TEXT MINING

Lecture 08

TEXT SIMILARITY

KEUNGOUI KIM

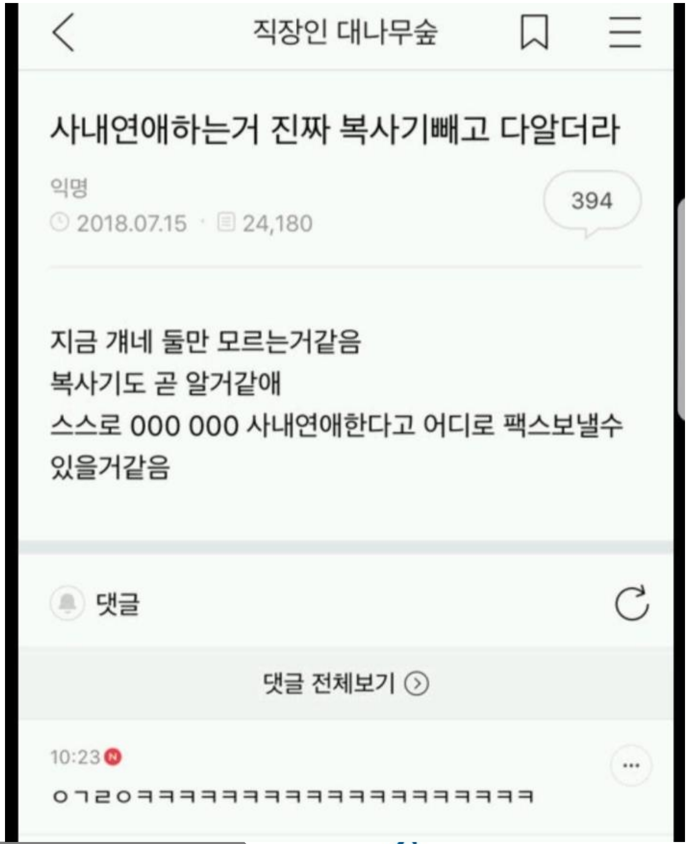
awekim@handong.edu

Text Co-occurrence

Co-occurrence Relations

- Co-occurrence relations
  - Co-occurrence: appears at the same time
  - Even if we do not know all about someone's human relationships, we can still guess who the closest person is → Someone with whom you talk frequently or someone with whom you spend time frequently
  - A high co-occurrence of two indicates either "closedness" or "similarity"

Everyone knows..
How?



https://www.youtube.com/watch?v=3XVqHs_SYe8
http://www.enuri.com/knowcom/detail.jsp?kbno=1675848

- Text co-occurrence relations
  - Often used in text analysis
  - With the text co-occurrence matrix, the relationship between texts can be observed
  - Ex) SNS replies
    - A: “That is so cool.” / B: “That sounds perfect.”

Matrix (DTM)

	that	is	so	…
A	1	1	1	…
B	1	0	0	…
…	…	…	…	…

Matrix (TDM)

	A	B	…
that	1	1	…
is	1	0	…
…	…	…	…

Pair table

From	To	Weight
that	is	1
is	so	1
so	cool	1

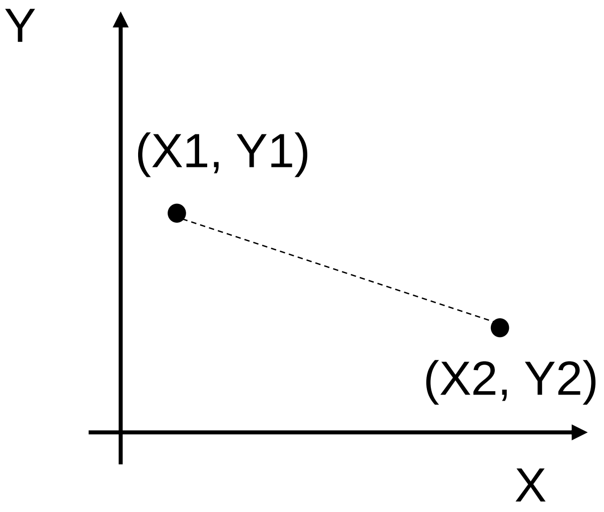
Text Co-occurrence

- Concept of co-occurrence
  - Frequency of occurrence of two elements
  - A higher co-occurrence between two elements indicates that two elements appear many times at the same time
  - Using the concept of co-occurrence, we can find which terms appear together more frequently.

- From the fact that terms appear together more frequently, we can assume that they are "more related"

Text Similarity

- Correlation
  - Correlation between variables (Pearson correlation)
  - In text analysis, correlation between terms or correlation between documents is used

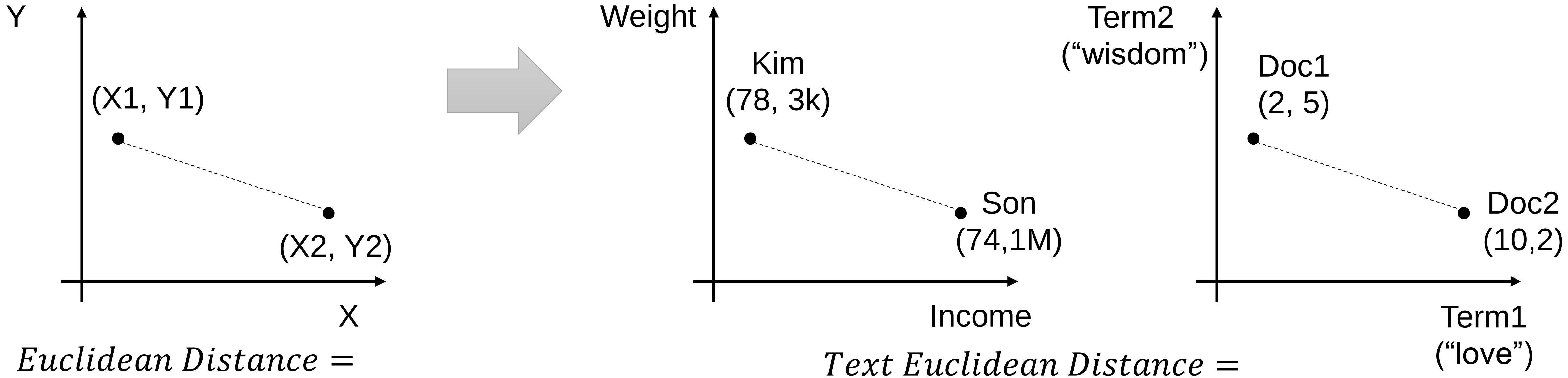


$$r = \frac{cov(X,Y)}{\sigma_X \sigma_Y} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

Negative correlation　　Zero (linear) correlation　　Positive correlation

-1-----------------------------0-----------------------------1

- Text similarity
  - In data analysis, the similarity between variables is understood with the concept of data dimension
  - Euclidean distance is the most simple and intuitive way of measuring similarity
  - Euclidean distance: the distance between two points in Euclidean space
  - Shorter the distance → greater similarity



$$Euclidean\ Distance = \sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2}$$

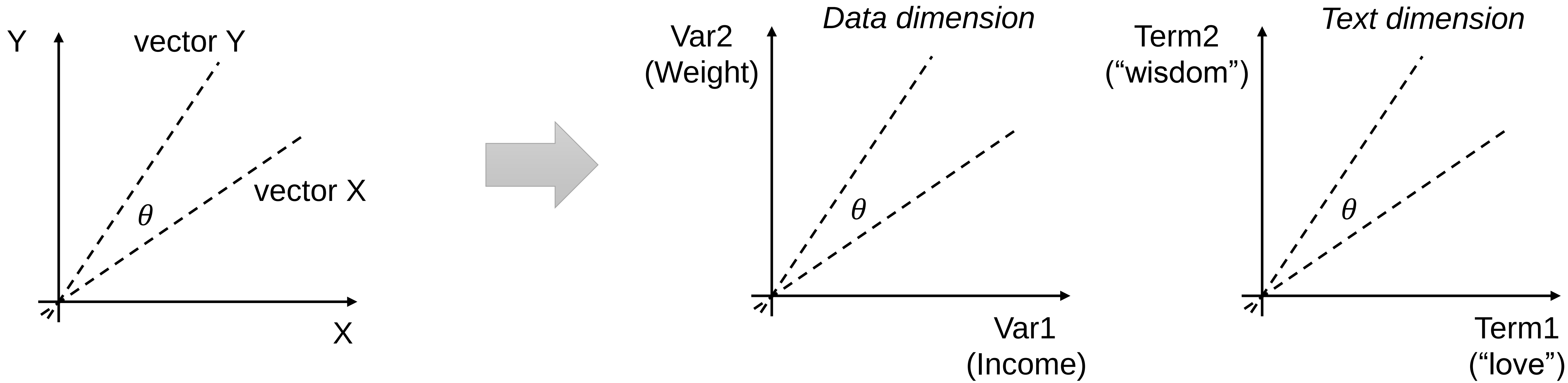
$$Text\ Euclidean\ Distance = \sqrt{(Term1_2 - Term1_1)^2 + (Term2_2 - Term2_1)^2}$$

- Limitation of Euclidean distance in text analysis
  - Low accuracy: Because of the sparsity of text data, the accuracy of Euclidean distance is low.
  - Biasedness: Euclidean distance is affected by the “size”
  - Ex) Doc1: “love” appeared 10 times / Doc2: “love” appeared 2 times

  → Doc1 is more related to “love”
  - Ex) Doc1: “love” appeared 10 times from1k words / Doc2: “love” appeared 2 times from 10 words

  → Doc2 is more related to “love”

- Cosine similarity
  - Instead, Cosine similarity if often used for measuring text similarity
  - Doc product & total frequency

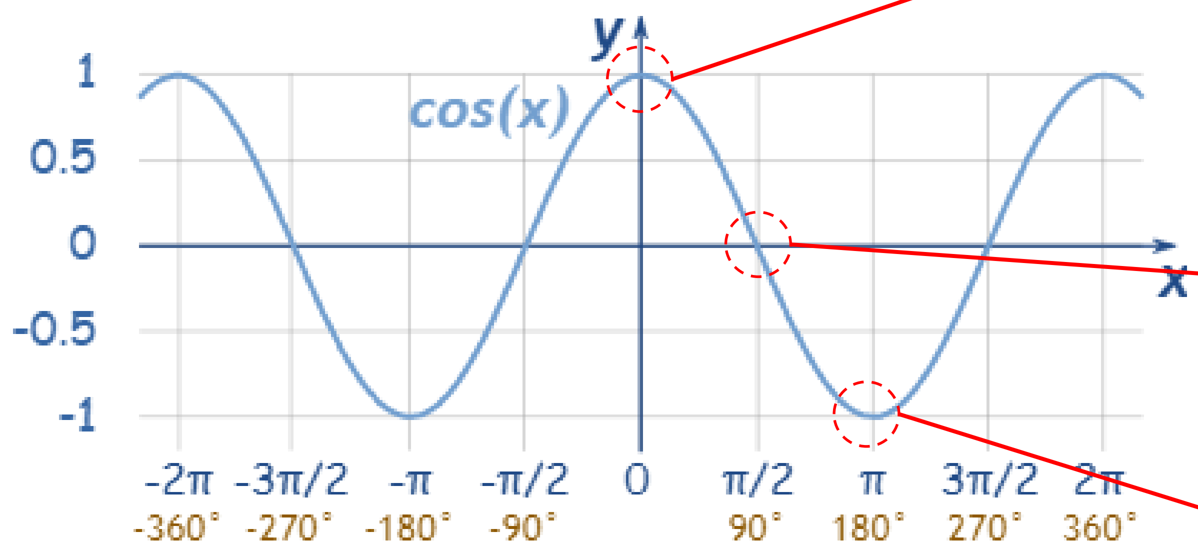


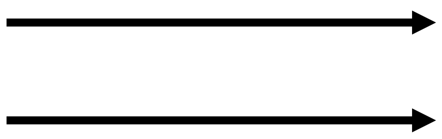
$$X \cdot Y = \|X\|\|Y\|cos\theta$$

$$cos\theta = \frac{X \cdot Y}{\|X\|\|Y\|} = \frac{\sum_{i=1}^{n} X_i Y_i}{\sqrt{\sum_{i=1}^{n} X_i^2}\sqrt{\sum_{i=1}^{n} Y_i^2}}$$

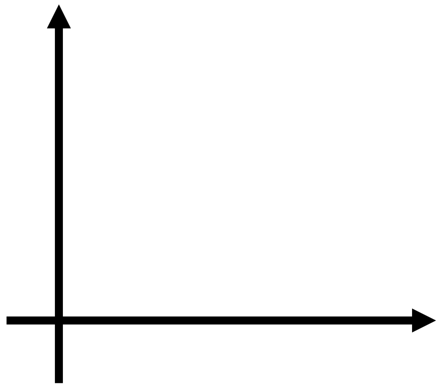
$$Text\ Similarity = \frac{Term1 \cdot Term2}{\|Term1\|\|Term2\|} = \frac{\sum_{i=1}^{n} Term1_i Term2_i}{\sqrt{\sum_{i=1}^{n} Term1_i^2}\sqrt{\sum_{i=1}^{n} Term2_i^2}}$$

- Cosine similarity
  - Range of cosine similarity: -1 ~ 1
  - Cosine similarity = 0 ~ Not similar at all
  - Cosine similarity = 1 ~ Identical

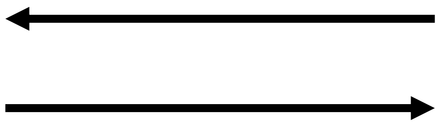




Cosine similarity = 1



Cosine similarity = 0



Cosine similarity = -1

https://www.mathsisfun.com/algebra/trig-sin-cos-tan-graphs.html

Text Similarity in R

- Import jfk_speech.docx

```
> library("officer")
> jfk.speech <-
+   read_docx("R file/R file_LEC08/jfk_speech_doc.docx")
> jfk.speech.sum <-
+   jfk.speech %>%
+   docx_summary %>%
+   rename(doc_id = doc_index) %>%
+   select(doc_id, text)
> jfk.speech.sum %>% head(1)
  doc_id
1      1
                                                                                                              text
1 President Pitzer, Mr. Vice President, Governor, Congressman Thomas, Senator Wiley, and Congressman Miller, M
r. Webb, Mr. Bell, scientists, distinguished guests, and ladies and gentlemen: I appreciate your president hav
ing made me an honorary visiting professor, and I will assure you that my first lecture will be very brief.
```

- Run simple text pre-processing
  - (Classic steps) remove punctuation, remove numbers, remove extra whitespaces, lemmatize, convert to lower case

```
> jfk.speech.corp <- jfk.speech.sum %>%
+   DataframeSource %>%
+   Corpus %>%
+   tm_map(removePunctuation) %>%
+   tm_map(removeNumbers) %>%
+   tm_map(stripWhitespace) %>%
+   tm_map(content_transformer(lemmatize_strings)) %>%
+   tm_map(content_transformer(tolower))
> jfk.speech.sum[2,]
  doc_id                                                                                         text
2      2 I am delighted to be here, and I'm particularly delighted to be here on this occasion.
> jfk.speech.corp[[2]]$content
[1] "i be delight to be here and im particularly delight to be here on this occasion"
```

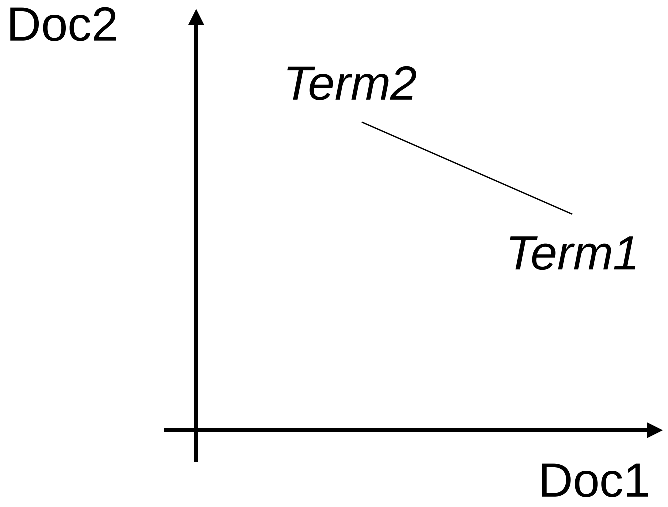
- Create DTM & TDM

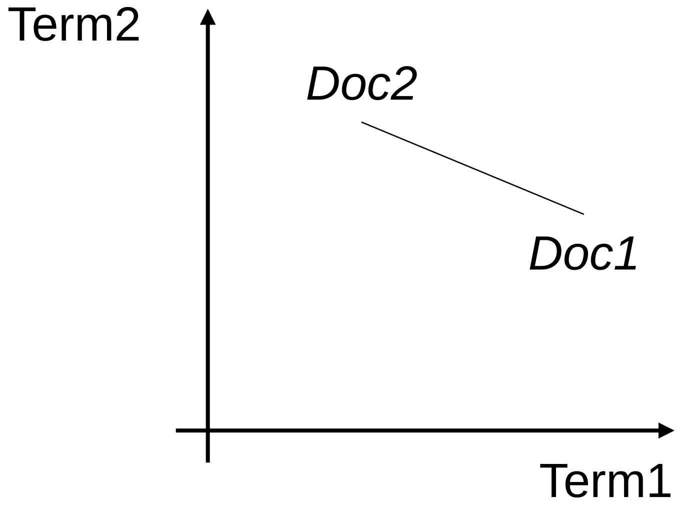
```
> jfk_speech.dtm <- jfk.speech.corp %>%
+   DocumentTermMatrix(control =
+                        list(wordLengths=c(1, Inf)))
> jfk_speech.dtm %>% inspect
<<DocumentTermMatrix (documents: 77, terms: 664)>>
Non-/sparse entries: 1682/49446
Sparsity           : 97%
Maximal term length: 14
Weighting          : term frequency (tf)
Sample             :
    Terms
Docs a and be in of that the this to we
  23 3   4  1  0  6    2   4    0  4  5
  25 0   3  0  3  1    0   2    2  5  0
  29 0   1  1  1  4    2   5    1  0  2
  3  6   4  0  6  4    0   0    0  0  3
  38 0   5  5  1  1    3   4    1  6  4
  42 3   1  2  1  2    0   4    1  2  1
  5  2   3  2  1  4    5   8    1  0  0
  59 1   5  0  3  1    0   3    3  5  0
  63 3   2  1  2  1    1   1    2  1  2
  64 4   5  6  1  7    2   8    3  4  2
```

```
> jfk_speech.tdm <- jfk.speech.corp %>%
+   TermDocumentMatrix(control =
+                        list(wordLengths=c(1, Inf)))
> jfk_speech.tdm %>% inspect
<<TermDocumentMatrix (terms: 664, documents: 77)>>
Non-/sparse entries: 1682/49446
Sparsity           : 97%
Maximal term length: 14
Weighting          : term frequency (tf)
Sample             :
      Docs
Terms  23 25 29 3 38 42 5 59 63 64
  a     3  0  0 6  0  3 2  1  3  4
  and   4  3  1 4  5  1 3  5  2  5
  be    1  0  1 0  5  2 2  0  1  6
  in    0  3  1 6  1  1 1  3  2  1
  of    6  1  4 4  1  2 4  1  1  7
  that  2  0  2 0  3  0 5  0  1  2
  the   4  2  5 0  4  4 8  3  1  8
  this  0  2  1 0  1  1 1  3  2  3
  to    4  5  0 0  6  2 0  5  1  4
  we    5  0  2 3  4  1 0  0  2  2
```

Text Similarity in R

- Measuring Euclidean distance between terms
  - proxy::dist(method = ‘Euclidean’)





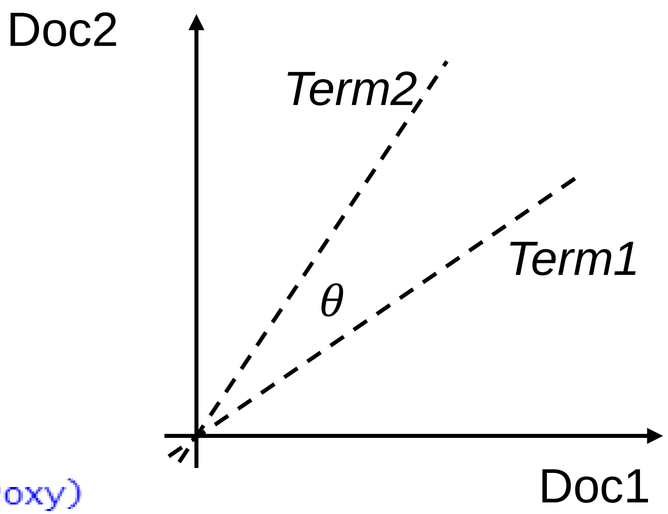

```
> term.euc <-
+   jfk_speech.tdm %>%
+   as.matrix %>%
+   proxy::dist(method = "euclidean") %>%
+   as.matrix
> term.euc[1:5,1:5]
                  a      and appreciate   assure       be
a           0.00000 16.27882   11.31371 11.31371 15.00000
and        16.27882  0.00000   17.91647 17.91647 14.62874
appreciate 11.31371 17.91647    0.00000  0.00000 14.59452
assure     11.31371 17.91647    0.00000  0.00000 14.59452
be         15.00000 14.62874   14.59452 14.59452  0.00000
```

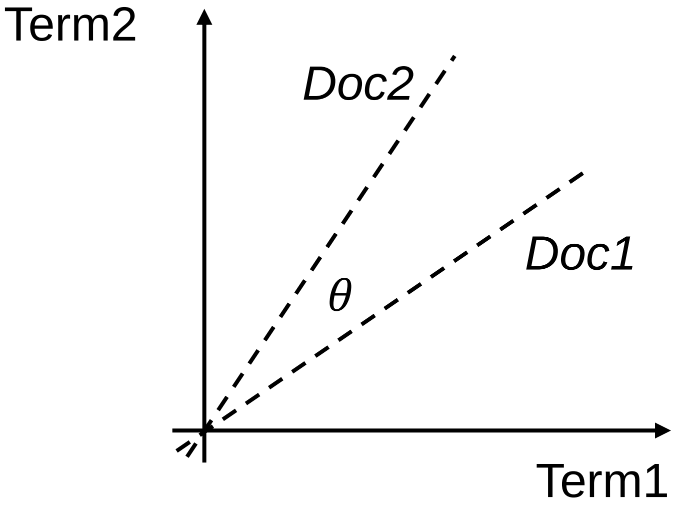
Euclidean distance of terms

```
> doc.euc <-
+   jfk_speech.dtm %>%
+   as.matrix %>%
+   proxy::dist(method = "euclidean") %>%
+   as.matrix
> doc.euc[1:5,1:5]
          1         2        3        4        5
1  0.000000  9.380832 14.10674  9.69536 14.52584
2  9.380832  0.000000 13.60147  6.63325 13.67479
3 14.106736 13.601471  0.00000 13.22876 15.81139
4  9.695360  6.633250 13.22876  0.00000 12.44990
5 14.525839 13.674794 15.81139 12.44990  0.00000
```

Euclidean distance of docs

- Measuring cosine similarity between terms
  - proxy::dist(method = ‘cosine’)






```
> library(proxy)
> term.cos <-
+   jfk_speech.tdm %>%
+   as.matrix %>%
+   proxy::dist(method = "cosine") %>%
+   as.matrix
> term.cos[1:5,1:5]
                   a       and appreciate    assure        be
a          0.0000000 0.5332982  0.9119549 0.9119549 0.6449003
and        0.5332982 0.0000000  0.7791369 0.7791369 0.3809866
appreciate 0.9119549 0.7791369  0.0000000 0.0000000 0.9316414
assure     0.9119549 0.7791369  0.0000000 0.0000000 0.9316414
be         0.6449003 0.3809866  0.9316414 0.9316414 0.0000000
```

Cosine similarity of terms

```
> doc.cos <-
+   jfk_speech.dtm %>%
+   as.matrix %>%
+   proxy::dist(method = "cosine") %>%
+   as.matrix
> doc.cos[1:5,1:5]
          1         2         3         4         5
1 0.0000000 0.8074178 0.8060752 1.0000000 0.8053502
2 0.8074178 0.0000000 0.9411510 1.0000000 0.8587480
3 0.8060752 0.9411510 0.0000000 0.9416126 0.7265176
4 1.0000000 1.0000000 0.9416126 0.0000000 0.6262825
5 0.8053502 0.8587480 0.7265176 0.6262825 0.0000000
```

Cosine similarity of docs

Text Similarity in R

- Measuring correlation between terms
  - cor.test()

```
> cor.test(as.vector(jfk_speech.dtm[,"space"]),
+          as.vector(jfk_speech.dtm[,"knowledge"]))

        Pearson's product-moment correlation

data:  as.vector(jfk_speech.dtm[, "space"]) and as.vector(jfk_speech.dtm[, "knowledge"])
t = 0.23214, df = 75, p-value = 0.8171
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.1983736  0.2492786
sample estimates:
       cor
0.02679581
```

- Finding similar terms
  - tm::findAssosc(*terms*, *corlimit*)

```
> jfk_speech.dtm %>%
+   findAssocs("space", 0.4)
$space
hostile  expect      or
   0.48    0.46    0.40
```

```
> jfk_speech.dtm %>%
+   findAssocs("knowledge", 0.6)
$knowledge
  change     fear     meet     note strength
     0.6      0.6      0.6      0.6      0.6
```

```
> jfk_speech.tdm %>%
+   findAssocs("space", 0.4)
$space
hostile  expect      or
   0.48    0.46    0.40
```

```
> jfk_speech.tdm %>%
+   findAssocs("knowledge", 0.6)
$knowledge
  change     fear     meet     note strength
     0.6      0.6      0.6      0.6      0.6
```

- Measuring correlation between documents
  - In our example, we measure the similarity between sentences…

(Docs23)
We mean to be a part of it--we mean to lead it. For the eyes of the world now look into space, to the moon and to the planets beyond, and we have vowed that we shall not see it governed by a hostile flag of conquest, but by a banner of freedom and peace. We have vowed that we shall not see space filled with weapons of mass destruction, but with instruments of knowledge and understanding.

(Docs25)
In short, our leadership in science and in industry, our hopes for peace and security, our obligations to ourselves as well as others, all require us to make this effort, to solve these mysteries, to solve them for the good of all men, and to become the world's leading space-faring nation.

```
> jfk_speech.tdm %>% inspect
       Docs
Terms   23 25 29 3 38 42 5 59 63 64
  a      3  0  0 6  0  3 2  1  3  4
  and    4  3  1 4  5  1 3  5  2  5
  be     1  0  1 0  5  2 2  0  1  6
  in     0  3  1 6  1  1 1  3  2  1
  of     6  1  4 4  1  2 4  1  1  7
  that   2  0  2 0  3  0 5  0  1  2
  the    4  2  5 0  4  4 8  3  1  8
  this   0  2  1 0  1  1 1  3  2  3
  to     4  5  0 0  6  2 0  5  1  4
  we     5  0  2 3  4  1 0  0  2  2
> cor.test(as.vector(jfk_speech.tdm[,"23"]),
+          as.vector(jfk_speech.tdm[,"25"]))

	Pearson's product-moment correlation

data:  as.vector(jfk_speech.tdm[, "23"]) and as.vector(jfk_speech.tdm[, "25"])
t = 9.1125, df = 662, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.2644795 0.3997794
sample estimates:
      cor
0.3338478
```

Text Similarity in R

- Similarity Correlation Matrix

```
doc.cor <-
  matrix(NA, nrow = length(colnames(jfk_speech.tdm)),
         ncol = length(colnames(jfk_speech.tdm)))
for(i in 1:length(colnames(jfk_speech.tdm))){
  for(j in 1:length(colnames(jfk_speech.tdm))){
    doc.cor[i,j] <-
      cor.test(as.vector(jfk_speech.tdm[,i]),
               as.vector(jfk_speech.tdm[,j]))$est
  }
}
> head(doc.cor)
            [,1]         [,2]       [,3]        [,4]      [,5]        [,6]       [,7]        [,8]
[1,]  1.00000000  0.172889542 0.16597986 -0.02103298 0.1591651 0.143576803 0.01363945 -0.03601731
[2,]  0.17288954  1.000000000 0.04086581 -0.01151801 0.1210349 0.007024491 0.12555346  0.11669837
[3,]  0.16597986  0.040865814 1.00000000  0.04361853 0.2496422 0.387197744 0.20218814  0.04830822
[4,] -0.02103298 -0.011518005 0.04361853  1.00000000 0.3633306 0.061488469 0.21560807 -0.01627157
[5,]  0.15916513  0.121034861 0.24964225  0.36333062 1.0000000 0.303535994 0.39975406  0.07528035
[6,]  0.14357680  0.007024491 0.38719774  0.06148847 0.3035360 1.000000000 0.26281268  0.12070754
          [,14]      [,15]       [,16]      [,17]       [,18]     [,19]     [,20]     [,21]
[1,] 0.09503523 0.07436693  0.08175807 0.13576411  0.10570584 0.2091305 0.1942505 0.1089334 0.113
[2,] 0.13222023 0.02640193  0.36623232 0.26421655  0.20571880 0.2240960 0.3604496 0.2260825 0.086
[3,] 0.18696982 0.11971981  0.11708764 0.24321772 -0.02200679 0.2870723 0.3138580 0.3470200 0.229
[4,] 0.02474442 0.09976546 -0.01610355 0.05820152 -0.01322441 0.2707493 0.1087173 0.2551629 0.264
[5,] 0.10127224 0.26418788  0.15182400 0.37543227  0.15007829 0.4334052 0.2825737 0.4679980 0.584
[6,] 0.14764746 0.11756449  0.09766928 0.23057825  0.01980921 0.1940884 0.1802787 0.3302682 0.281
```

Association Analytics

- Association analytics
  - Finding associations and relations among variables
  - Simply speaking, finding (frequently occurring) patterns using **association rules**

- Association Rules
  - Must be apparent, useful, and applicable
  - Trivial rules (well known, obvious or not explainable) should be avoided.

$$\{Condition\} \rightarrow \{Result\}$$

Under what condition, would we get the result?

- Market based analysis
  - One of the representative approach based on association rule analysis
  - Used for recommendation services or product display

단순 4개의 T를 가지고 가정을(패턴) 연관성을 발견하게됨
그 연관성을 가지고 마켓에서 묶음으로 판매

위스키 사는사람이 담배를 많이 산다 가정1)

<Market based analysis>

Transaction	Items
t_1	Whisky, Cigarette
t_2	Milk, Eggs, Candy
t_3	Milk, Eggs, Kimchi
t_4	Soju, Whisky, Cigarette

Item: $i_1, i_2, i_3, \ldots$

Item set: $I = \{i_1, i_2, i_3, \ldots, i_j\}$

Transaction: $T = \{t_1, t_2, t_3, \ldots, t_k\}$

"Person who buys Whisky is likely to purchase Cigarette."
{Whisky} → {Cigarette}

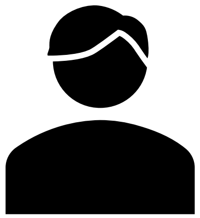
"Person who buys Milk is likely to purchase Eggs."
{Milk} → {Eggs}

$X \Rightarrow Y \text{ where } X \subset I, Y \subset I \text{ and } X \cap Y = 0$

- *antecedent: X (LHS)*
- *consequent: Y (RHS)*

조건 결과
이것을 보여주는것 연관분석
이것으 랑ㄹ고리즘적으로 볼수
있다.

"I want to buy Milk, Orange, Kimchi."



Recommend "Egg"

Apriori Algorithm

- Apriori algorithm
  - An algorithm for *frequent item set mining* and *association rule* learning over relational databases [Wikipedia]

- Text association analytics
  - Finding associations and relations among texts

Apriori Algorithm

DocID	Text
Doc1	love, passion
Doc2	love, passion, sweet
Doc3	hungry

지금은 전처리가 끝이난 아이템(단어) 들
어떤 단어가 나올때 어떤 단어가 나온다 어떤 단어가 더 연관이있다 확인

{love} → {passion}

- Support 지지도 어떤것들이 출현하는 정도
  - Probability of occurrences
  - [0, 1]
  - $Support(X,Y) = Support(X \rightarrow Y) = P(X,Y) = \frac{Frequency(X,Y)}{Total\ \#\ of\ Document}$

DocID	Text
Doc1	love, passion, sweet
Doc2	love, passion, hungry
Doc3	love, anger, sweet
Doc4	anger, disgrace, passion

Itemset (Doc1)	Support
{love}	3 / 4 = 0.75
{passion}	3 / 4 = 0.75
{sweet}	2 / 4 = 0.5
{love, passion}	2 / 4 = 0.5
{passion, sweet}	1 / 4 = 0.25
{love, sweet}	2 / 4 = 0.5
{love, passion, sweet}	1 / 4 = 0.25

Love and passion appear in 50% of the whole document

love와passion이 둘다 나오는것

- Support 지지도
  - Low Support value of *termX* indicates that the *termX* itself does not occur frequently → *termX* is unlikely to have high association with other terms
  - High Support value of *termX* indicates that the *termX* itself does occur frequently.

  - Instead of considering all terms regardless of the support value, the apriori algorithm suggests considering only cases that are likely to occur
  - Use threshold to filter cases with higher support

Itemset (Doc1)	Support
{love, passion}	2 / 4 = 0.5
~~{passion, sweet}~~	~~1 / 4 = 0.25~~
{love, sweet}	2 / 4 = 0.5
~~{love, passion, sweet}~~	~~1 / 4 = 0.25~~

10번중에 4번도 안나오는애들은 안볼꺼야 하고 거는것

Setting threshold = 0.4 will keep {love, passion} and {love, sweet} → *This means that we will not consider any cases including {love, passion}, {love, passion, sweet}*

0.4미만은 다 제거함

- Confidence 신뢰도 쉽게 말해서 조건부 확률 A가 나타났을때 B가 나타날 확률
  - Conditional probability of B occurring when A is occurred.
  - [0, 1]
  - $Confidence(X \rightarrow Y) = \frac{Support(X,Y)}{Support(X)} = P(Y|X) = \frac{P(X,Y)}{P(X)}$

X가 나타났을때Y가 나타날 확률

Independent condition

$$\boxed{P(Y)} = \boxed{P(Y|X)} = \frac{P(Y \cap X)}{P(X)}$$

$$P(Y)P(X) = P(Y \cap X)$$

Itemset (Doc1)	Support
{love}	3 / 4 = 0.75
{passion}	3 / 4 = 0.75
{sweet}	2 / 4 = 0.5
{love, passion}	2 / 4 = 0.5
~~{passion, sweet}~~	~~1 / 4 = 0.25~~
{love, sweet}	2 / 4 = 0.5
~~{love, passion, sweet}~~	~~1 / 4 = 0.25~~

Itemset (Doc1)	Confidence
{love, passion}	$\frac{support(love, passion)}{support(love)} = \frac{0.5}{0.75} = 0.66$
{love, passion}	$\frac{support(love, passion)}{support(passion)} = \frac{0.5}{0.75} = 0.66$
{love, sweet}	$\frac{support(love, sweet)}{support(love)} = \frac{0.5}{0.75} = 0.33$
{love, sweet}	$\frac{support(love, sweet)}{support(sweet)} = \frac{0.5}{0.5} = 1$
~~{love, sweet}~~	
~~{love, passion, sweet}~~	

When love occurred, passion occurred in 70% of the total cases

- Confidence 신뢰도
  - Use threshold to filter cases with higher confidence

연관분석 핵심은 신뢰도이다
A와B가 얼마나 연관성이 있지?
threshold는 몇으로 해야할까?

Itemset (Doc1)	Confidence
{love, passion}	$\frac{support(love, passion)}{support(love)} = \frac{0.5}{0.75} = 0.66$
{passion, love}	$\frac{support(love, passion)}{support(passion)} = \frac{0.5}{0.75} = 0.66$
~~{love, sweet}~~	$\cancel{\frac{support(love, sweet)}{support(love)} = \frac{0.5}{0.75} = 0.33}$
{love, sweet}	$\frac{support(love, sweet)}{support(sweet)} = \frac{0.5}{0.5} = 1$

Setting threshold = 0.5 will keep {love, passion}, {passion, love}, {love, sweet}

Confidence = 1 means that they occur together in all cases. This may be the probability of co-incidence

- Coverage
  - Also known as cover or LHS-support
  - Support of the left-hand-side of the rule X => Y, support(X)
  - It represents a measure of how often the rule can be applied.
  - [0,1]
  - $Coverage(X \to Y) = \frac{Support(X,Y)}{Confidence(X,Y)}$

- “How much does the occurrence of X contribute to the occurrence of Y?”

Lift와 레버리지가 전달하고자 하는것은
X가 (출현)이 Y의 출현에 얼마나 많이 기여하는가
이것을 리프트와 레버리지 두 지표를 쓰는데 리프트는 비율로써 비교한다 ratio로...
support (y)분의 cofidence

- Lift 향상도
  - Compare the two cases of obtaining Y with and without restriction with *the form of proportion (ratio)*
  - Confidence / Support

조건이 되는것

Case with condition (Confidence)

조건이 없을때마다 조건이
있을때가 더 많이 나타난다

  - $Lift(X \rightarrow Y) = \frac{Confidence(X \rightarrow Y)}{Support(Y)} = \frac{Suport(X,Y)}{Support(X)} \frac{1}{Support(Y)} = \frac{P(Y|X)}{P(Y)} = \frac{P(X,Y)}{P(X)P(Y)}$
  - [0, ∞]

Case without condition (Support)

x랑 Y가 독립이라는 말이다.

[확인]

  - Lift = 1 → Independent, P(X,Y) = P(X)P(Y)
  - Lift > 1 → Probability that **Y occurred when X occurred** is higher than when **Y occurred without restriction**.

1보다 크다는것 분자가 크다
1도다 작다 분자가 작다
리프트가 1보다 커? 조건이 있을때 얘가더 많이 나타나는구나
love라는 단어가 나올때 jesus가 혼자 나올때보다 더 많이 나온다.

Apriori Algorithm

- "How much does the occurrence of X contribute to the occurrence of Y?"

- Leverage 레버리지

빼기를 한것이다.

  - Compare the two cases of obtaining Y with and without restriction using *the form of subtraction*
  - $Leverage(X \rightarrow Y) = Support(X,Y) - \left(Support(X)Support(Y)\right)$
    $= P(X,Y) - \left(P(X)P(Y)\right)$

조건이 있는것 조건이 없는것

  - [-1, 1]
  - Leverage = 0 → Independent, P(X,Y) = P(X)P(Y)
  - Leverage > 0 → Case with restriction is higher than the case without restriction
  - Leverage < 0 → Case without restriction is higher than the case with restriction

0보다 작은것은 혼자 단일로 나오는것이 더 잘 나온다

- Advantages
  - Simple and easy to apply

연관분석 장점은 단순하고 쉽게 적용가능
굉장히 빠르게 아이디어를 얻을수있다.

- Limitations
  - Computation load → necessity of threshold

굉장히 단순 확률계산같지만 텍스트 특성상 굉장히 많은 확률 계산해야하는 이슈를 threshold를 통해 해결해야하는데 threshold넣는 이유가 두가지
1. computaiton 부담을 덜기위해 2. 어차피 확률 떨어지는(잘안나오는) 것들은 내가 중요하지 않다고 생각해서.

Apriori in R

Apriori in R

- arules::apriori()

Create a list with word vector

```
> library('arules')
> mydoc <- list(
+   c("love","passion","sweet"),
+   c("love","passion","hungry"),
+   c("love","anger","sweet"),
+   c("anger","disgrace","passion")
+ )
> mydoc
[[1]]
[1] "love"    "passion" "sweet"

[[2]]
[1] "love"    "passion" "hungry"

[[3]]
[1] "love"  "anger" "sweet"

[[4]]
[1] "anger"    "disgrace" "passion"

> mydoc %>%
+   as('transactions') %>% inspect
    items
[1] {love, passion, sweet}
[2] {hungry, love, passion}
[3] {anger, love, sweet}
[4] {anger, disgrace, passion}
```

```
> mydoc %>%
+   as('transactions') %>% inspect
Error in UseMethod("inspect", x) :
  no applicable method for 'inspect' applied to an object of class "c('transactions', 'itemMatrix')"
```

If you see this error, type ?inspect to check in which package inspect() is used… If it's tm's inspect, do as follows…

>>> detach("package:tm")

- arules::apriori()

Use parameter to set the "threshold"

```
> mydoc.ap <-
+   mydoc %>%
+   apriori(parameter=list(supp=0, conf=0))
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen maxlen target  ext
          0    0.1    1 none FALSE            TRUE       5       0      1     10  rules TRUE

Algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE

Absolute minimum support count: 0

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[6 item(s), 4 transaction(s)] done [0.00s].
sorting and recoding items ... [6 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 done [0.00s].
writing ... [96 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
> mydoc.ap %>%
+   inspect
     lhs            rhs          support confidence coverage lift      count
[1]  {}          => {hungry}     0.25    0.2500000  1.00     1.0000000 1
[2]  {}          => {disgrace}   0.25    0.2500000  1.00     1.0000000 1
[3]  {}          => {sweet}      0.50    0.5000000  1.00     1.0000000 2
[4]  {}          => {anger}      0.50    0.5000000  1.00     1.0000000 2
[5]  {}          => {passion}    0.75    0.7500000  1.00     1.0000000 3
[6]  {}          => {love}       0.75    0.7500000  1.00     1.0000000 3
```

```
> mydoc
[[1]]
[1] "love"    "passion" "sweet"

[[2]]
[1] "love"    "passion" "hungry"

[[3]]
[1] "love"  "anger" "sweet"

[[4]]
[1] "anger"    "disgrace" "passion"
```

- arules::apriori()
  - Set threshold with the parameter option
  - supp ~ Support
  - conf ~ Confidence

Support = 0.4, Confidence = 0.7

```
> mydoc.ap.1 <-
+   mydoc %>%
+   apriori(parameter=list(supp=0.4, conf=0.7))
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support mi
        0.7    0.1    1 none FALSE            TRUE       5     0.4

Algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE

Absolute minimum support count: 1

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[6 item(s), 4 transaction(s)] done [0.00s].
sorting and recoding items ... [4 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 done [0.00s].
writing ... [3 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
> mydoc.ap.1 %>%
+   inspect
    lhs        rhs       support confidence coverage lift     count
[1] {}      => {passion} 0.75    0.75       1.0      1.000000 3
[2] {}      => {love}    0.75    0.75       1.0      1.000000 3
[3] {sweet} => {love}    0.50    1.00       0.5      1.333333 2
```

When the word sweet occurs, love is likely to occur

- arules::apriori()
  - Use the appearance option to find associated terms and their results

```
> mydoc.ap.2 <-
+   mydoc %>%
+   apriori(parameter=list(supp=0.1, conf=0.1),
+           appearance=list(rhs="love"))

> mydoc.ap.2 %>%
+   inspect
    lhs                  rhs    support confidence coverage lift      count
[1] {}                => {love} 0.75    0.7500000  1.00     1.0000000 3
[2] {hungry}          => {love} 0.25    1.0000000  0.25     1.3333333 1
[3] {sweet}           => {love} 0.50    1.0000000  0.50     1.3333333 2
[4] {anger}           => {love} 0.25    0.5000000  0.50     0.6666667 1
[5] {passion}         => {love} 0.50    0.6666667  0.75     0.8888889 2
[6] {hungry, passion} => {love} 0.25    1.0000000  0.25     1.3333333 1
[7] {anger, sweet}    => {love} 0.25    1.0000000  0.25     1.3333333 1
[8] {passion, sweet}  => {love} 0.25    1.0000000  0.25     1.3333333 1
```

Collect the result that "love" occurs as consequent

Terms that are likely to occur as a condition of "love"

- arules::apriori()
  - Use the appearance option to find associated terms and their results

장바구니 핵심은 연관성을 찾는것이 핵심이다.

```
> mydoc.ap.3 <-
+   mydoc %>%
+   apriori(parameter=list(supp=0.1, conf=0.1),
+           appearance=list(lhs="love"))
```

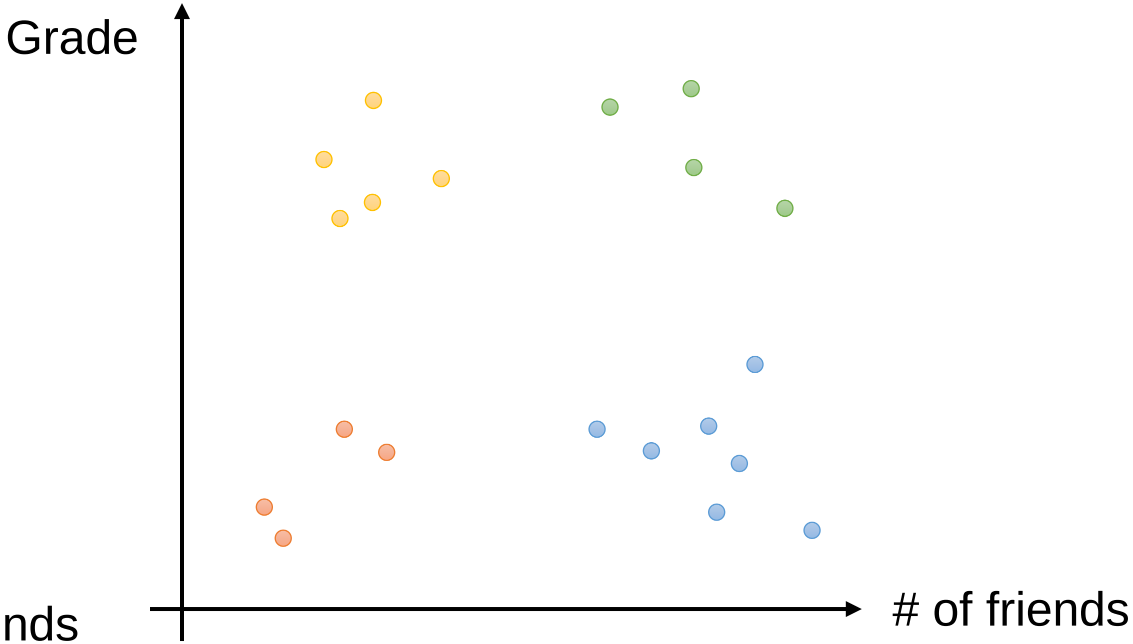
Collect the result that "love" occurs as an antecedent

```
> mydoc.ap.3 %>%
+   inspect
    lhs       rhs         support confidence coverage lift      count
[1] {}     => {hungry}    0.25    0.2500000  1.00     1.0000000 1
[2] {}     => {disgrace}  0.25    0.2500000  1.00     1.0000000 1
[3] {}     => {sweet}     0.50    0.5000000  1.00     1.0000000 2
[4] {}     => {anger}     0.50    0.5000000  1.00     1.0000000 2
[5] {}     => {passion}   0.75    0.7500000  1.00     1.0000000 3
[6] {love} => {hungry}    0.25    0.3333333  0.75     1.3333333 1
[7] {love} => {sweet}     0.50    0.6666667  0.75     1.3333333 2
[8] {love} => {anger}     0.25    0.3333333  0.75     0.6666667 1
[9] {love} => {passion}   0.50    0.6666667  0.75     0.8888889 2
```

Terms that are likely to occur as the result of "love"

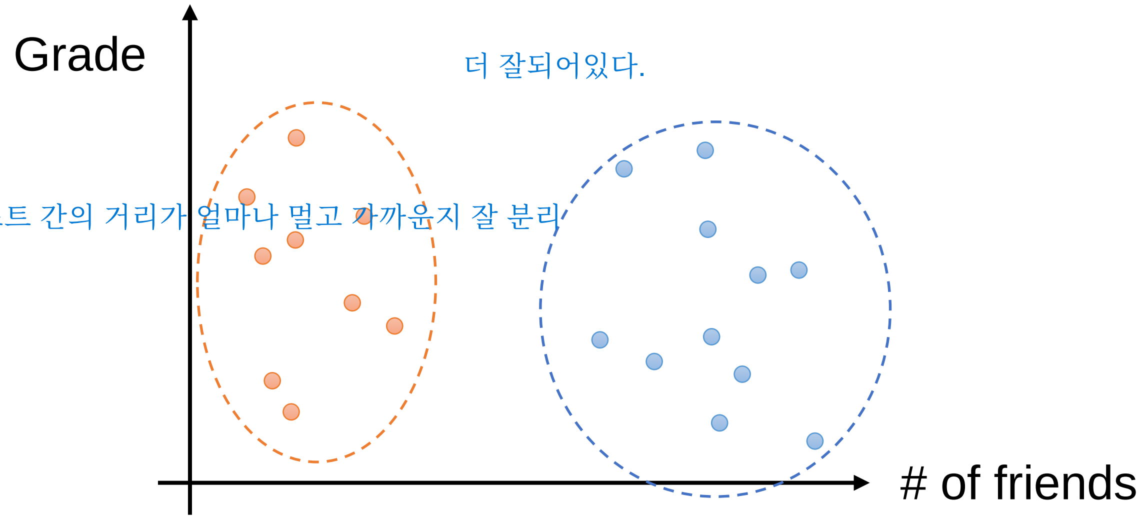
Clustering Principle

- Clustering or clustering analysis
  - Task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense) to each other than to those in other groups (clusters)
  - When the label of the data is not provided, clustering analysis can be used to add labels or identify the grouping pattern of the variable
  - *We want to find a group or cluster*



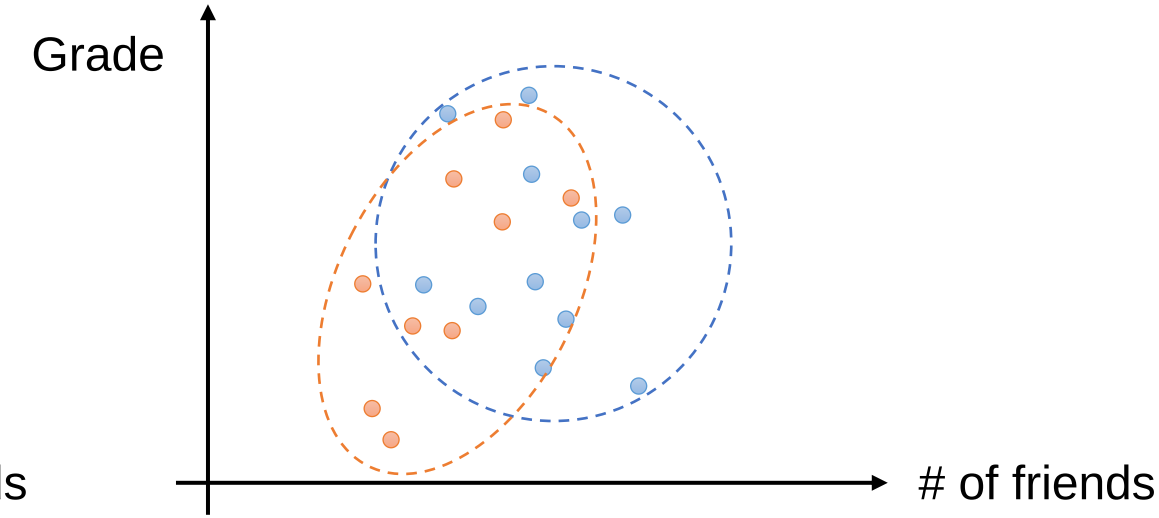
Clustering

- Which is a better cluster?



더 잘되어있다.

클러스트 간의 거리가 얼마나 멀고 가까운지 잘 분리



- Condition for a good cluster (or group)?
  - Maximizes the inter-cluster distance (or variance)
  - Minimizes the intra-cluster distance (or variance)

  - Inter-cluster relationship: One between objects inside the cluster
  - Intra-cluster relationship: One between objects outside the cluster

Hierarchical Clustering

Hierarchical Clustering

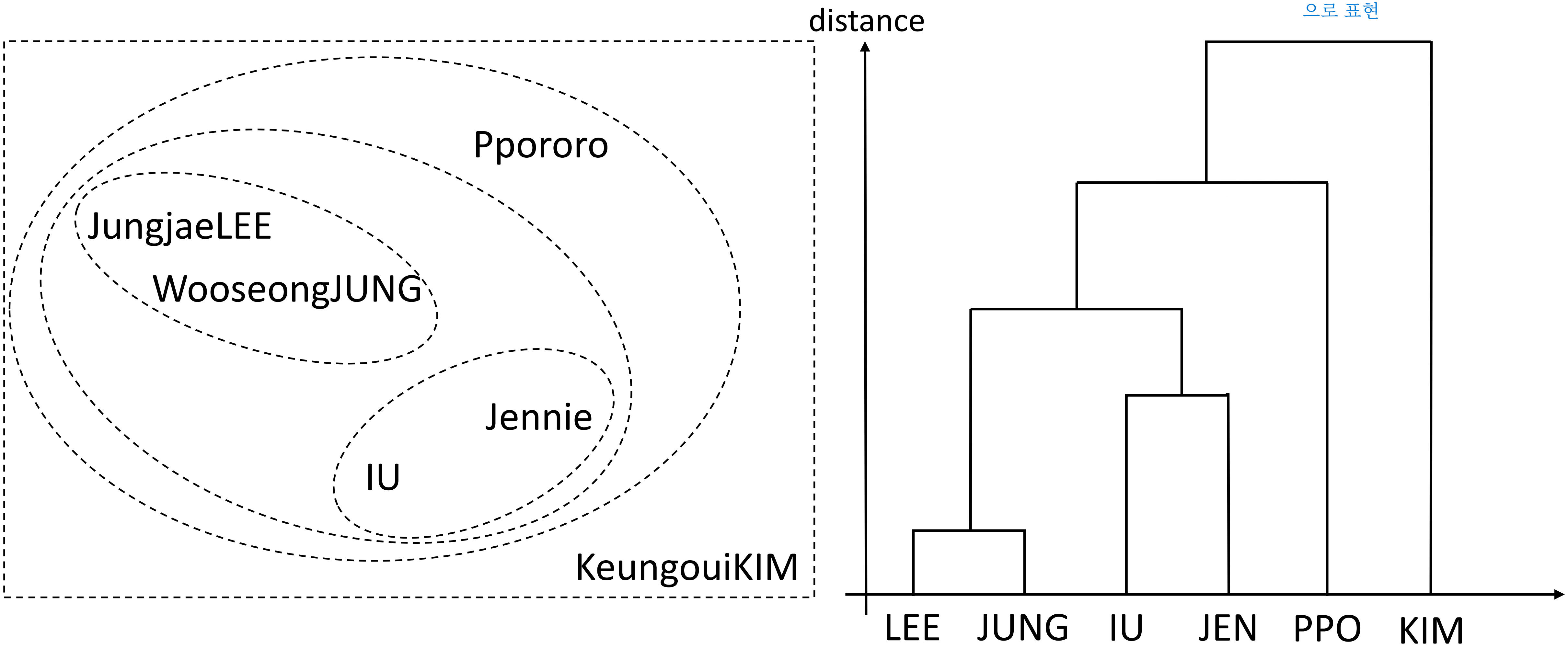
- Hierarchical clustering
  - No need to assume the number of clusters. But we still can increase the number of clusters

- Agglomerative clustering
  - points (individual cluster) → cluster
  - Starting from a single point, clustering algorithm merges and creates a cluster until it reaches the one final cluster

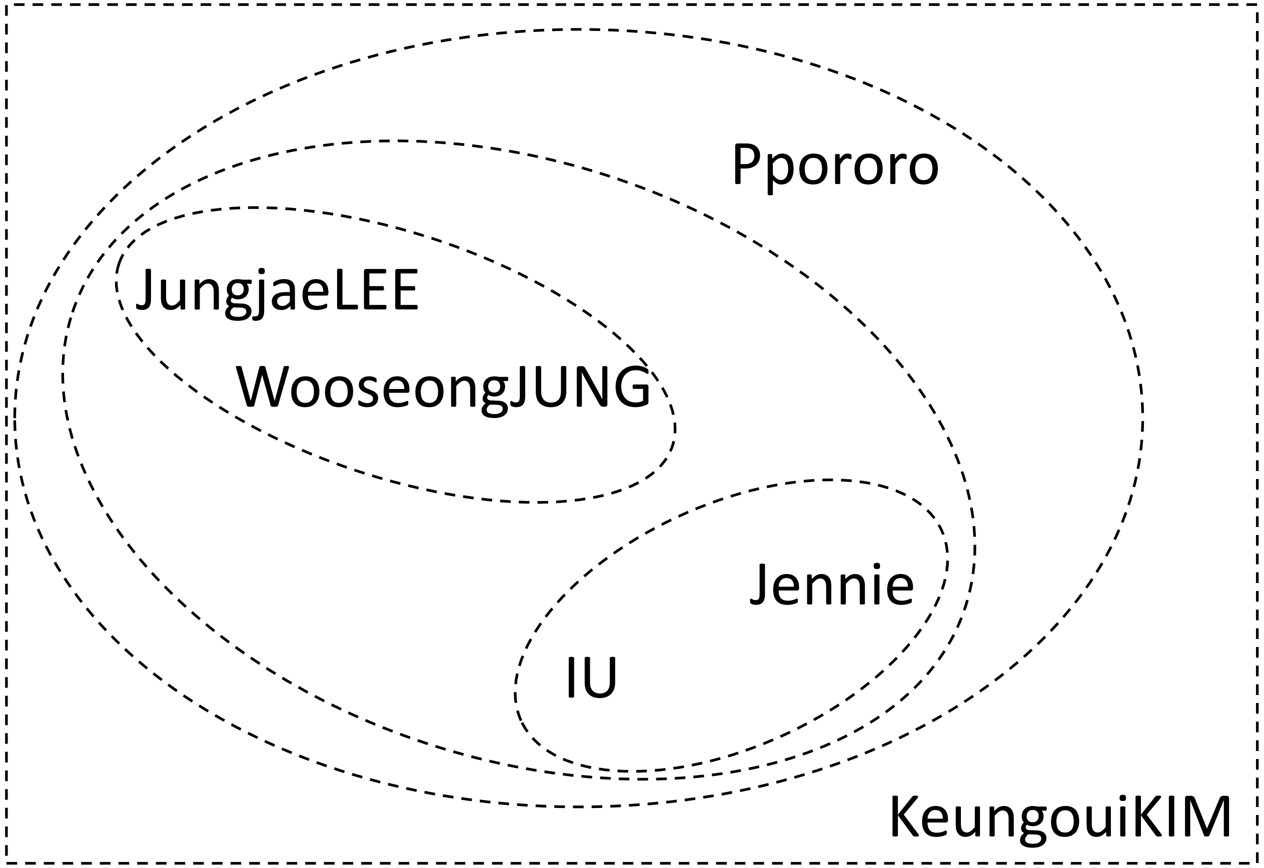
- Divisive clustering
  - cluster → points (individual cluster)
  - Starting from the whole, it splits a cluster until each cluster contains a point

- Hierarchical clustering 계층적 분석
  - A set of nested clusters organized as a <u>hierarchical tree</u>

높이에 따라 서로가 얼마나 밀집되어있는지를 보고 어떻게 묶이는지를 계층적으로 표현



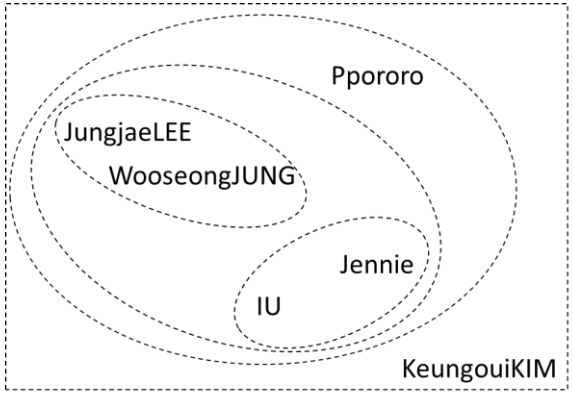
- Hierarchical clustering procedures
  - Step 1. Create a distance matrix



먼저 distance matrix을 만들어 놓고

	LEE	JUNG	IU	JEN	PPO	KIM
LEE		4	18	20	31	81
JUNG			19	21	32	82
IU				8	33	92
JEN					34	93
PPO						79
KIM						

Hierarchical Clustering



- Hierarchical clustering
  - Step 2. starting from the minimum distance, merge the pairs
  - Example) Single link

	LEE	JUNG	IU	JEN	PPO	KIM
LEE		4	18	20	31	81
JUNG			19	21	32	82
IU				8	33	92
JEN					34	93
PPO						79

  - Step 3. Update the cluster distance matrix. Repeat step 2 & 3, until it reaches the one big cluster

	LJ	IU	JEN	PPO	KIM
LJ		18	20	31	81
IU			8	33	92
JEN				34	93
PPO					79

Clustering Method

- Clustering Methods: Defining distance between clusters

- Single link (single-nearest distance)
  - Distance between the closest element of the two clusters
  - Clustering focused on the closest distance.

$$D(c_1, c_2) = \min_{x_1 \in c_1, x_2 \in c_2} D(x_1, x_2)$$

- Complete link (complete farthest distance)
  - Distance between the farthest elements of the two clusters
  - Clustering focused on the farthest distance.

$$D(c_1, c_2) = \max_{x_1 \in c_1, x_2 \in c_2} D(x_1, x_2)$$

- Clustering Methods: Defining distance between clusters

- Average link
  - Clustering focused on the average distance.

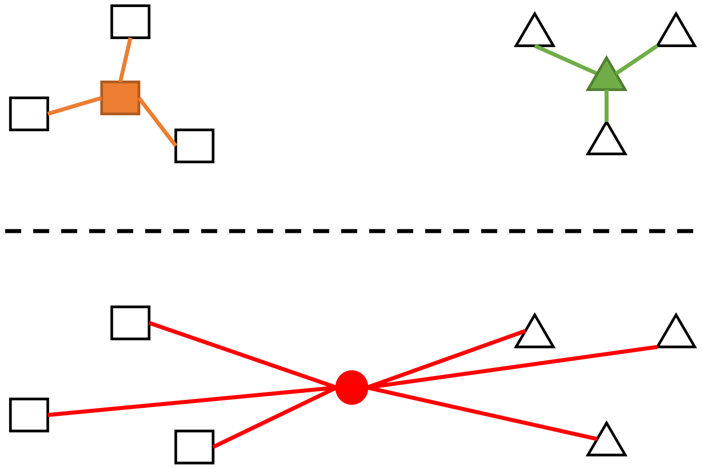
$$D(c_1, c_2) = \frac{1}{|c_1|}\frac{1}{|c_2|}\sum_{x_1 \in c_1}\sum_{x_2 \in c_2} D(x_1, x_2)$$

- Median link
  - Clustering focused on the median distance.
- Centroids link
  - Clustering focused on the distance between the centroids of two clusters

- Ward method (minimum variance method)
  - Popular in linguistics
  - Creates compact, even-sized clusters (Szmrecsanyi, 2012)
  - Measures the similarity of two clusters based on the increase in squared error when two clusters are merged
  - Simply speaking, it compares the SSE (sum of square error) before and after clustering, and selects the one where the SSE increases less.

Before clustering

After clustering



SSE – (SSE + SSE)

Text Clustering

Clustering

클러스터링 하는게 중심 졸업전에 명확하게 이해하자 그러나 강하게는 이야기X robust 한 과정이 필요하기때문

- Clustering applications
  - Find the pattern in the data
  - Understand the data
  - Summarize the data

- Clustering? Classification?
  - Classification → supervised learning
  - Clustering → unsupervised learning

- Thinking of the complexity of knowledge, clustering helps us to gain a better understanding of the data or sometimes get a new insight

Clustering

- Three key factors needed to be identified
  - Clustering target (ex. Classmates)
  - Clustering relation (ex. SNS)
  - Clustering method (partitional clustering, hierarchical clustering)

- Clustering results are just “clustering results”
  - In other words, the clustering algorithm separates the objects into groups, without knowing any further information beyond the clustering target, relation, and method
  - The important part is the “interpretation” of the result
  - There is no optimal solution for clustering, which requires an understandable explanation and description.

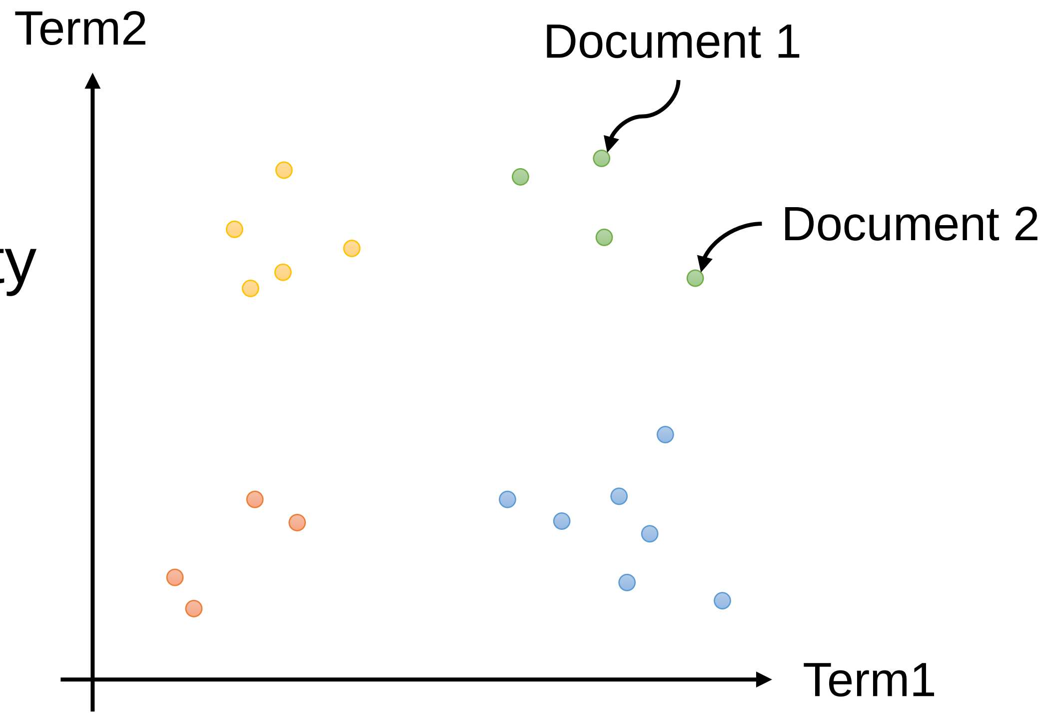
Clustering

- Clustering types
  - Hard clustering: No overlapping clusters
  - Soft clustering: Overlapping clusters

- Clustering process
  - Feature selection
  - Distance measure
  - Clustering algorithm
  - Interpretation

- Text clustering
  - Exploring the clustering pattern between documents (or terms)
  - Based on the text data, it allows us to classify the groups among text

- Text clustering process
  - Feature selection → Text Pre-processing
  - Distance measure → Measuring text similarity
  - Clustering algorithm → Clustering
  - Interpretation



Text Clustering in R

Distance Matrix

- Create a distance matrix

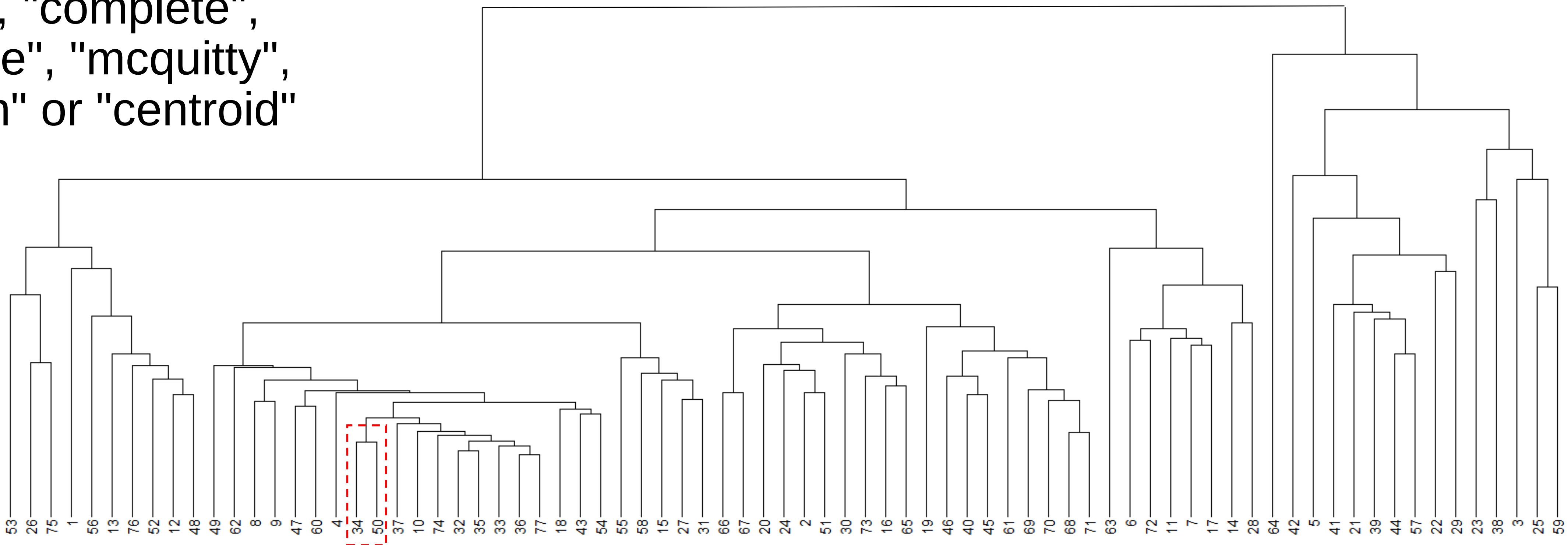
기말에 무엇을 물어보느냐에따라 위에꺼가 될수있고 아래될수있다.
위 는 문서와 문서
아래는 단어와 단어

```
> jfk_speech.dtm %>%
+   stats::dist(method="euclidean") %>%
+   as.matrix %>%
+   .[1:10,1:10]
           1         2        3         4        5         6         7         8         9        10
1   0.000000  9.380832 14.10674  9.695360 14.52584  9.899495 10.723805  9.797959  9.055385  9.219544
2   9.380832  0.000000 13.60147  6.633250 13.67479  8.124038  7.810250  6.324555  6.000000  5.916080
3  14.106736 13.601471  0.00000 13.228757 15.81139 11.789826 13.038405 13.228757 12.609520 13.114877
4   9.695360  6.633250 13.22876  0.000000 12.44990  7.211103  6.855655  5.830952  5.477226  4.795832
5  14.525839 13.674794 15.81139 12.449900  0.00000 12.767145 12.165525 13.601471 13.453624 13.490738
6   9.899495  8.124038 11.78983  7.211103 12.76715  0.000000  7.681146  7.071068  6.633250  6.708204
7  10.723805  7.810250 13.03840  6.855655 12.16553  7.681146  0.000000  6.708204  6.708204  6.782330
8   9.797959  6.324555 13.22876  5.830952 13.60147  7.071068  6.708204  0.000000  4.898979  4.582576
9   9.055385  6.000000 12.60952  5.477226 13.45362  6.633250  6.708204  4.898979  0.000000  4.123106
10  9.219544  5.916080 13.11488  4.795832 13.49074  6.708204  6.782330  4.582576  4.123106  0.000000
> jfk_speech.tdm %>%
+   stats::dist(method="euclidean") %>%
+   as.matrix %>%
+   .[1:10,1:10]
                   a      and appreciate    assure       be      bell     brief congressman distinguish     first
a            0.00000 16.27882  11.313708 11.313708 15.00000 11.313708 11.313708    11.35782   11.313708 11.618950
and         16.27882  0.00000  17.916473 17.916473 14.62874 17.916473 17.916473    17.77639   17.916473 17.606817
appreciate  11.31371 17.91647   0.000000  0.000000 14.59452  0.000000  0.000000     1.00000    0.000000  3.872983
assure      11.31371 17.91647   0.000000  0.000000 14.59452  0.000000  0.000000     1.00000    0.000000  3.872983
be          15.00000 14.62874  14.594520 14.594520  0.00000 14.594520 14.594520    14.62874   14.594520 14.282857
bell        11.31371 17.91647   0.000000  0.000000 14.59452  0.000000  0.000000     1.00000    0.000000  3.872983
brief       11.31371 17.91647   0.000000  0.000000 14.59452  0.000000  0.000000     1.00000    0.000000  3.872983
congressman 11.35782 17.77639   1.000000  1.000000 14.62874  1.000000  1.000000     0.00000    1.000000  4.000000
distinguish 11.31371 17.91647   0.000000  0.000000 14.59452  0.000000  0.000000     1.00000    0.000000  3.872983
first       11.61895 17.60682   3.872983  3.872983 14.28286  3.872983  3.872983     4.00000    3.872983  0.000000
```

Hierarchical Cluster

- Create a hierarchical cluster object
  - hclust(x, *method*)
  - method: "ward.D", "ward.D2", "single", "complete", "average", "mcquitty", "median" or "centroid"

```
> library(factoextra)
> jfk_speech.dtm.cluster <-
+   jfk_speech.dtm %>%
+   stats::dist(method="euclidean") %>%
+   hclust(method="ward.D2")
> jfk_speech.dtm.cluster %>%
+   fviz_dend
```



두개를 비교했더니
빈도수를 기반으로 유사도를 재는것
ANd they may 나오고 그러니깐 얘가 인식이 될수있다 비슷하다고

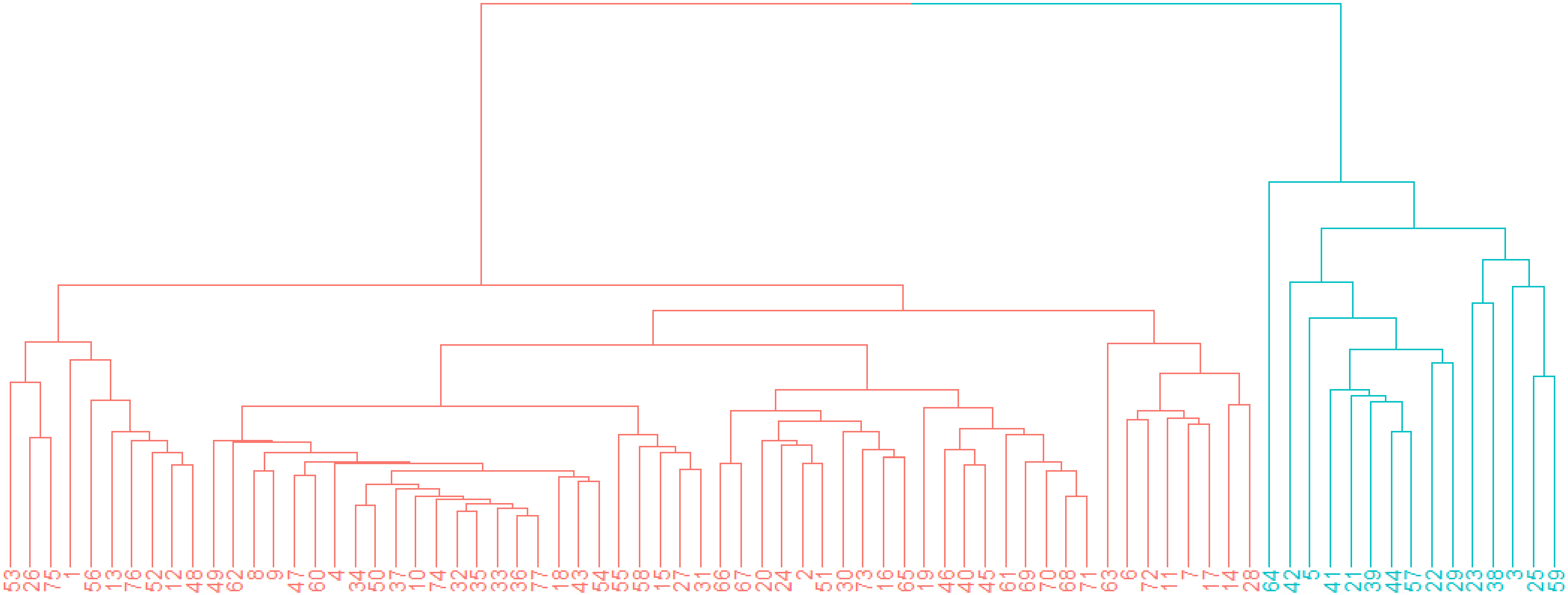
```
> jfk.speech.sum[34,]
   doc_id                                                   text
34     34 And they may well ask why climb the highest mountain?
> jfk.speech.sum[50,]
   doc_id                        text
50     50 And they may be less public.
```

Hierarchical Cluster

- Add a “cluster”

10개이상의 클러스터를 만드는것을 추천하지 않는다.

```
> jfk_speech.dtm.cluster %>%
+   cutree(k=2)
 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 
 1  1  2  1  2  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  2  2  2  1  2  1  1  1  2 
30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 
 1  1  1  1  1  1  1  1  2  2  1  2  2  1  2  1  1  1  1  1  1  1  1  1  1  1  1  2  1 
59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 
 2  1  1  1  1  2  1  1  1  1  1  1  1  1  1  1  1  1  1 
> jfk_speech.dtm.cluster %>%
+   fviz_dend(k=2)
```



Distance Matrix

- Create a distance matrix

```
> data("crude")
> crude.cleaned <- crude %>%
+   tm_map(removePunctuation) %>%
+   tm_map(removeNumbers) %>%
+   tm_map(removeWords, stopwords('en')) %>%
+   tm_map(stripWhitespace) %>%
+   tm_map(content_transformer(lemmatize_strings)) %>%
+   tm_map(content_transformer(tolower))
```

From Lecture 7

```
> crude.dtm.dist <-
+   crude.cleaned %>%
+   DocumentTermMatrix() %>%
+   stats::dist(method="euclidean")
> crude.dtm.dist
          127       144       191       194       211       236       237       242       246       248
144 28.390139
191  9.695360 30.133038
194 10.000000 30.692019  7.348469
211 13.674794 30.740852 11.357817 12.206556
236 25.258662 27.604347 26.907248 27.349589 27.730849
237 24.799194 33.749074 24.677925 24.919872 24.698178 31.160873
242 13.379088 28.407745 12.206556 13.152946 13.856406 25.436195 24.657656
246 22.561028 33.481338 22.561028 23.000000 22.494444 30.282008 28.600699 23.065125
248 22.135944 27.092434 24.819347 25.377155 25.748786 23.065125 31.288976 23.302360 29.410882
273 26.532998 31.048349 27.531800 27.568098 28.442925 27.349589 32.140317 27.404379 32.449961 27.784888
349 12.529964 28.827071 10.908712 12.041595 13.564660 25.238859 24.859606 12.489996 22.978251 23.345235
352 11.575837 27.820855 11.661904 13.038405 14.247807 25.219040 25.238859 12.922848 23.086793 18.439089
353 11.789826 26.925824 10.344080 11.532563 12.884099 22.825424 24.535688 11.916375 22.271057 22.248595
368 13.564660 30.397368 11.916375 12.961481 13.601471 27.928480 24.879711 14.106736 23.473389 26.229754
489 13.304135 29.614186 12.529964 13.304135 12.806248 26.776856 24.576411 14.560220 23.151674 24.433583
502 14.662878 29.546573 14.525839 15.132746 14.628739 27.110883 24.657656 16.062378 23.790755 24.919872
543  9.486833 29.597297  9.380832  9.273618 12.609520 26.076810 24.718414 13.674794 23.086793 23.790755
704 23.237900 33.406586 22.803509 22.934690 24.020824 32.062439 30.838288 23.130067 30.248967 29.664794
708 12.806248 30.692019  9.899495 11.045361 11.445523 27.712813 25.159491 13.000000 22.561028 25.884358
```

Create Euclidean distance Matrix

- Create a hierarchical cluster object

```
> crude.dtm.cluster <-
+   crude.dtm.dist %>%
+   hclust(method="ward.D2")
> crude.dtm.cluster

Call:
hclust(d = ., method = "ward.D2")

Cluster method   : ward.D2
Distance         : euclidean
Number of objects: 20

> crude.dtm.cluster %>%
+   fviz_dend
```

같은단어가 많이 나오는 케이스

- Add a “cluster”

```
> crude.dtm.cluster %>%
+   fviz_dend(k=3)
```