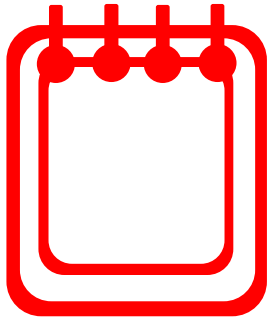
Statistical Computing and Simulation -Final Report



Professor: 翁久幸

Student: 統研一 董承

Procedure

- EM & K-means
- NIPS Clustering

EM & K-means

EM Algorithm

1. 初始化參數

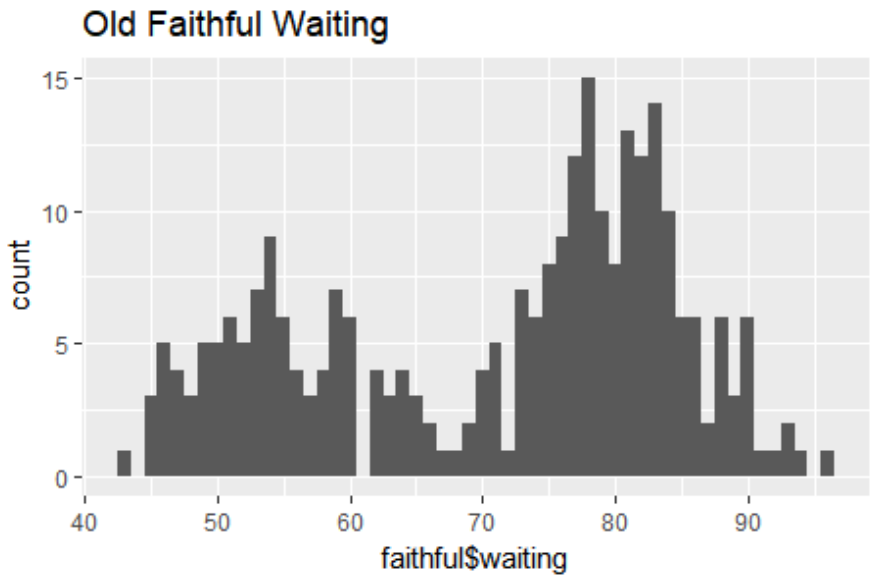
2. 重複直到收斂:

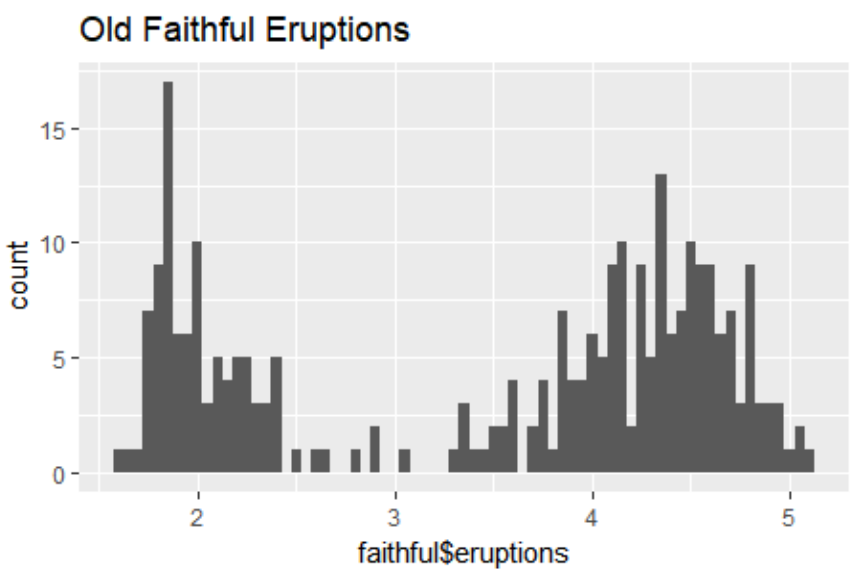
    E步驟：估計未知參數的期望值，給出當前的參數估計。

    M步驟：重新估計分布參數，以使得最大概似函數最大，得到未知變量的期望估計。

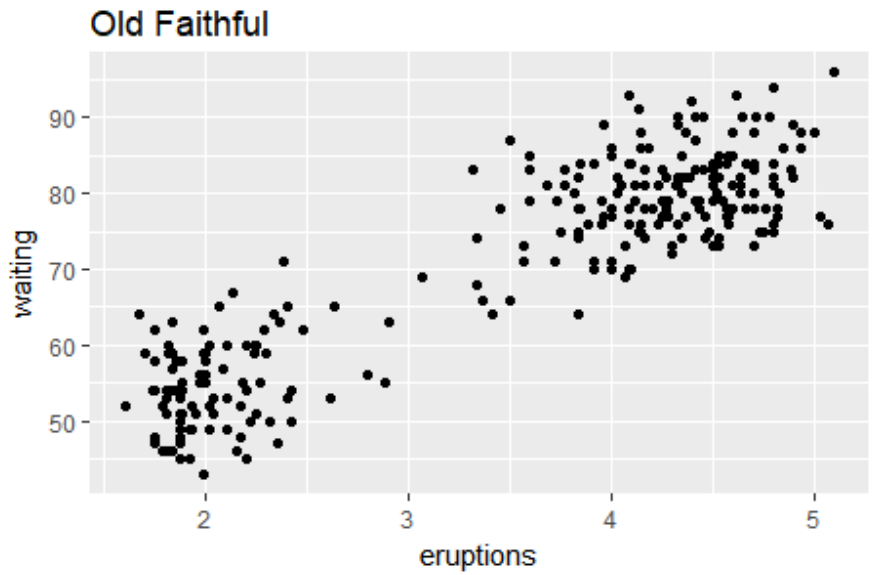
Old Faithful Dataset

- 利用美國黃石國家公園的老忠實噴泉作為比較K-means & EM 資料，包含兩個變數，分別是eruptions(噴發持續時間)、waiting(兩次噴發之間隔時間)，這裡利用視覺化探索資料得到資料分布圖形：





Old Faithful Dataset



- 可以看出這筆資料大致有兩個群，分別是eruptions(噴發持續時間)及waiting(噴發間隔時間)”短”，和eruptions(噴發持續時間)及waiting(噴發間隔時間)”長”。

Expectation–maximization

- 作為missing data的指標變數為

$$Y_i = \begin{cases} 1 & W_i \text{ belongs to distribution of shorter waiting times} \\ 0 & W_i \text{ belongs to distribution of longer waiting times} \end{cases}$$

- Yi服從二項分配，參數為p，接著，利用EM演算法找到Yi的期望值

- 透過重覆疊代得到Yi的參數p，我們可以得到期望值函式，此即為EM演算法的E-step ，再利用微分技巧找出每個參數，使得概似函數最大，為EM演算法的M-step
- 而初始值的假設是透過資料探索的圖做為參考，假設如下

$$p^{(0)} = 0.4, \mu_1^{(0)} = 40, \mu_2^{(0)} = 90, \sigma_1^{(0)} = 4, \sigma_2^{(0)} = 4$$

Expectation–maximization

- 分群結果:

Expectation–maximization

```
$params
                Comp.1    Comp.2
means      54.195591 80.353490
variances   4.956753  7.547971

means      54.194080 80.35394
variances   4.946035  7.53402

$clusterpobs
  probMembership
1      0.3083102
2      0.6916898
  probMembershipFlex
1          0.3087248
2          0.6912752
```

```
$params
                Comp.1    Comp.2
means      4.2733696 2.0186404
variances 0.4378327 0.2361129

means      2.0186078 4.2733434
variances 0.2356218 0.4370631

$clusterpobs
  probMembership
1      0.3484046
2      0.6515954
  probMembershipFlex
1          0.6507353
2          0.3492647
```

- 來自”短”群的waiting與eruptions之平均值分別為54.1956, 4.2733，

而資料分別由waiting與eruptions的變數下，

屬於”短”群的機率分別為0.3083, 0.3484，反之亦然。

K-means

- 給定K 個clusters,指標變數為

$$r_{i,k} = \begin{cases} 1 & \text{if } \mathbf{x}_i \text{ belongs to cluster } k \\ 0 & \text{otherwise} \end{cases}$$

- 由於K means 是屬於 hard clustering，

也就是每筆資料只有屬於或不屬於k-th 群的選擇，

每一點都被分配到一個且是唯一的一個群

K-means

- 首先，隨機取 k 個 seed point，將與第 i 個 seed point 的點距離最近的點分配至群 i，至所有點分配完為止。這邊所使用的距離測量方式為 Euclidean distance

$$\sum_{\mathbf{x}\in S_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2$$

- 指標變數可以改寫成：

$$r_{i,k} = \begin{cases} 1 & \text{if } d(\mathbf{x}_i, \boldsymbol{\mu}_k) = \min\{d(\mathbf{x}_i, \boldsymbol{\mu}_1), d(\mathbf{x}_i, \boldsymbol{\mu}_2), \ldots, d(\mathbf{x}_i, \boldsymbol{\mu}_K)\} \\ 0 & \text{otherwise} \end{cases}$$

K-means

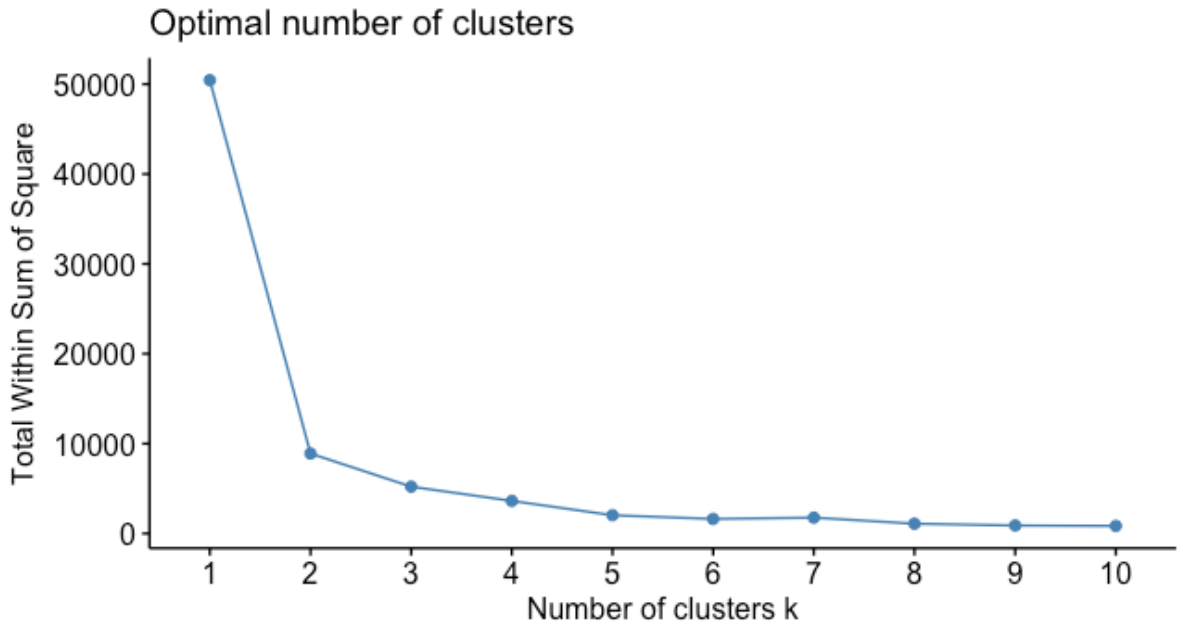
- 計算所形成的群中更新過後的centroids，k = 1, . . . ,K

$$\mu_k = \frac{1}{\sum_{i=1}^{N} r_{i,k}} \sum_{i=1}^{N} r_{i,k} \mathbf{x}_i$$

- 重複上述過程直到每個 centroids 收斂至一特定值。

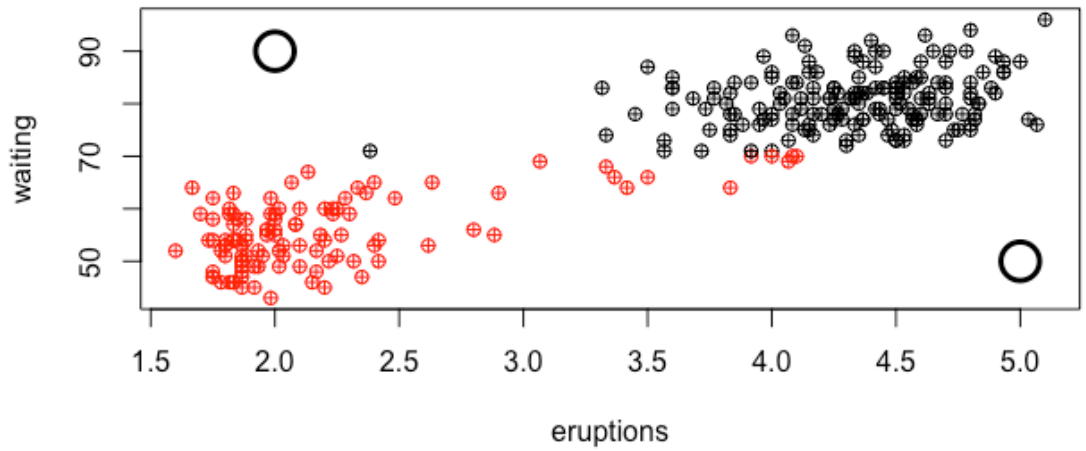
K-means

- 因為K 是執行之前要先給定的值，通常會用組內平方和(within group sum of squares)來決定K 的取值，以下是給定K=1到10 中的組內平方和：



K-means

- 隨著K 值增加，組內平方和會下降，但其中又以 K=2 時，下降幅度最大且與 K=3 時差異不會太大，因此我們選定 K=2，任取2個點(2,90) ,(5,50)作為起始位置，經過一次迭代，得到：

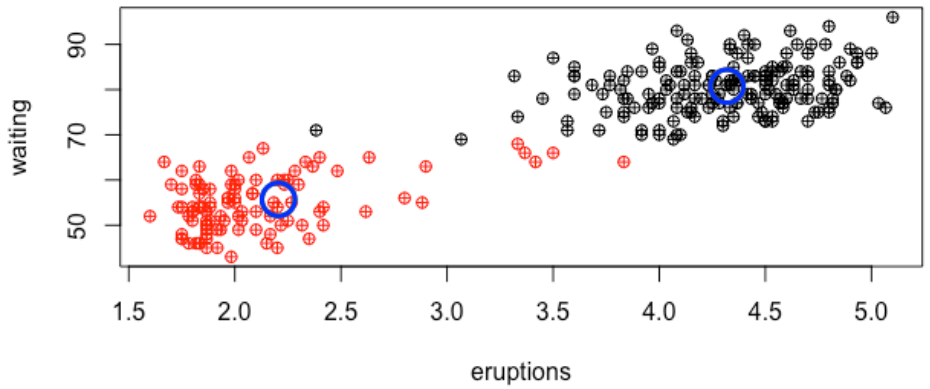


K-means

- 根據此次分群結果再重新計算此兩群內的新中心點：

1. .(2.205607 , 55.71028)　　2 (4.319255 , 80.74545)

- 計算其他所有點到這兩個中心分別的 Euclidean distance，取距離近的加入該群，第二次迭代後得到的結果如下：

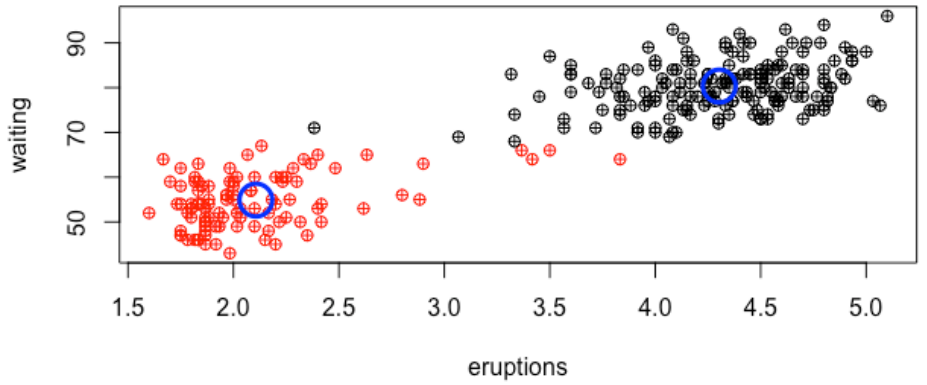


K-means

- 此次分群結果兩群內的新中心點為：

1. (2.106594 , 54.88119)　　2. (4.303573 , 80.35673)

- 執行第三次迭代得到以下結果：

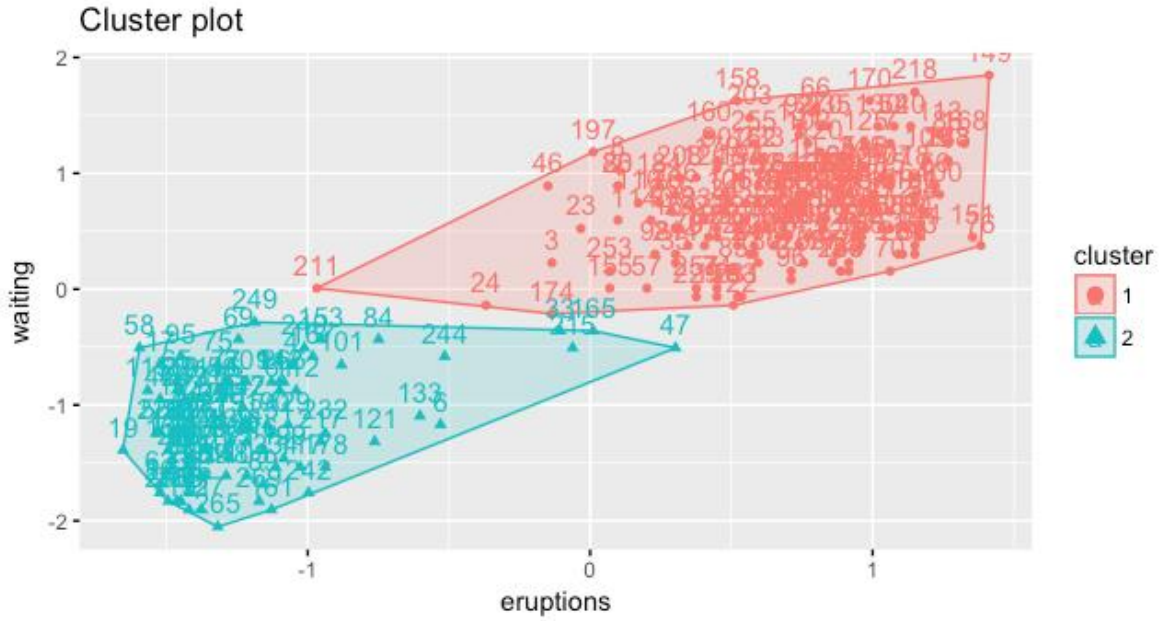


K-means

- 由此三次迭代可以發現中心點位置漸漸收斂，使用R 內建 kmeans() 執行 old faithful(老忠實)噴泉分群，得出以下視覺畫圖型及分群結果：

```
> kmeans.cluster$centers
  eruptions  waiting
1   4.29793 80.28488
2    2.09433 54.75000

> kmeans.cluster$size
[1] 172 100
> kmeans.cluster$withinss
[1] 5445.591 3456.178
```



Summary

- EM method優點：

1. 可靠地找到局部最優的收斂值
2. 比K-means穩定、準確

- EM method缺點：

1. 對初始值敏感
2. 計算複雜且收斂慢

- K-means 優點：

1. 原理簡單，實現容易
2. 收斂速度快
3. 算法的可解釋度比較強
4. 只有參數k 需要調整

- K means 缺點：

1. k 值不易選取
2. 採用迭代方法，得到的結果只是局部最優
3. 對outliers 較為敏感
4. 群集不能重疊
5. 數據分布情況、群集中心的初始位置都會影響重複次數

NIPS Clustering

NIPS

- 資料集為NIPS(Conference on Neural Information Processing Systems)神經資訊處理系統研討會的文件資料
- 包含11463個詞以及5812份文件的Term-Document Matrix
- 透過斷詞、移除停用字符，記錄每篇文件中出現詞的頻率，並且只保留那些出現次數高於50次的詞
- 稀疏矩陣(Sparse Matrix)

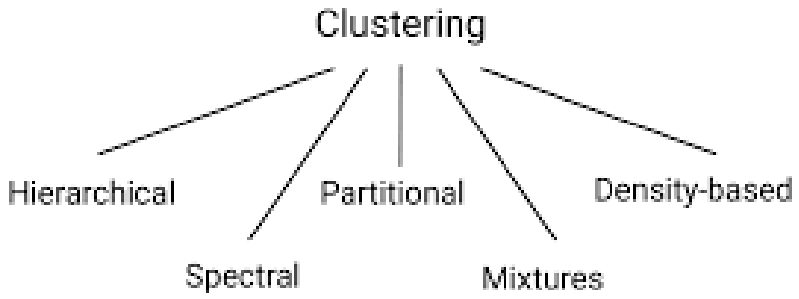
NIPS

- 資料集為NIPS(Conference on Neural Information Processing Systems)
  神經資訊處理系統研討
- 包含11463個詞以及5
- 透過斷詞、移除停用字
  出現次數高於50次的詞
- 稀疏矩陣(Sparse Mat

	A	B	C	D	E	F	G	H	I	J
1		1987_1	1987_2	1987_3	1987_4	1987_5	1987_6	1987_7	1987_8	1987_9
2	abalone	0	0	0	0	0	0	0	0	0
3	abbeel	0	0	0	0	0	0	0	0	0
4	abbott	0	0	0	0	0	0	0	0	0
5	abbreviate	0	0	0	0	0	0	0	0	0
6	abbreviate	0	0	0	0	0	0	0	0	0
7	abc	0	0	0	0	0	0	0	0	0
8	abeles	0	0	0	0	0	0	0	0	0
9	abernethy	0	0	0	0	0	0	0	0	0
10	abilistic	0	0	0	0	0	0	0	0	0
11	abilities	0	1	2	0	1	0	0	0	0
12	ability	2	0	0	0	0	1	0	0	0
13	ables	0	0	0	0	0	0	0	0	0
14	abnormal	0	0	0	0	0	0	0	0	0

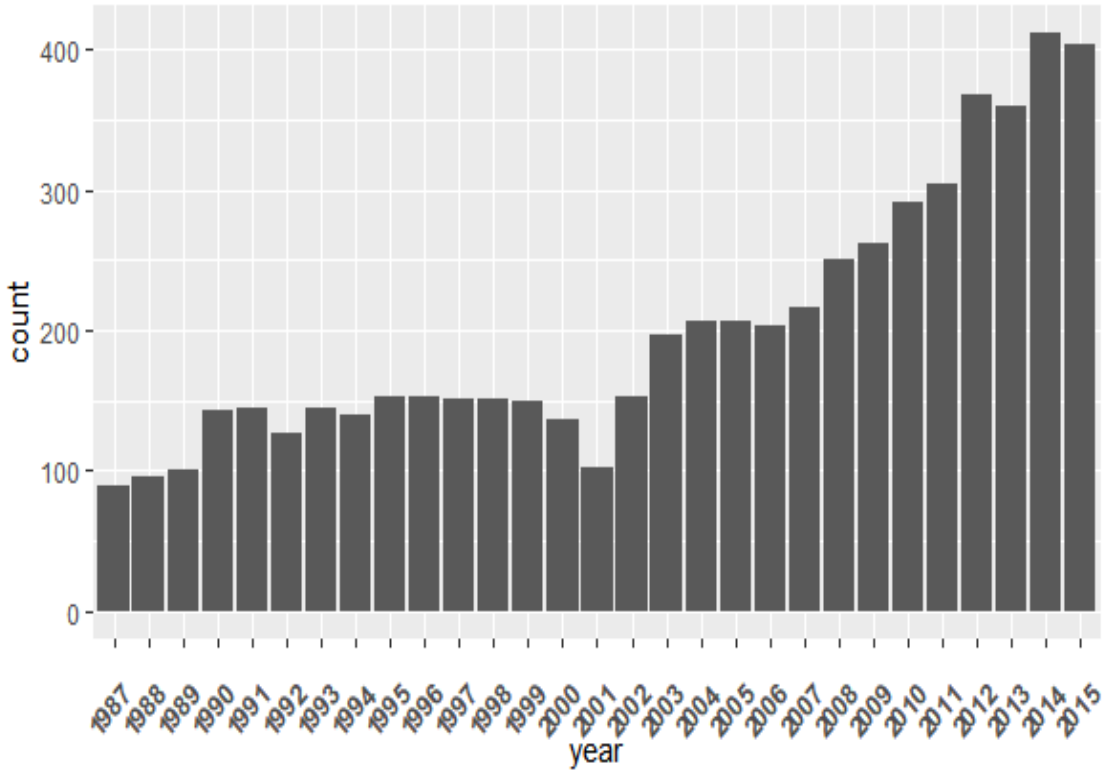
Target

- Cluster the documents by term : Documents with similar key terms are supposed to be in same group.



EDA(Exploratory Data Analysis)

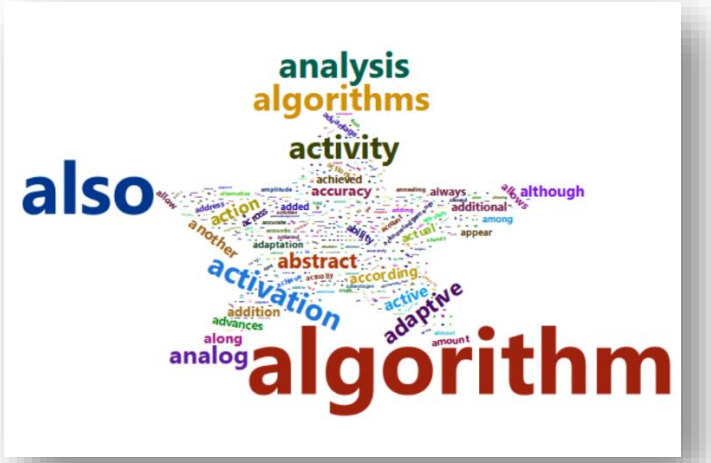
- Number of documents since 1987-2015



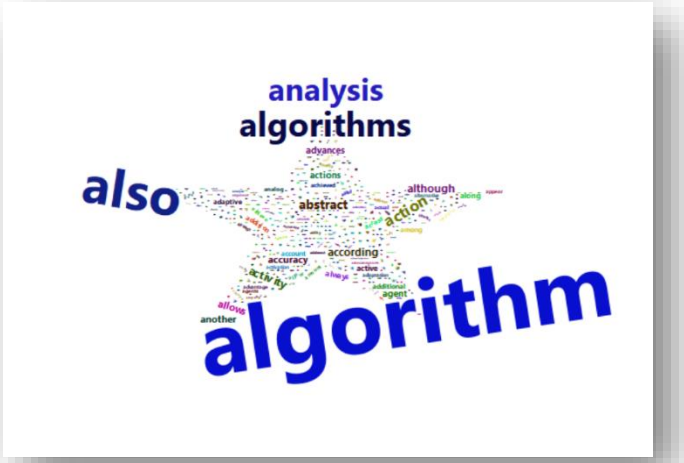
Wordcloud

- R package : wordcloud2

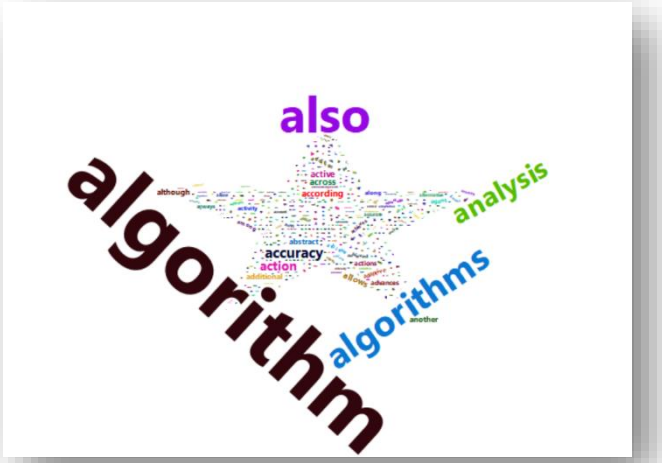
Since 1987-1995



Since 1996-2005



Since 2006-2015



TF-IDF

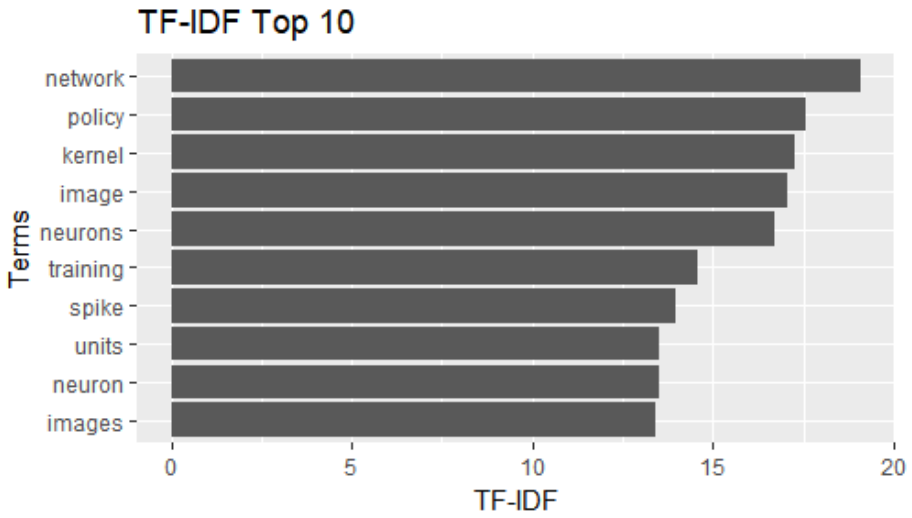
- TF-IDF(Term Frequency–Inverse Document Frequency)
- 詞頻（Term Frequency，TF）逆向文件頻率（Inverse Document Frequency，IDF）

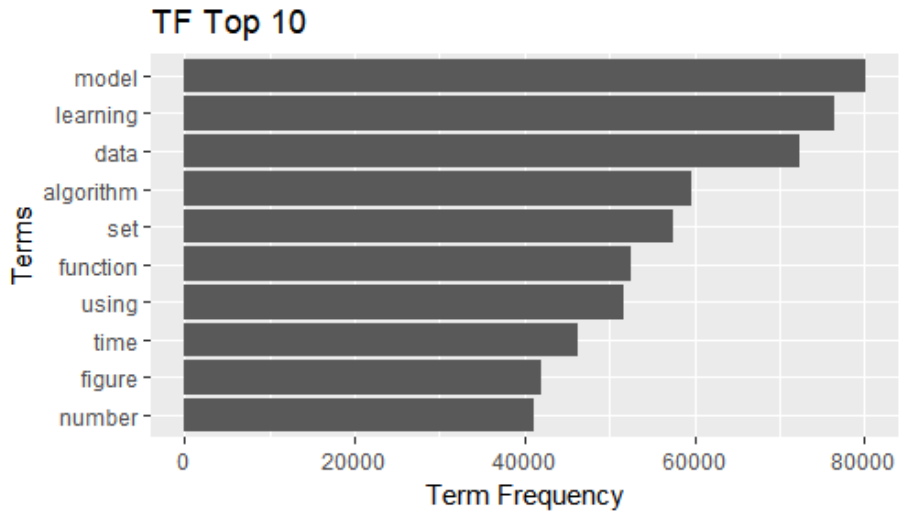
$$TFIDF=TF\times IDF=n\_w\text{^}d\times \log\_2〚N/N\_W〛$$

- 其中, $n_w\text{^}d$表示在文章d中，文字w出現的次數，N表示總共有幾篇文章，N_W表示有文字w的文章有幾篇。

TF-IDF

- R package : tm (text mining)
- By sum the TF-IDF given all documents

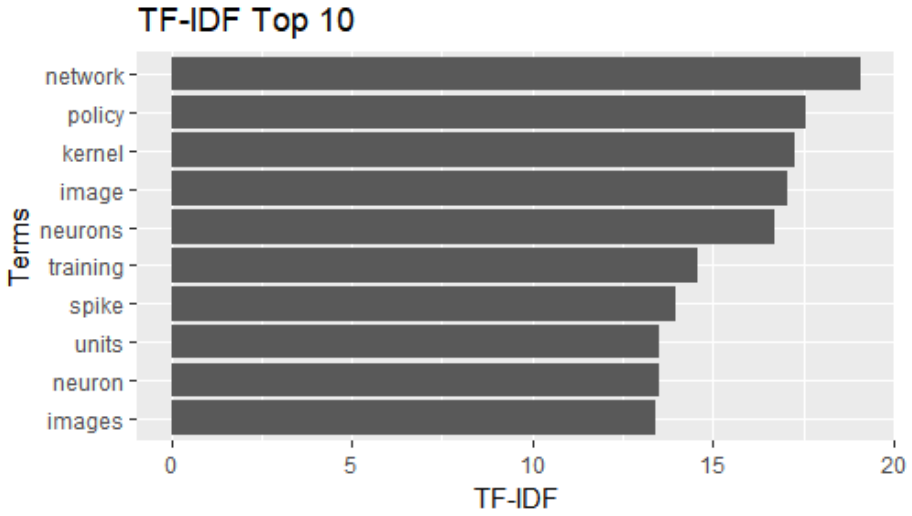




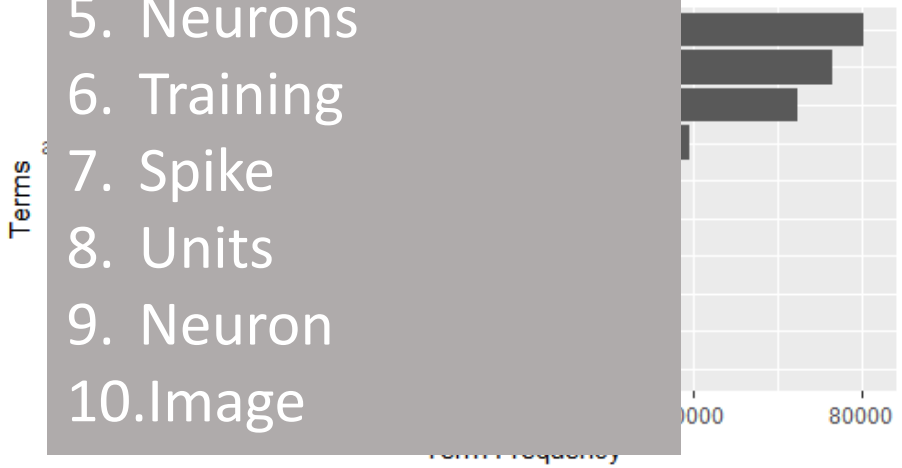
$$\begin{bmatrix} & \end{bmatrix}_{11463\times5811}$$

TF-IDF

- R package : tm (text mining)
- By sum the TF-IDF given all documents



1. Network
2. Policy
3. Kernel
4. Image
5. Neurons
6. Training
7. Spike
8. Units
9. Neuron
10. Image



11463×5811

TF-IDF

- See correlated words

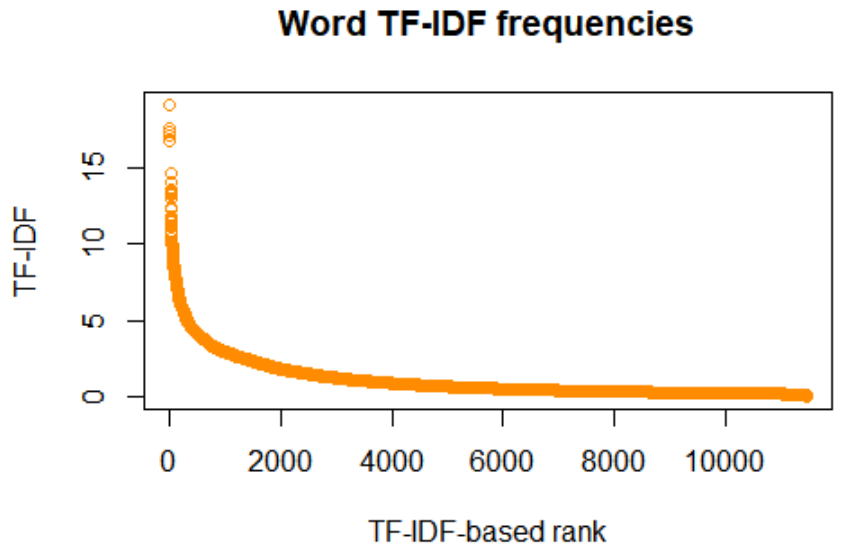
```
> # find correlated terms
> findAssocs(tdm.tfidf, "network", 0.5)
$network
networks   neural
    0.65     0.52

> findAssocs(tdm.tfidf, "spike", 0.5)
$spike
spikes trains
  0.71   0.67

> findAssocs(tdm.tfidf, "units", 0.5)
$units
  unit hidden
  0.69   0.54
```

Data Manipulation

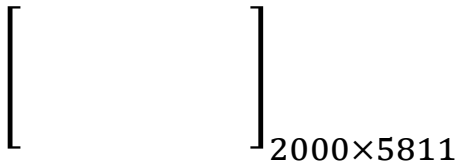
- Sum the value of TF-IDF given all documents



Data Manipulation

- Stemming & Lemmatisation
- R command : textstem::lemmatize_words()
- c(“slowly” , “ran” , “seen” , "eats") %>% lemmatize_words()

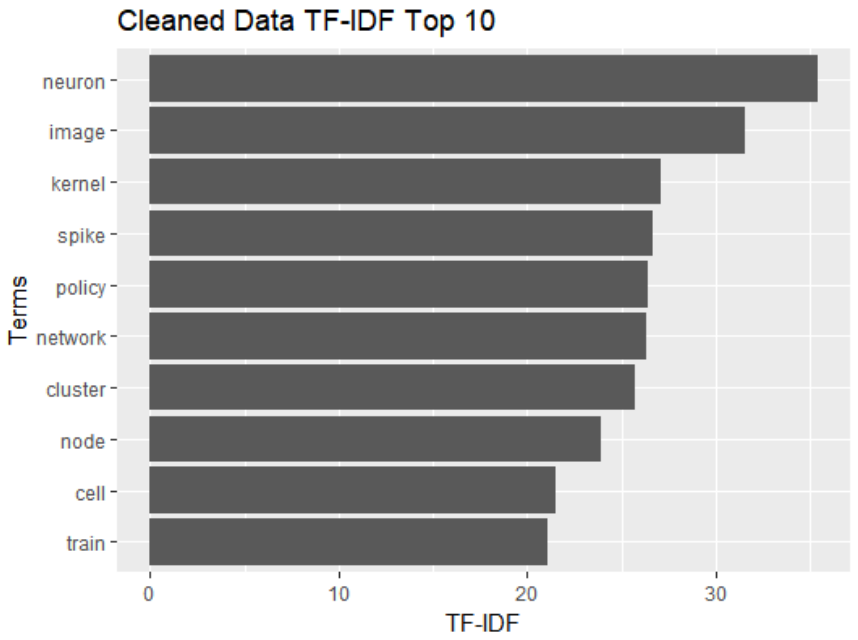
  c("slowly", "run", "see", "eat")



- R command : plyr::ddply

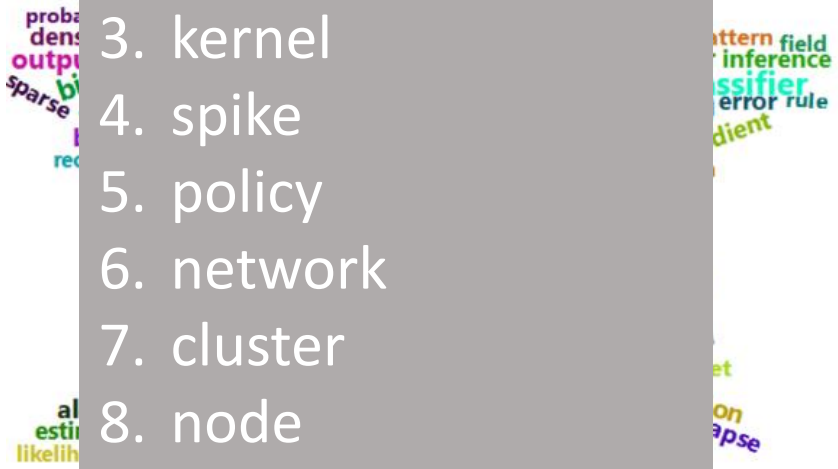
Data Manipulation

- WordCloud & Top 10 TF-IDF Terms

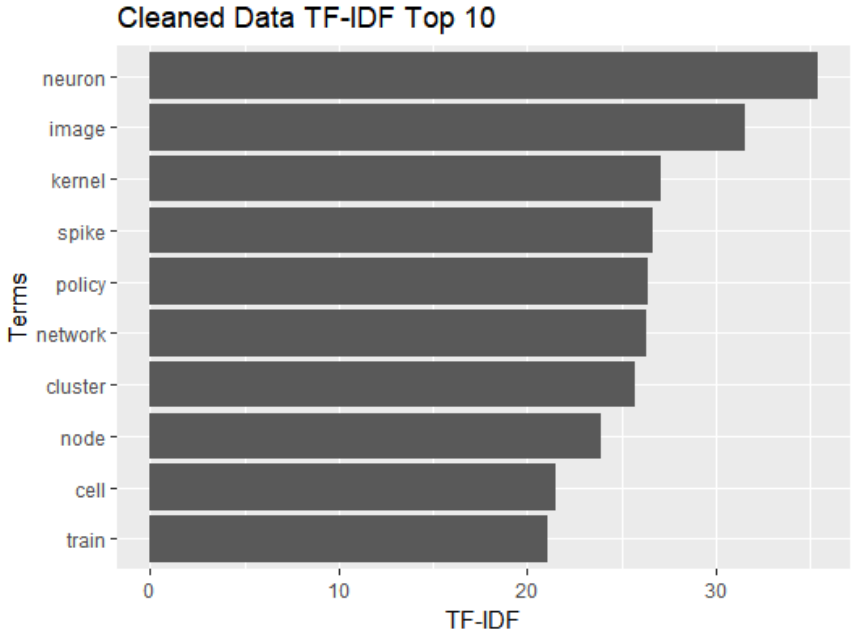


Data Manipulation

- WordCloud & Top 10 TF-IDF Terms



1. neuron
2. image
3. kernel
4. spike
5. policy
6. network
7. cluster
8. node
9. call
10. train



Vector Space Model

- 餘弦相似性(Cosine Similarity)是通過測量兩個向量的夾角的餘弦值來度量它們之間的相似性。0度角的餘弦值是1 (兩個向量有相同的指向時)；當兩個向量夾角為90°時，餘弦相似度的值為0；而兩個向量指向完全相反的方向時，餘弦相似度的值為-1。其計算方式如下

$$\text{Similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}}$$

$$\begin{bmatrix} \\ \\ \end{bmatrix}_{5811 \times 5811}$$

Vector Space Model

- Top 20 Documents Related to X1987_1

```
> aa[5792:5811,1:2]  # see top 20 documents
related to X1987_1
            Docs
Docs           X1987_1   X1987_2
  X2003_158 0.4181841 0.3280711
  X1992_123 0.4187696 0.2599045
  X2002_133 0.4204830 0.2879627
  X2005_94  0.4226493 0.3560670
  X2015_230 0.4356350 0.5413217
  X2000_1   0.4404999 0.3223251
  X2000_2   0.4409303 0.3220780
  X2006_77  0.4445143 0.4487613
  X1997_100 0.4486664 0.5279541
  X2014_120 0.4549942 0.2833302
  X1993_65  0.4638252 0.4223435
  X2007_52  0.4646202 0.4228044
  X1995_30  0.4720592 0.3845963
  X1999_16  0.4723698 0.3793990
  X1992_71  0.4899850 0.3513951
  X1993_62  0.4958996 0.3648531
  X1987_30  0.5029856 0.3273626
  X1992_100 0.5141574 0.4492241
  X1990_143 0.5305098 0.4073333
  X1987_1   1.0000000 0.3300572
```

Clustering

- 階層式分群法（Hierarchical Clustering）

Names	Formula
Euclidean distance	$\|a-b\|_2 = \sqrt{\sum_i (a_i - b_i)^2}$
Squared Euclidean distance	$\|a-b\|_2^2 = \sum_i (a_i - b_i)^2$
Manhattan distance	$\|a-b\|_1 = \sum_i \|a_i - b_i\|$
Maximum distance	$\|a-b\|_\infty = \max_i \|a_i - b_i\|$
Mahalanobis distance	$\sqrt{(a-b)^\top S^{-1}(a-b)}$ where *S* is th

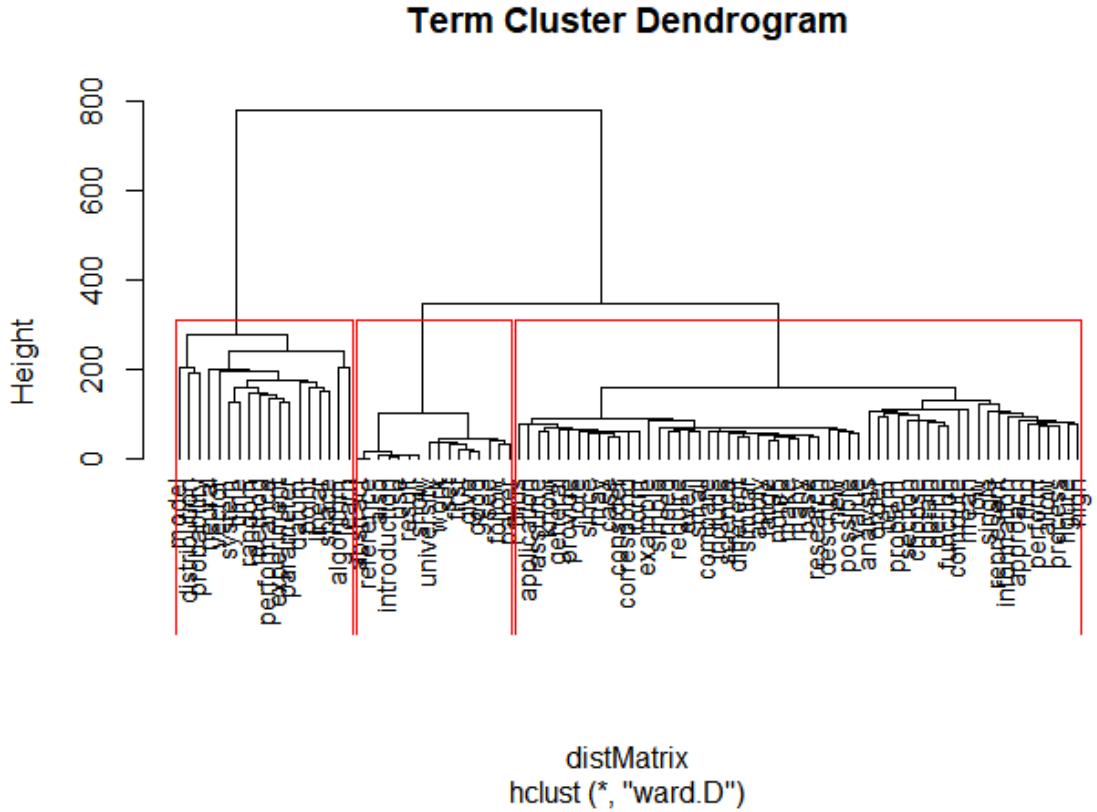
Names	Formula
Maximum or complete-linkage clustering	$\max\{\, d(a,b) : a \in A,\, b \in B \,\}.$
Minimum or single-linkage clustering	$\min\{\, d(a,b) : a \in A,\, b \in B \,\}.$
Mean or average linkage clustering, or UPGMA	$\frac{1}{\|A\|\|B\|}\sum_{a\in A}\sum_{b\in B} d(a,b).$
Centroid linkage clustering, or UPGMC	$\|c_s - c_t\|$ where c_s and c_t are the centroids of clus respectively.
Minimum energy clustering	$\frac{2}{nm}\sum_{i,j=1}^{n,m}\|a_i - b_j\|_2 - \frac{1}{n^2}\sum_{i,j=1}^{n}\|a_i - a_j\|_2 -$

where *d* is the chosen metric. Other linkage criteria include:

- The sum of all intra-cluster variance.
- The increase in variance for the cluster being merged (Ward's criterion).[7]
- The probability that candidate clusters spawn from the same distribution function (V-linkage).
- The product of in-degree and out-degree on a k-nearest-neighbour graph (graph degree linkage).

Clustering

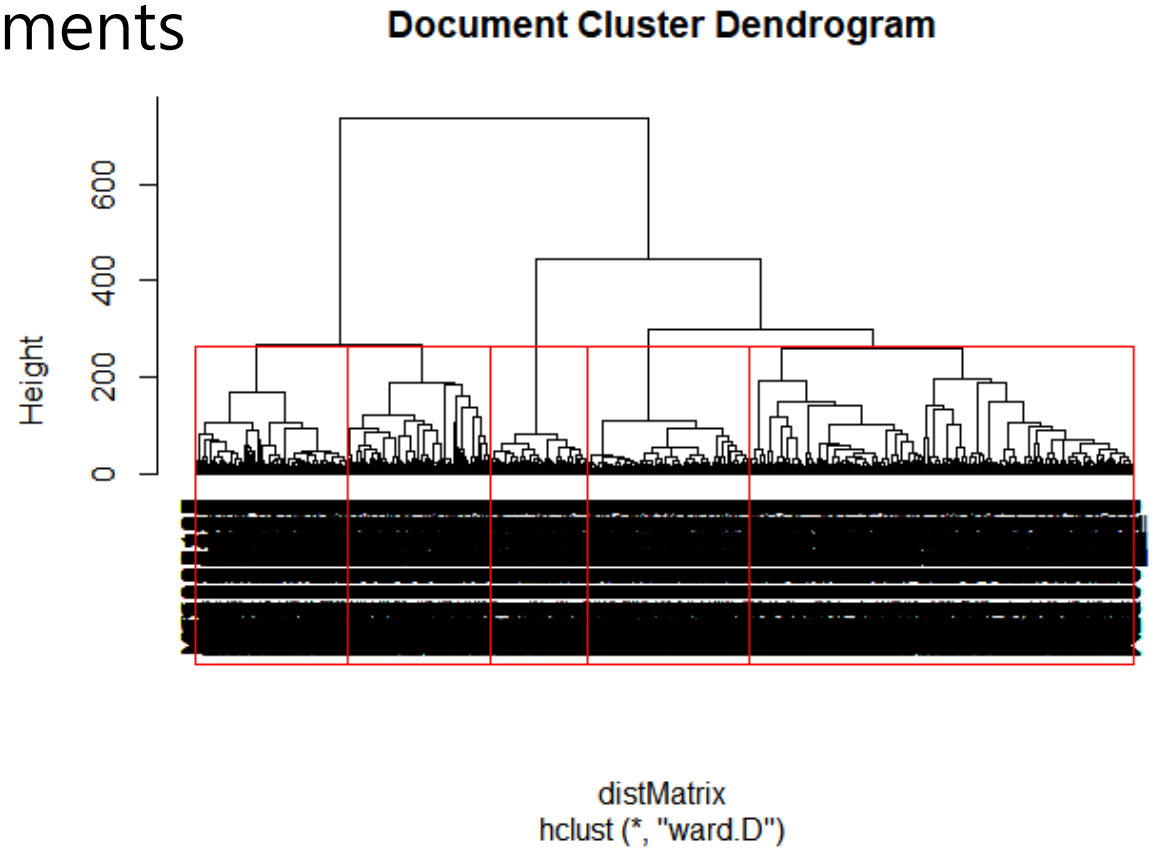
- Hierarchical clustering on Terms



Clustering

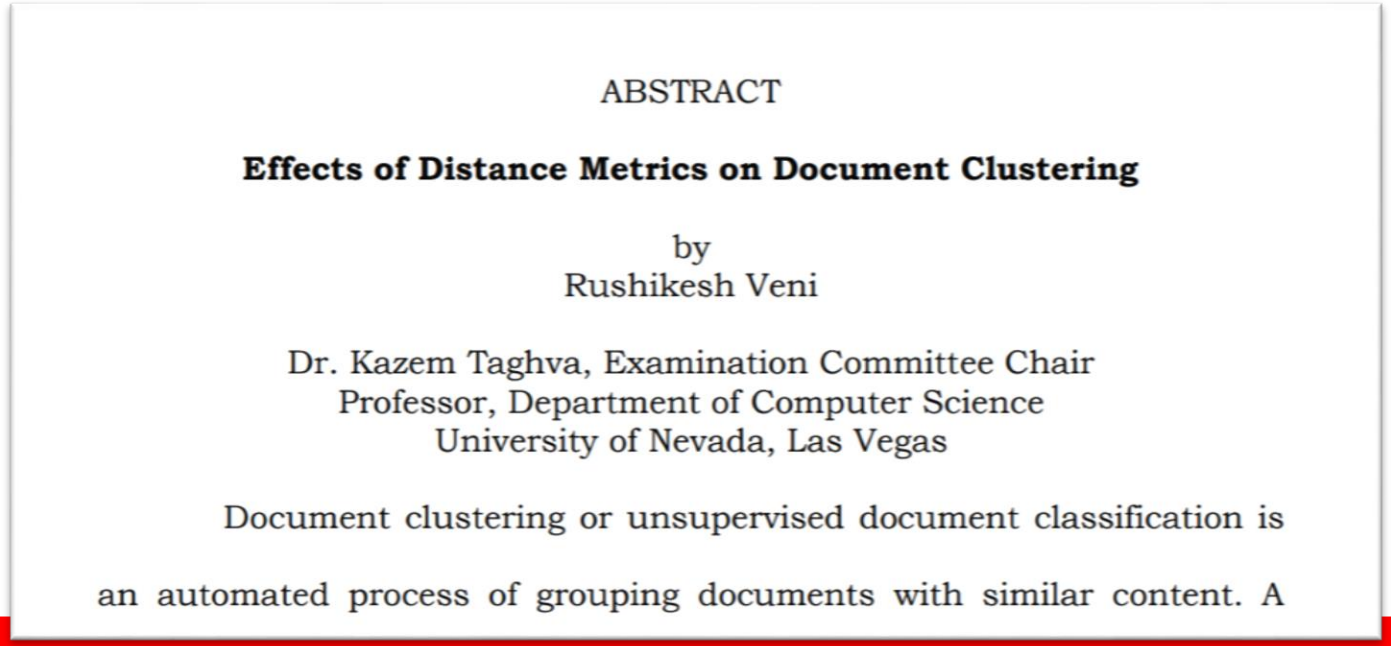
- Hierarchical clustering on Documents

```
> bb[which(cut.h.cluster == 1),1] %>% names()  # X1987_1
  [1] "X1987_1"   "X1987_5"   "X1987_7"   "X1987_9"   "X1987_10"
  [6] "X1987_11"  "X1987_19"  "X1987_25"  "X1987_27"  "X1987_33"
 [11] "X1987_34"  "X1987_37"  "X1987_39"  "X1987_41"  "X1987_42"
 [16] "X1987_43"  "X1987_45"  "X1987_46"  "X1987_49"  "X1987_51"
 [21] "X1987_55"  "X1987_57"  "X1987_58"  "X1987_63"  "X1987_66"
 [26] "X1987_68"  "X1987_79"  "X1987_81"  "X1987_88"  "X1987_90"
 [31] "X1988_15"  "X1988_21"  "X1988_22"  "X1988_23"  "X1988_24"
 [36] "X1988_25"  "X1988_29"  "X1988_33"  "X1988_35"  "X1988_38"
 [41] "X1988_39"  "X1988_42"  "X1988_46"  "X1988_51"  "X1988_53"
 [46] "X1988_61"  "X1988_62"  "X1988_64"  "X1988_65"  "X1988_66"
 [51] "X1988_73"  "X1988_74"  "X1988_80"  "X1988_81"  "X1988_84"
 [56] "X1988_90"  "X1988_92"  "X1988_93"  "X1989_9"   "X1989_12"
 [61] "X1989_13"  "X1989_16"  "X1989_20"  "X1989_23"  "X1989_25"
 [66] "X1989_30"  "X1989_32"  "X1989_42"  "X1989_46"  "X1989_47"
 [71] "X1989_57"  "X1989_58"  "X1989_61"  "X1989_62"  "X1989_73"
 [76] "X1989_77"  "X1989_79"  "X1989_85"  "X1989_86"  "X1989_90"
 [81] "X1990_1"   "X1990_4"   "X1990_7"   "X1990_8"   "X1990_13"
 [86] "X1990_32"  "X1990_33"  "X1990_34"  "X1990_40"  "X1990_41"
 [91] "X1990_42"  "X1990_44"  "X1990_48"  "X1990_56"  "X1990_61"
```



Summary

- Data manipulation 清理副詞
- NLP 分群的部分可以做語意分析(LSA)
- Effects of Distance Metrics on Document Clustering



ABSTRACT

Effects of Distance Metrics on Document Clustering

by
Rushikesh Veni

Dr. Kazem Taghva, Examination Committee Chair
Professor, Department of Computer Science
University of Nevada, Las Vegas

Document clustering or unsupervised document classification is

an automated process of grouping documents with similar content. A

THANK YOU

:)