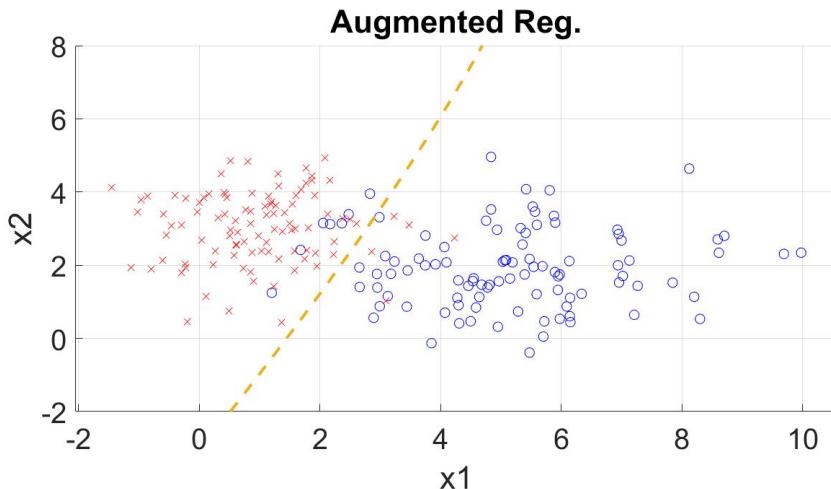


圖 4.7: Augmented Model

而此例之曲線弧度雖不明顯，但仍可看出與圖 4.4之間的不同。

同樣地，我們一樣關心著此分類器的錯誤率高低，成效如何，因此透過 MATLAB 實作得出此模型錯誤率依然為 0.06，因此我們得知在此資料集當中，簡單線性迴歸的分類器與加廣型迴歸之分類器，錯誤率一樣，成效相去不遠，然而簡單線性迴歸在實作上較容易，效率較高，因此最此資料集而言，透過簡單線性迴歸的方式可能更好。

4.2 模擬資料

我們剛才所用的資料集都是既有的資料集，然而，在日常生活中，以有限的金錢與時間考量下，我們可能無法取得大量資料集供分析實驗，因此必須考量到如何透過 MATLAB 程式，自行模擬出資料集，而再透過資料集進行分析與實驗。

在 MATLAB 中，透過 **mvnrnd** 指令可以幫助我們產生多維的常態分佈資料，而我們以下也將利用此指令幫助我們模擬新的資料集，並且進一步討論此資料集對於研究的優劣程度，指令如下所示：

MATLAB 語法:

```
n1=200;n2=200;mu1=[0 1];mu2=[4 6];
S1=[1 0;0 1];S2=[1 0;0 1];y=[zeros(n1,1);ones(n2,1)];
A=mvnrnd(mu1,S1,n1);B=mvnrnd(mu2,S2,n2);
X=[A;B]
gscatter(X(:,1),X(:,2),y,'br','ox')
```

此程式先定義樣本數 n_1 與 n_2 ，接著定義常態分配之參數 μ 與 σ ，還有所要分的類別 y ，最後透過 **mvnrnd** 以及參數，產生多維常態分佈的資料，最後即可利用此資料集繪圖，如圖 4.8：

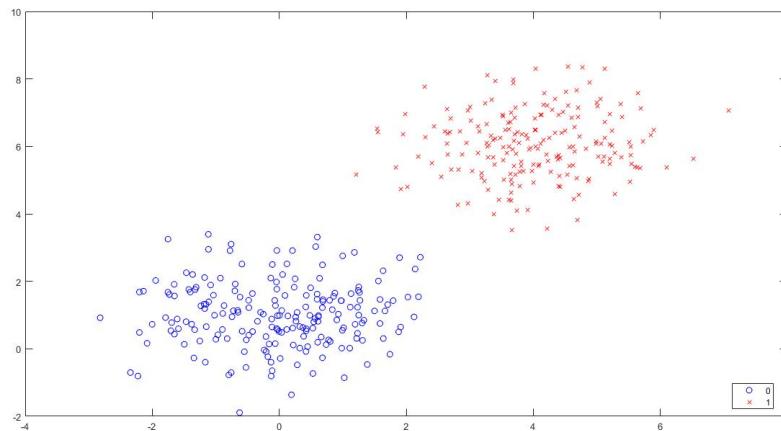


圖 4.8: 模擬資料之散佈圖

然而，這樣的資料對於本文所探討之分類器優劣並無幫助，由於兩群能明顯區分開來，無論何種分類器，最終所得出的錯誤率都極低甚至為零，鑑別度不高，因此我們利用同樣的方式，反覆實作出以下幾種

不同的資料集，如圖 4.9 所示：

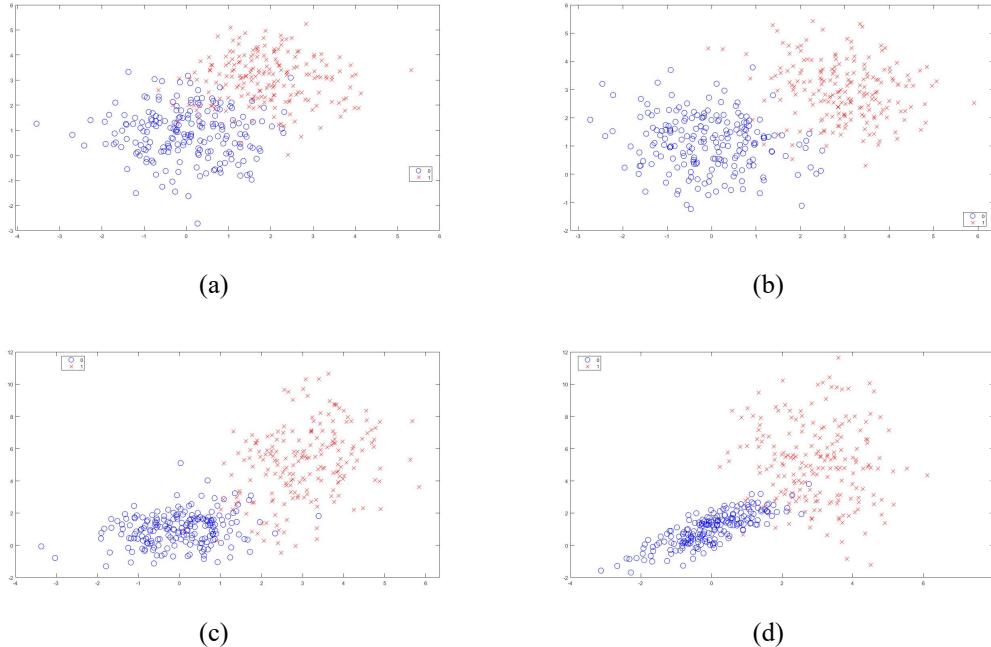


圖 4.9: 模擬不同資料集

圖 4.9 (a) 中，資料太密集，較不易區分，而圖 4.9 (b) 中，兩群分太開，分類器鑑別度也較低，因此以散佈圖直觀判斷，若要做分類器優劣比較，圖 4.9 (c) 與 (d) 可能會是較優先的選項，而以下也將示範如何透過自己模擬的資料集，比較上節討論的分類器：

- 以圖 4.9 (c) 資料集為例：我們同樣透過程式碼先觀察資料集大致分布，如圖 4.10：

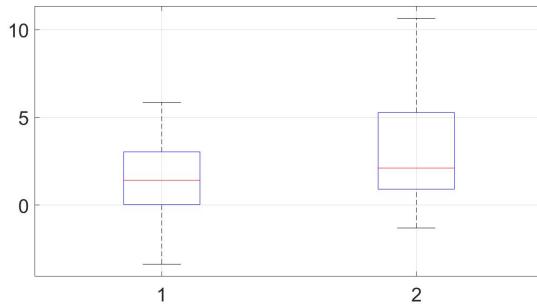


圖 4.10: 模擬資料大致分布

初步分析資料，並無離群值，也沒有較特別的異狀，且變數 x_2 較 x_1 變異程度高，而無異狀後我們直接進行模型配適，而在此不同於先前建模方式，我們以 MATLAB 本身語法進行，如下所示：

MATLAB 語法:

```
set(gca,'fontsize',30);
D = load('myData2.mat');x=D.X;y=D.y;
boxplot([x(:,1),x(:,2)]);
figure,hold on;
gscatter(x(:,1),x(:,2),y,'br','ox',10);
mdl=fitlm(x,y);
b = mdl.Coefficients.Estimate;
f=@(x) (-b(2)/b(3))*x+(0.5-b(1))/b(3);
fplot(f,'LineWidth',3);
hold off;legend('hide');grid;
```

以指令 **fitlm** 即可取得模型，再透過 MATLAB 整理好的物件，將 β 也取得，最後顯示圖形如圖 4.11：

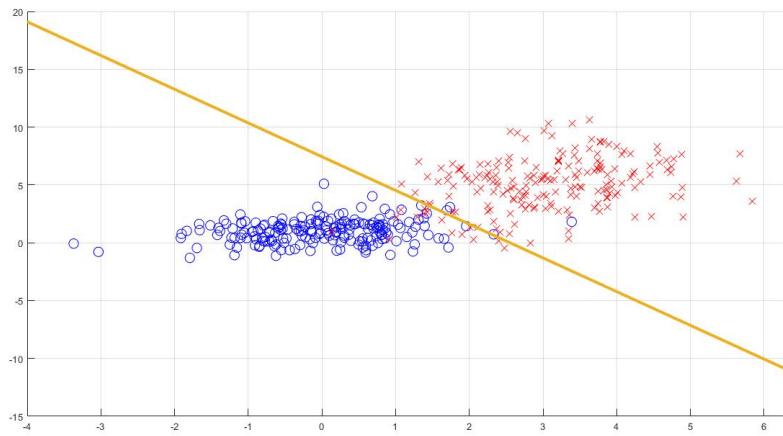


圖 4.11: 模擬資料之簡單迴歸分類器

而為了進行比較，我們計算其錯誤率為 0.0475，在 400 筆資料中，有 19 筆判斷錯誤，而接著我們以加廣型分類器進行建模，程式如下：

MATLAB 語法:

```

set(gca,'fontsize',30);
D = load('myData2.mat');
x=D.X;y=D.y;n=size(x,1);
figure,hold on;
gscatter(x(:,1),x(:,2),y,'br','ox',10);
mdl=fitlm(x,y,'quadratic');
Ab = mdl.Coefficients.Estimate;
g=@(x1,x2) Ab(1)-0.5+Ab(2)*x1+Ab(3)*x2+Ab(4)*...
x1.*x2+Ab(5)*x1.^2+Ab(6)*x2.^2;
fimplicit(g,'LineWidth',3,'LineStyle','-' );
hold off;legend('hide');grid;
```

而在此例，加廣型明顯配適出一條分類曲線，如圖 4.12：

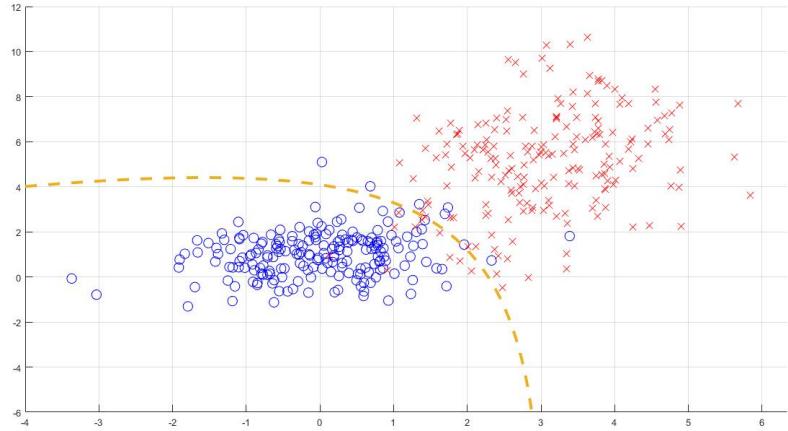


圖 4.12: 模擬資料之加廣型迴歸分類器

而其錯誤筆數為 18，錯誤率 0.045，因此我們可以得知，以訓練資料集的正確率來說，加廣型迴歸的表現比較好，但兩者之間判斷錯誤僅差一筆，因此效率可能仍要討論，而圖 4.13 彙整兩種圖形，可看出兩分類器在圖形表現上的差異。

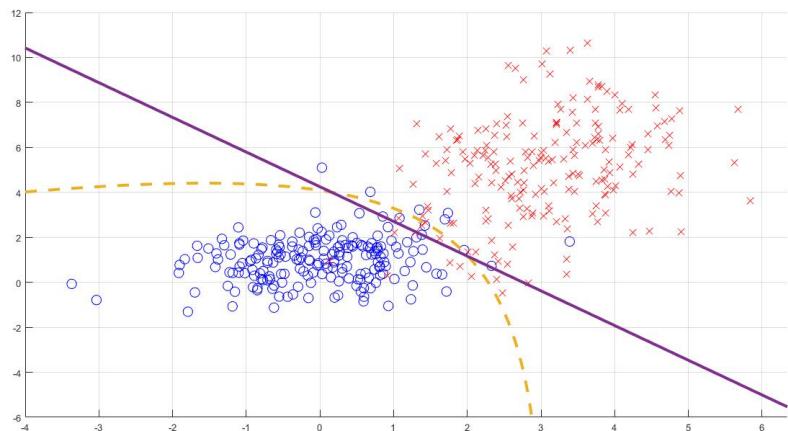


圖 4.13: 兩分類器表現差異

- 以圖 4.9 (d) 資料集為例：在圖 4.9(d) 資料集中，能明先觀察出兩群散佈程度有明顯的差異，而透過盒形圖觀察資料如圖 4.14 可見 x_2 散佈明顯較廣，且有離群值存在，但離群值不多，因此資料來源可能仍為正常：

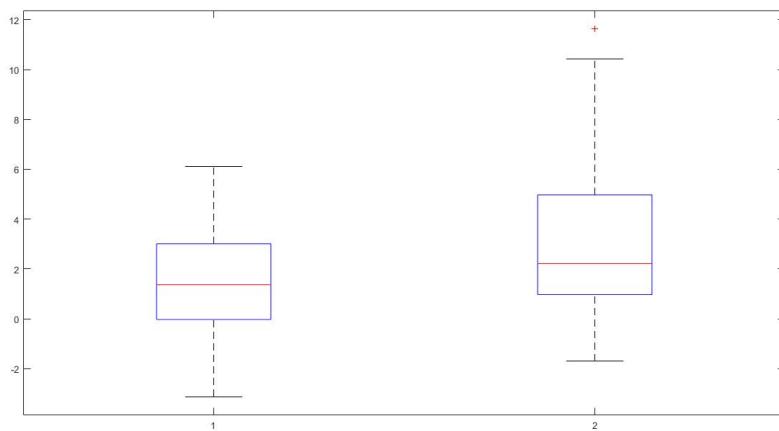


圖 4.14: 圖 4.9 (d) 資料散佈情形

接著透過模型配適，並利用簡單迴歸分類，觀察分類器的成效，如圖 4.15：

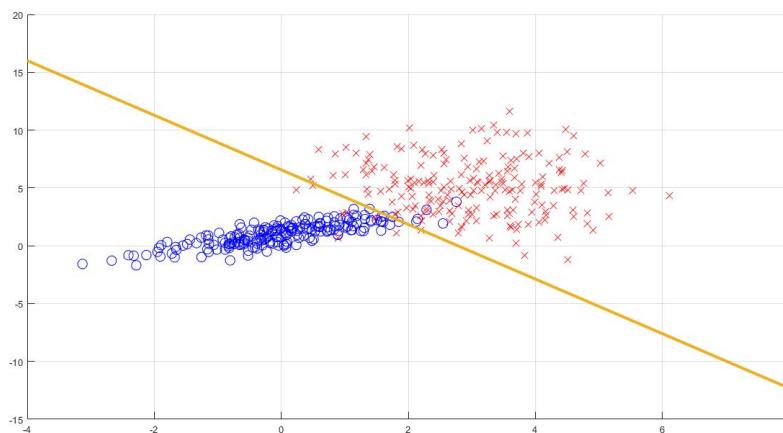


圖 4.15: 圖 4.9 (d) 資料集之簡單迴歸分類器

而由程式運算結果，其在 400 筆資料中，有 14 筆錯誤資料，其中錯誤率為 0.035，而程式碼如下所示：

MATLAB 語法:

```
set(gca,'fontsize',30);
D = load('myData1.mat');
x=D.X;y=D.y;
n=size(x,1);
boxplot([x(:,1),x(:,2)]);
figure,hold on;
gscatter(x(:,1),x(:,2),y,'br','ox',10);
mdl=fitlm(x,y);
b = mdl.Coefficients.Estimate;
f=@(x) (-b(2)/b(3))*x+(0.5-b(1))/b(3);
fplot(f,'LineWidth',3);
hold off;legend('hide');grid;
yHat = mdl.Fitted;
yHat(yHat<0.5)=0;yHat(yHat>0.5)=1;
errorMdl = sum(abs(y-yHat))
errorMdl = sum(abs(y-yHat))/n
```

接著利用加廣型迴歸模型，實驗結果，錯誤筆數為 12 筆，錯誤率為 0.03，而函數圖形如圖 4.16

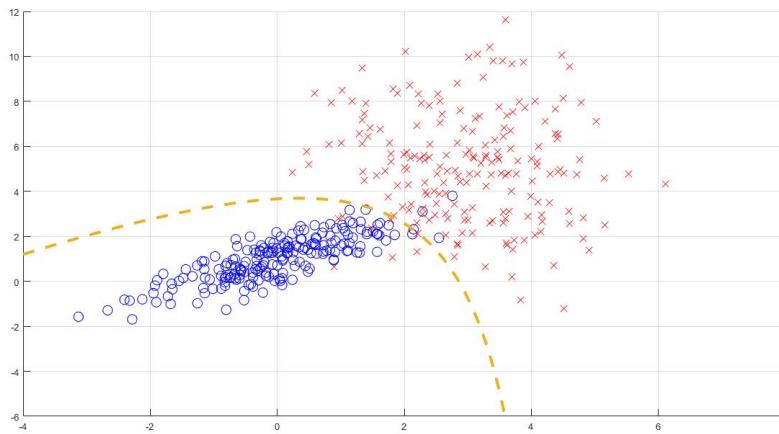


圖 4.16: 圖 4.9 (d) 資料集之加廣型迴歸分類器

而在此例中，以訓練資料集的正確率來說，加廣型較簡單迴歸優異，其錯誤筆數少兩筆，因此在此例中，以加廣型迴歸做分類可能較優異。

最後，我們討論當資料超過兩群時，要如何以上述的迴歸線進行分類，我們也以簡單線性迴歸實作，如圖 4.17：

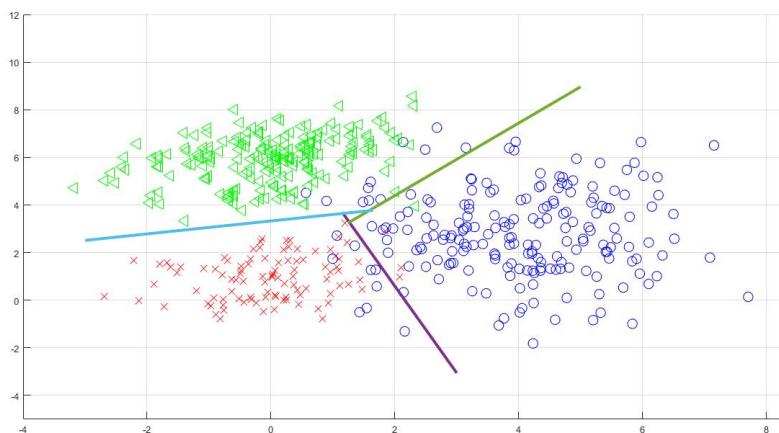


圖 4.17: 三群簡單迴歸分類器

其中，我們是以三次兩群分類繪圖，因此在程式中有三個模型，以便繪出三種不同的分割線，如下：

MATLAB 語法:

```
D=load("myData3.mat");
x=D.X;y=D.y;
figure,hold on;
gscatter(x(:,1),x(:,2),y,'rbg','xo <',10)

mdl=fitlm(x(1:300,:),y(1:300,:));
b = mdl.Coefficients.Estimate;
f=@(x) (-b(2)/b(3))*x+(0.5-b(1))/b(3);
fplot(f,[1.15 3],'LineWidth',3);

mdl2=fitlm(x(101:end,:),y(101:end,:));
b = mdl2.Coefficients.Estimate;
f=@(x) (-b(2)/b(3))*x+(1.5-b(1))/b(3);
fplot(f,[1.25 5],'LineWidth',3);

mdl3=fitlm([x(1:100,:);x(301:end,:)], [y(1:100,:);y(301:end,:)]);
b = mdl3.Coefficients.Estimate;
f=@(x) (-b(2)/b(3))*x+(1-b(1))/b(3);
fplot(f,[-3 1.65],'LineWidth',3);

hold off;legend('hide');
ylim([-5 12])grid
```

其中，仔細觀察圖 4.17，圖中有部分區域屬於三條線之外，而若是

新的資料於此，可能透過此分類器無法有效判斷，因此我們嘗試以加廣型分類器，來實測是否有機會解決此問題，而實測結果如圖 4.18：

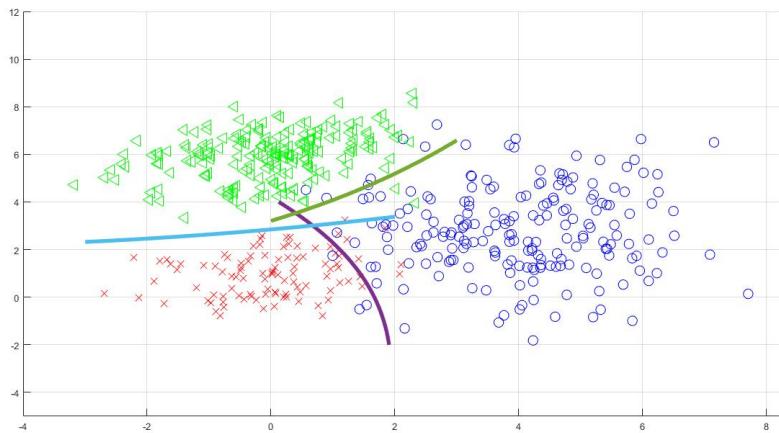


圖 4.18: 三群加廣型分類器

圖 4.18 大致和圖 4.17 差不多，也有三條分類線無法判斷之處，因此可能仍需另找其他有效方式，解決此問題。

4.3 結論

在簡單線性迴歸分類器，以及加廣型分類器上，兩種方法對於解決兩群以上的分類問題時，成效都不盡理想，但對於兩群來說，此二分類器的表現，無論是理論或是圖形呈現，都能達到高準確度，而同時也能藉由分類來預測新資料的歸屬，其中最重要的除了程式設計上的邏輯能力，以及語法熟悉度外，更需要掌握其中迴歸方程式的理論，以及參數的估計，兼具理論以及實作能力，展現統計與監督式學習，正是本文所想探討的。

第 5 章

監督式學習之判別式與 KNN

在監督式學習中，除了能用傳統統計的迴歸方式作為分群的依據之外，判別式 (**Discriminant**) 與 **K-Nearest Neighbors(KNN)** 都是此學習模式中良好的分類器，而相較於迴歸強硬的規範分類線的型態，利用判別式的方式更著重在機率的比較，同樣建構在統計的基礎上，此方式是對資料有著分配的假設，比較新的資料在哪一種類別的機率較高，而讓我們判斷類別機率高低的依據，即是本文的主題之一，判別式。而在最後，也將討論在監督式學習中常見的演算法之一，KNN，有別於統計的基礎，KNN 在不需經由任何假設的情況下，透過機器學習的概念做基本預測與分類，以下也將逐一探討這幾種分類方式。

5.1 Discriminant

在日常生活中，假設我們有一群資料包含自變數 (x_1, x_2) 以及應變數 $(y, \forall y \in \{0, 1\})$ ，其中 y 屬於類別變數，而我們有興趣知道新的一筆資料是屬於哪種類別，亦即有興趣想知道新資料中， y 是 0 還是 1，而基於機率的角度思考，我們更想知道新資料中， $y = 0$ 的機率高，還是 $y = 1$ 的機率高，因此由直觀角度思考，便是求 $P(y = 0 | X = newX)$ 以及 $P(y = 1 | X = newX)$ 其中 $newX$ 為新資料，而我們透過統計中的貝式定理改寫此機率，如式 (5.1)：

$$P(y = 0 \mid X = newX) = \frac{P(X = newX \mid y = 0)P(y = 0)}{P(X = newX)} \quad (5.1)$$

我們將原本需進行比較的機率利用貝式定理改成式 (5.1) 的型態後，分母因為兩者都相同，比較時並不需考量，因此又可改形成式 (5.2)：

$$P(X = newX \mid y = 0)P(y = 0) \quad (5.2)$$

其中式 (5.2) 的乘積，前者又可稱為概似函數值 ($P(X = newX \mid y = 0)$)，倘若我們做出一項大膽的假設，即假設資料服從常態分佈，其概似函數值就能透過概似函數求出，而此概似函數在此例便是二維常態分配，如式 (5.3)，因此可以透過估計方式求得，而後者 $P(y = 0)$ 亦可從樣本資料分布估計，因此我們便可從出機率值加以比較大小。

$$f_k(x) = \frac{1}{(2\pi)^{\frac{p}{2}} \left| \sum_k \right|^{\frac{1}{2}}} e^{-\frac{1}{2}(X - \mu_0)^T \sum_k^{-1} (X - \mu_0)} \quad (5.3)$$

其中若假設每群間的共變異數矩陣相同，式 (5.3) 中的 \sum_k 便可改寫成 \sum ，並且透過對數轉換及取極值化簡後，即可得出最終判別式，如式 (5.4)：

$$\delta_k(x) = X^T \sum^{-1} \mu_0 - \frac{1}{2} \mu_0^T \sum^{-1} \mu_0 + \log P(y = 0) \quad (5.4)$$

最後，我們令 $P(y = 0 \mid X = newX) = P(y = 1 \mid X = newX)$ 即可得到分類線，而在此時，因為我們假設了每群共變異數相同，因此線段呈現直線，稱作 **LDA**(Linear Discriminant Analysis)，而倘若部基於此假設情形下建模，線段將會呈現曲線狀，稱作 **QDA**(Quadrati

Discriminant Analysis)，而此二分類器皆可以利用 MATLAB 實作，以下也將一一討論。

5.1.1 LDA(Linear Discriminant Analysis)

在上述我們已經介紹完 LDA 的數學內涵以及統計性質，還有其基本的假設，而以下將透過 MATLAB 展示其在分類上的成效，以及圖形上的呈現，而我們以既有的資料集為例，包含兩變項 (x_1, x_2) 以及類別變數 ($y, \forall y \in \{0, 1\}$)，資料集名稱為”Demo”，作為實作中的模擬資料集，而實驗步驟如下：

Step 1：取得並了解資料集

在取得資料集時，我們往往會先簡單觀察資料是否來源正常，會不會有過多離群值，資料大致分佈狀況，以及有無遺失值等等，先檢視資料，前(預)處理資料後，使得進行之後分析，而在此也不例外，由於我們已知資料為模擬資料，並無缺失值存在，因此接著透過盒形圖進行簡單觀察資料分佈以及離群值存在多寡，如圖 5.1：

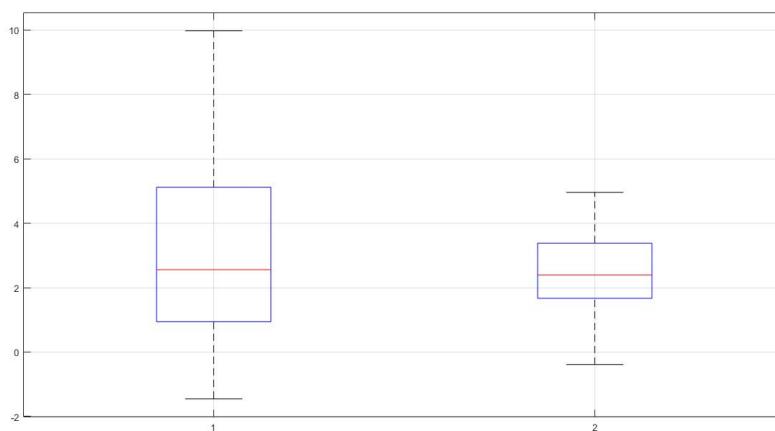


圖 5.1: 資料大致分布

由圖 5.1 可看出並無離群值存在，並且資料中兩變項分佈相去不遠，僅有些許變異上不同，中位數也大致坐落同樣位置，而資料並無太大問題，因此我們略過資料前(預)處理，直接進行統計分析。

Step 2：求矩陣 μ 與矩陣 Σ

接著，我們需要透過 MATLAB 程式語法幫助我們求出矩陣 μ 與矩陣 Σ ，而 μ 矩陣，便是要分別求出當 $y = 0$ 之下， x_1 和 x_2 的平均，以及 $y = 1$ 之下，兩者的平均，語法如下：

MATLAB 語法:

```
D=load('Demo.txt');
X=D(:,1:2); y=D(:,3);
gscatter(X(:,1),X(:,2),y,'br','>>');
x1=X(y==0,:);x2=X(y==1,:);
mu1 = mean(x1)';mu2 = mean(x2)';
text(mu1(1),mu1(2),'O','FontSize',20);
text(mu2(1),mu2(2),'O','FontSize',20);
```

接著求出平均後，讓其平均值標記於資料之散佈圖上，觀察是否平均合理，如圖 5.2：

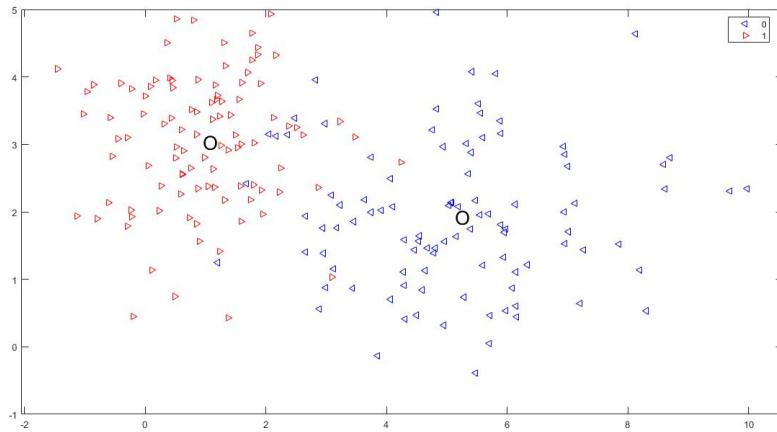


圖 5.2: 平均值坐落位置

圖 5.2 中，平均值為點”O”，大致能夠坐落在兩群之中心位置，因此判斷平均值可能錯誤機會不大，而接著開始計算 Σ 。

在前文提及，LDA 是基於假設每群變異數矩陣 Σ 皆相等的情形下，所建構之模型，而此 Σ 在兩群數量相等時， $\Sigma = \frac{(\Sigma_A + \Sigma_B)}{2}$ ，其中 A, B 代表兩群分別為群 A 以及群 B ，在此例中，僅需將 $y = 0$ 和 $y = 1$ 的共變異數矩陣求出，相加後除 2 即可得到共同 Σ ，而在程式碼上，僅需加入”sigma = (cov(x1)+cov(x2))/2”，即可求出。

Step 3：建立判別式

最後，在資料都準備齊全後，即可開始建立式 (5.4)，建立判別式後即可得知當取得新資料時，他的類別歸屬，在此我們舉例新資料為 $[0, 3]^T$ ，而其在座標軸上如圖 5.3：

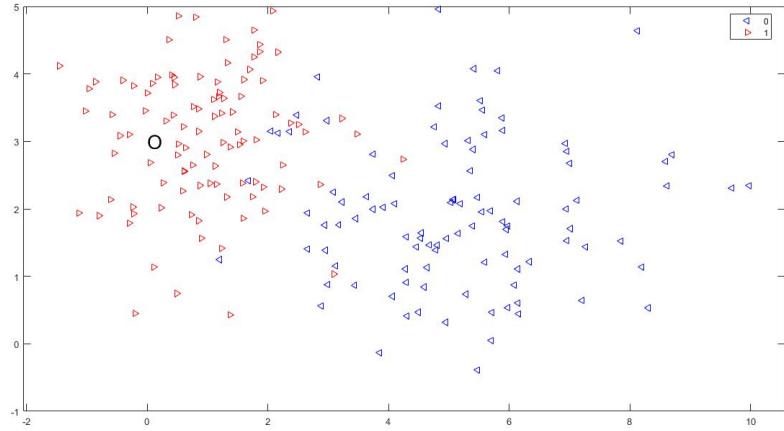


圖 5.3: 新資料 (0,3) 位置

圖 5.3，可看出新資料大約落在群 1 之中，而由判別式計算後，我們得出，落在群 0 的可能性有 -3.2742 單位，遠小於落在群 1 的可能 3.4207 單位，因此由判別式我們也可得知，此新資料較可能屬於群 1。其中語法如下：

MATLAB 語法:

```
D=load('Demo.txt');X=D(:,1:2); y=D(:,3);
gscatter(X(:,1),X(:,2),y,'br','><');
x1=X(y==0,:);x2=X(y==1,:);
mu1 = mean(x1);mu2 = mean(x2)';
text(0,3,'O','FontSize',20);
sigma = (cov(x1)+cov(x2))/2;x=[0,3]
N1=sum(y==0);N2=sum(y==1);
pi1 = N1/(N1+N2);pi2 = N2/(N1+N2);
delta0 = x'*inv(sigma)*mu1-0.5*mu1'*inv(sigma)*mu1+log(pi1)
delta1 = x'*inv(sigma)*mu2-0.5*mu2'*inv(sigma)*mu2+log(pi2)
```

最後，我們也可利用 MATLAB，將此判別之分類線繪製於圖中，如圖 5.4：

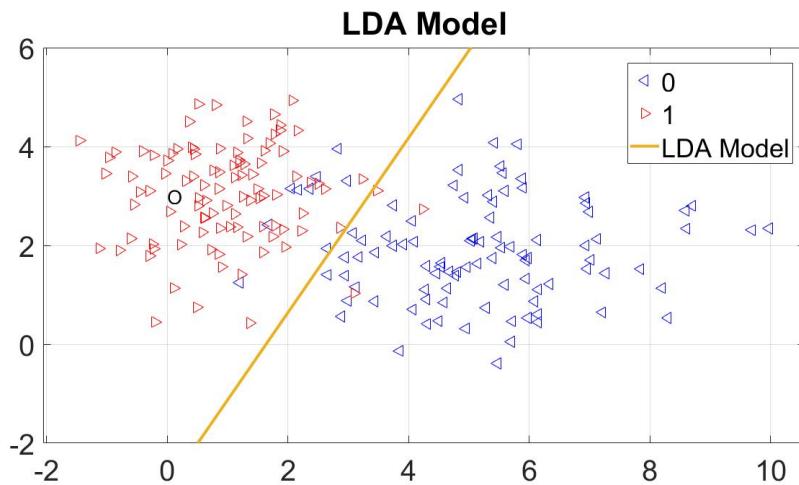


圖 5.4: LDA 分類線

圖 5.4 中 LDA Model 即是令 $P(y = 0 | X = newX) = P(y = 1 | X = newX)$ 求出之分類線，而透過此分類線，也可以明顯判斷出新資料 $(0, 3)$ 屬於群 1。

5.1.2 QDA(Quadratic Discriminant Analysis)

然而，在日常生活中有許多資料都不滿足 LDA 的變異一致性假設，因此對於那些不滿足假設的資料，LDA 模型並不能做良好的分類，而在面對這樣資料的處理上，我們便拿掉變異數一致性的假設，形成新的分類模型，QDA，而在此模型上，背後理論基礎只建立在資料服從常態分配上，因此較 LDA 來說，更能處理較多型態的資料，而本文也將透過上述”Demo”資料集，利用 MATLAB 做實作，展示 QDA 在實驗上的成效，以及程式碼操作的處理，還有最後圖形上的展示。

在實作上，我們已經知道資料大致的分佈模樣，因此直接透過 MATLAB 語法進行 QDA 建模，而 MATLAB 中，也有對應的語法，不需要向先前透過理論基礎，土法煉鋼的建立模型，而是透過”fitcdiscr”來完成模型建立，其程式語法如下：

MATLAB 語法:

```
D=load('Demo.txt');
n=size(D,1);g=cell(n,1);X=D(:,1:2);
g(D(:,3)==0)='Group A';
g(D(:,3)==1)='Group B';
gscatter(D(:,1),D(:,2),g,'br','ox');
QDA = fitcdiscr(X,g,'DiscrimType','quadratic');
k=QDA.Coeffs(1,2).Const;
Q=QDA.Coeffs(1,2).Quadratic
L=QDA.Coeffs(1,2).Linear
f=@(x1,x2) k+L(1)*x1+L(2)*x2+Q(1,1)*x1^2+Q(1,2)*x1*x2+Q(2,2)*x2^2
hold on;fimplicit(f,'LineWidth',3)
hold off;
```

我們取得資料後，透過’cell’ 將原本屬於 {0, 1} 的資料改成’Group A’ 和’Group B’，接著透過’fitcdiscr’ 建立模型，其中參數’DiscrimType’ 設定為’quadratic’ 也就是代表著目前建立的模型為 QDA 模型，最後透過語法，將 QDA 物件內的各個屬性取出，包含’Const’、’Quadratic’ 等，以建立 QDA 分類線，而最後展示圖形如圖 5.5：

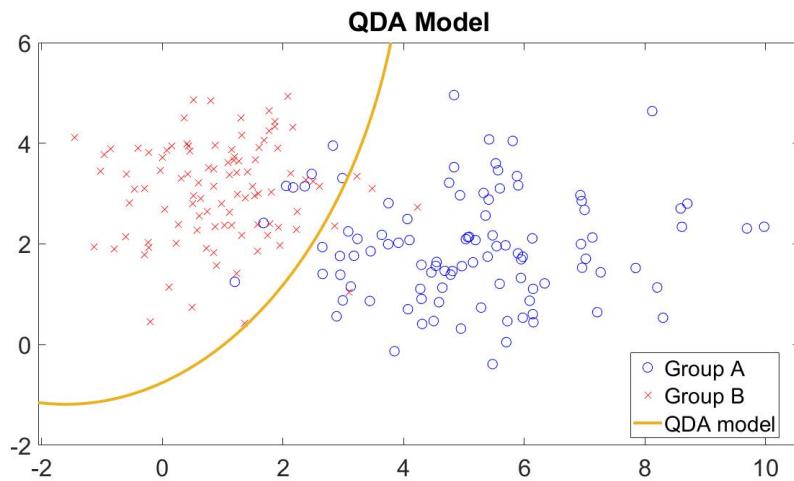


圖 5.5: QDA Model

最後，我們也同樣以新資料 $[0, 3]$ 作為測試，利用語法'predict'來預測新資料坐落類別，而實驗結果顯示，新資料屬於群”Group B”也就是和 LDA 結果相同為群 1，而圖形顯示也明顯可看出資料位於”Group B”之位置，如圖 5.6：

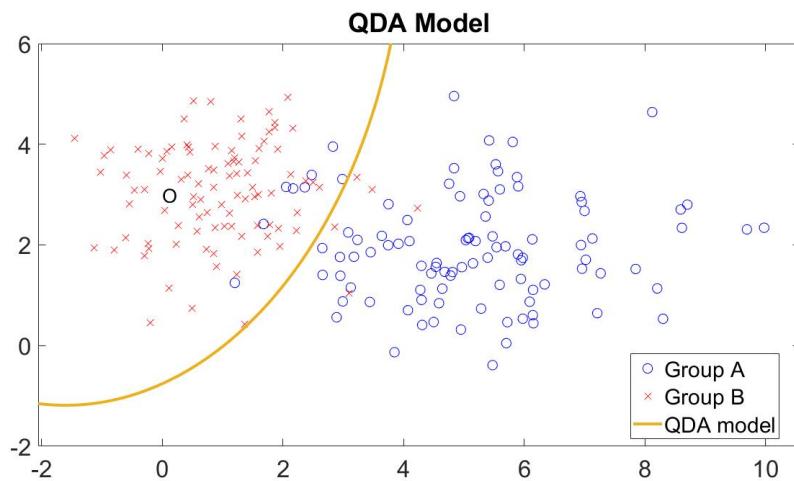


圖 5.6: QDA Model Predict

圖 5.6 中，點”O”即為新資料位置。

而在此二模型中，以”Demo” 資料集為例，透過 MATLAB 計算可知，兩者在訓練資料上都有超過 9 成的正確率，其中 LDA 的方式更是展現了 94% 的高正確率，高於 QDA 0.5% 的水準，因此可以猜測此資料集變異數可能較一致，因此以 LDA 有較簡單，且高效率的結果。

5.2 K-Nearest Neighbors(KNN)

KNN 在機器學習中佔有一席重要之地，因為其能夠以直觀的演算概念，呈現高水準的正確程度，其中不需要有任何假設的基礎，即能完成預測，而此概念即是在某個新資料周圍，找出 K 個離此新資料最近的此 K 已知資料，並計算此 K 個資料所屬群組，最後將新資料點判定給此 K 個資料所屬較多的群組，如圖 5.7 所示；

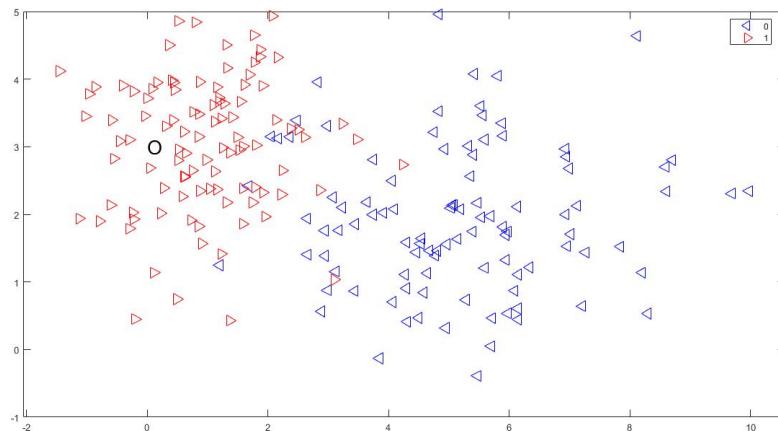


圖 5.7: KNN Predict

我們將 K 設定成 10，亦即計算最近的 10 個點的類別，而此例最近 10 個點中，最多的類別是群 1，因此判定新資料較有可能是群 1。

然而，K 設定成 10 並無任何依據證明是最佳解，也可設定成其他

數值，而 K 要設定多少變成了 KNN 演算法的問題之一，而在此資料集中，我們反覆測試 k 為 5、10、15、20 等等，發現在 K 為 5 時有著最好的訓練正確率為 95%，而其他正確率分別為 93%、94%、93.5%，因此我們以 $K = 5$ 繪製 KNN 之分類線，如圖 5.8；

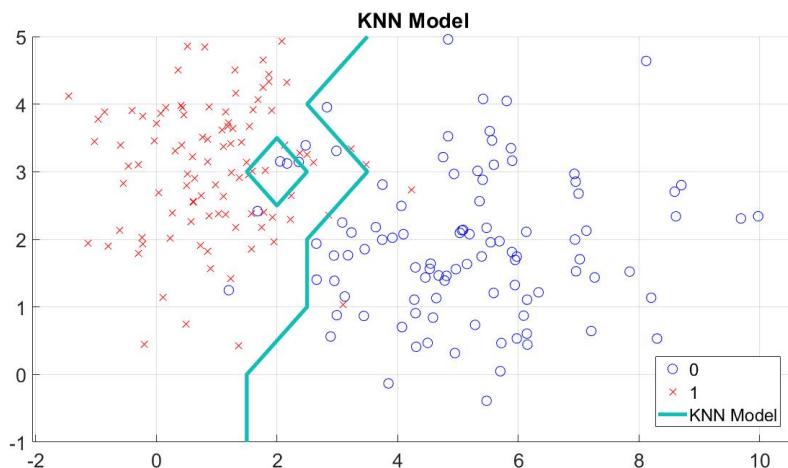


圖 5.8: KNN 分類線

透過輪廓線，將 0, 1 區隔開來，其中所使用到的關鍵函數為”contour”，是能繪出平面上座標值的輪廓，而此例刻意將平面每個以 0.05 為間隔的點做預測，因此將近預測整個平面座標，再將預測結果和每個點的位置當作參數傳給”contour”，如此一來，此函數就知道平面座標每個點的高低 (0,1)，接著透過此高低繪製輪廓線，類似等高線，而若是將此預測完結果也以 MATLAB 繪製出來，圖形將如圖 5.9 所示：

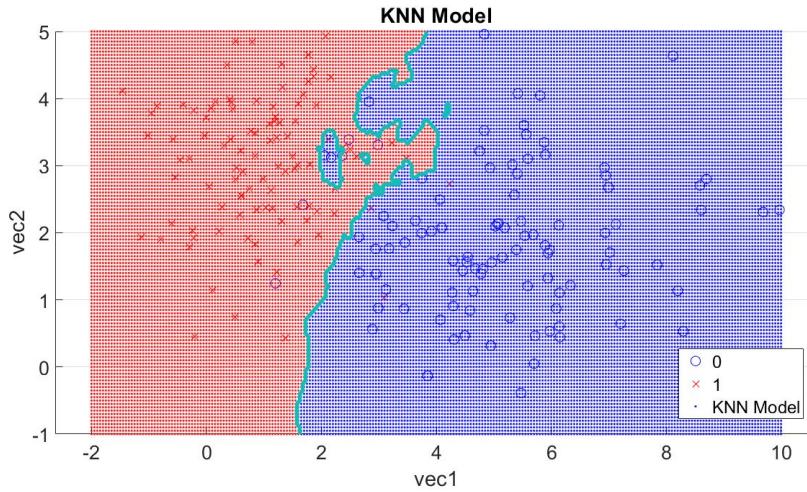


圖 5.9: KNN Meshgrid

其中，程式碼如下：

MATLAB 語法:

```

figure,hold on;
D=load('Demo.txt');
knn5 = fitcknn(D(:,1:2),D(:,3),'NumNeighbors',5);
gscatter(D(:,1),D(:,2),D(:,3),'br','ox',10)
[matrix1,matrix2] = meshgrid(-2:0.05:10,-1:0.05:5);
vec1 = matrix1(:);vec2 = matrix2(:);
m = predict(knn5,[vec1,vec2]);[mm,nn]=size(matrix1);
z=reshape(m,mm,nn);gscatter(vec1,vec2,m,'br','..');
contour(matrix1,matrix2,z,[0.5 0.5],'LineWidth',4)
title('KNN Model');grid;legend('0','1','KNN Model');
set(gca,'fontsize',20);hold off;

```

而此亦可以用立體圖來顯示此預測結果，僅須加上”mesh(matrix1,matrix2,z)”即可完成立體圖形，如圖 5.10：

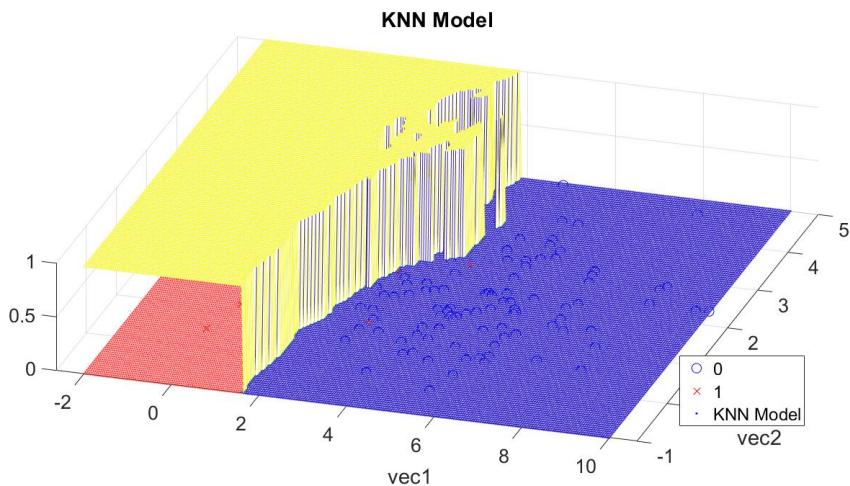


圖 5.10: KNN 立體圖

圖 5.10 中，透過立體圖形，可以更明顯看出來兩群間的差異，利用 z 座標，將群 1 撐高，而群 0 仍維持平面。

5.3 模型比較

綜合以上三種不同的模型，我們對於分類器已有相當程度的了解，但哪種分類器能夠有較好的成效，以下我們將做測試，比較 LDA，QDA，KNN(5)，KNN(15)，其中 KNN 的 K 數量選擇，暫時先依據上述例子的正確率較高兩者決定，因此先由 5 與 15 建模。

再者，在資料選擇中，我們模擬出以下兩種資料，如圖 5.11：

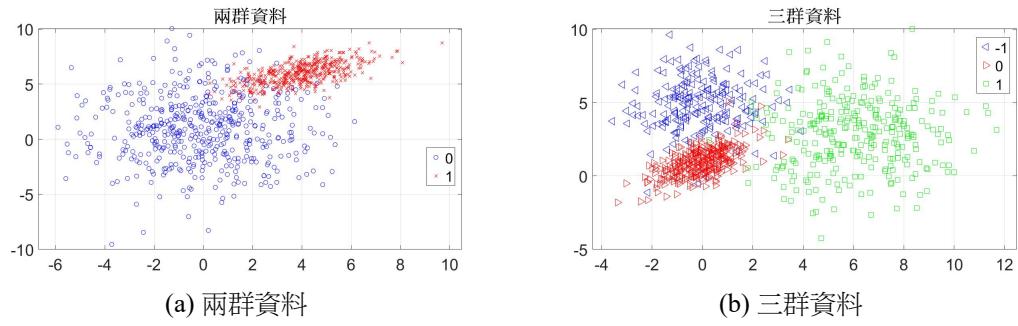


圖 5.11: 實驗資料集

圖 5.11 中，包含兩種類別的資料，資料筆數為 1000 筆，以及三種類別的資料，資料筆數為 900 筆，分別用來測試模型成效，而在資料備妥後，先以兩種類別資料做實驗。

1. 兩種類別實驗

在兩種類別資料實驗中，我們將資料讀入後，隨機將資料區分為訓練資料集和測試資料集，其中兩者比例為 7 : 3，利用 70% 的資料訓練模型，其餘 30% 的資料測試模型正確與否，程式如下所示：

MATLAB 語法:

```

load Class2Data.mat;
n=size(x,1)
p=0.7;
trainNum = n*p;
testNum = n - trainNum;
index = randperm(n);
trainX = x(index(1:trainNum),:);
trainG = y(index(1:trainNum),:);
testX = x(index(trainNum+1:end),:);
testG = y(index(trainNum+1:end),:)

```

如此一來，訓練資料集便是變數”trainX” 和”trainG”，其中”G”代表群組別，接著我們透過 MATLAB 內建函數，建立 LDA、QDA、KNN 模型，並且計算其模型之正確率，而建模與計算正確率之語法如下；

MATLAB 語法:

```

LDA = fitcdiscr(trainX,trainG);
QDA = fitcdiscr(trainX,trainG,'DiscrimType','quadratic');
knn5 = fitcknn(trainX,trainG,'NumNeighbors',5);
knn15 = fitcknn(trainX,trainG,'NumNeighbors',15);
trainAccLDA=1-resubLoss(LDA);trainAccknn5=1-resubLoss(knn5)
trainAccQDA=1-resubLoss(QDA);trainAccknn15=1-
resubLoss(knn15)

```

其中”trainAcc”開頭變數，即是四種模型之訓練正確率，即”training accuracy”，且實驗結果如表 5.1：表 5.1 可以看出，KNN(5) 的訓

表 5.1: 兩種類別之訓練正確率

LDA	QDA	KNN(5)	KNN(15)
0.9275	0.9612	0.9637	0.9587

練正確率最高，而 LDA 之訓練正確率最低，但這僅僅只是訓練資料，不一定代表此四種模型最後的表現，因此，我們用測試資料集再做一次預測，結果呈現如表 5.2；從表 5.2 可以看出測試

表 5.2: 兩種類別之測試正確率

LDA	QDA	KNN(5)	KNN(15)
0.9350	0.9700	0.9650	0.9650

資料結果，QDA 為正確率最高之模型，其次是 KNN，其中 KNN 之 K 為 5 和 15 在此次範例並無差異，最後表現最差的為 LDA，而我們觀察函數圖形，如圖 5.12；

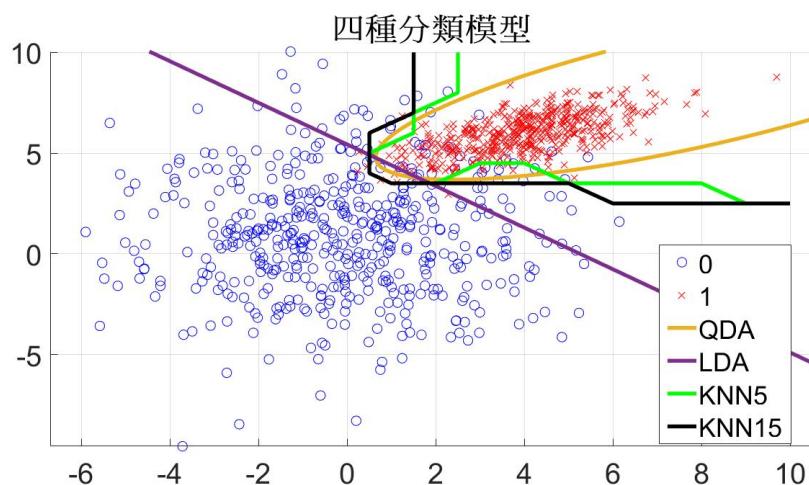


圖 5.12: 四種分類模型

圖 5.12 可以非常明顯看出資料散佈程度差異非常大，不符合變異數一致性之假設，因此 LDA 在此資料集表現非常差，而 QDA 的曲線完美配適此資料分佈位置，最後 KNN 以不規則形狀做分類，雖然也能大致區分不同類別，但在表現上較不如 QDA，而最終繪圖之程式碼如下所示：

MATLAB 語法:

```

figure,hold on;gscatter(x(:,1),x(:,2),y,'br','ox',10);
k=QDA.Coeffs(1,2).Const;L=QDA.Coeffs(1,2).Linear;
Q=QDA.Coeffs(1,2).Quadratic;
f=@(x1,x2)k+L(1)*x1+L(2)*x2+Q(1,1)*x1^2+
(Q(1,2)+Q(2,1))*x1*x2+Q(2,2)*x2^2
fimplicit(f,'LineWidth',4);
k=LDA.Coeffs(1,2).Const;L=LDA.Coeffs(1,2).Linear;
f=@(x1,x2)k+L(1)*x1+L(2)*x2;
fimplicit(f,'LineWidth',4);[matrix1,matrix2]= meshgrid(-6:10,-8:10);
vec1 = matrix1(:);vec2 = matrix2(:);
m = predict(knn5,[vec1,vec2]);[mm,nn]=size(matrix1);
z=reshape(m,mm,nn);
contour(matrix1,matrix2,z,[0.5 0.5],'LineWidth',4,'color','g');
m = predict(knn15,[vec1,vec2]);
z=reshape(m,mm,nn);
contour(matrix1,matrix2,z,[0.5 0.5],'LineWidth', 4,'color','black');
hold off;legend('0','1','QDA','LDA','KNN5','KNN15')
title(' 四種分類模型');grid;set(gca,'fontsize',30);

```

而將資料分割，建模，繪圖三種程式碼彙整，即是本文所實驗之

完整程式碼。

2. 三種類別實驗

在三種類別之實驗，我們和先前一樣，隨機將資料分割成訓練資料集與測試資料集，接著進行建模與計算訓練正確率，如表 5.3：表 5.3 可見此次模型中，訓練正確率最高的為 QDA 模型，其次

表 5.3: 三種類別之訓練正確率

LDA	QDA	KNN(5)	KNN(15)
0.9196	0.9643	0.9625	0.9589

是 KNN，而表現最差的依舊是 LDA，由圖 5.11 (b) 可見，三種類別的模型，散佈程度依舊差距極大，應該理論上來說，LDA 確實在此依舊不具備良好分類器之條件，而 KNN 在不基於任何假設下，也維持高水準之正確率。

再者，我們一樣透過語法檢視模型測試正確率，如表 5.4：

表 5.4: 三種類別之測試正確率

LDA	QDA	KNN(5)	KNN(15)
0.9167	0.9625	0.9500	0.9583

由表 5.4 之測試正確率可見，QDA 在三種類別之分群上，表現最為優異，而值得討論的是，KNN(5) 在訓練資料時以 0.9625 之正確率高於 KNN(15)，但在測試時，卻是由 KNN(15) 以 0.9583 之正確率優過 KNN(5)，可見在訓練資料時，正確率高不一定能反映在測試時，若是訓練資料正確率極高，卻在測試時有差強人意

的表現，很可能其中有 **overfitting** 之問題存在。

而 overfitting 顧名思義就是過度學習訓練資料，變得無法順利去預測或分辨不是在訓練資料內的其他資料，然而，機器學習的目標就是要訓練機器擁有人類的思考，並且擁有解決一般問題的能力，即使看到沒有包含在訓練資料的資料，也是要可以正確辨識的，因此，回到實驗主題中，KNN(5) 雖然擁有良好的訓練正確率，但若與 KNN(15) 比較，我們依舊偏好有著較高測試正確率之 KNN(15)。

5.4 結論

綜合以上幾種實驗，我們可以得到表 5.5 之彙整資料：

表 5.5: 綜合兩群及三群之測試正確率

	兩群實驗		三群實驗	
	訓練正確率	測試正確率	訓練正確率	測試正確率
LDA	92.8%	93.5%	92.0%	91.7%
QDA	96.1%	97.0%	96.4%	96.3%
KNN(5)	96.4%	96.5%	96.3%	95.0%
KNN(15)	95.9%	96.5%	95.9%	95.8%

表 5.5 中 QDA 在兩實驗都扮演最優異之分類器，因為其高測試正確率，在分類成效上良好，但 QDA 在訓練資料上並非皆為最佳，在兩群中我們可以看到 KNN(5) 在訓練上表現較為優異，而 KNN 分類表現雖然在測試上都並非頂尖，但也一直扮演優良的分類角色，由於其概念簡單，正確率上又不亞於 QDA 許多，因此在實務上也能占據重要地

位。至於 LDA 在此次實驗中皆未有良好分類表現，由於此次實驗之模擬資料，變異程度相差甚高，因此 LDA 在理論上已經無法滿足，在實務呈現理所當然也會成效不佳，這也驗證了 LDA 確實是需要基於各個群組共變異數皆為一致，LDA 方能有所表現。

第 6 章

監督式學習之類神經網路

類神經網路，又稱人工神經網路 (Artificial Neural Network，簡稱 ANN)，在機器學習和認知科學領域，是一種模仿生物神經網路（動物的中樞神經系統，特別是大腦）的結構和功能的數學模型或計算模型，用於對函式進行估計或近似。而此概念也能夠透過各種程式語言實現，例如 MATLAB、PYTHON，但在實作類神經網路之前，必須先探討其中之理論，以及運作方式，之後方能進行實務上之運用。

6.1 類神經網路 (ANN)

一般神經網路的階層上，包含輸入層、隱藏層、輸出層，而中間的隱藏層可以有一層以上。其中，兩層（含）以上隱藏層的神經網路，通常會被泛稱為深度神經網路 (Deep Neural Network，DNN)，如圖 6.1 所示。¹

¹資 料 來 源：<https://medium.com/marketingdatascience/%E5%BF%AB%E9%80%9F%E5%8F%8D%E6%87%89%E6%A9%9F%E5%88%B6-%E9%A1%9E%E7%A5%9E%E7%B6%93%E7%B6%B2%E8%B7%AF-a3bbdee4a6f6>

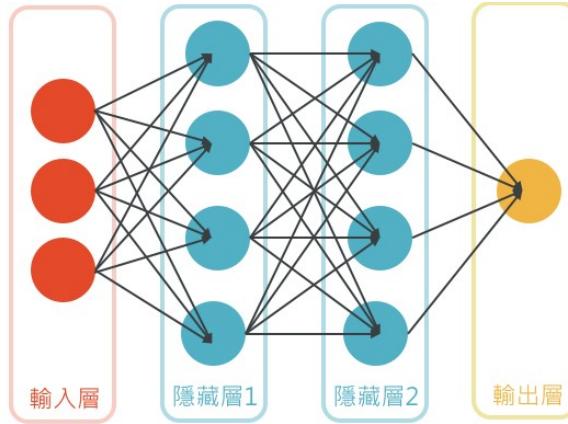


圖 6.1: 類神經網路之概念圖

假設輸入層有 p 個變數， x_1, x_2, \dots, x_p ，輸出層有 r 個變數， y_1, y_2, \dots, y_r 類神經網路的概念就是討論輸入層之變數 x 與輸出層之變數 y 之間的非線性函數關係，如式 (6.1)：

$$y_k = f_k(x_1, x_2, x_3, \dots, x_p) \quad (6.1)$$

由式 (6.1) 可看出，類神經網路認為每項 y 都可以由 x_1, x_2, \dots, x_p 共同反應而得，而其中之反應，也是隱藏層內部的非線性函數轉換，換言之，將 x_1, x_2, \dots, x_p 經由隱藏層內之函數做非線性組合，便可以得到預測結果 y ，而圖 6.1 中，包含兩個隱藏層，可見其中輸入值經由兩種不同函數作用，方能得到輸出值 y 。

然而，在輸入值進入隱藏層，以及隱藏層準備做輸出之間，還有一項控制因素，權重 (weight)，考量到一項輸出值 y 是由 x_1, x_2, \dots, x_p 共同作用而來，但並不能確保每項 x 對於 y 都有相同的影響力，因此將每個輸入值 x 做權重調整 w 後才做函數轉換，舉例來說，若要做出一項動作 (y)，是透過數種神經元 x 共同觸發而得，但並不一定每條神

經元在傳導(接觸)的效用相同，可能有的神經元觸發效果大，而有的神經元小，因此才需要權重作調整，如圖 6.2 所示。²

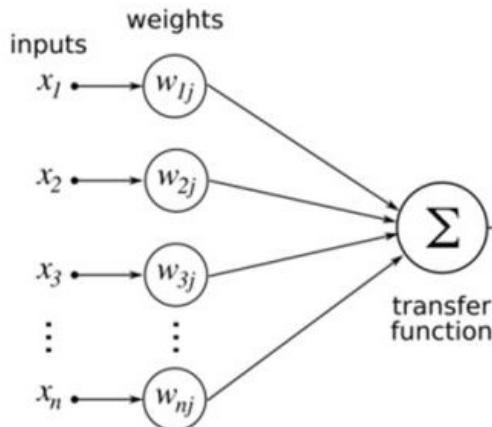


圖 6.2: 權重調整輸入值

再者，經由函數轉換後的非線性組合，將再加上殘差值 (bias) 後再傳給下一個神經元，如此便完成了該節點的輸出，而此節點的輸出，即為下個神經元之輸入值，如此一層一層傳導，直到最後的輸出層，便產生預測結果，如此方式在學術上，稱為「前向傳播法 (Forward-Propagation)」。

最後，透過前向傳播法，我們得到了預測結果，便可以透過此預測結果和真實值做相減再平方最後取偏微分，如式 (6.2)，透過此類似最小平方法之概念，便可找出能讓殘差最小之權重。

$$\nabla \sum (y_i - \hat{y}_i)^2 \quad (6.2)$$

因此，類神經網路可以透過大量的資料集訓練，並且經由複雜的數學計算預測最終結果，而其中訓練資料便是能影響最後預測結果的關鍵之一，若是訓練資料集具有代表性，且足夠多，最終便會產生較小的

²圖片來源：http://www.cc.ntu.edu.tw/chinese/epaper/0038/20160920_3805.html

誤差，反之，若是訓練資料瑕疵較多，最終測試結果便會差強人意。

6.2 ANN 實務應用——機器手臂

而此類神經模型，在實務也被廣泛應用，例如商業分析、金融預測、圖像辨識以及機器人，都是當前常會出現的議題，而本文也將淺談機器人中的機器手臂運動，作為類神經之應用，並且將透過 MATLAB 作為實作工具，了解其中語法，以及利用圖像理解。

假設在平面空間中，有項手臂運動，其運動範圍之控制因素由手臂角度 (θ_1) 和關節角度 (θ_2) 所控制，而若有項物體在 (x_1, x_2) 點上，手臂要如何改變角度才能觸碰到此物體，也就是說，假設給定 (x_1, x_2) ，希望求 (θ_1, θ_2) 為何，而此問題可以更簡單想成式 (6.3)：

$$\begin{aligned}\theta_1 &= f_1(x_1, x_2) \\ \theta_2 &= f_2(x_1, x_2)\end{aligned}\tag{6.3}$$

式 (6.3) 中，可看出在已知 (x_1, x_2) 的情況下，透過某種函數轉換，能得出最終結果 *theta*，而此問題模式，正如同類神經網路所討論，透過函數轉換輸入值，並且經由權重，殘差所調整後，得出預測值，而要如何判斷此模型好壞，或是如何增加預測正確度，便是從大量資料之訓練學習而得，因此回到主題，在進行建模之前，需要先有一組訓練資料集，而由上述提到，訓練資料集的多寡以及是否具有代表性，有可能影響最終預測結果，因此我們在機器手臂可能的運動範圍內，討論以下幾種模擬資料，來討論訓練資料集之優劣：

1. 樣本數 N=100

根據上述問題，我們先假設機器手臂能活動的範圍為圖 6.3 之灰色區塊：

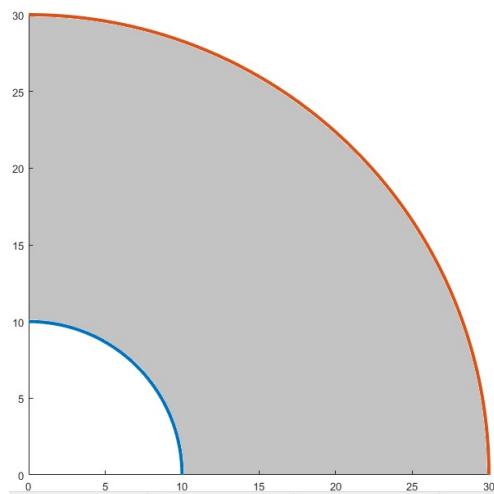
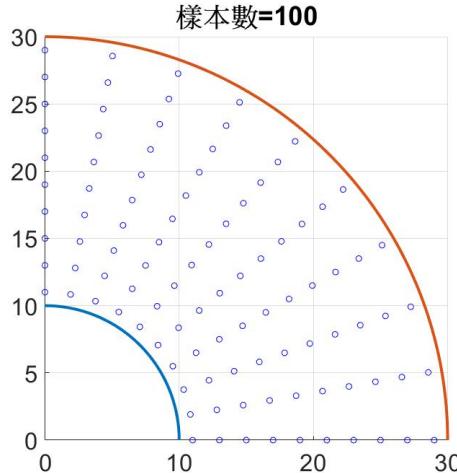


圖 6.3: 機器手臂運動範圍 (灰色區域)

那模擬資料之抽樣，亦會在此黑色區塊內做抽樣，而若是以樣本數為 100 做抽樣，則會如圖 6.4 所示：

圖 6.4: 機器手臂之訓練資料集 ($N=100$)

可見資料散佈程度較稀疏，仍有許多運動空間尚未訓練，理論上來說可能無法有良好的預測，然而，我們仍此資料作為實驗，來觀察最終結果是否符合理論。

2. 樣本數 N=400

而在對照組的部分，我們以樣本數為 400 來做測試，由圖 6.5 可見，樣本數在 400 時，已經能涵蓋大部分之運動範圍，但在外圍的部分，可能仍有訓練較為不足之處，因此可能在預測時外圍之誤差會較高，同樣我們列為本次實驗之資料集，以實作來觀察結果是否和預期一樣，也和另一組資料做比較，討論樣本數多寡時所造成的誤差。

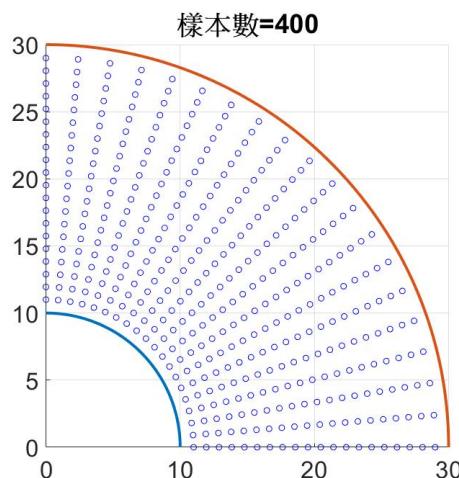


圖 6.5: 機器手臂之訓練資料集 (N=400)

在正式進入實驗之前，必須先設定類神經網路中隱藏層內的層數 (q)，若層數愈高，其中非線性程度也愈高，我們在此透過程式來測試不同層數最後的預測成效，而可以先觀察圖 6.6，由於目前要進行為監督式學習，因此在學習前須要先具備已知資料 $(x_1, x_2, \theta_1, \theta_2)$ ，而透過方程式轉換，我們將模擬資料之 (x_1, x_2) 轉換為目標變數 (θ_1, θ_2) ，如此便有真實值可做監督式學習之建模。

接著再透過 MATLAB 內建之類神經學習器 ("Neural Net Fitting") 進

行建模實驗，而在本次實驗中，我們除了將測試兩種不同數量之資料集以外，同時測試當隱藏層為 10 至 20 時，正確率會有何變化。

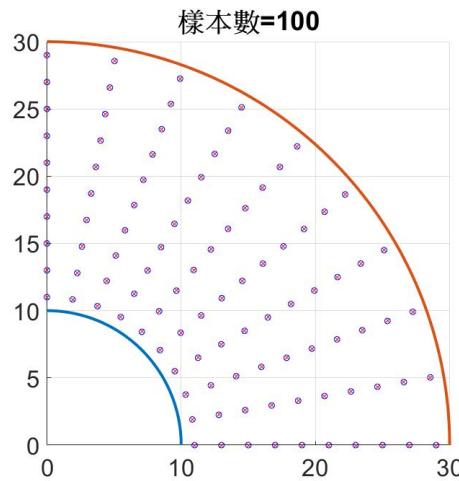


圖 6.6: 目標變數之真實值資料

而結果如表 6.1 所示：由表 6.1 可看出 $L=12$ 與 20 之 performance 最高，

表 6.1: 不同隱藏層之 performance($N=100$)

$L=10$	$L=12$	$L=14$	$L=16$	$L=18$	$L=20$
0.0021	0.0028	0.0017	0.0017	0.0013	0.0028

而 $L=18$ 最低，當隱藏層數不同時，每個模型表現亦會有差距，而接著我們測試當 $n=400$ 時，各個模型之表現，結果如表 6.2：

表 6.2: 不同隱藏層之 performance($N=400$)

$L=10$	$L=12$	$L=14$	$L=16$	$L=18$	$L=20$
0.0010	0.0011	0.0015	0.0011	0.0014	0.0011

表 6.2 可見當樣本數大時，大部分 performance 都下降，可見當樣本數愈大時，誤差愈小，估計愈準確，而 performance 最小則是發生在

$L=10$ 處。

而透過此機器手臂之實驗，我們可以知道樣本數對於預測之重要性，而此例是將資料透過類似系統式抽樣的方式，每隔一段半徑及一個角度抽取一個樣本，如此抽樣也是希望能夠不遺漏所有母體內之資料，但倘若以隨機抽樣的方式，均勻分佈在母體上，可能也會有較好的成效。

6.3 ANN 實務應用——分類器

類神經網路亦可以用在分類上，例如已知 x_1, x_2 ，預測 y 的類別，而在此透過既有資料”Demo”來做實作 ANN 之分類，同樣我們先觀察資料集基本分佈狀況，如圖 6.7：

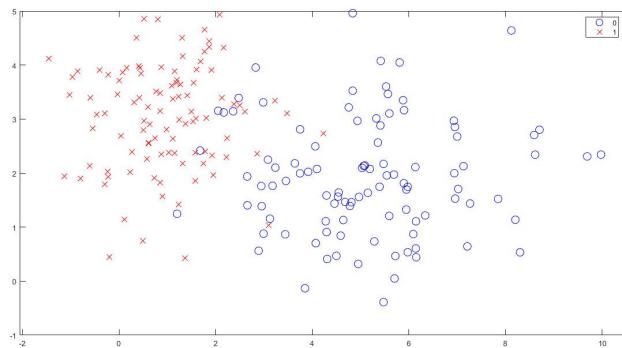


圖 6.7: 模擬資料集之散佈圖

而透過 MATLAB 中的”Neural Net Pattern Recognition”便可以直接預測結果，如圖 6.8：

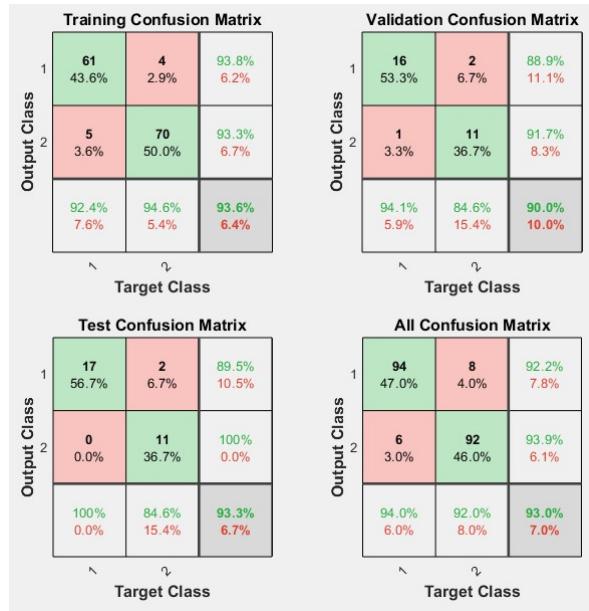


圖 6.8: confusion matrix

由圖 6.8 可看出，訓練正確率為 93.6%，而測試正確率為 93.3%，在分類上已有不錯表現，再者，我們透過不同的隱藏層，觀察資料正確率變化，如表 6.3：

表 6.3: 不同隱藏層之錯誤率以兩群為例

L=10	L=12	L=14	L=16	L=18	L=20
0.065	0.125	0.060	0.070	0.070	0.060

表 6.3 則顯示當隱藏層愈多時，錯誤率表現不一定愈低，在 L=20 與 L=14 有同樣的表現，而在此例中，L=12 反而有最高之錯誤率存在。而除了兩群分類，本文亦討論在三群時 ANN 之表現如何。

在三類分群上，我們用相同的方式進行分類，而其中 Confusion Matrix 如圖 6.9：

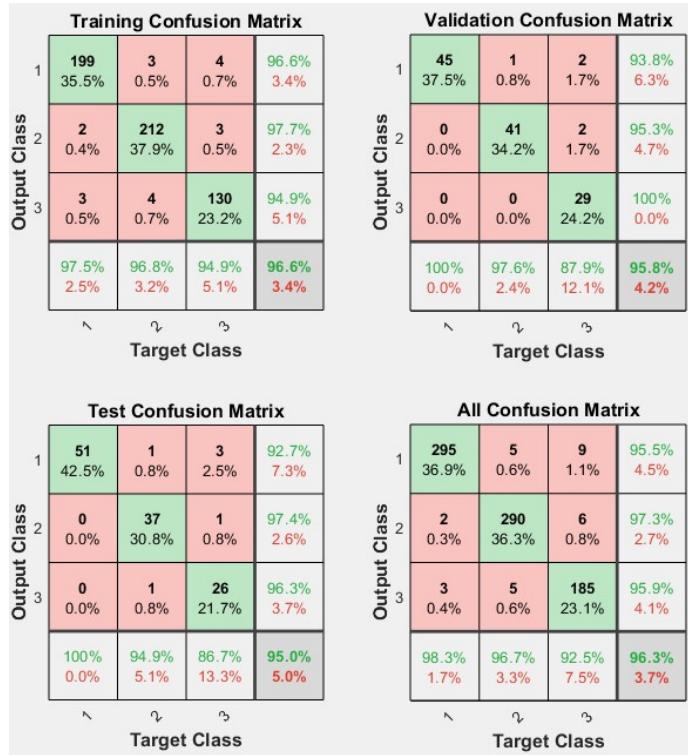


圖 6.9: confusion matrix (三群)

由圖 6.9 可看出在三群分類上，訓練正確率高達 96%，而測試正確率也有 95% 的高正確率，可見 ANN 在三群分類上亦有相當成效，雖然無法像 LDA，QDA 等等擁有線段繪圖，以輔佐分類理解，但 ANN 却也帶給分類之高效益，最後，我們同樣測試不同隱藏層之表現，如表 6.4：表 6.4 可見隨著隱藏層愈多，錯誤率愈高，由於隱藏層愈多會導

表 6.4: 不同隱藏層之錯誤率以三群為例

L=10	L=12	L=14	L=16	L=18	L=20
0.0387	0.0325	0.0413	0.0413	0.0437	0.0450

致非線性成分愈高，因此就此例而言，有可能在分類上不需要太高之非線性函數，因此在隱藏層數為 12 時就已經達到高水準的正確率，而

由此二例我們都可以觀察的出來，不同的隱藏層導致不同的錯誤率存在，而隱藏層的選擇也是 ANN 中需要考量的重點之一。

6.4 結論

在本文中，僅是粗淺的談論類神經網路，若是在數學上討論，會包含更多函數存在，而當隱藏層愈多，愈接近深度學習，其中變化也愈高，而本文透過簡單的機器手臂預測點，以及分類來說明 ANN 之用途，並且討論不同層數之錯誤率成效，而由實驗也可知，樣本數若是無法大致涵蓋母體，將會導致正確率下降，因此樣本數的多寡以及樣本數分佈情形，對於最終預測結果相當重要，且隱藏層數也須經由測試得知效果。

第 7 章

MATLAB 分類方法總結

本文歸納了多種不同的分類模型，其中在分類成效上各有優缺，並在理論觀點上也各不相同，有的基於分配假設下建立分類器，有的則是建立在模型假設下，判斷機率高低，也有的分類模型僅在直觀的基礎架構下，做出類別的預測，而本章主要總合所有分類器，一併討論其中優劣。

7.1 資料集建立

在比較所有模型之前，透過 MATLAB 先建立不同種類的資料集，而在此分別探討 3 種不同的資料集之型態，並且最後透過以下 3 種不同資料集，進行各個模型的測試，比較優劣。

1. 資料集一常態分配且變異數相同

資料集一中，我們建立了常態分配的資料，並且令其變異數矩陣皆相同，最後透過散佈圖顯示如圖 7.1：

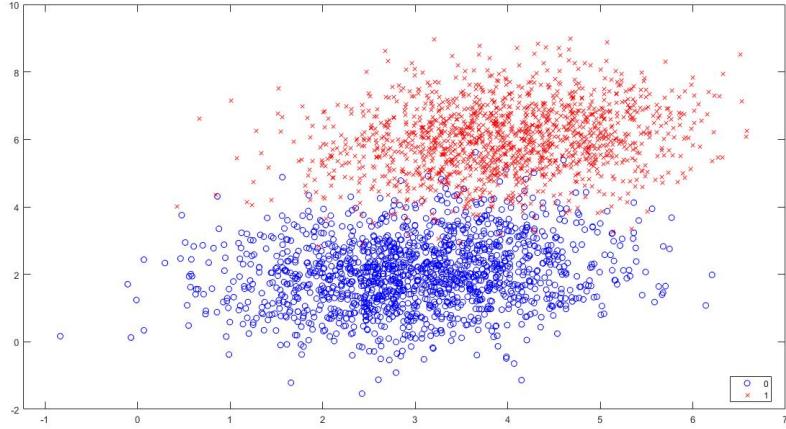


圖 7.1: 常態分配且變異數相同之資料集

資料集一，利用最常見之常態分配建立資料，並且作最單純的變異數矩陣相同，測試模型最後的效能如何，觀察是否在假設常態分配下的分類器，會有較優異的表現，或是假設變異數矩陣相同的模型，能展現低誤差。

其中在模擬資料集一之建立下，令 $\mu_1 = (3, 2)$, $\mu_2 = (4, 6)$ 且變異數矩陣如式 (7.1)：

$$\Sigma = \begin{bmatrix} 1 & 0.2 \\ 0.2 & 1 \end{bmatrix} \quad (7.1)$$

可見兩群資料皆有共變異存在，而第二群資料之平均數較第一群高。

2. 資料集二常態分配但變異數不同

接著，在資料集二當中，主要測試在常態分配下，若是變異數矩陣不同，各個分類器會有何表現，因此我們先令 $\mu_1 = (3, 2)$, $\mu_2 =$

(4, 6) 和資料集一相同，保持不變，但變異數矩陣改成式 (7.2)：

$$\sum_1 = \begin{bmatrix} 1 & 0.1 \\ 0.1 & 1 \end{bmatrix} \quad (7.2)$$

$$\sum_2 = \begin{bmatrix} 8 & 0.4 \\ 0.4 & 3 \end{bmatrix}$$

可由式 (7.2) 看出兩共變異數矩陣有所差別，第一組資料之變異程度較小，而第二組資料相較變異程度較大，且對於 x_1 有較高變異，因此散佈圖可能也會呈現較廣的分佈，如圖 7.2；

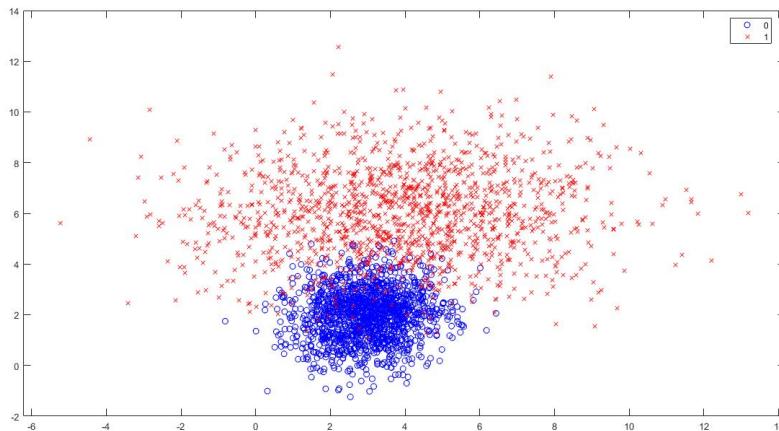


圖 7.2: 常態分配但變異數不同之資料集

圖 7.2 則呈現資料一較為密集，而資料二如上述所分析，分佈較廣，並且兩資料重疊部分較高，而此資料集主要希望測試是否共變異數矩陣不同時，加廣型迴歸方式能有較好的表現。

3. 資料集三非常態分配

最後，在資料集三中，我們討論到當兩資料來自不同分配，且都不是常態分配時，模型的表現如何，而在此令第一組資料來自卡

方分配，其中透過標準常態分配之平方轉換，第二組資料來自伽馬分配，程式碼如下所示：

```
Ntrain=1000;Ntest=1000;
n1=Ntrain+Ntest;n2=n1;n=n1+n2;
mu1=[0 0];
sigma=[1 0;0 1];
A=mvnrnd(mu1,sigma,n1);
B=gamrnd(4,1.5,[2000 2]);
A=A.^2;
x=[A;B];y=[zeros(n1,1);ones(n2,1)];
gscatter(x(:,1),x(:,2),y,'br','ox');
xlim([0 20]);ylim([0 20]);
index = randperm(n);
D.test.x=x(index(1:Ntest),:);
D.test.labels=y(index(1:Ntest),:);
D.train.x=x(index(Ntest+1:end),:);
D.train.labels=y(index(Ntest+1:end),:);
```

令第一組資料集為卡方分配，且自由度為一，第二組資料集為伽馬分配，且參數 α, β 分別為 (4, 1.5)，最後透過訓練資料集繪製圖形，如圖 7.3：

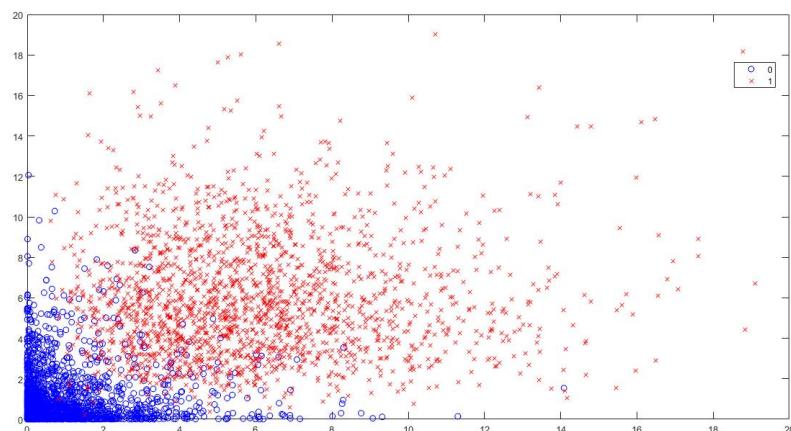


圖 7.3: 非常態分配之資料集

圖 7.3 可看出卡方分配散落在座標軸附近，而伽馬分配則是分佈較廣，但兩者仍有較明顯之分類線存在，以此資料集，測試是否建立在常態分配之假設下的分類器，會有較高的錯誤率。

7.2 模型測試

先前對於分類，預測之模型，總共包含以下幾種，REG、REG*、LDA、QDA、KNN、ANN，其中 KNN 又可由附近 K 個值做參數的變動，因此由 KNN 又可分出 KNN(10)，KNN(20)，總共有 7 種分類器，而 ANN 則是以隱含層數為 10 進行測試比較，以下將分別由三種資料集，進行各個模型之比較：

1. 資料集一：

資料集一我們了解資料分佈皆為常態，且變異數矩陣相同，以肉眼觀察可能能用直線做分類，因此 REG 與 LDA 可能會有較高的表現，透過迴圈，重複 100 次實驗後取平均值之結果如表 7.1：

表 7.1: 所有分類器比較 _ 以資料集一為例

正確率	REG	REG*	LDA	QDA	KNN(10)	KNN(20)	ANN
訓練	97.43%	97.48%	97.43%	97.42%	97.44%	97.41%	95.73%
測試	98.24%	98.14%	98.24%	98.23%	98.19%	98.30%	92.33%

由表 7.1 可見，訓練正確率最高者為加廣型迴歸，而測試正確率最高則是 KNN(20)，而其中 LDA 與 REG 表現為次高者，可能由於資料集一中，既符合常態，也服從共變異數一致之假設，因此有較好的表現，而 ANN 則是處於墊底狀態，由於 ANN 較屬於高度非線性模型，因此在分類線類似直線的情形下，ANN 成效可能較差，由圖 7.4 更可清楚觀察出差異所在。



圖 7.4: 所有模型正確率以資料集一為例

圖 7.4 可見 ANN 之正確率相較於其他模型下非常低，可見高度非線性模型並不適用於線性分類上，至於其他模型之正確率則是相去不遠。

2. 資料集二：

接著，我們以資料集二做實作，資料集二雖也屬於常態，卻是不同共變異數矩陣之情形，同樣以重複 100 次做實驗，而得到之結果如表 7.2：

表 7.2: 所有分類器比較 _ 以資料集一為例

正確率	REG	REG*	LDA	QDA	KNN(10)	KNN(20)	ANN
訓練	93.23%	94.76%	93.23%	95.69%	95.17%	94.90%	94.04%
測試	93.53%	94.75%	93.53%	95.17%	95.00%	95.00%	92.39%

在表 7.2 中，訓練正確率最高為 QDA 模型，而測試正確率最高同樣也是 QDA 模型，可見在常態分配以及共變異數矩陣不等時，完全符合 QDA 的假設前提，因此與理論相符，擁有最好的分類

表現，反而 LDA 相較之下，正確率則是低了不少，由圖 7.5 可見所有模型之間的差異。

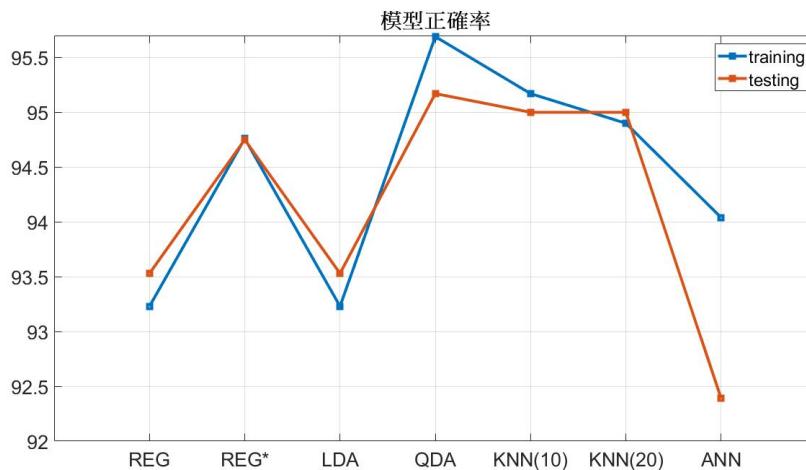


圖 7.5: 所有模型正確率以資料集二為例

而 ANN 在圖 7.5 同樣能清楚看出，測試時表現不理想，但訓練時的正確率卻不低，可能在訓練模型上，非線性程度過高，導致過度訓練的情形發生，因此才有此現象。

3. 資料集三：

最後，我們以資料集三做實作，資料集三屬於兩非常態資料集，目測觀察可能分類線一樣是非線性曲線，而由 MATLAB 執行結果如表 7.3：

表 7.3: 所有分類器比較 _ 以資料集一為例

正確率	REG	REG*	LDA	QDA	KNN(10)	KNN(20)	ANN
訓練	93.67%	95.86%	93.67%	93.76%	96.05%	95.63%	94.58%
測試	92.92%	95.03%	92.92%	93.22%	95.55%	95.28%	93.80%

表 7.3 可見，KNN 在此範例資料集表現良好，反而同樣建立在常

態分配假設下的 LDA 與 QDA 相對而言就無法有較高的正確率，而在此兩者皆為非常態模型下，KNN 亦能準確預測，並且在前兩資料集中，KNN 表現相較之下正確率也偏高，可見在無任何假設下的 KNN 能對更廣泛的資料集有好的預測，由圖 7.6 也可以明顯看出 KNN 正確率之高。

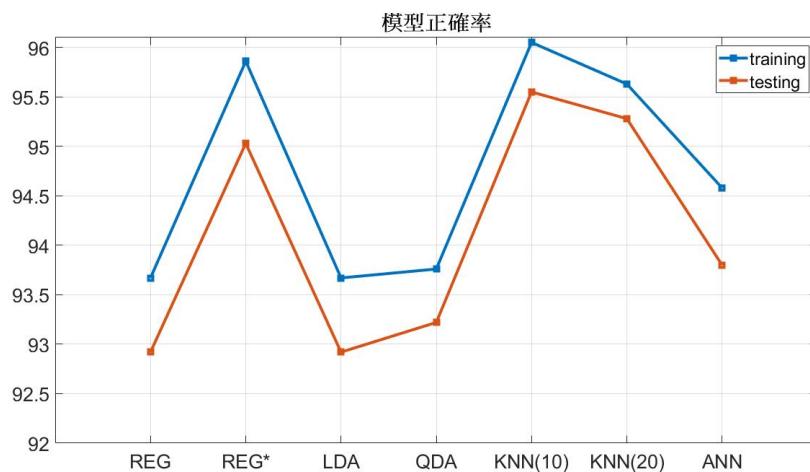


圖 7.6: 所有模型正確率以資料集三為例

而圖 7.6 也可見，加廣型迴歸也有高水準之正確率表現，由於其僅假設模型為非線性模型，因此對於此資料集而言，也較為適合。

7.3 結論

結合三種資料集之所有模型正確率，可以得到表 7.4 如下所示：

表 7.4: 所有分類器之正確率

正確率	資料集一		資料集二		資料集三	
	訓練	測試	訓練	測試	訓練	測試
REG	97.43%	98.24%	93.23%	93.53%	93.67%	92.92%
REG*	97.48%	98.14%	94.76%	94.75%	95.86%	95.03%
LDA	97.43%	98.24%	93.23%	93.53%	93.67%	92.92%
QDA	97.42%	98.23%	95.69%	95.17%	93.76%	93.22%
KNN(10)	97.44%	98.19%	95.17%	95.00%	96.05%	95.55%
KNN(20)	97.41%	98.30%	94.90%	95.00%	95.63%	95.28%
ANN	95.73%	92.33%	94.04%	92.39%	94.58%	93.80%

KNN 在所有資料集中的正確率，都在水準之上，而需要建構在假設下的 LDA 與 QDA 則是有著起起伏伏的正確率表現，而 ANN 在此三種資料集中，也不盡理想，可能原因是三種資料集都無法讓 ANN 展現其高度非線性之模型，而由表 7.4 我們便可對於先前所講到的分類器，有著更深入的認知，在不同資料情形下，分類器之選擇也是極為重要的議題之一，倘若在分類器正確率差異不遠的情形下，也可以選擇成本，效率較好的分類器，如 REG 或 LDA，由此例同時也可證明 MATLAB 在不同模型進行下，能夠快速進行多次建模預測之處理，並且同時繪圖以讓大眾理解，在進行理論數學探討的同時，以實作圖形輔助，驗證理論並且加強記憶，不僅僅對於分類器之學習，對於其他廣泛議題仍有相當高的助益。