**統計計算與模擬 期末報告**

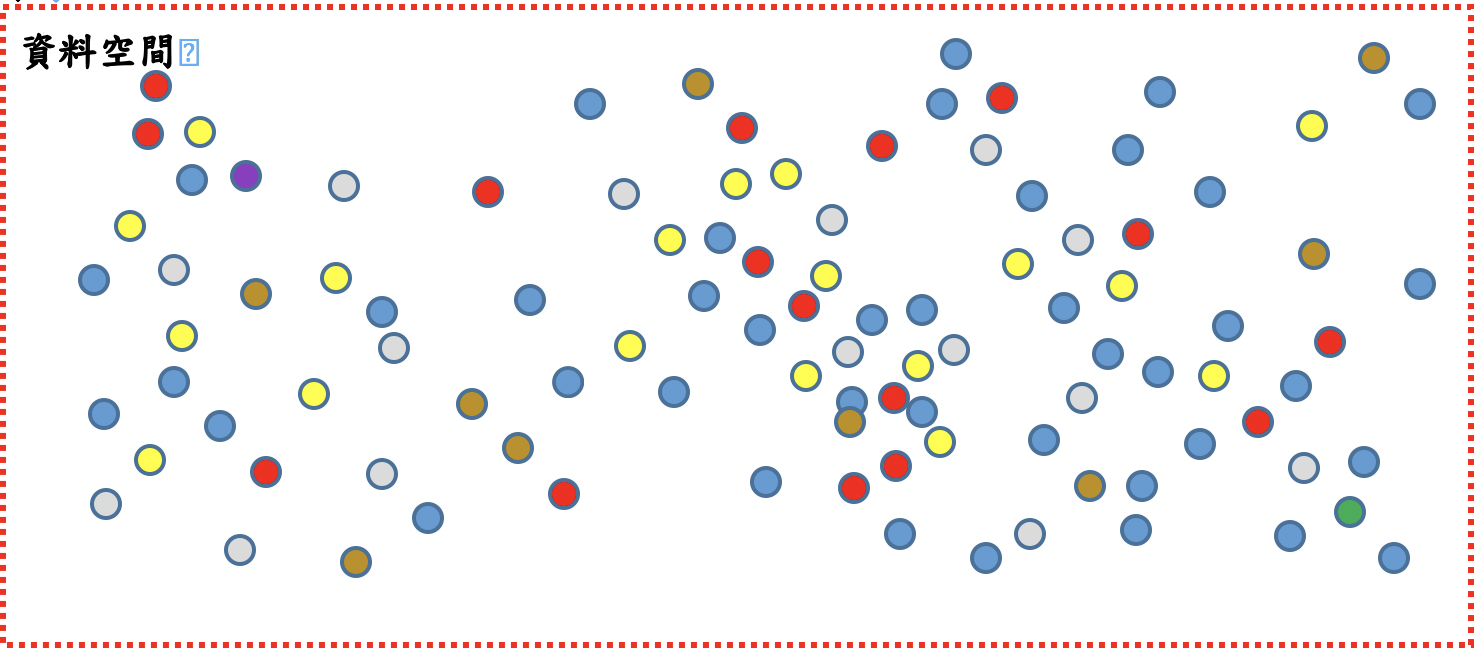
第10組

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：

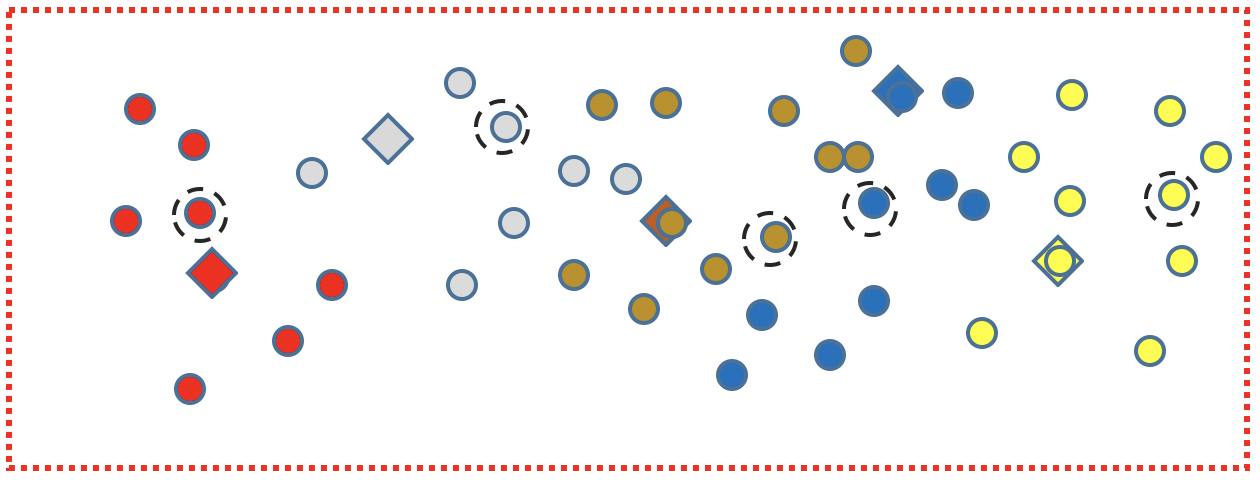


註：圖中每一種顏色表示一個群集(clusters),圖中共有五種顏色

1. **K-means**

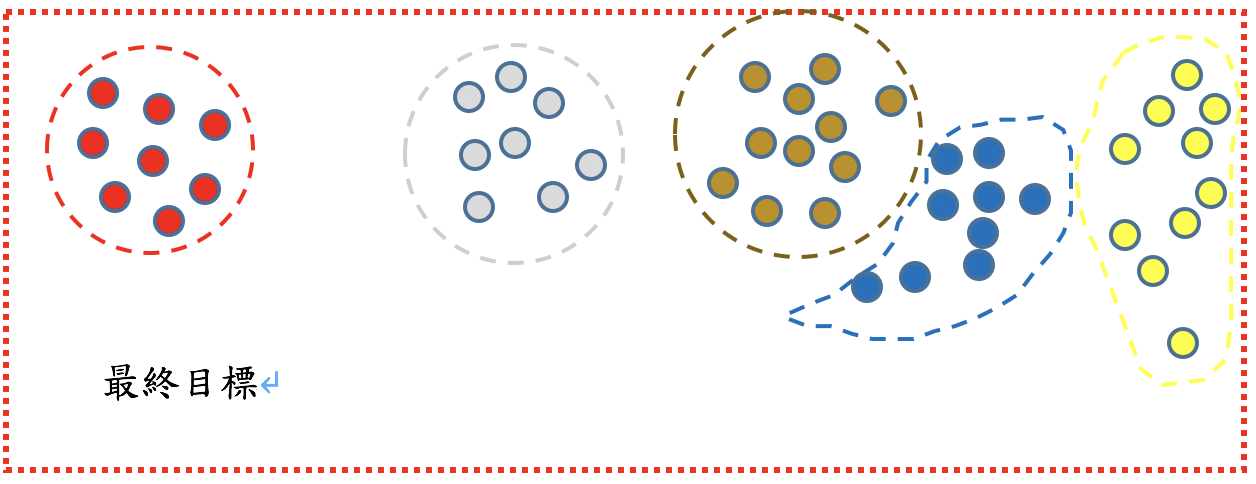
**˙步驟**

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。

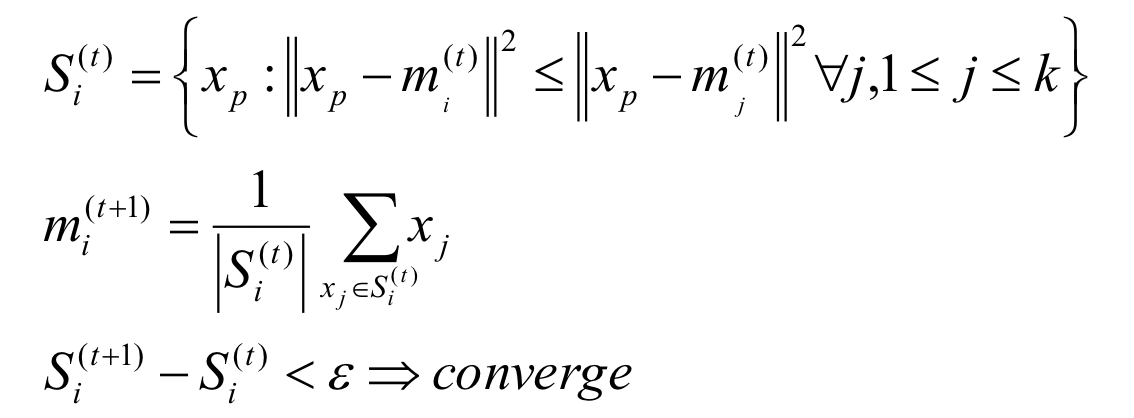


圖中 : 「更新」 後新的平均數

: 「舊」 的平均數



**˙演算法**

****

*xp*為某一資料位置  
*mi* 為第*i*群之平均值

*ε* 為誤差值

**˙優點**

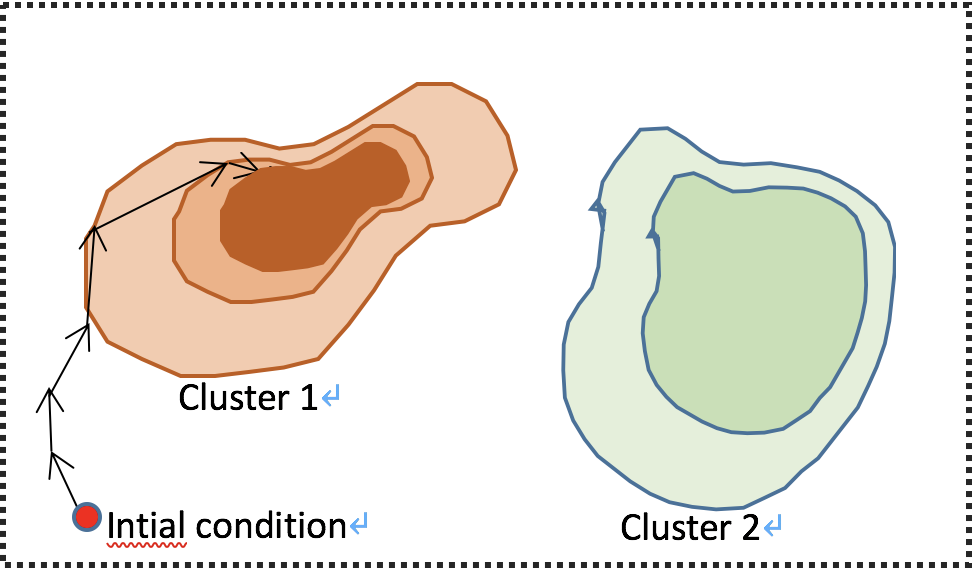
1. 以群內變異小、群間變異大為分群的核心概念。
2. 想法簡單，收斂速度快(因為與平方有關係)。
3. 不需要分佈假設。

**˙缺點**

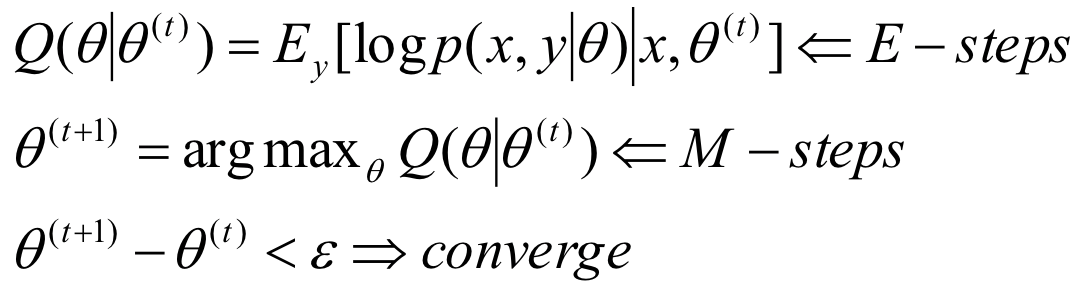
1. 實務上，global optimization的解不太容易求得，大部分皆為local optimization。
2. 最終收斂結果與起始值選取有關。
3. 倘若資料本身有重疊部分,則重疊部分的資料有可能被分到錯誤類別。
4. 易受到極端值影響，雖此缺陷有機會透過K-Mediods法改善，也就是改以中位數而非平均數去作為衡量距離的基準點，但同時收斂速度也會變得較慢。
5. 邊界是由直線分割的，有時在fit data的過程中較難達到精準的效果。
6. K值（群數）需要事先選定，選值的最適方法較難掌握，然而不合適的K值可能影響分類效果。
7. **EM for mixture normal**

**˙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**

****

註：*θ* 為參數估計值（可以表示成一個或一組參數）

˙**優點**

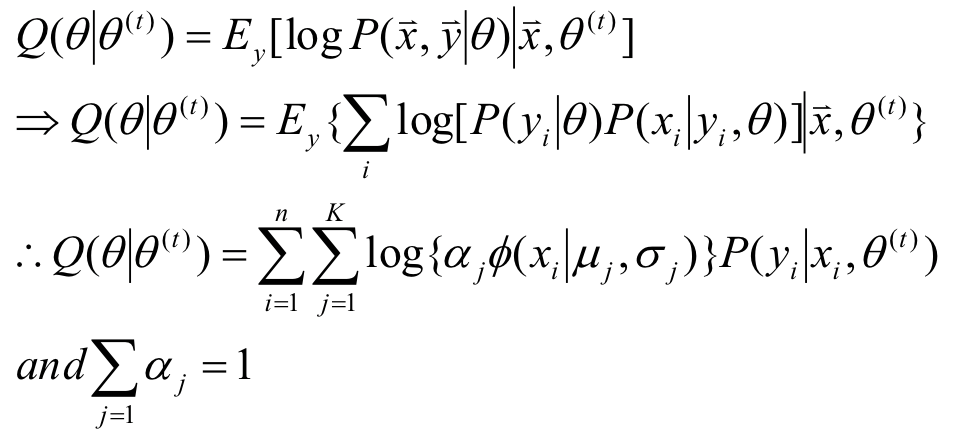
1. 因為EM演算法主要是利用likelihood function，因此對於有重疊性質的資料，會利用比較機率大小來判定該筆資料屬於那一群，因此錯判可能性較低。
2. 因為該演算法建立在常態模型之上，因此可以利用一些文獻上的方法去推估該筆資料有幾個modes。
3. EM演算法可利用於遺失值的插補，E-step是對遺漏值利用likelihood function做最佳的估計，M-step則求出MLE、再進行取代，重複迭代直到估計值的變化可以被忽略為止。

**˙缺點**

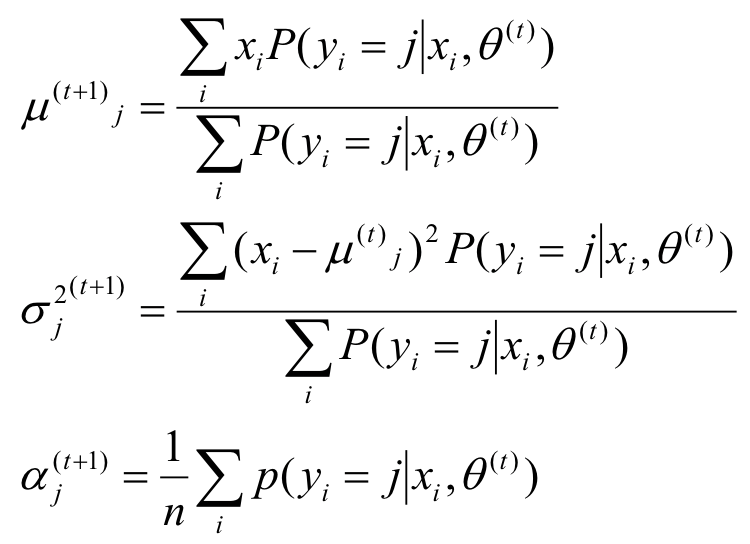
1. 起始值選取會直接影響最後收斂結果。
2. 實務上，其收斂結果大部分為local optimization。
3. 倘若這些樣本非i.i.d，其likelihood function不容易被推導。

**Remark: Case of Mixture Normal distribution**

1. E-steps



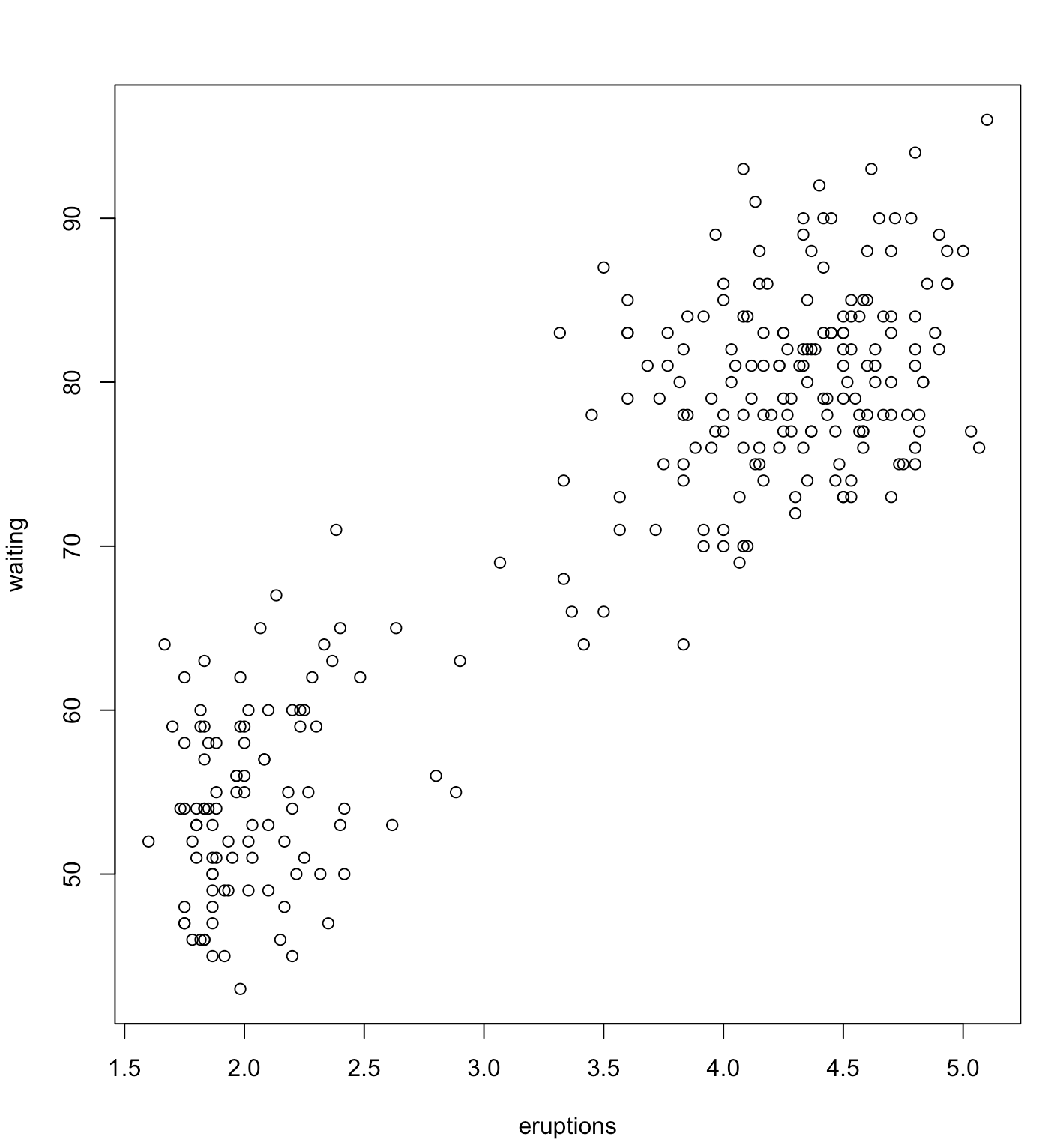
1. M-steps



**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。

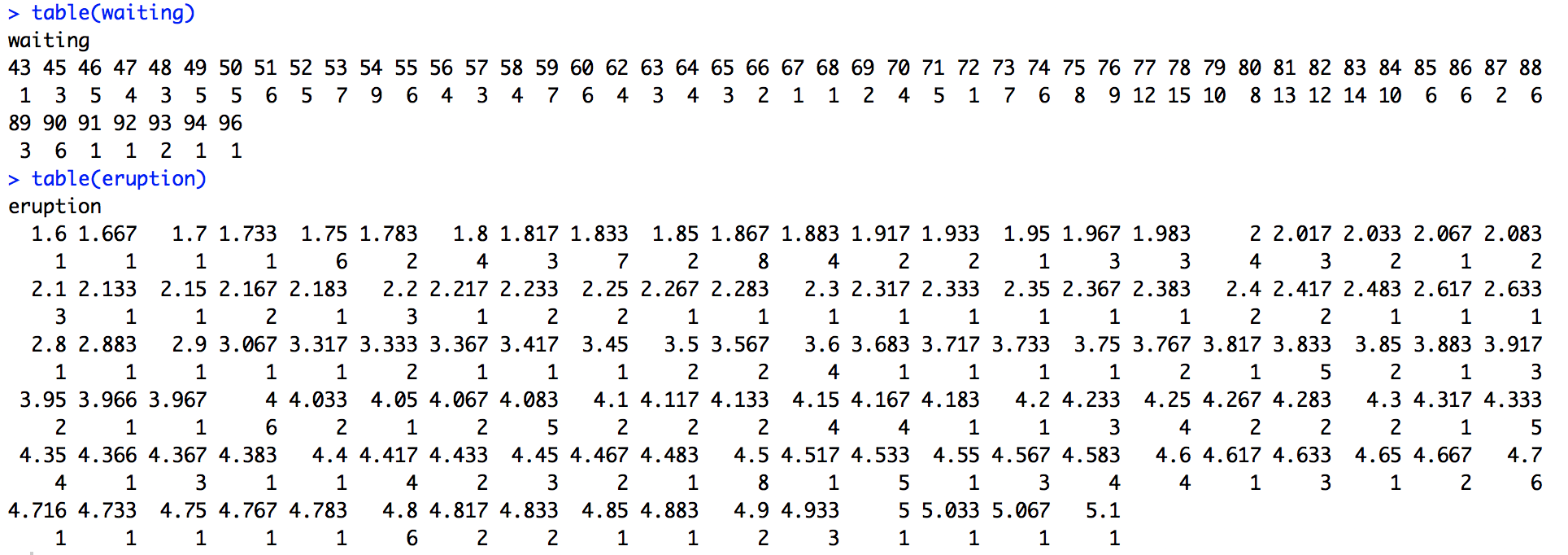
**˙Histograms & Frequency tables**

|  |  |
| --- | --- |
| Waiting | Eruption |
|  |  |

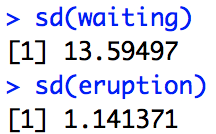
由上圖所示，我們可以發現兩個變數分佈皆呈現明顯的雙峰型態，與散佈圖所呈現的資訊相呼應。

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：



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：



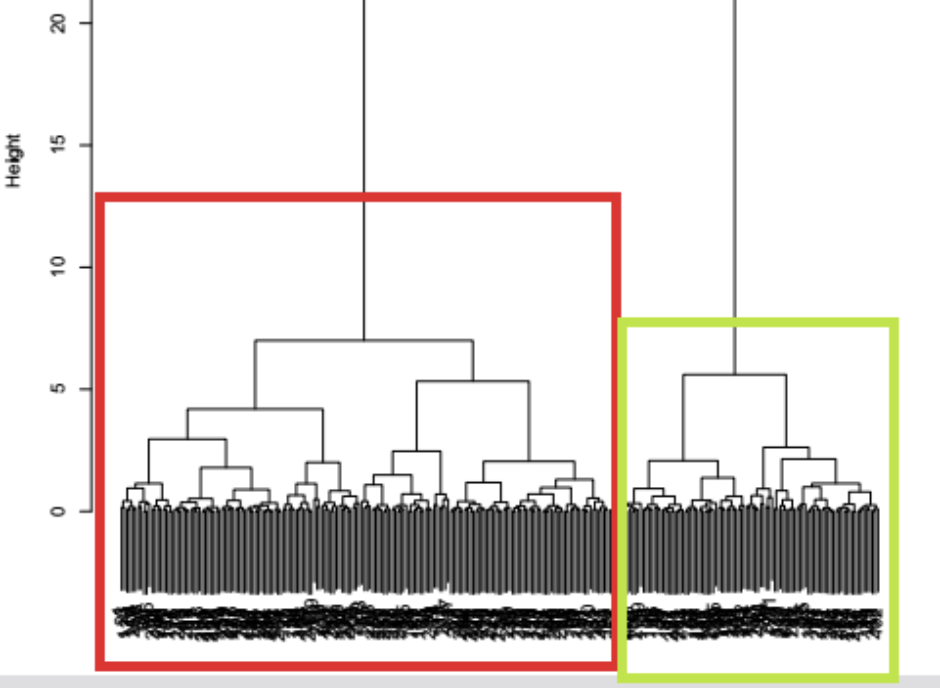
1. **By Hierarchical Clustering**

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** |
| **Ward method** |  |
| **Divisive method** |  |

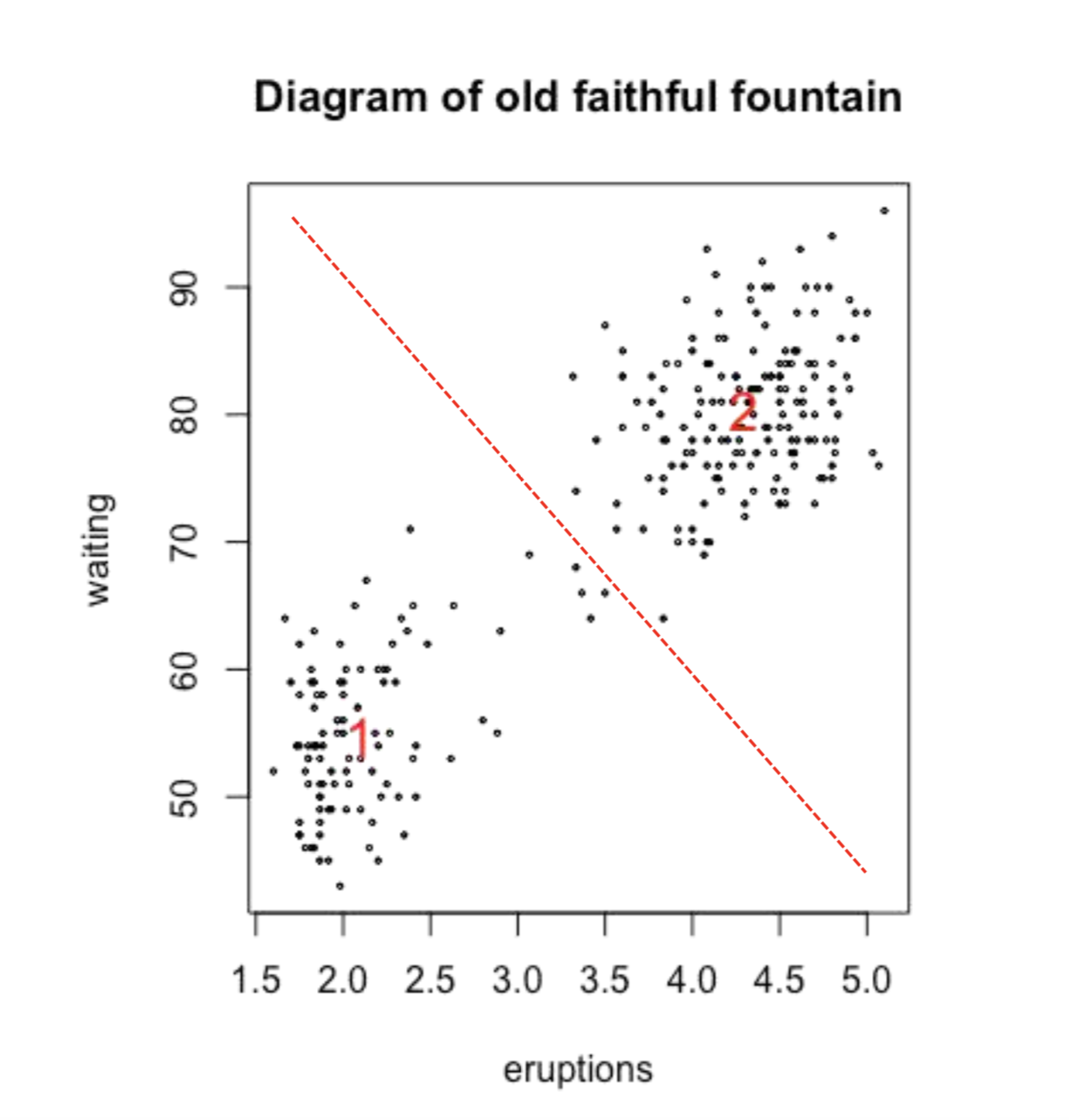
由以上Banner plot所示，無論使用哪種linkage，分為兩群依舊是我們的最佳選擇，為追求高標準的分群結果，我們比較了以上Hierarchical Clustering的其中兩種linkage，發現Ward的AC值趨近於1，因此我們決定以Ward method的分群結果為評斷標準。

Ward的分群結果如下圖所示（紅、綠兩群）：

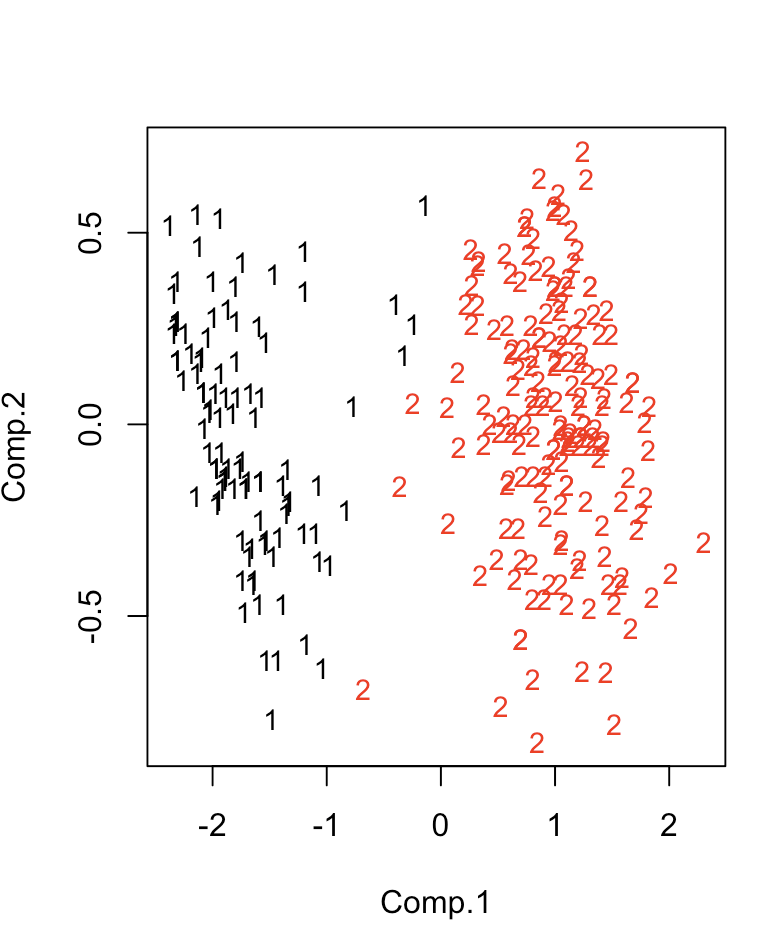


1. **By K-means**

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）：

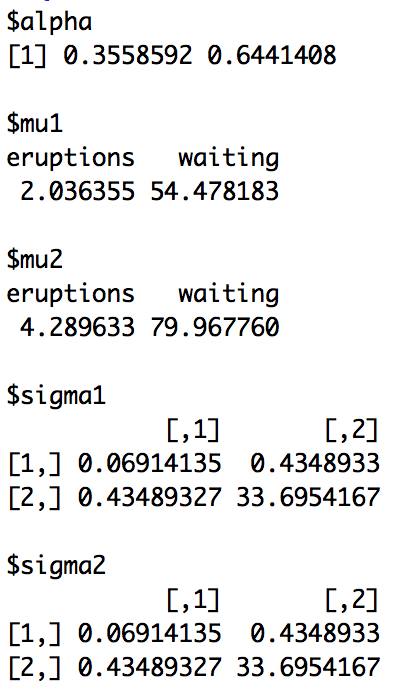


另外，我們也做了主成份分析，並將以上的分類結果投影在以前兩個主成份為軸的圖上：

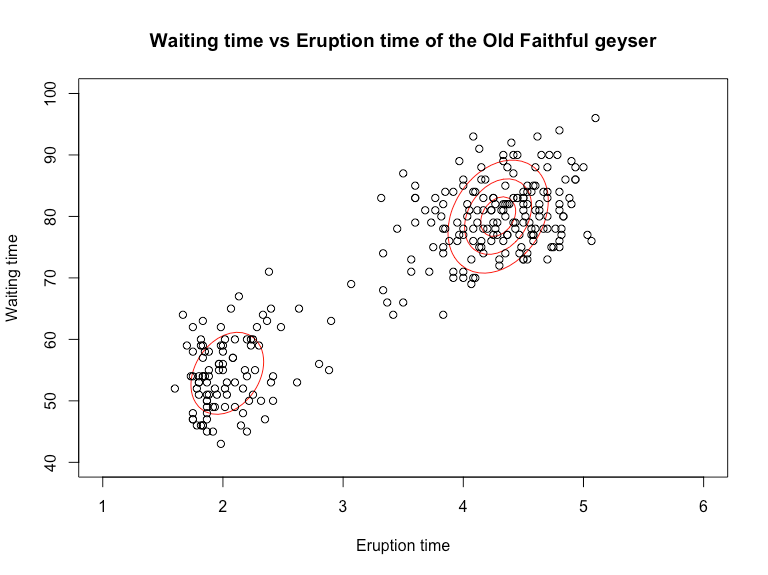


1. **By EM**

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：

****

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 |
| abalone | 0 | 0 | 0 | 0 | 0 | 0 |
| abbeel | 0 | 0 | 0 | 0 | 0 | 0 |
| abbott | 0 | 0 | 0 | 0 | 0 | 0 |
| abbreviate | 0 | 0 | 0 | 0 | 0 | 0 |
| abbreviated | 0 | 0 | 0 | 0 | 0 | 0 |
| abc | 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。

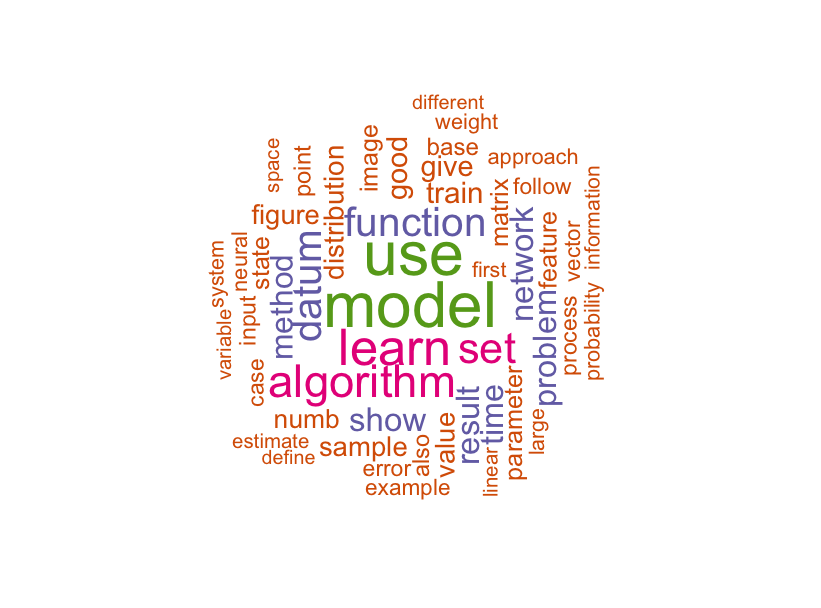
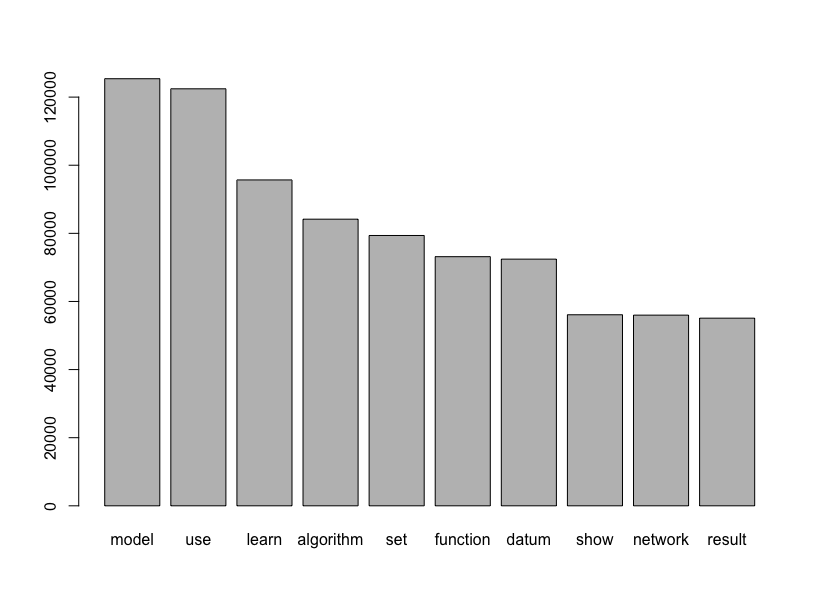
我們先檢測是否有單字數全為0的文章，發現共有7篇文章的單字數全為0，我們決定將這7篇文章移除以便分析，因此剩下5804篇。接著清除單字時態的問題（將同單字不同時態或詞性合併為同一單字），透過R package “textstem”，將這11463個單字縮減為8426個。

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：

1. 前2000個總出現次數的單字。

次數最多的為model 125399個，這100個中次數最少的單字為apply 17501個。其中前10名出現在最多的單字如下表所示：

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| model | use | learn | algorithm | set | function | datum | show | network | result |
| 125399 | 122437 | 95682 | 84159 | 79371 | 73146 | 72427 | 56092 | 55984 | 55095 |



**文字雲**

**直方圖**

我們使用這100個單字對這5804篇文章進行k-means分群並投影到PCA。

1. 使用tf-idf去選擇重要度較高的單字。

˙**詞頻**（term frequency，TF）

詞頻指的是某一個給定的詞語在該文件中出現的頻率，第 t個詞出現在第 ~d~ 篇文件的頻率記做。

**˙逆向文件頻率**（inverse document frequency，IDF）

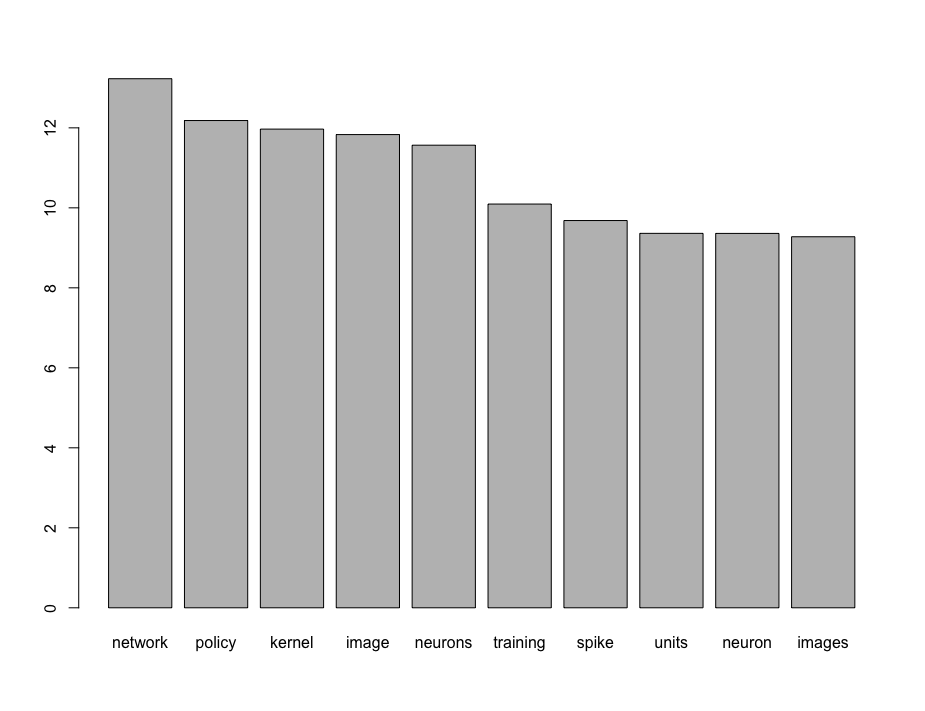
假設詞彙 t總共在  篇文章中出現過，則詞彙 t的 IDF 定義成。比如說，假設文字 1 總共出現在 25 篇不同的文件，則 。如果詞彙 t在非常多篇文章中都出現過，就代表  很大，此時 就會比較小。

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** |
| **model** | 0.000237 | 0.000431 | 0.000398 | 0.002418 | 0.000287 | 0.000094 |
| **use** | 0.000034 | 0.000032 | 0.000032 | 0.000055 | 0.000005 | 0.000027 |
| **learn** | 0.002797 | 0.003047 | 0.001253 | 0.002551 | 0.000967 | 0.001854 |
| **algorithm** | 0 | 0.000594 | 0.000962 | 0.000102 | 0.000849 | 0.001628 |
| **set** | 0.000076 | 0.000042 | 0.000103 | 0.000019 | 0.000119 | 0.000395 |
| **function** | 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，並取總合中前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：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| network | policy | kernel | image | neurons |
| 13.2283 | 12.1828 | 11.9686 | 11.8317 | 11.5664 |
| training | **spike** | **units** | **neuron** | **images** |
| 10.0947 | 9.6824 | 9.362 | 9.3606 | 9.2769 |



**直方圖**



**文字雲**

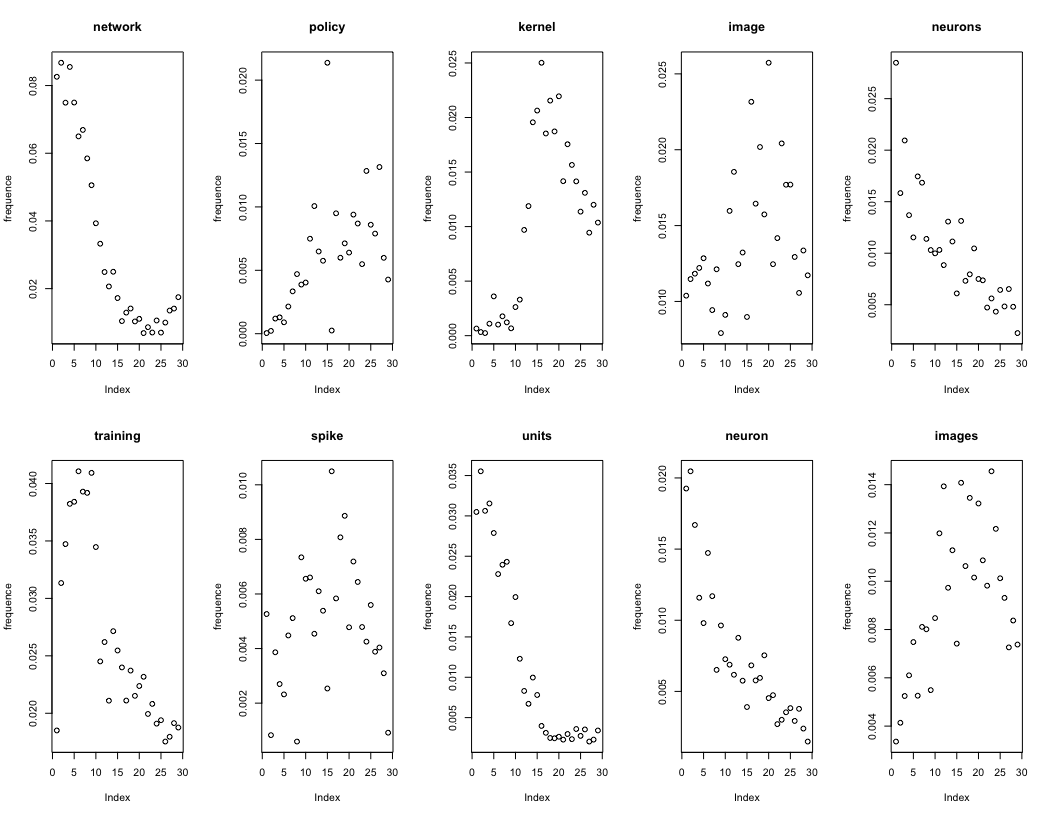
然而，考慮到可能有些文字每篇文章都出現，造成字數累計上比其他單字來得多，最後我們**採用case 2用tf-idf去取用最重要的前100個單字當作分群用資料**。

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** |
| network | 1647 | 1572 | 1571 | 2185 | 2165 | 1668 | 1922 | 1679 | 1501 | 1169 |
| policy | 1 | 4 | 25 | 33 | 26 | 55 | 96 | 135 | 115 | 120 |
| kernel | 13 | 6 | 5 | 28 | 104 | 26 | 51 | 35 | 20 | 78 |
| image | 207 | 208 | 248 | 312 | 371 | 287 | 271 | 348 | 235 | 271 |
| neurons | 568 | 287 | 439 | 350 | 333 | 448 | 484 | 327 | 306 | 297 |
| training | 369 | 568 | 728 | 977 | 1109 | 1054 | 1129 | 1125 | 1215 | 1025 |
| spike | 105 | 15 | 81 | 69 | 67 | 115 | 147 | 17 | 218 | 195 |
| units | 608 | 644 | 642 | 806 | 805 | 585 | 688 | 698 | 496 | 593 |
| neuron | 384 | 371 | 350 | 296 | 283 | 378 | 336 | 187 | 286 | 216 |
| images | 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 |
| network | 0.0826 | 0.0867 | 0.0749 | 0.0855 | 0.075 | 0.065 | 0.0669 | 0.0585 | 0.0506 | 0.0393 |
| policy | 0.0001 | 0.0002 | 0.0012 | 0.0013 | 0.0009 | 0.0021 | 0.0033 | 0.0047 | 0.0039 | 0.004 |
| kernel | 0.0007 | 0.0003 | 0.0002 | 0.0011 | 0.0036 | 0.001 | 0.0018 | 0.0012 | 0.0007 | 0.0026 |
| image | 0.0104 | 0.0115 | 0.0118 | 0.0122 | 0.0128 | 0.0112 | 0.0094 | 0.0121 | 0.0079 | 0.0091 |
| neurons | 0.0285 | 0.0158 | 0.0209 | 0.0137 | 0.0115 | 0.0175 | 0.0168 | 0.0114 | 0.0103 | 0.01 |
| training | 0.0185 | 0.0313 | 0.0347 | 0.0382 | 0.0384 | 0.0411 | 0.0393 | 0.0392 | 0.0409 | 0.0345 |
| spike | 0.0053 | 0.0008 | 0.0039 | 0.0027 | 0.0023 | 0.0045 | 0.0051 | 0.0006 | 0.0073 | 0.0066 |
| units | 0.0305 | 0.0355 | 0.0306 | 0.0315 | 0.0279 | 0.0228 | 0.0239 | 0.0243 | 0.0167 | 0.0199 |
| neuron | 0.0193 | 0.0205 | 0.0167 | 0.0116 | 0.0098 | 0.0147 | 0.0117 | 0.0065 | 0.0096 | 0.0073 |
| images | 0.0034 | 0.0041 | 0.0052 | 0.0061 | 0.0075 | 0.0053 | 0.0081 | 0.008 | 0.0055 | 0.0085 |

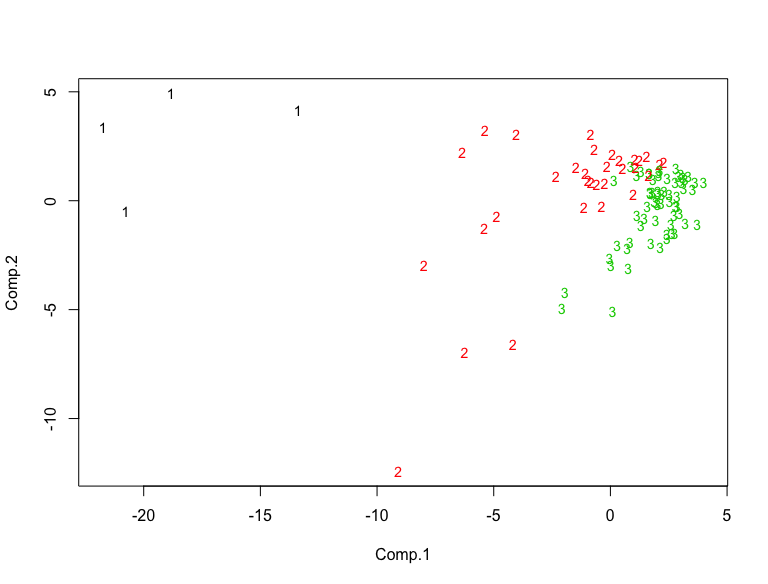
將這10個字在每一年度的使用頻率繪製散佈圖，發現有些字使用頻率逐年下降，有些字使用頻率反而上升，也有些字使用頻率持平。



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：

|  |  |
| --- | --- |
| **使用頻率上升** | Policy, kernel, image, matrix … 共 50個字 |
| **使用頻率下降** | Network, neurons, training, unit … 共33個字 |
| **使用頻率持平** | 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 |
| network | 2 |
| policy | 3 |
| kernel | 2 |
| image | 2 |
| neurons | 3 |
| training | 2 |
| spike | 3 |
| units | 3 |
| neuron | 3 |
| images | 3 |

**四、Appendix**

˙R code

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

˙參考資料來源

1. <https://taweihuang.hpd.io/2017/03/01/tfidf/>
2. <https://rpubs.com/saqib/DocumentClustering>
3. <https://commons.wikimedia.org/wiki/File:Em_old_faithful.gif>