

# 統計計算與模擬 期末報告

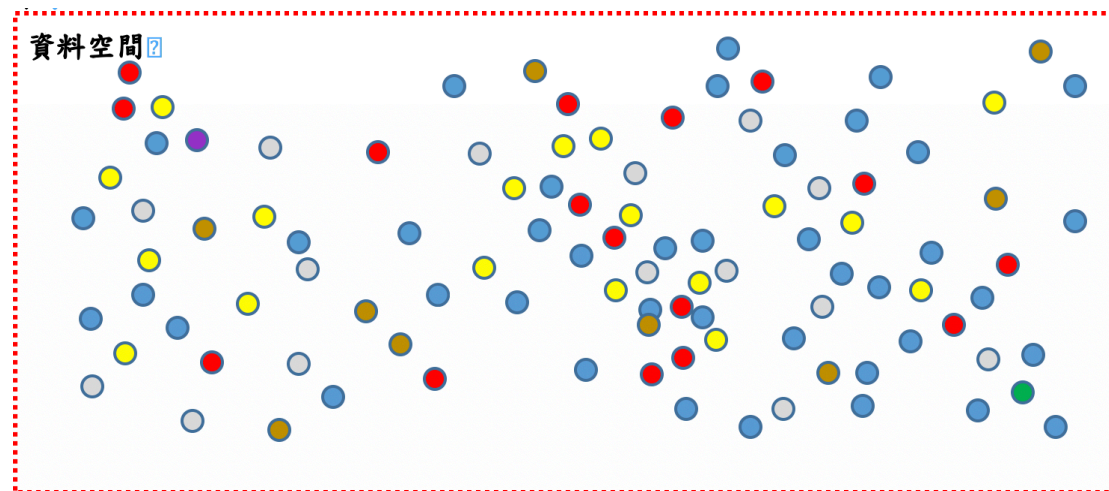
## 第 10 組

林健宏、陳初勝、許晉瑋

### 一、K-means 與 EM for mixture normal 的比較

給定一組資料(並不知道資料的群集特性)

例：

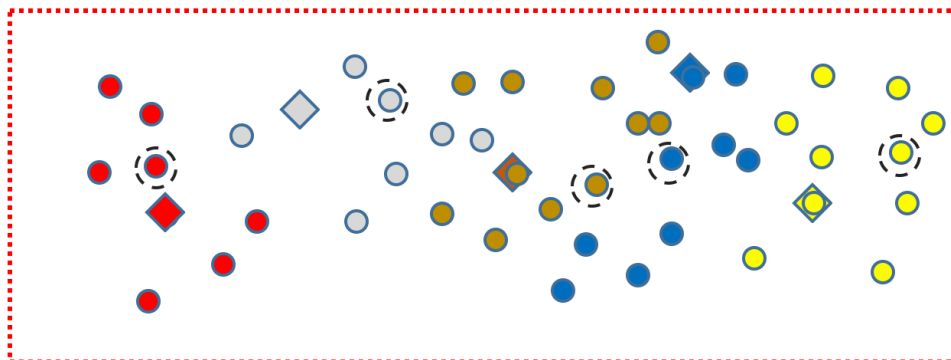


註：圖中每一種顏色表示一個群集(clusters),圖中共有五種顏色

#### 1. K-means

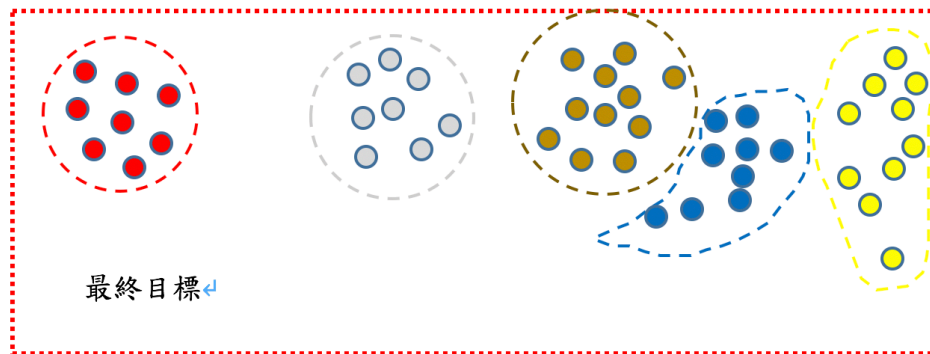
##### • 步驟

- (1) 給定一組初始值(用於決定多少群，且該值為每一群的中心值/平均數)
- (2) 加入資料後，計算每一資料與中心值的距離,並將資料分配到與其自身相小距離的中心值的群裡。
- (3) 重新計算找出群的新中心。
- (4) 重複(2)與(3)的步驟直到新平均值與舊平均值之間的差異可被忽略為止。



圖中 ◆：「更新」後新的平均數

○：「舊」的平均數



### • 演算法

$$S_i^{(t)} = \left\{ x_p : \|x_p - m_i^{(t)}\|^2 \leq \|x_p - m_j^{(t)}\|^2 \forall j, 1 \leq j \leq k \right\}$$

$$m_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j$$

$$S_i^{(t+1)} - S_i^{(t)} < \varepsilon \Rightarrow \text{converge}$$

$x_p$  為某一資料位置  
 $m_i$  為第  $i$  群之平均值  
 $\varepsilon$  為誤差值

### • 優點

- (1) 以群內變異小、群間變異大為分群的核心概念。
- (2) 想法簡單，收斂速度快(因為與平方有關係)。
- (3) 不需要分佈假設。

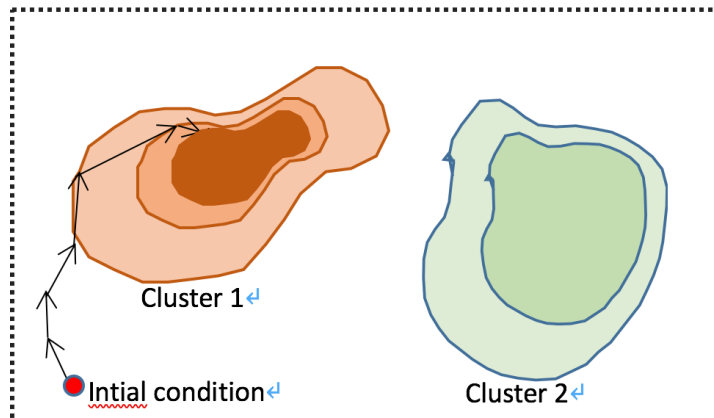
### • 缺點

- (1) 實務上，global optimization 的解不太容易求得，大部分皆為 local optimization。
- (2) 最終收斂結果與起始值選取有關。
- (3) 倘若資料本身有重疊部分，則重疊部分的資料有可能被分到錯誤類別。
- (4) 易受到極端值影響，雖此缺陷有機會透過 K-Medoids 法改善，也就是改以中位數而非平均數去作為衡量距離的基準點，但同時收斂速度也會變得較慢。
- (5) 邊界是由直線分割的，有時在 fit data 的過程中較難達到精準的效果。
- (6) K 值（群數）需要事先選定，選值的最適方法較難掌握，然而不合適的 K 值可能影響分類效果。

## 2. EM for mixture normal

### • 步驟

- (1) 決定資料混合模型。
- (2) 給定一組起始值(起始值為假設分佈模型的參數值)。
- (3) 寫下 likelihood function 並求其期望值。(E-steps)
- (4) 求其參數使得 likelihood function 最大值。(M-steps)
- (5) 重複(3)與(4)直到新的參數估計值與舊的參數估計值相差不大。



### • 演算法

$$Q(\theta|\theta^{(t)}) = E_y[\log p(x, y|\theta)|x, \theta^{(t)}] \Leftarrow E - steps$$

$$\theta^{(t+1)} = \arg \max_{\theta} Q(\theta|\theta^{(t)}) \Leftarrow M - steps$$

$$\theta^{(t+1)} - \theta^{(t)} < \varepsilon \Rightarrow converge$$

註： $\theta$  為參數估計值（可以表示成一個或一組參數）

### • 優點

- (1) 因為 EM 演算法主要是利用 likelihood function，因此對於有重疊性質的資料，會利用比較機率大小來判定該筆資料屬於那一群，因此錯判可能性較低。
- (2) 因為該演算法建立在常態模型之上，因此可以利用一些文獻上的方法去推估該筆資料有幾個 modes。
- (3) EM 演算法可利用於遺失值的插補，E-step 是對遺漏值利用 likelihood function 做最佳的估計，M-step 則求出 MLE、再進行取代，重複迭代直到估計值的變化可以被忽略為止。

• 缺點

- (1) 起始值選取會直接影響最後收斂結果。
- (2) 實務上，其收斂結果大部分為 local optimization。
- (3) 倘若這些樣本非 i.i.d，其 likelihood function 不容易被推導。

**Remark: Case of Mixture Normal distribution**

(1) E-steps

$$\begin{aligned}
 Q(\theta|\theta^{(t)}) &= E_y[\log P(\bar{x}, \bar{y}|\theta)|\bar{x}, \theta^{(t)}] \\
 \Rightarrow Q(\theta|\theta^{(t)}) &= E_y\left\{\sum_i \log[P(y_i|\theta)P(x_i|y_i, \theta)]|\bar{x}, \theta^{(t)}\right\} \\
 \therefore Q(\theta|\theta^{(t)}) &= \sum_{i=1}^n \sum_{j=1}^K \log\{\alpha_j \phi(x_i|\mu_j, \sigma_j)\} P(y_i|x_i, \theta^{(t)}) \\
 \text{and } \sum_{j=1} \alpha_j &= 1
 \end{aligned}$$

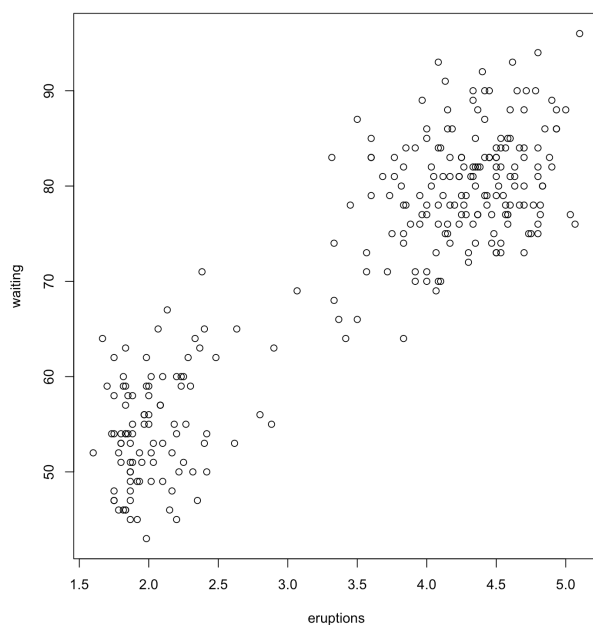
(2) M-steps

$$\begin{aligned}
 \mu_j^{(t+1)} &= \frac{\sum_i x_i P(y_i = j|x_i, \theta^{(t)})}{\sum_i P(y_i = j|x_i, \theta^{(t)})} \\
 \sigma_j^{2(t+1)} &= \frac{\sum_i (x_i - \mu_j^{(t)})^2 P(y_i = j|x_i, \theta^{(t)})}{\sum_i P(y_i = j|x_i, \theta^{(t)})} \\
 \alpha_j^{(t+1)} &= \frac{1}{n} \sum_i P(y_i = j|x_i, \theta^{(t)})
 \end{aligned}$$

## 二、Clustering Old Faithful data

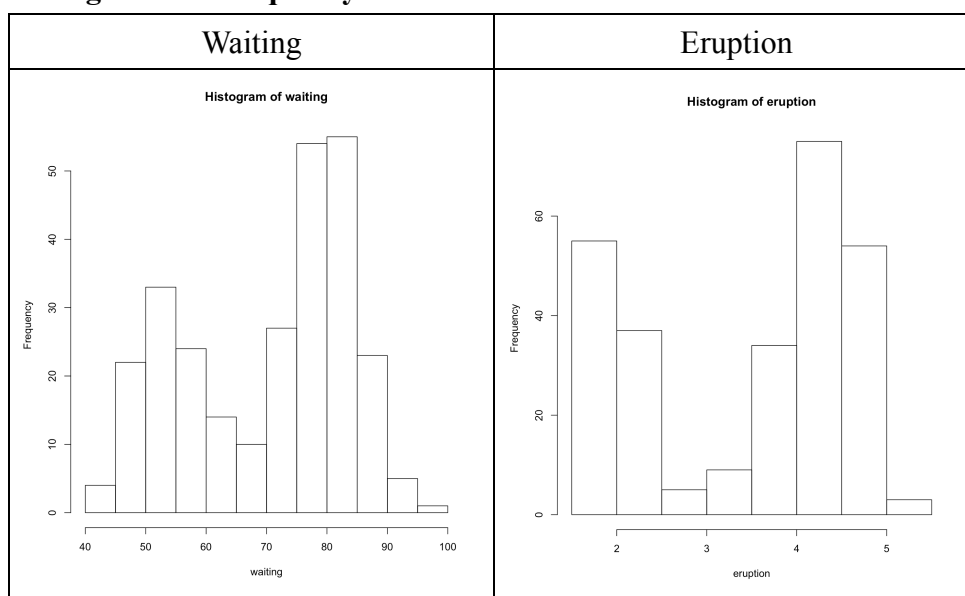
### 1. Descriptive statistics

#### • Scatter plot



根據以上 eruptions 與 waiting 兩變數的散布圖，我們可以初步推測將 data 分為兩群應是個不錯的選項。

#### • Histograms & Frequency tables



由上圖所示，我們可以發現兩個變數分佈皆呈現明顯的雙峰型態，與散佈圖所呈現的資訊相呼應。

此外，有一個值得思考的問題是，若我們接下來打算依循前面介紹的兩種分群方法（K-means 與 EM）來進行分群，都必須事先選取一組

起始值，且不論使用哪種方法，起始值的選取皆容易影響最終結果，故起始值的選取成了一個重要的課題，因此我們希望起始值能夠透過一個較有根據的方式選出，這也是我們在此事先進行敘述統計分析的一大主因。

像是若我們要用 K-means 來做分群，透過前面的觀察，我們會選擇分兩群 ( $K=2$ )，且兩變數的 histogram 也皆呈現雙峰，因此，我們可以分別從那兩個高峰中取出適當的起始值。例如：第一個起始點，waiting 的部分可以選擇從 50~55 的區間取出，相對的，eruption 的部分可以從 0~2 的區間取出；第二個起始點的 waiting 可從 75~85 的區間抽出，eruption 可從 4~4.5 的區間抽出。如果我們希望再取得更精準一些，也可以觀察以下的 frequency table，用紅色框住的部分代表明顯的高峰地帶：

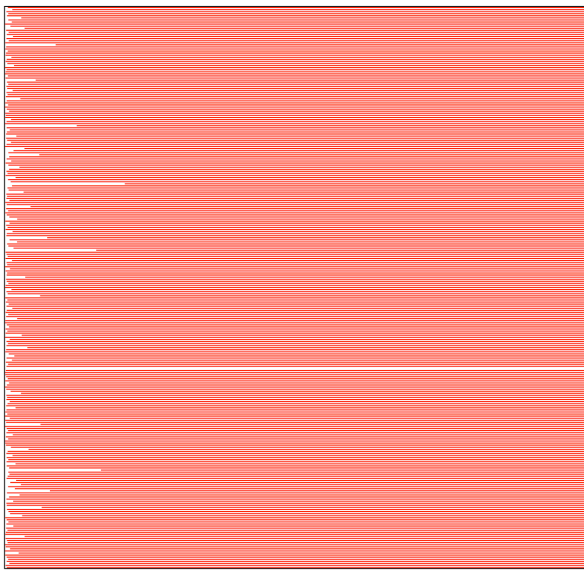
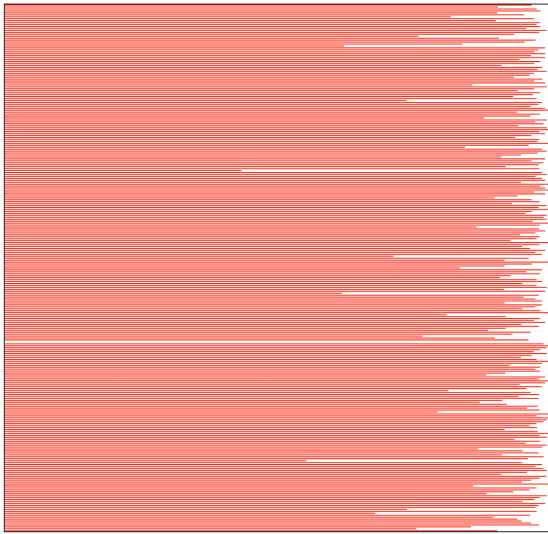
```
> table(waiting)
waiting
43 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88
1 3 5 4 3 5 5 6 5 7 9 6 4 3 4 7 6 4 3 4 3 2 1 1 2 4 5 1 7 6 8 9 12 15 10 8 13 12 14 10 6 6 2 6
89 90 91 92 93 94 96
3 6 1 1 2 1 1
> table(eruption)
eruption
1.6 1.667 1.7 1.733 1.75 1.783 1.8 1.817 1.833 1.85 1.867 1.883 1.917 1.933 1.95 1.967 1.983 2 2.017 2.033 2.067 2.083
1 1 1 1 6 2 4 3 7 2 8 4 2 2 1 3 3 4 3 2 1 2
2.1 2.133 2.15 2.167 2.183 2.2 2.217 2.233 2.25 2.267 2.283 2.3 2.317 2.333 2.35 2.367 2.383 2.4 2.417 2.483 2.617 2.633
3 1 1 2 1 3 1 2 2 1 1 1 1 1 1 1 1 2 2 1 1 1
2.8 2.883 2.9 3.067 3.317 3.333 3.367 3.417 3.45 3.5 3.567 3.6 3.683 3.717 3.733 3.75 3.767 3.817 3.833 3.85 3.883 3.917
1 1 1 1 1 2 1 1 1 2 2 4 1 1 1 2 1 5 2 1 3
3.95 3.966 3.967 4 4.033 4.05 4.067 4.083 4.1 4.117 4.133 4.15 4.167 4.183 4.2 4.233 4.25 4.267 4.283 4.3 4.317 4.333
2 1 1 6 2 1 2 5 2 2 2 4 4 1 1 3 4 2 2 2 1 5
4.35 4.366 4.367 4.383 4.4 4.417 4.433 4.45 4.467 4.483 4.5 4.517 4.533 4.55 4.567 4.583 4.6 4.617 4.633 4.65 4.667 4.7
4 1 3 1 1 4 2 3 2 1 8 1 5 1 3 4 4 1 3 1 2 6
4.716 4.733 4.75 4.767 4.783 4.8 4.817 4.833 4.85 4.883 4.9 4.933 5 5.033 5.067 5.1
1 1 1 1 1 6 2 2 1 1 2 3 1 1 1 1
```

若我們選擇使用 EM for mixture normal，則我們需選定的起始值即為 normal 分配的 mean 以及 standard deviation，既然我們決定分兩群，因此 mean 的部份我們一樣可以參考兩個高峰值，至於 standard deviation 的部分我們就選擇兩個變數分別計算出來的標準差，結果如下：

```
> sd(waiting)
[1] 13.59497
> sd(eruption)
[1] 1.141371
```

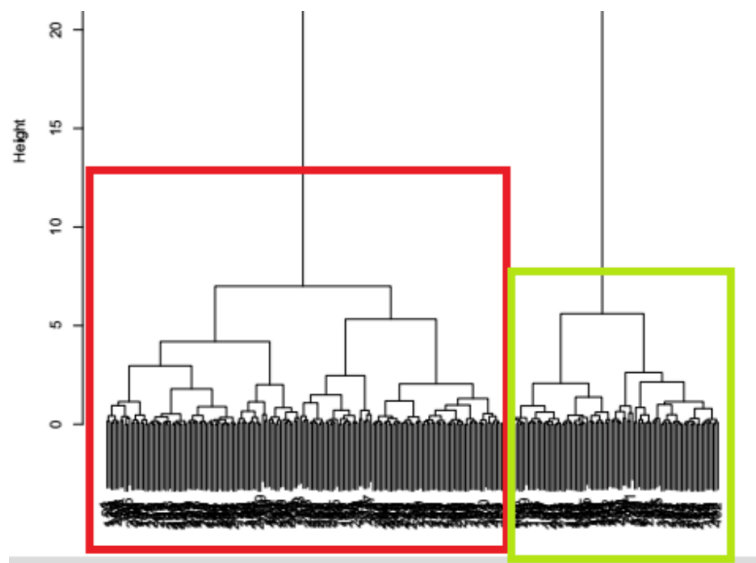
## 2. By Hierarchical Clustering

若我們想比較 K-means 與 EM 的分類效果好壞，但分群的結果並無明確的標準答案，因此我們需要尋求一個評斷的標準，依此為標準答案，分別與 K-means 與 EM 的分群結果做比較。我們選擇採用的是 Hierarchical Clustering method：

Linkage	Banner plot
<p><b>Ward method</b></p>	<p><b>Banner of agnes(x = x, metric = "euclidean", method = "ward")</b></p>  <p>Agglomerative Coefficient = 1</p>
<p><b>Divisive method</b></p>	<p><b>Banner of diana(x = x, metric = "euclidean")</b></p>  <p>Divisive Coefficient = 0.98</p>

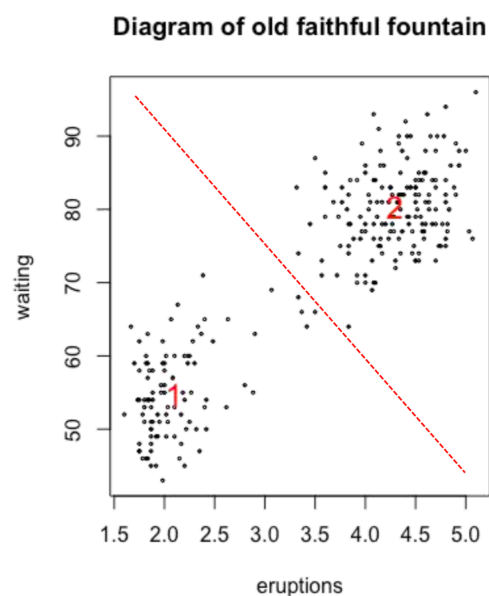
由以上 Banner plot 所示，無論使用哪種 linkage，分為兩群依舊是我們的最佳選擇，為追求高標準的分群結果，我們比較了以上 Hierarchical Clustering 的其中兩種 linkage，發現 Ward 的 AC 值趨近於 1，因此我們決定以 Ward method 的分群結果為評斷標準。

Ward 的分群結果如下圖所示（紅、綠兩群）：



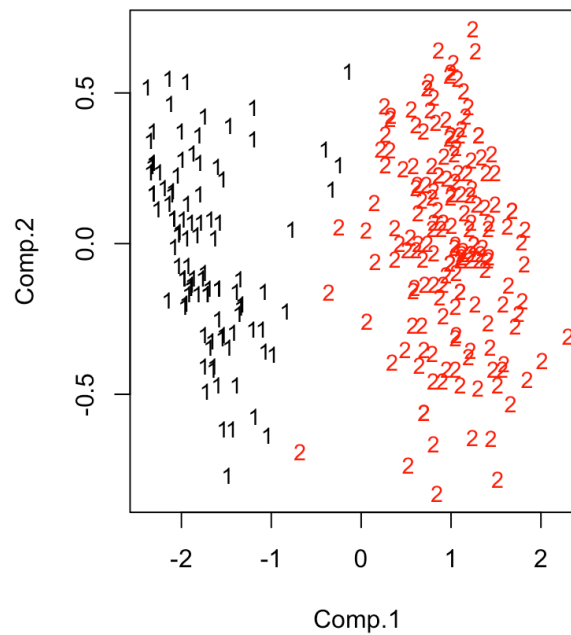
### 3. By K-means

根據前面的種種觀察，我們這裡依然選擇分為兩群，也就是  $K=2$ 。選出兩個起始點後，利用計算距離取其短的方式分群，分完一次後會得到一組新的起始點，再依此作分群，以此類推，迭代 30 次後，得到最終的中心點為  $(2.09433, 54.75)$  與  $(4.29793, 80.28488)$ ，最終的中心點以及分群結果示意圖如下所示，1 與 2 代表兩群個別的中心點，紅色筆直虛線代表的是分隔線（因 K-means 分群皆是以直線型邊界為基礎）：



另外，我們也做了主成份分析，並將以上的分類結果投影在以前兩個主成份為軸的圖上：





#### 4. By EM

若依據 EM for mixture normal 為分群準則，透過本報告第一部分所述的步驟，進行 10 次迭代，得到最終參數組合如下所示：

```
$alpha
[1] 0.3558592 0.6441408

$mu1
eruptions waiting
2.036355 54.478183

$mu2
eruptions waiting
4.289633 79.967760

$sigma1
      [,1]      [,2]
[1,] 0.06914135 0.4348933
[2,] 0.43489327 33.6954167

$sigma2
      [,1]      [,2]
[1,] 0.06914135 0.4348933
[2,] 0.43489327 33.6954167
```

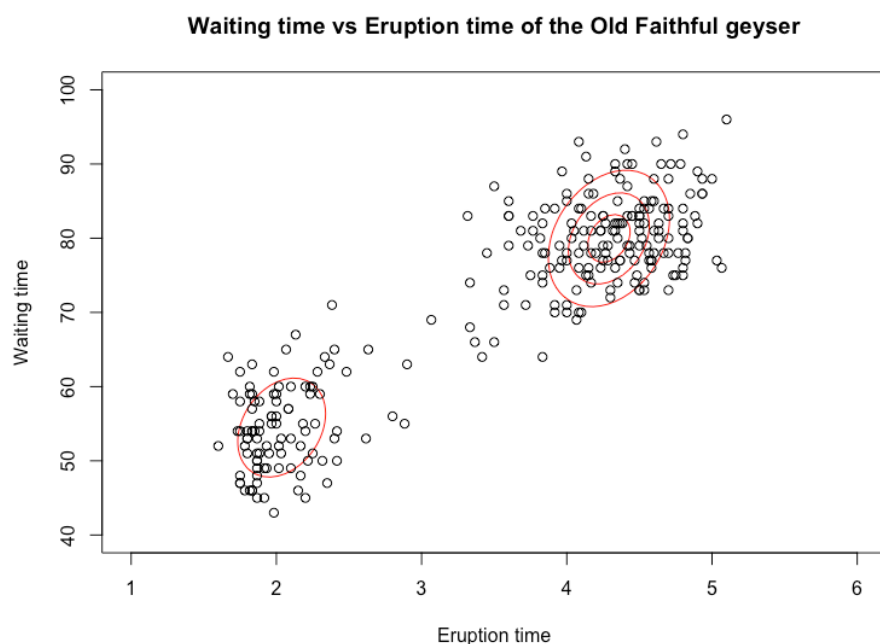


$$\alpha_1 = 0.3559, \alpha_2 = 0.6441$$

$$\vec{\mu}_1 = (2.0364, 54.4782), \vec{\mu}_2 = (4.2896, 79.9678)$$

$$\vec{\sigma}_1^2 = \vec{\sigma}_2^2 = \begin{bmatrix} 0.0691 & 0.4349 \\ 0.4349 & 33.6954 \end{bmatrix}$$

分群結果如下圖所示：



### 三、Clustering NIPS\_1987-2015

文字檔屬於文章與單字的聯合次數表，以下展示出前 6 筆單字與前 6 篇文章的次數矩陣：

	X1987_1	X1987_2	X1987_3	X1987_4	X1987_5	X1987_6
<b>abalone</b>	0	0	0	0	0	0
<b>abbeel</b>	0	0	0	0	0	0
<b>abbott</b>	0	0	0	0	0	0
<b>abbreviate</b>	0	0	0	0	0	0
<b>abbreviated</b>	0	0	0	0	0	0
<b>abc</b>	0	0	0	0	0	0

橫軸代表第幾篇文章，例如 X1987\_1 指的是 1987 年的第一篇文章，而縱軸為該篇可能使用的單字，格子內為該單字在該篇文章內的使用次數。文章年份從 1987 到 2015，總篇數為 5811 篇，而單字數則為 11463 個。

我們先檢測是否有單字數全為 0 的文章，發現共有 7 篇文章的單字數全為 0，我們決定將這 7 篇文章移除以便分析，因此剩下 5804 篇。接著清除單字時態的問題（將同單字不同時態或詞性合併為同一單字），透過 R package “textstem”，將這 11463 個單字縮減為 8426 個。

但縮減後的單字量依然太大，因此我們依照 2 個不同的標準去篩選分析用的單字：



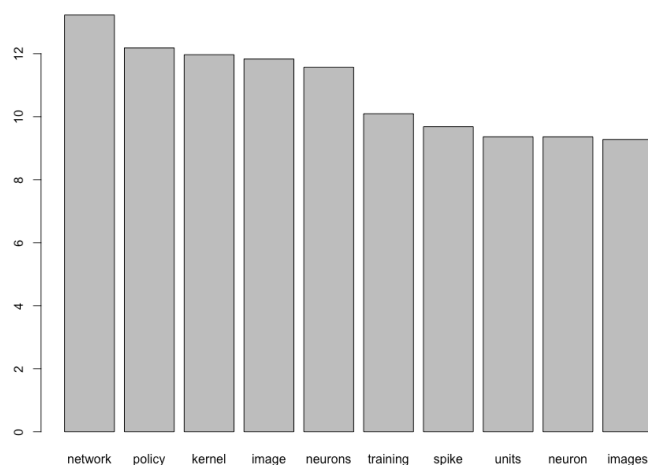
詞彙  $t$  在非常多篇文章中都出現過，就代表  $d_t$  很大，此時  $idf_t$  就會比較小。

→ 我們使用 tf 與 idf 的相乘為重要度，相乘的值越大，代表該字對於該篇文章越重要，反之亦是。得到下表各個單字對於每一篇文章的重要度，以下僅顯示前 6 個單字及前 6 筆文章的重要度值：

	X1987_1	X1987_2	X1987_3	X1987_4	X1987_5	X1987_6
<b>model</b>	0.000237	0.000431	0.000398	0.002418	0.000287	0.000094
<b>use</b>	0.000034	0.000032	0.000032	0.000055	0.000005	0.000027
<b>learn</b>	0.002797	0.003047	0.001253	0.002551	0.000967	0.001854
<b>algorithm</b>	0	0.000594	0.000962	0.000102	0.000849	0.001628
<b>set</b>	0.000076	0.000042	0.000103	0.000019	0.000119	0.000395
<b>function</b>	0.001314	0.000186	0.000196	0.000763	0.001111	0.00029

透過將每個字對於每篇文章的重要度相加，得到每個字的重要度總和，並取總和中前 100 的單字，重要度數值最高的為 neuron 18.61700 以及重要度最低的為 score 5.808552。以下為重要度前 10 高的單字：

<b>network</b>	<b>policy</b>	<b>kernel</b>	<b>image</b>	<b>neurons</b>
13.2283	12.1828	11.9686	11.8317	11.5664
<b>training</b>	<b>spike</b>	<b>units</b>	<b>neuron</b>	<b>images</b>
10.0947	9.6824	9.362	9.3606	9.2769



直方圖

文字雲

然而，考慮到可能有些文字每篇文章都出現，造成字數累計上比其他單字來得多，最後我們採用 case 2 用 tf-idf 去取用最重要的前 100 個單字當作分群用資料。

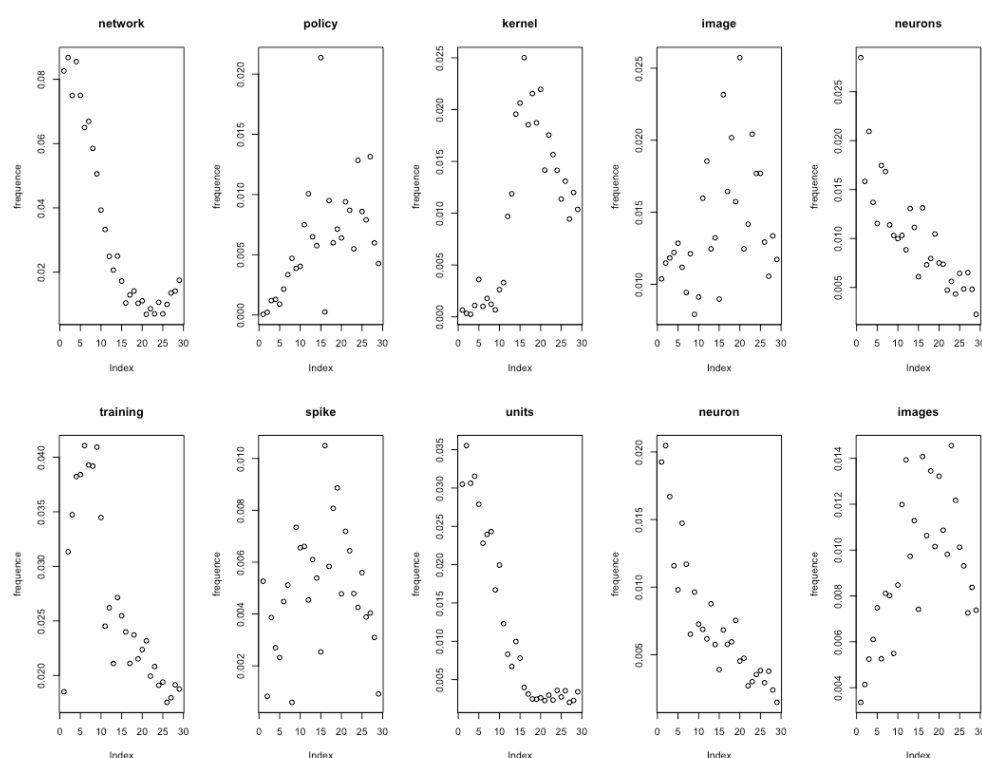
首先，加總每個字在該年度使用次數得到下表，並計算該字在該年度使用頻率：

每年該字使用的次數 (僅顯示前 10 個單字在前 10 年的使用次數)										
	1987	1988	1989	1990	1991	1992	1993	1994	1995	1996
<b>network</b>	1647	1572	1571	2185	2165	1668	1922	1679	1501	1169
<b>policy</b>	1	4	25	33	26	55	96	135	115	120
<b>kernel</b>	13	6	5	28	104	26	51	35	20	78
<b>image</b>	207	208	248	312	371	287	271	348	235	271
<b>neurons</b>	568	287	439	350	333	448	484	327	306	297
<b>training</b>	369	568	728	977	1109	1054	1129	1125	1215	1025
<b>spike</b>	105	15	81	69	67	115	147	17	218	195
<b>units</b>	608	644	642	806	805	585	688	698	496	593
<b>neuron</b>	384	371	350	296	283	378	336	187	286	216
<b>images</b>	67	75	110	156	216	135	233	230	163	252

每年該字使用的頻率(tf) (僅顯示前 10 個單字在前 10 年的使用次數)										
	1987	1988	1989	1990	1991	1992	1993	1994	1995	1996
<b>network</b>	0.0826	0.0867	0.0749	0.0855	0.075	0.065	0.0669	0.0585	0.0506	0.0393
<b>policy</b>	0.0001	0.0002	0.0012	0.0013	0.0009	0.0021	0.0033	0.0047	0.0039	0.004
<b>kernel</b>	0.0007	0.0003	0.0002	0.0011	0.0036	0.001	0.0018	0.0012	0.0007	0.0026
<b>image</b>	0.0104	0.0115	0.0118	0.0122	0.0128	0.0112	0.0094	0.0121	0.0079	0.0091

<b>neurons</b>	0.0285	0.0158	0.0209	0.0137	0.0115	0.0175	0.0168	0.0114	0.0103	0.01
<b>training</b>	0.0185	0.0313	0.0347	0.0382	0.0384	0.0411	0.0393	0.0392	0.0409	0.0345
<b>spike</b>	0.0053	0.0008	0.0039	0.0027	0.0023	0.0045	0.0051	0.0006	0.0073	0.0066
<b>units</b>	0.0305	0.0355	0.0306	0.0315	0.0279	0.0228	0.0239	0.0243	0.0167	0.0199
<b>neuron</b>	0.0193	0.0205	0.0167	0.0116	0.0098	0.0147	0.0117	0.0065	0.0096	0.0073
<b>images</b>	0.0034	0.0041	0.0052	0.0061	0.0075	0.0053	0.0081	0.008	0.0055	0.0085

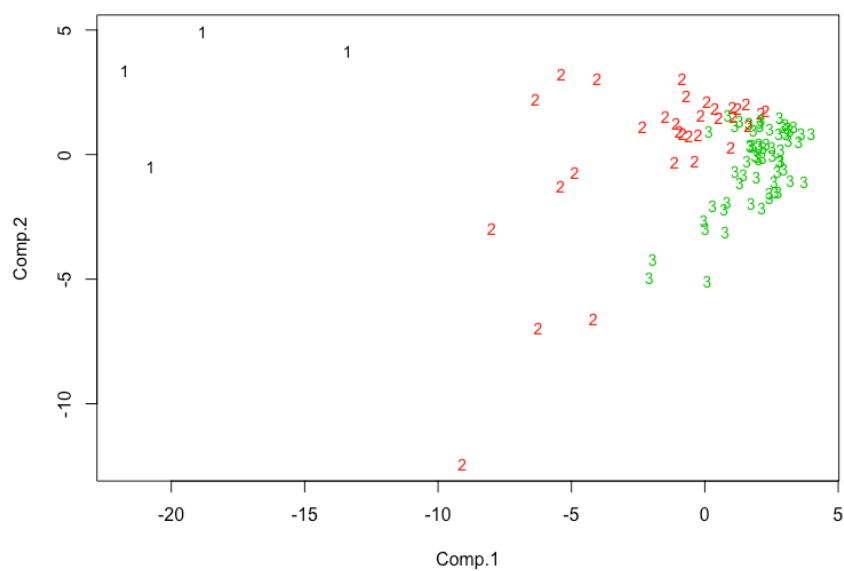
將這 10 個字在每一年度的使用頻率繪製散佈圖，發現有些字使用頻率逐年下降，有些字使用頻率反而上升，也有些字使用頻率持平。



我們將這 100 個單字依照使用頻率上升、下降以及持平分為三類：

<b>使用頻率上升</b>	Policy, kernel, image, matrix ... 共 50 個字
<b>使用頻率下降</b>	Network, neurons, training, unit ... 共 33 個字
<b>使用頻率持平</b>	Spike, action, node, object ... 共 17 個字

另外，我們亦用 K-means 方法，以年份為變數將文字分成 3 群，經過迭代 50 次，並將文字分群後的結果投影到 PCA 的第一、第二主成份上，以圖呈現分群結果：



前 10 個單字分群結果	
	Cluster
<b>network</b>	2
<b>policy</b>	3
<b>kernel</b>	2
<b>image</b>	2
<b>neurons</b>	3
<b>training</b>	2
<b>spike</b>	3
<b>units</b>	3
<b>neuron</b>	3
<b>images</b>	3

#### 四、Appendix

- R code

<https://github.com/kevinpiger/StatisticalComputingandSimulation-Final>

- 參考資料來源

1. <https://tawei Huang.hpd.io/2017/03/01/tfidf/>
2. <https://rpubs.com/saqib/DocumentClustering>
3. [https://commons.wikimedia.org/wiki/File:Em\\_old\\_faithful.gif](https://commons.wikimedia.org/wiki/File:Em_old_faithful.gif)