대통령 연설문 분석하기

가벼운 Text Analytics을 중심으로

유충현

Updated: 2017/09/19

Overview

1. Learning Outline

2. Scraping Data

3. 수집 데이터 전처리

4. Text Analytics 개요

5. Frequency Analysis

6. Association Rules

7. Clustering / Topic Analytics

Learning Outline

First Topic

웹에 게시된 데이터를 수집하는 scraping 기능은, 데이터 획득의 주요한 방법이다. HTML tag의 이해가 따르지만 R을 이용하여 간단하게 원하는 데이터를 추출하여 수집하는 방법을 생각해 보자.

R에서 데이터를 scraping하는 기술은 데이터 조작 함수를 구사하는 능력에 비례한다. 그 이유는 원하는 부분의 데이터만 취하기 위해서는 데이터 조작이 필수적이기 때문이다. 물론 원천 데이터가 웹 채널에 있기 때문에 HTML 문법과 이를 긁어올 수 있는 rvest 패키지 등의 사용법도 중요하다.

Second Topic

텍스트 분석 빛 좋은 개살구다. 노가다를 방불케 하는 데이터 전처리가 따르며, 그 결과도 우리가 알고 있는 fancy한 모습과 거리가 멀다.

Text Analytics는 NLP(Natural language processing)를 위한 전문적인 도구와 기술이 필요하다. 그러나 다행이도 일반적인 데이터 분석 기법을 적용할 수도 있고, 최근에는 Deep Learning 기법이 응용되는 분야이기도 하다. 일반적인 데이터 분석 기법을 적용하여 Text Analytics을 이해해 보자.

한글 텍스트 데이터가 필요한데 ?!

"나는 데이터를 조작하기 위해서 그 많은 연산자와 함수를 익혀야만 하였다. 그리고 왕년에는 웹 어플케이션도 개발해 보았다. 원치 않았지만, 이것들이 scraping에 유용하게 활용될 수 있음을 깨달았다."

당신 또한 이 기능을 익혀야할 수도 있다.!!!

- regular expression:
  - `Pattern Matching and Replacement, grep, gsub.`
- rvest package: HTML 문법 숙지가 어느 정도 필요함
  - `read_html()`: 원하는 URL의 웹 페이지 수집.
  - `html_nodes()`: 특정 노드의 값들을 추출.
  - `html_children()`: 자식 노드들을 추출.
  - `html_attrs()`: HTML Tag의 attributes 추출.

알수록 어렵고, 할수록 부족해!!!

"Text Analytics 전문가가 아니라 한계가 느껴지고, 할수록 어려운 분야임을 실감하고 있다. 그러나 장난 수준을 결과는 도출할 수 있게 되었다."

여러분들은 아니길 바랍니다.!!!

- NLP:
  - 형태소분석
    - KoNLP package, RMeCab package
- Document Term Matrix:
  - `Basic Architecture`
    - Term Frequency, Boolean Term Frequency, TF-IDF
- 분석 기법의 TA 응용:
  - `Association Rules`
  - `Clustering` : Hireracy Clustering, Kmeans Clusterig
  - `Topic Model` : LDA (Latent Dirichlet Allocation)

나의 작업환경

```
               _
platform       x86_64-apple-darwin13.4.0
arch           x86_64
os             darwin13.4.0
system         x86_64, darwin13.4.0
status
major          3
minor          3.2
year           2016
month          10
day            31
svn rev        71607
language       R
version.string R version 3.3.2 (2016-10-31)
nickname       Sincere Pumpkin Patch
```

데이터 준비하기

- 대통령기록연구실 홈페이지
  - http://www.pa.go.kr/research/contents/speech/index.jsp

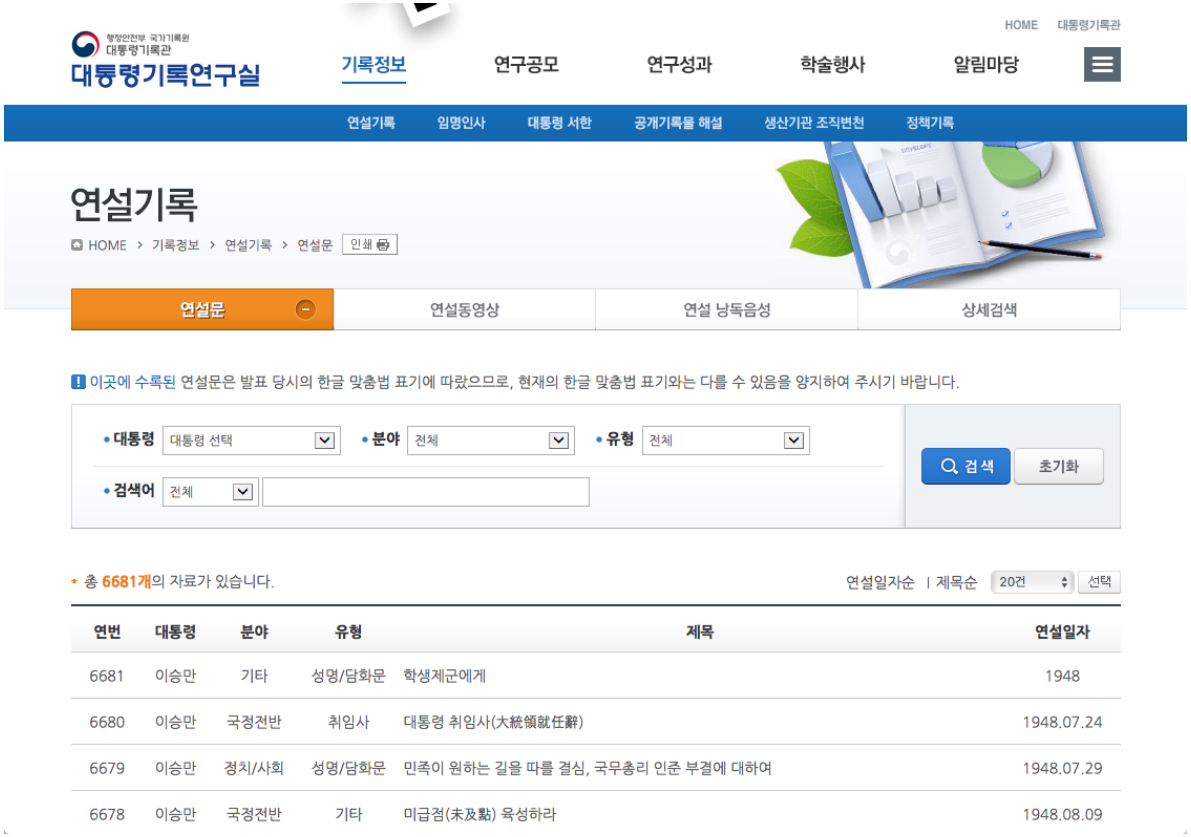


그림: 연설기록 제공 화면

Scraping Data

웹 페이지 소스 보기

웹 브라우저의 기능을 이용한 소스 보기

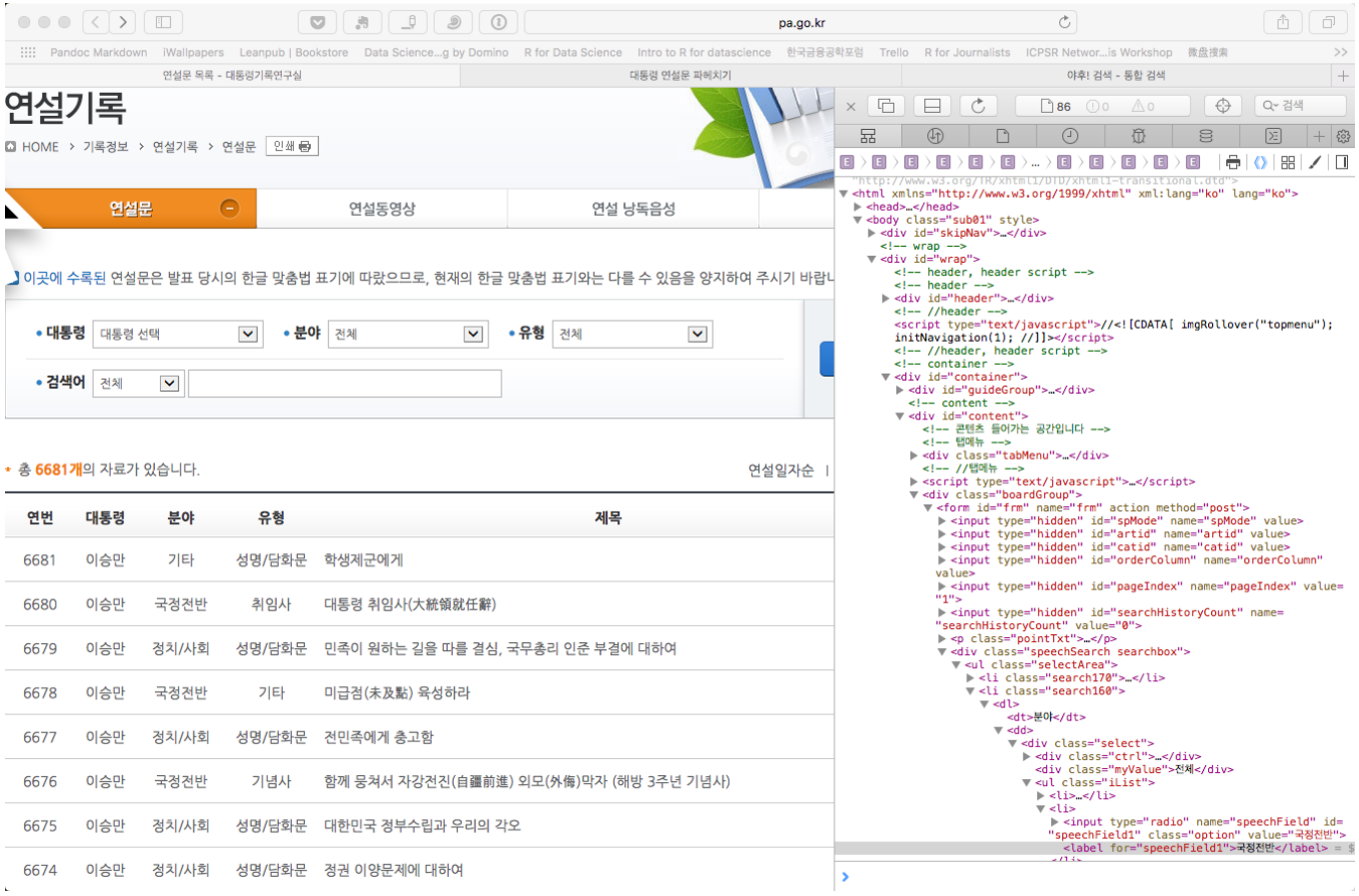
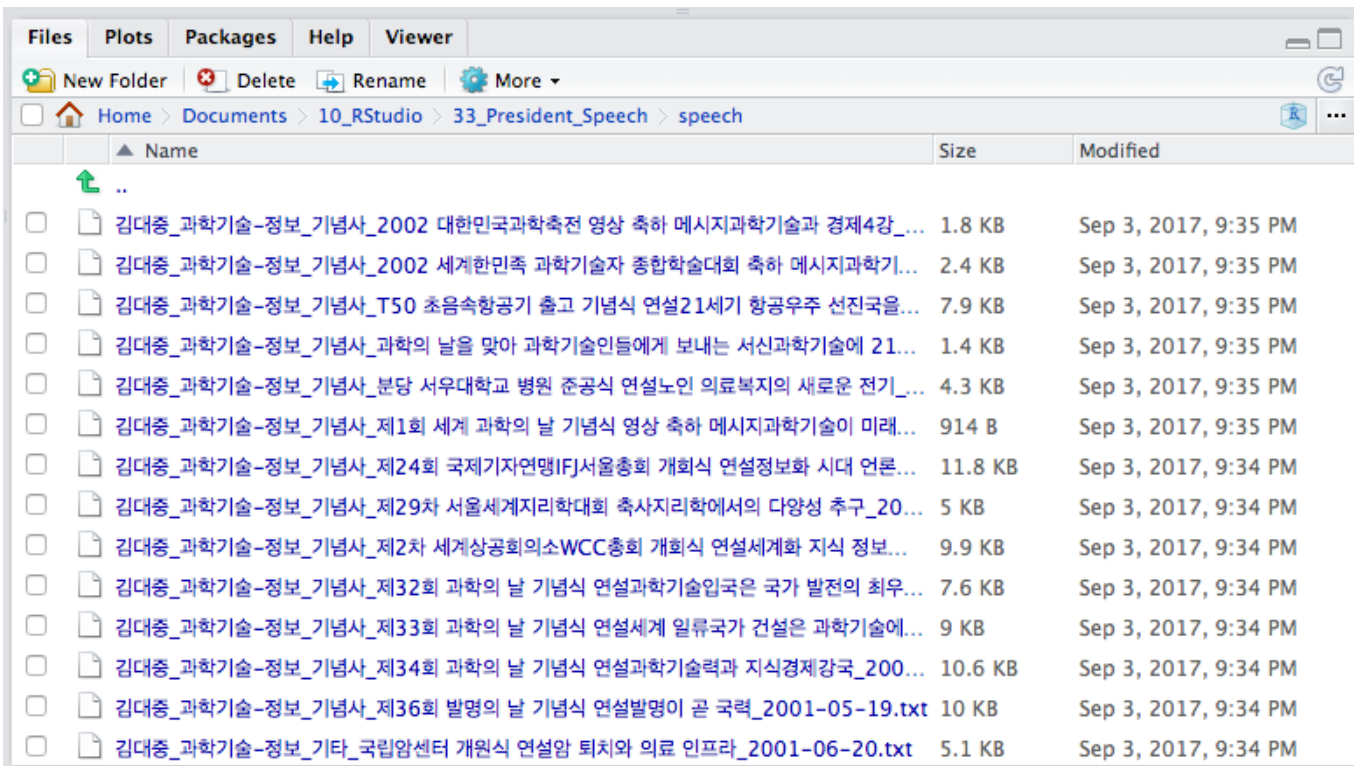


그림: 사파리에서 소스보기

대통령 연설문 수집하기

- 김대중, 노무현, 이명박 전 대통령의 연설문 수집
- 수집 데이터의 형태
  - 연설문 단위로 파일에 저장
    - "성명_연설분야_연설유형_연설제목_연설날짜.txt"
    - working directory 하위의 "speech"라는 디렉토리에 저장



연설문 목록 정보 추출하기

1. 목록조회 URL 생성
   - URL Main
     - http://www.pa.go.kr/research/contents/speech/index.jsp
   - URL Arguments
     - ?pageIndex=1&presidents=노무현
2. read_html() 함수로 html code 읽어오기
3. html code를 parsing하여 목록 페이지 개수 산정
4. pageIndex에 페이지 번호를 순차적으로 기술하면서 반복,
   - html_nodes() 함수 등을 이용하여 연설문 정보를 추출하여 파일 이름 정의
   - 연설문 보기 링크를 발췌하여 연설문 읽기 (이후 설명)

연설문 목록 정보 추출하기 - 화면

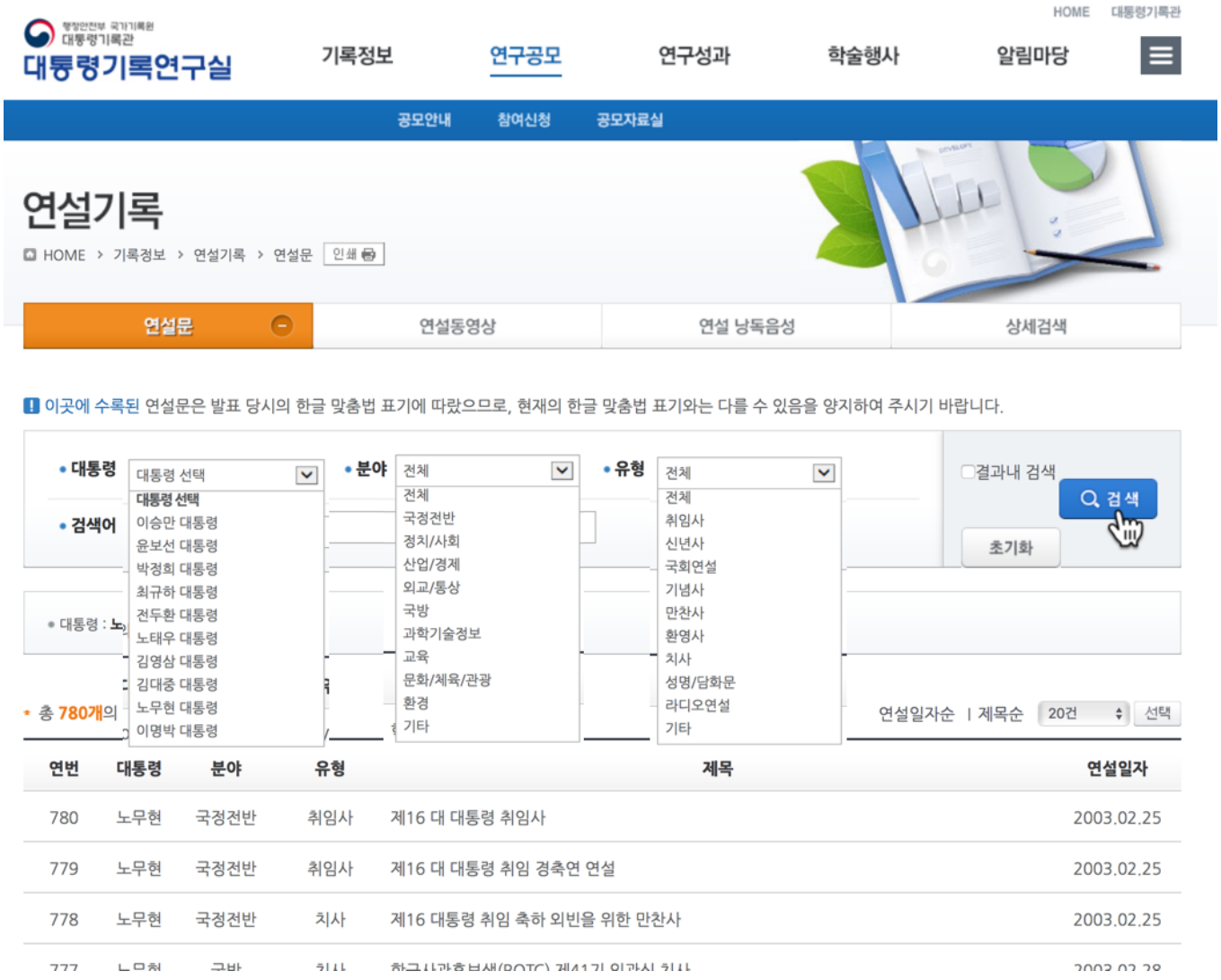


그림: 연설문 목록 정보

연설문 내용 추출하기

1. 연설문보기 링크의 URL 발췌
   - URL Main
     - http://www.pa.go.kr/research/contents/speech/index.jsp
   - URL Arguments
     - ?spMode=viewcatid=c_pa02062artid=1309347
2. read_html() 함수로 html code 읽어오기
3. html_nodes() 함수 등일 이용해서 연설문의 내용 추출
4. 연설문 내용 파일에 저장
   - file(), cat(), close() 함수 등을 이용

연설문 내용 추출하기 - 화면

그림: 연설문 내용

연설문 목록 정보 추출하기 - 화면

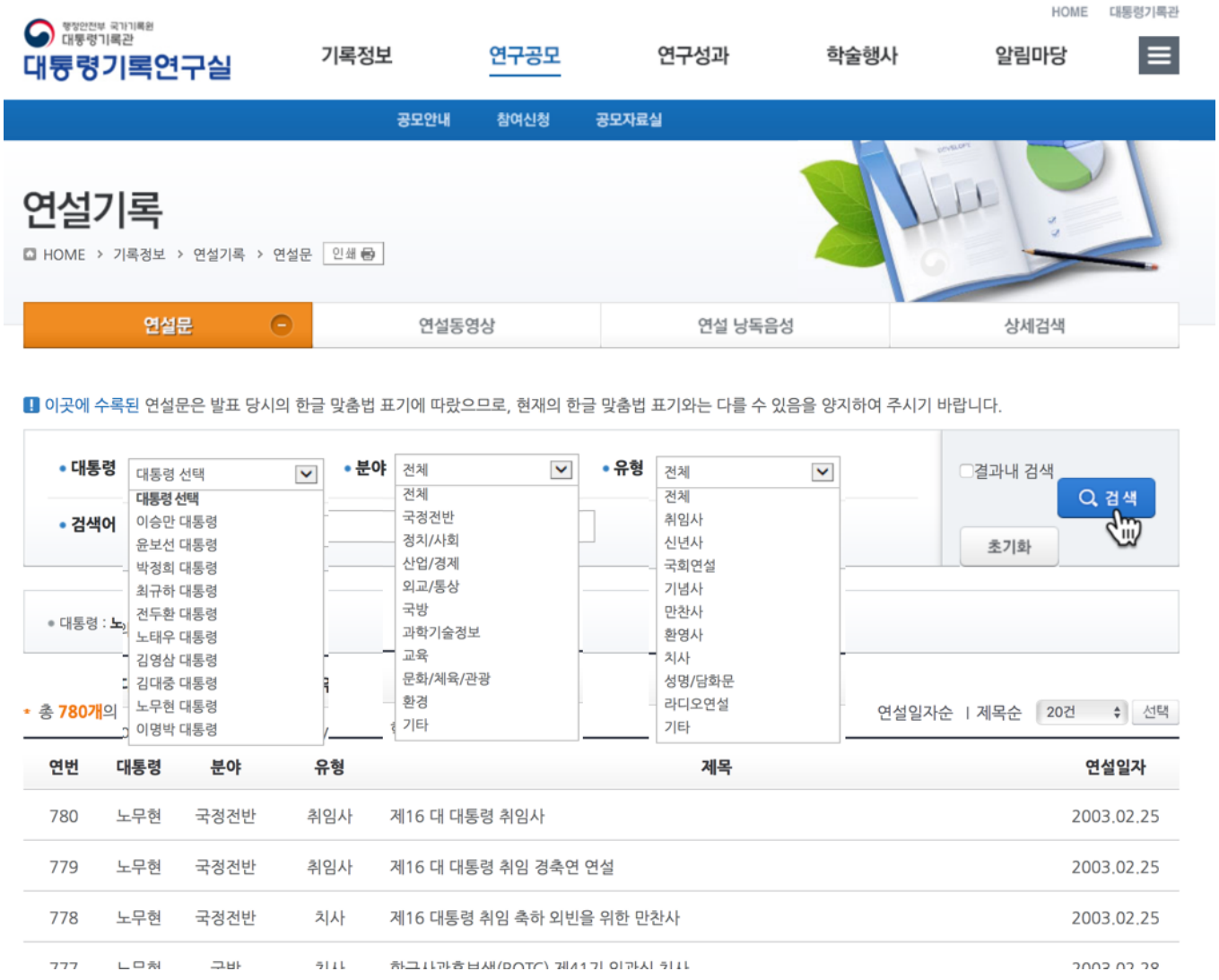


그림: 연설문 목록 정보

홈페이지 html 읽어오기

"홈페이지의 내용을 담은 html을 수집하자"

- read_html() 함수
  - read_html(URL, ...)
  - 당연히, 인터넷에 연결되어 있는 환경에서 수행
  - 수행 결과는 xml_document 클래스 객체
- exercises
  - 노무현 전 대통령의 연설문 목록화면을 수집하자.

exercises: 연설문 목록화면 수집

```
> library(rvest)
>
> URL0 <- "http://www.pa.go.kr/research/contents/speech/index.
> pres <- "노무현"
> URL <- sprintf("%s?pageIndex=1&damPst=%s", URL0, pres)
> pageTxt <- read_html(URL)
> pageTxt

{xml_document}
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="ko" lang=
[1] <head>\n<meta http-equiv="Content-Type" content="text/html
[2] <body class="sub01">\n<div id="skipNav">\n\t<ul>\n<li><a h
```

특정 값 추출하기

"xml_document 객체에서 특정 값을 추출하자"

- html_nodes() 함수 : 노드를 검색으로 추출하기
  - html_nodes(x, css, xpath)
- html_children() 함수 : 자식노드 추출
  - html_children(x)
- html_attrs() 함수 : 노드의 속성값 추출
  - html_attrs(x)
- exercises
  - 연설문 목록에서 페이지 목록의 개수 추출

exercises: 페이지 목록의 개수 추출

```
> pages <- html_nodes(pageTxt, "form#frm div.boardPage li") %>%
+   html_children %>%
+   .[length(.)] %>%
+   html_attrs %>%
+   unlist %>%
+   .[2] %>%
+   gsub(pattern = "[^[:digit:]]", replace = "") %>%
+   as.integer
>
> pages

[1] 39
```

첫 페이지의 목록 내용 추출

"첫 페이지의 목록 내용을 추출하자"

- exercises
  - 노무현 전 대통령의 연설문 목록의 첫 페이지를 추출하자.

1. URL 생성 후 read_html() 함수로 수집
   - pageIndex: 페이지 번호
   - damPst: 대통령 이름
2. html_nodes() 함수로 해당 노드 추출
3. html_children() 자식 노드 추출

exercises: 첫 페이지 목록 내용 추출

```
> URL <- sprintf("%s?pageIndex=%s&damPst=%s", URL0, 1, pres)
>
> tableTxt <- read_html(URL)
> page <- html_nodes(tableTxt, "form#frm tbody") %>%
+   html_children()
> length(page)

[1] 20

> page[1]

{xml_nodeset (1)}
[1] <tr>\n<td scope="row">780</td>\n\t\t\t\t\t\t\t<td><a href=
```

첫 페이지 첫 연설 내용 추출

"첫 페이지 첫 연설 내용을 추출하자"

- exercises
  - 노무현 전 대통령의 첫 연설 내용을 추출하자.

1. 목록의 첫 노드에서 앵커를 추출 후 연설문 URL 추출
   - html_nodes("a")
   - html_attr(name = "href")
2. URL 생성 후 read_html() 함수로 수집
3. html_nodes() 함수로 연설문이 들어 있는 노드 추출
4. html_text() 함수로 연설문 텍스트 추출
5. gsub() 함수로 특수문자 제거

exercises: 첫 페이지 첫 연설 내용 추출

```
> value <- html_nodes(page[1], "a") %>%
+   .[1] %>%
+   html_attr(name = "href")
>
> URL <- sprintf("%s%s", URL0, value)
> contentTxt <- read_html(URL)
>
> contents <- html_nodes(contentTxt, "div#content div.conTxt")
+   html_text %>%
+   gsub(pattern = "[[:cntrl:]]", replacement = "")
> substr(contents, 1, 50)

[1] "존경하는 국민 여러분.오늘 저는 대한민국의 제16대 대통
령에 취임하기 위해 이 자리에 섰습"
```

수집 데이터 전처리

데이터 전처리 1

```
> library(dplyr)
> fname <- system("ls ./speech", intern = TRUE) %>%
+   iconv(from ="utf-8-mac", to = "utf-8")
> fname2 <- strsplit(fname, "_")
> fnames <-
+   data.frame(president = sapply(fname2, "[", 1),
+              category = sapply(fname2, "[", 2),
+              type = sapply(fname2, "[", 3),
+              title = sapply(fname2, "[", 4),
+            date = sub("\\.txt", "", sapply(fname2, "[", 5)),
+              stringsAsFactors = FALSE)
> dim(fnames) # 몇 건의 연설문을 수집하였는가?


[1] 2410    5
```

데이터 전처리 2

```
> lst <- fnames[1:3, ]
> lst$title <- substr(lst$title, 1, 7)
> knitr::kable(lst)
```

president	category	type	title	date
노무현	외교-통상	치사	2005 한일	2005-01-27
노무현	외교-통상	치사	2007 한중	2007-01-01
노무현	외교-통상	기타	2007 남북	2007-10-02

```
> # 대통령별로 몇 건의 연설문을 수집했는가?
> table(fnames$president)



김대중 노무현 이명박
   822    780    808
```

데이터 전처리 3

```
> # 데이터 프레임에 연설내용 붙이기
> getDocs <- function(x) {
+   readtext::readtext(x) %>%
+   dplyr::select(text) %>% as.character
+ }
> fnames$doc <- sapply(paste("speech", fname, sep = "/"), getDo
> fnames <-
+   data.frame(docid = sprintf("DOC_%04d", seq(NROW(fnames))),
+              fnames, stringsAsFactors = FALSE)
> fnames[1, 1:3]

     docid president  category
1 DOC_0001    노무현 외교-통상
```

Text Analytics 개요

RMeCab 패키지에 대해서

"RMeCab 패키지는 MeCab 형태소 분석기를 인터페이스하는 패키지다. MeCab 형태소 분석기는 일본에서 개발된 오픈소스로 국내에서는 은전한닢 프로젝트로 한글 형태소 분석기로 포팅하여 배포되고 있다."

- 장점
  - 수행속도가 빠르다. : C++ 기반
  - 띄어쓰기에 민감하지 않다.
- 단점
  - 유지보수가 상대적으로 쉽지 않다.

수행속도 - 데이터 준비

```
> doc <- "노약자를 비롯한 소외받는 사람들에게 더 많은 관심을 기울이는 따뜻한 사회를 만들어야 합니다. 이를 위해 복지 정책을 내실화하고자 합니다. 모든 종류의 불합리한 차별을 없애 나가겠습니다. 양성평등사회를 지향해 나가겠습니다. 개방화 시대를 맞아 농어업과 농어민을 위한 대책을 강구하겠습니다. 고령사회의 도래에 대한 준비에도 소홀함이 없도록 하겠습니다. 반칙과 특권이 용납되는 시대는 이제 끝나야 합니다. 정의가 패배하고 기회주의자가 득세하는 굴절된 풍토는 청산되어야 합니다. 원칙을 바로 세워 신뢰사회를 만듭시다. 정정당당하게 노력하는 사람이 성공하는 사회로 나아갑시다. 정직하고 성실한 대다수 국민이 보람을 느끼게 해드려야 합니다."
> doc10 <- paste(doc, doc, doc, doc, doc,
+                doc, doc, doc, doc, doc)
```

수행속도 - 리소스 준비

```
> library(KoNLP)
> library(RMeCab)
> library(microbenchmark)
> library(dplyr)
>
> useNIADic()

Backup was just finished!
983012 words dictionary was built.

> useKoNLP <- function(x) SimplePos22(x)
> useRMeCab <- function(x) RMeCabC(x)
```

수행속도 - 리소스 준비

```
> result <- microbenchmark(useKoNLP(doc10),
+     useRMeCab(doc10))
> print(result)

Unit: milliseconds
           expr       min        lq      mean    median        uq
  useKoNLP(doc10) 258.774159 269.009951 395.512450 273.833662 283.787
 useRMeCab(doc10)   6.829906   7.551557   8.867694   8.457187   9.6104
       max neval cld
 8663.5093   100   b
   29.3356   100  a
```

띄어쓰기 - 정상적인 문장

```
> str1 <- "아버지가 방에 들어가신다."
> sapply(useKoNLP(str1), c)

                    아버지가                                 방에
          "아버지/NC+가/JC"                                    "
방/NC+에/JC"
                    들어가신다                                .
"들/PV+어/EC+가/PX+시/EP+ㄴ다/EF"                             "./SF"

> sapply(useRMeCab(str1), c)

     NNG      JKS      NNG      JKB       VV    EP+EF       SF
"아버지"        "가"       "방"       "에"  "들어가"       "
신다"       "."
```

띄어쓰기 - 비정상적인 문장

```
> str2 <- "아버지가방에들어가신다."
> sapply(useKoNLP(str2), c)

                                                    아버지가방에들어가신다
"아버지가방/NC+에/JC+이/JP+들/EP+어/EC+가/PX+시/EP+ㄴ다/EF"
                                                                        .
                                                                     "./SF"

> sapply(useRMeCab(str2), c)

     NNG       JKS       NNG       JKB        VV    EP+EF        SF
"아버지"        "가"       "방"        "에"  "들어가"      "
신다"       "."
```

띄어쓰기가 형편없는 문서의 띄어쓰기 보정 1

```
> ### 정상적인 문장
> str3 <- "못찾겠다. 꾀꼬리 꾀꼬리 나는야 오늘도 술래."
>
> sapply(useRMeCab(str3), c)

    MAG      VV      EP      EF      SF     NNG     NNG      NP
   "못"     "찾"    "겠"    "다"     "." "꾀꼬리"  "
꾀꼬리"    "나"
     JX      IC     NNG      JX     NNG      SF
   "는"    "야"  "오늘"    "도"  "술래"     "."
```

띄어쓰기가 형편없는 문서의 띄어쓰기 보정 2

```
> ### 띄어쓰기비정상적인 문장
> str4 <- "못찾 겠다.꾀꼬 리꾀꼬리나는 야오 늘도술래."
>
> sapply(useRMeCab(str4), c)

    MAG      VV      EP      EF      SF      VV      EC     NNG
   "못"     "찾"     "겠"     "다"      "."     "꾀"     "
꼬"     "리"
    NNG      NP      JX      IC     MAG      JX     NNG      SF
"꾀꼬리"     "나"     "는"   "야오"     "늘"     "도"   "
술래"      "."
```

띄어쓰기가 형편없는 문서의 띄어쓰기 보정 3

```
> ### 모든 문자를 붙인 문장
> str5 <- "못찾겠다.꾀꼬리꾀꼬리나는야오늘도술래."
>
> sapply(useRMeCab(str5), c)

    MAG      VV      EP      EF      SF     NNG     NNG      NP
  "못"     "찾"    "겠"    "다"     "."  "꾀꼬리"  "꾀꼬리"  "나"
   JX      IC     NNG      JX     NNG      SF
 "는"    "야"  "오늘"    "도"  "술래"     "."
```

Mecab 형태소분석기의 품사 Tag (1/3)

tag	pos	class	large
NNG	일반 명사	실질형태소	체언
NNP	고유 명사	실질형태소	체언
NNB	의존 명사	실질형태소	체언
NNBC	단위를 나타내는 명사	실질형태소	체언
NR	수사	실질형태소	체언
NP	대명사	실질형태소	체언
VV	동사	실질형태소	용언
VA	형용사	실질형태소	용언
VX	보조 용언	실질형태소	용언
VCP	긍정 지정사	실질형태소	용언
VCN	부정 지정사	실질형태소	용언
MM	관형사	실질형태소	수식언
MAG	일반 부사	실질형태소	수식언
MAJ	접속 부사	실질형태소	수식언
IC	감탄사	실질형태소	독립언

Mecab 형태소분석기의 품사 Tag (2/3)

	tag	pos	class	large
16	JKS	주격 조사	형식형태소	관계언
17	JKC	보격 조사	형식형태소	관계언
18	JKG	관형격 조사	형식형태소	관계언
19	JKO	목적격 조사	형식형태소	관계언
20	JKB	부사격 조사	형식형태소	관계언
21	JKV	호격 조사	형식형태소	관계언
22	JKQ	인용격 조사	형식형태소	관계언
23	JX	보조사	형식형태소	관계언
24	JC	접속 조사	형식형태소	관계언
25	EP	선어말 어미	형식형태소	선어말어미
26	EF	종결 어미	형식형태소	어말어미
27	EC	연결 어미	형식형태소	어말어미
28	ETN	명사형 전성 어미	형식형태소	어말어미
29	ETM	관형형 전성 어미	형식형태소	어말어미
30	XPN	체언 접두사	형식형태소	접두사

Mecab 형태소분석기의 품사 Tag (3/3)

	tag	pos	class	large
31	XSN	명사 파생 접미사	형식형태소	접미사
32	XSV	동사 파생 접미사	형식형태소	접미사
33	XSA	형용사 파생 접미사	형식형태소	접미사
34	XR	어근	NA	어근
35	SF	마침표, 물음표, 느낌표	NA	부호
36	SE	줄임표 …	NA	부호
37	SSO	여는 괄호 (, [	NA	부호
38	SSC	닫는 괄호),]	NA	부호
39	SC	구분자 , · / :	NA	부호
40	SY		NA	부호
41	SL	외국어	NA	한글이외
42	SH	한자	NA	한글이외
43	SN	숫자	NA	한글이외

오늘 다룰 Text Analytics 내용

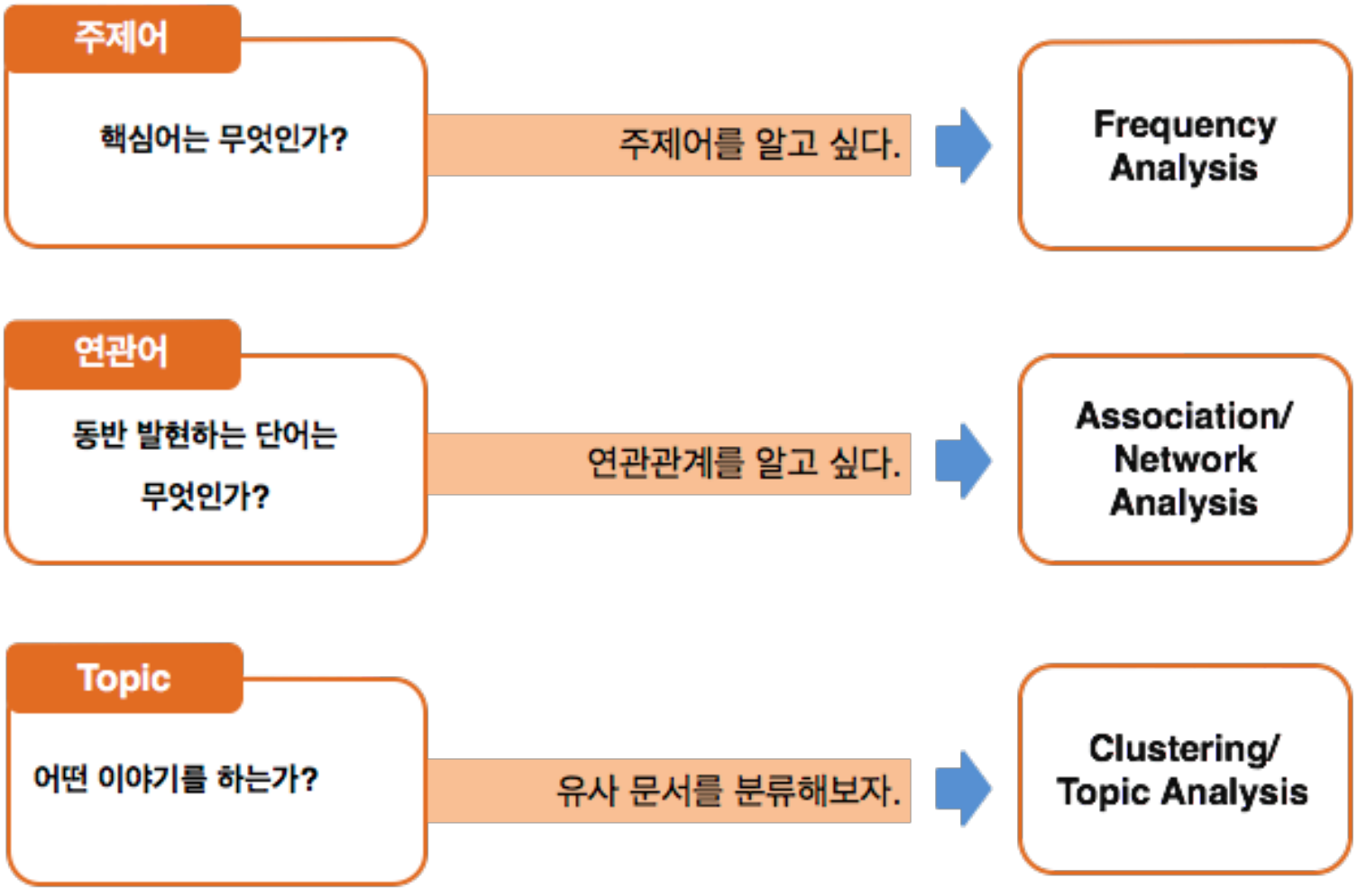


그림: 오늘 다룰 Text Analytics 내용

Frequency Analysis

Frequency Analysis 들어가기 : 학습 목표

- 데이터 구조
  - Document Term Matrix 이해하기
- 빈발단어
  - Term Frequency, Document Term Frequency 이해하기
  - 빈발단어 시각화하기
- 핵심단어
  - TF-IDF 이해하기

Document Term Matrix

- Documents(문서)에 발현하는 Term(단어 등)을 행렬의 구조로 나타낸 것
  - 행의 차원은 문서 ID
  - 열의 차원은 Term ID

● **발현 건수 행렬 (Frequency)**

	Term 1	Term 1	...	Term n
Doc 1	3	0	...	1
Doc 2	0	0	...	2
...	...	...	...	
Doc m	4	1	...	0

● **발현 여부 행렬 (Boolean Frequency)**

	Term 1	Term 1	...	Term n
Doc 1	1	0	...	1
Doc 2	0	0	...	1
...	...	...	...	
Doc m	1	1	...	0

그림: Document Term Matrix 구조

Term Frequency

- 개별 문서 내에서 발현한 Term의 Frequency
- 활용
  - 개별 문서 안에서 관심 있는 Term이 몇 번 발현되었는가?

	Doc 1	Doc 1	Doc 3
Term 1	3	0	4
Term 2	0	0	1
Term 3	1	2	0

Doc 1에서 Term 1의 TF는 3임

그림: Term Frequency 구조

Overall Term Frequency

- 문서와 상관없이 전체 문서들 중에서 발현한 Term의 Frequency
- 활용
  - 전체 문서에서 관심 있는 Term이 몇 번 발현되었는가?

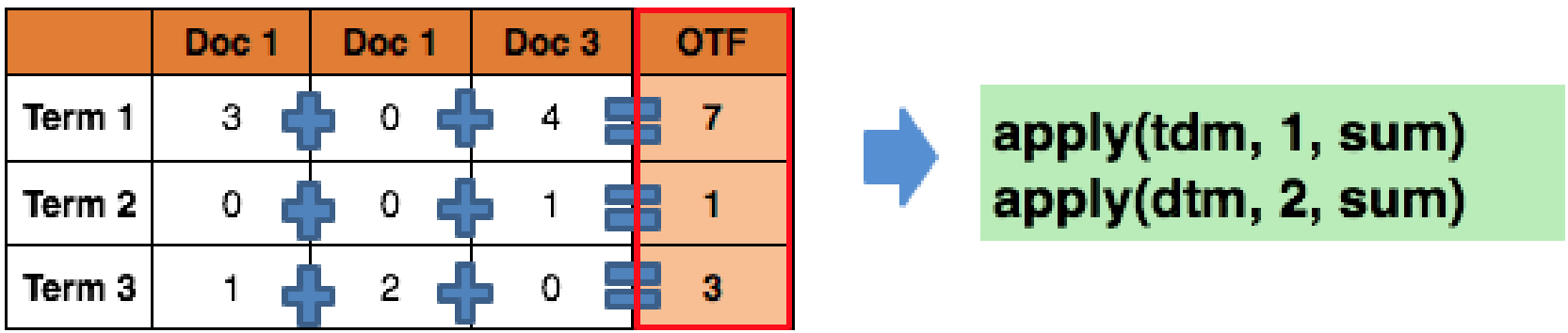


그림: Overall Term Frequency 구조

Document Frequency

- 개별 Term이 발현된 문서들의 Frequency
- 활용
  - 관심 있는 Term이 포함된 문서의 개수는 몇개인가?

- **DF → Boolean Frequency 이용**

	Doc 1	Doc 1	Doc 3	OTF
Term 1	1	0	1	2
Term 2	0	0	1	1
Term 3	1	1	0	2

```
apply(tdm, 1, sum)
apply(dtm, 2, sum)
```

그림: Document Frequency 구조

TF-IDF

- Term Frequency – Inverse Document Frequency
- 활용
  - 관심 있는 Term이 각각의 문서에서 차지하는 중요도는 얼마인가?

- **TF-IDF → TF와 IDF를 이용**

	Doc 1	Doc 1	Doc 3
Term 1	X.XXX	X.XXX	X.XXX
Term 2	X.XXX	X.XXX	X.XXX
Term 3	X.XXX	X.XXX	X.XXX

TF-IDF = TF * log(N / DF)
N : 전체 문서의 개수

그림: TF-IDF 구조

준비하기 - 사용자 정의 함수

```
> ## 형태소분석 - 품사 추출하기 정의
> getMorpMecab <- function(x, type = c("morpheme", "noun",
+                                      "noun2", "verb", "adj")[2]) {
+   library(RMeCab)
+   morpheme <- RMeCabC(x)
+   if (type == "morpheme") return(morpheme)
+   if (type == "noun") pattern <- "NNG|NNP"
+   if (type == "noun2") pattern <- "^N"
+   if (type == "verb") pattern <- "^VV"
+   if (type == "adj") pattern <- "^VA"

+   idx <- grep(pattern, sapply(morpheme, names))
+   return(unlist(morpheme[idx]))
+ }
```

준비하기 - Corpus 생성

```
> library(tm)
>
> corpus_kim <- fnames %>%  ## 김대중 대통령 Corpus
+   filter(president %in% c("김대중")) %>%
+   select(doc) %>% unlist %>%
+   VectorSource %>% Corpus
>
> corpus_noh <- fnames %>% ## 노무현 대통령 Corpus
+   filter(president %in% c("노무현")) %>%
+   select(doc) %>% unlist %>%
+   VectorSource %>% Corpus
>
> corpus_lee <- fnames %>% ## 이명박 대통령 Corpus
+   filter(president %in% c("이명박")) %>%
+   select(doc) %>% unlist %>%
+   VectorSource %>% Corpus
```

준비하기 - Corpus 생성

```
> ctrl_tf <- list(tokenize = getMorpMecab,
+                 weighting = weightTf,
+                 wordLengths = c(2, Inf))
>
> ctrl_bin <- list(tokenize = getMorpMecab,
+                  weighting = weightBin,
+                  wordLengths = c(2, Inf))
>
> ctrl_tfidf <- list(tokenize = getMorpMecab,
+                    weighting = weightTfIdf,
+                  wordLengths = c(2, Inf))
```

Term Document Matrix 생성

```
> ## Term Frequency
> tdmtf_kim <- TermDocumentMatrix(corpus_kim, control = ctrl_tf)
> tdmtf_noh <- TermDocumentMatrix(corpus_noh, control = ctrl_tf)
> tdmtf_lee<- TermDocumentMatrix(corpus_lee, control = ctrl_tf)
>
> ## Boolean Term Frequency
> tdmbtf_kim <- TermDocumentMatrix(corpus_kim, control = ctrl_bin)
> tdmbtf_noh <- TermDocumentMatrix(corpus_noh, control = ctrl_bin)
> tdmbtf_lee<- TermDocumentMatrix(corpus_lee, control = ctrl_bin)
>
> ## TF-IDF
> tdmtfidf_kim <- TermDocumentMatrix(corpus_kim, control = ctrl_tfid
> tdmtfidf_noh <- TermDocumentMatrix(corpus_noh, control = ctrl_tfid
> tdmtfidf_lee<- TermDocumentMatrix(corpus_lee, control = ctrl_tfidf
```

가장 많이 발현되는 명사

```
> ## Overall Term Frequency
> otf_kim <- apply(tdmtf_kim, 1, sum)
> otf_noh <- apply(tdmtf_noh, 1, sum)
>
> ## 가장 많이 발현되는 명사 상위 10 - 김대중
> sort(otf_kim, decreasing = TRUE)[1:10]

국민 경제 세계 한국 협력 나라 발전 정부 국가 생각
5222 4798 4458 3252 3109 2759 2624 2580 2213 2182

> ## 가장 많이 발현되는 명사 상위 10 - 노무현
> sort(otf_noh, decreasing = TRUE)[1:10]

국민 정부 경제 생각 문제 사회 발전 한국 사람 협력
3323 2963 2862 2669 2279 1950 1873 1836 1775 1733
```

가장 많은 문서에서 발현된 명사

```
> ## Overall Term Frequency (Boolean)
> botf_kim <- apply(tdmbtf_kim, 1, sum)
> botf_noh <- apply(tdmbtf_noh, 1, sum)
>
> ## 가장 많은 문서에서 발현된 명사 상위 10 - 김대중
> sort(botf_kim, decreasing = TRUE)[1:10]

국민 세계 발전 감사 경제 노력 나라 생각 국가 협력
 722  717  688  678  657  646  645  626  603  601

> ## 가장 많은 문서에서 발현된 명사 상위 10 - 노무현
> sort(botf_noh, decreasing = TRUE)[1:10]

감사 발전 국민 세계 정부 생각 나라 경제 노력 말씀
 636  599  557  505  500  491  463  458  450  442
```

빈발 Term 검색하기 - 1

```
> kim <- findFreqTerms(tdmbtf_kim, lowfreq = 650)
> kim

[1] "감사" "경제" "국민" "발전" "세계"

> noh <- findFreqTerms(tdmbtf_noh, lowfreq = 550)
> noh

[1] "감사" "국민" "발전"
```

빈발 Term 검색하기 - 2

```
> lee <- findFreqTerms(tdmbtf_lee, lowfreq = 550)
> lee

[1] "감사" "경제" "국가" "국민" "나라" "생각" "세계"

> ## 연설문에서 공통 발현된 단어는?
> intersect(intersect(kim, noh), lee)

[1] "감사" "국민"
```

빈발 term의 시각화 1

```
> library(wordcloud)
> library(RColorBrewer)
>
> pal <- brewer.pal(8,"Dark2")
>
> ## 김대중 대통령의 연설문에서 발현되는 단어
> wordcloud(word = names(otf_kim), freq = otf_kim,
+           min.freq = 150, colors = pal)
>
> ## 김대중 대통령의 연설문에서 발현되는 단어 - 연설문의 규모
> wordcloud(word = names(botf_kim), freq = botf_kim,
+           min.freq = 50, colors = pal)
```

빈발 term - Term Frequency 규모

빈발 term - 발현 Documents 규모

검색엔진 흉내내기

"TF-IDF를 이용해서 일자리와 관련이 높은 문서 조회"

```
> tfidf_noh <- as.matrix(tdmtfidf_noh)
> search_doc <- sort(tfidf_noh[row.names(tfidf_noh) %in% "일자
리", ],
+                    decreasing = TRUE)[1:3]
> search_doc

      514       364       579
0.4828965 0.3727622 0.1748761

> search_doc_01 <- fnames %>%
+   filter(docid == "DOC_0514") %>%
+   dplyr::select(doc) %>% unlist
```

검색엔진 흉내내기 - 결과

```
> search_doc_01


"  시청자 여러분, 안녕하십니까,  지금 일을 하고 싶어도 일자리를 찾지 못하고 있는 많은 분들이 계십니다. 그 분들의 안타까운 심정을 누가 대신해줄 수 있겠습니까, 가족들의 고통 또한 얼마나 크겠습니까,  그러나 결코 낙담하거나 포기하지 마십시오. 정부도 최선을 다하고 있습니다. 올해부터 5년 동안 200만개의 새로운 일자리를 창출하도록 하겠습니다.  공공부문부터 솔선해서 일자리를 만들 것입니다. 노,사,정도 일자리 창출을 위해 손을 맞잡았습니다. 충분하진 않지만 기업들이 작년보다 많은 인력을 뽑겠다고 합니다. 유한킴벌리처럼 일자리를 나누면서도 생산성은 획기적으로 높아진 성공사례들이 확산되어 갈 것입니다.  일할 기회는 반드시 찾아옵니다. 준비된 사람부터 그 기회를 잡게 될 것입니다. 희망과 자신감을 가지고 우리 함께 노력해 봅시다.  뜻깊은 프로그램을 제작해주신 KBS에 감사드리며, 이를 계기로 많은 분들이 새로이 일자리를 찾게 되기를 바랍니다. 감사합니다. "
```

Association Rules

Association Rules 들어가기 : 학습 목표

- Correlation
  - tm::findAssocs() 함수 이해하기
- Association Rules
  - Association Rules Analytics 이해하기
- Network 시각화
  - 빈발 Term의 Network 시각화하기

Correlation Analysis

"상관관계가 높은 단어 검색"

```
> tm::findAssocs(tdmbtf_kim, terms = "통일", corlimit = 0.4)

$통일
이산가족     공존     남북     분단     장차     화해     교류
    0.45     0.44     0.44     0.43     0.43     0.43     0.41

> tm::findAssocs(tdmbtf_noh, terms = "통일", corlimit = 0.3)

$통일
    북방     신라   부의장 평화통일
    0.36     0.36     0.33     0.33

> tm::findAssocs(tdmbtf_lee, terms = "통일", corlimit = 0.35)
```

Correlation Analysis

"상관관계가 높은 단어 검색"

```
> tm::findAssocs(tdmbtf_lee, terms = "통일", corlimit = 0.35)


$통일
      북한       남북     한반도       평화     남북한 
      0.46       0.43       0.43       0.40       0.39 
      분단       도발 자유민주주의 
      0.38       0.36       0.35 
```

Association Rules

"Association Rules 생성하기"

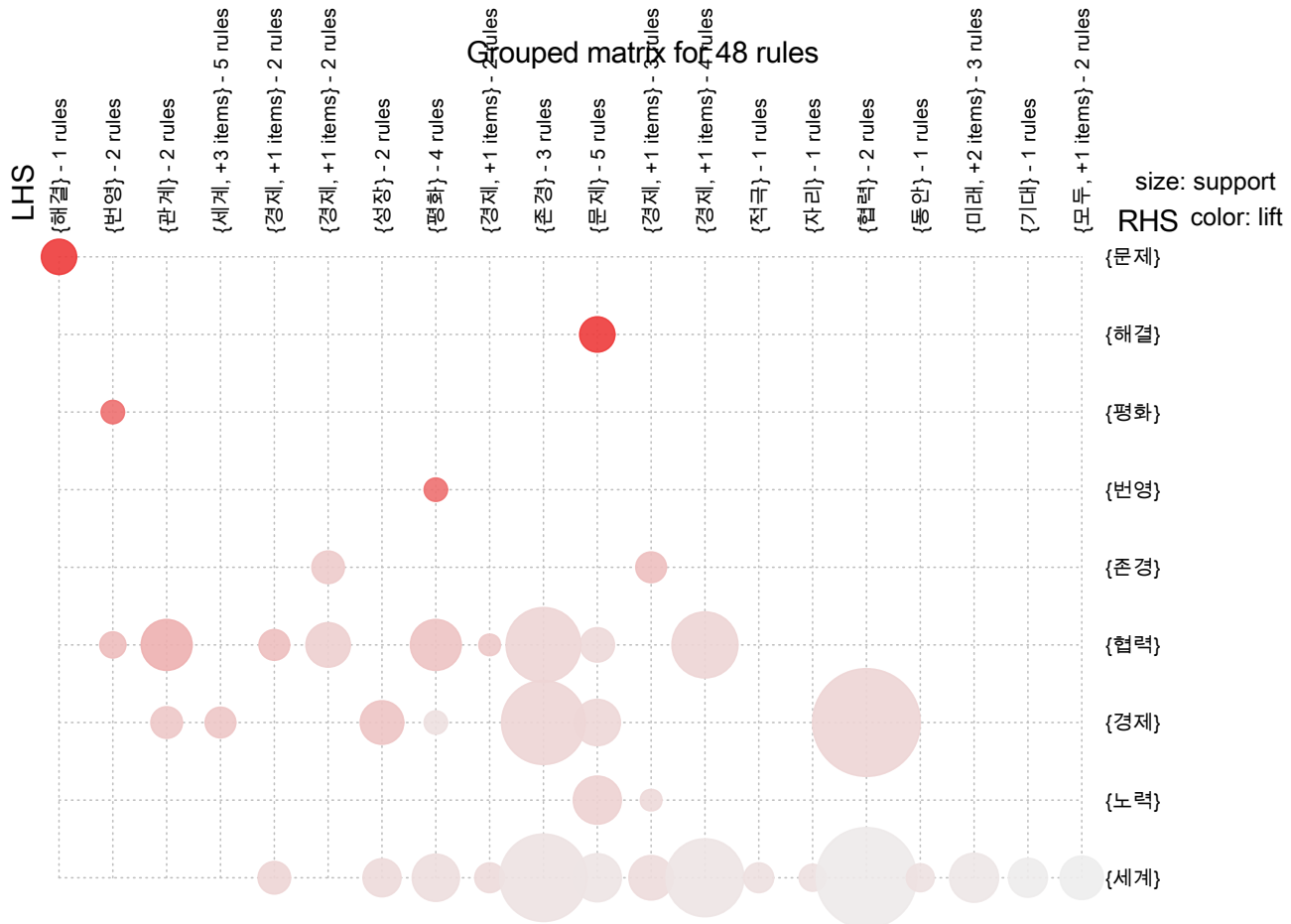
```
> library(arules)
> library(arulesViz)
> sw <- c("국민", "감사", "발전", "정부", "나라", "말씀",
+         "생각", "한국", "국가", "진심", "축하", "기원")
> tdm_bin <- tdmbtf_noh[!row.names(tdmbtf_noh) %in% sw, ]
> trans <- as(t(as.matrix(tdm_bin)), "transactions")
> rules <- apriori(trans,
+                  parameter = list(supp = 0.3,
+                                   conf = 0.7, target = "rules"),
+                  control = list(verbose = FALSE))
```

"Inspect Association Rules"

```
> inspect(rules[1:7])


    lhs        rhs    support   confidence lift
[1] {번영} => {평화} 0.3025641 0.8082192  1.870656
[2] {평화} => {번영} 0.3025641 0.7002967  1.870656
[3] {번영} => {협력} 0.3064103 0.8184932  1.444400
[4] {이번} => {협력} 0.3038462 0.7117117  1.255962
[5] {이번} => {세계} 0.3076923 0.7207207  1.113192
[6] {적극} => {세계} 0.3115385 0.7738854  1.195308
[7] {평화} => {협력} 0.3410256 0.7893175  1.392913
```

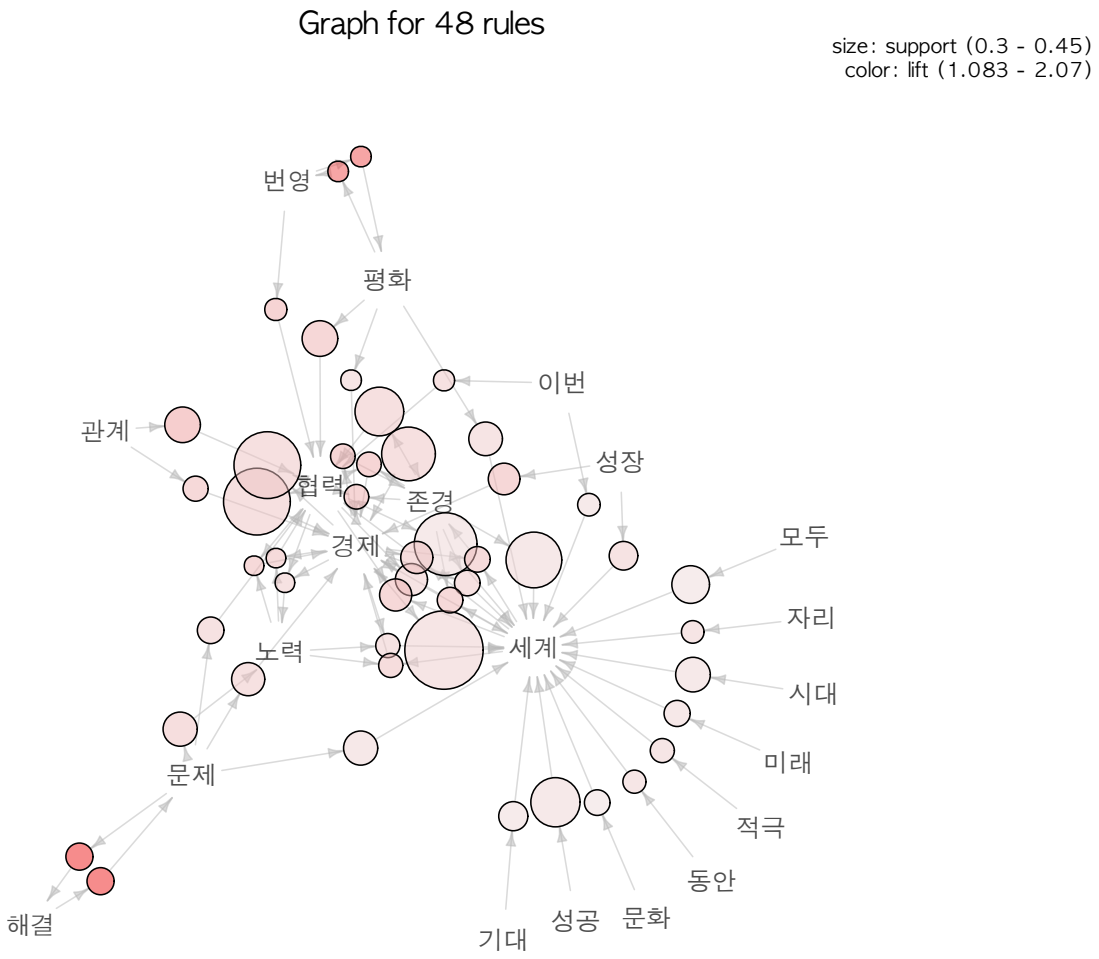
Association Rules 시각화 - grouped

```
> plot(rules, method = "grouped")
```



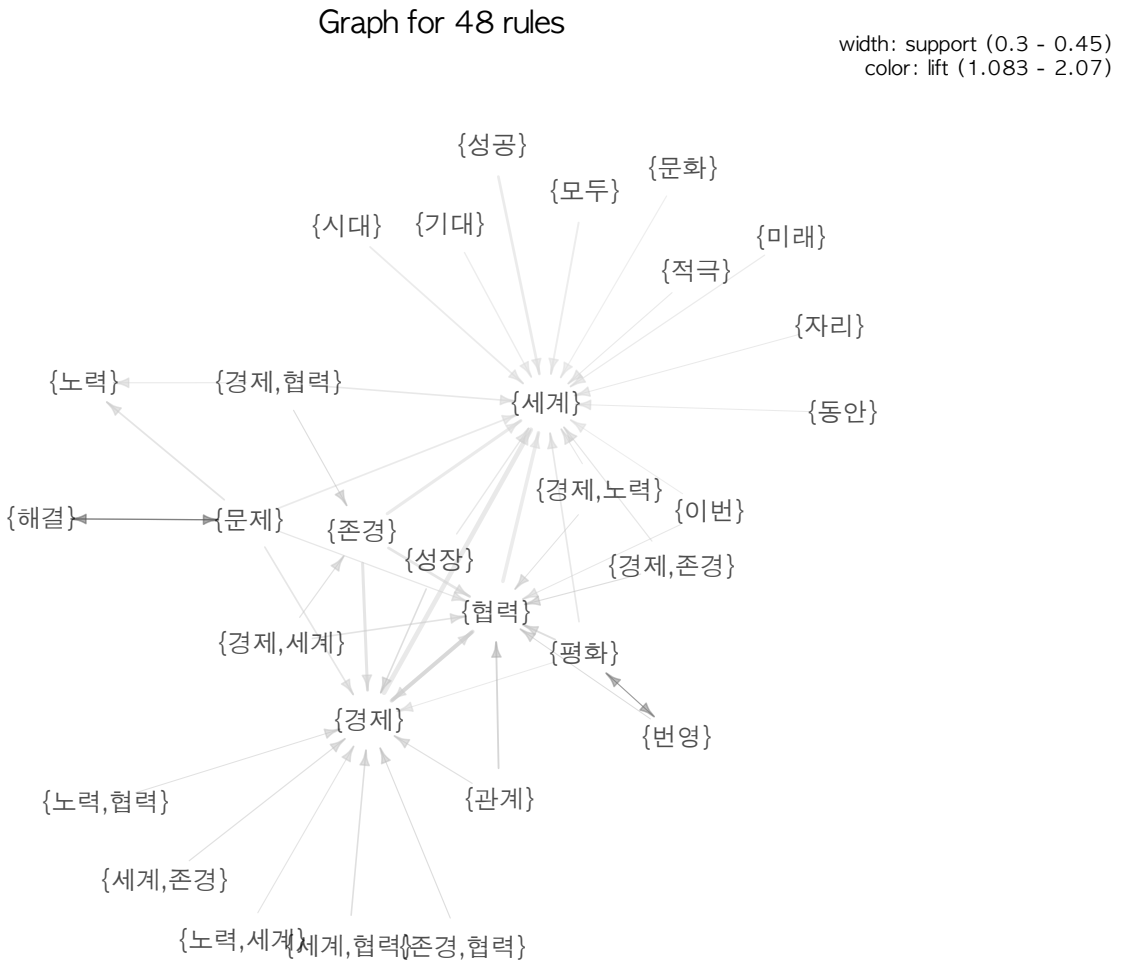
Association Rules 시각화 - graph

```
> plot(rules, method = "graph")
```

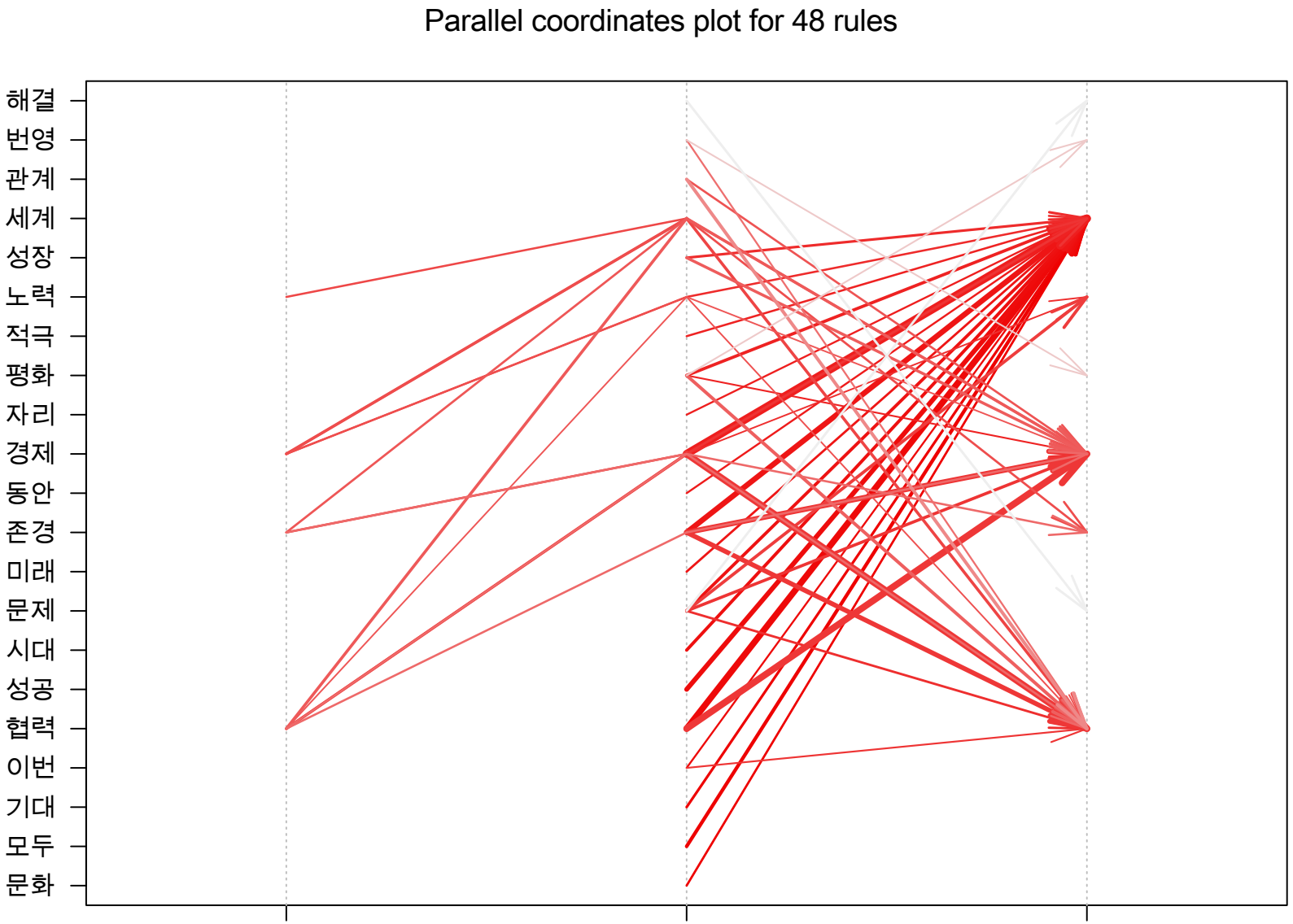


Association Rules 시각화 - graph with control

```
> plot(rules, method = "graph", control = list(type = "itemsets
```



Association Rules 시각화 - paracoord



Clustering / Topic Analytics

Clustering / Topic Analytics 들어가기 : 학습 목표

- Hierarchy Clustering
  - Hierarchy Clustering 이해하기
- Kmeans Clustering
  - Kmeans Clustering 이해하기
- Topic Analytics
  - LDA(Latent Dirichlet Allocation) 이해하기

Hierarchy Clustering : Analytics Process

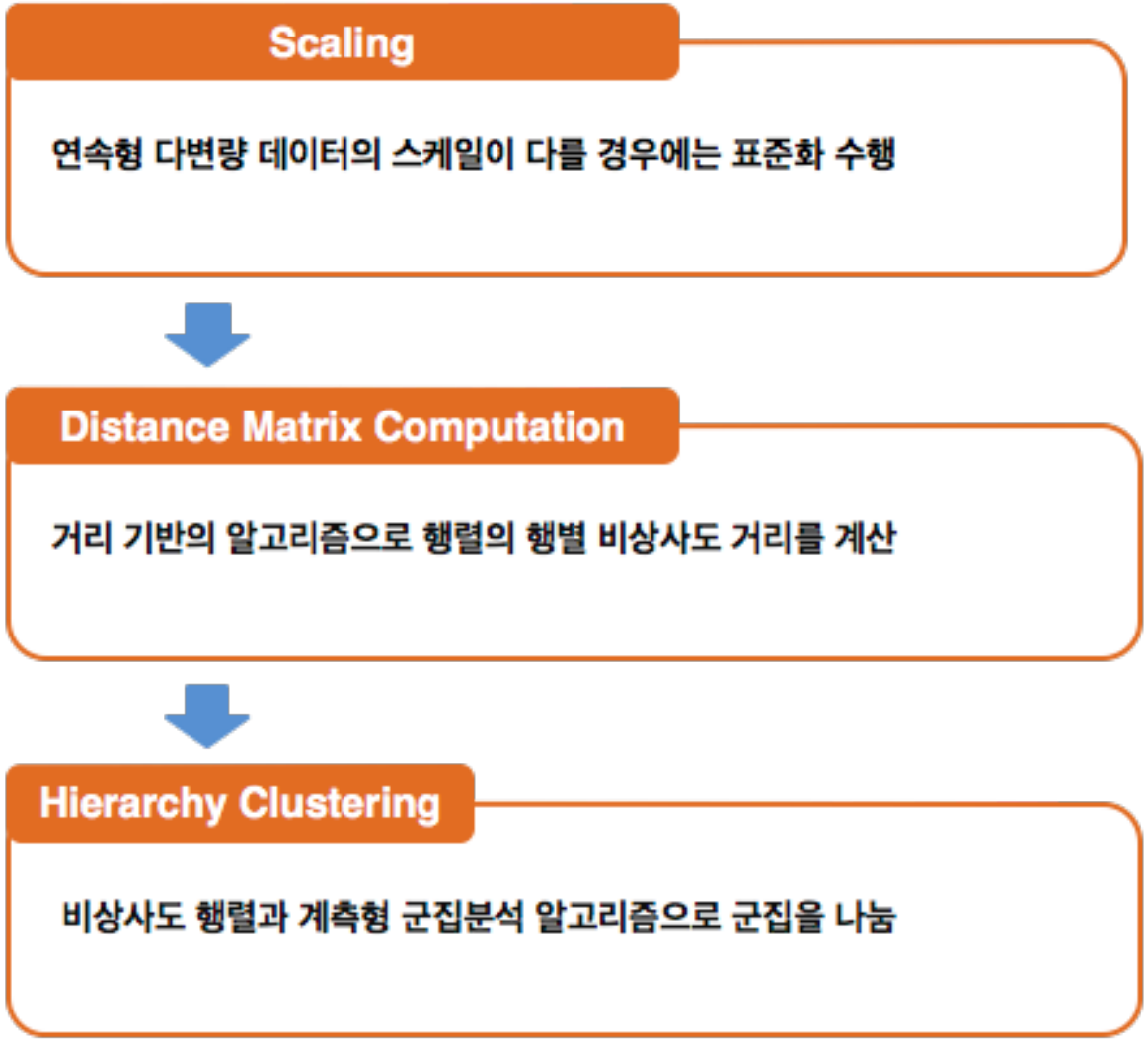


그림: Hierarchy Clustering : Analytics Process

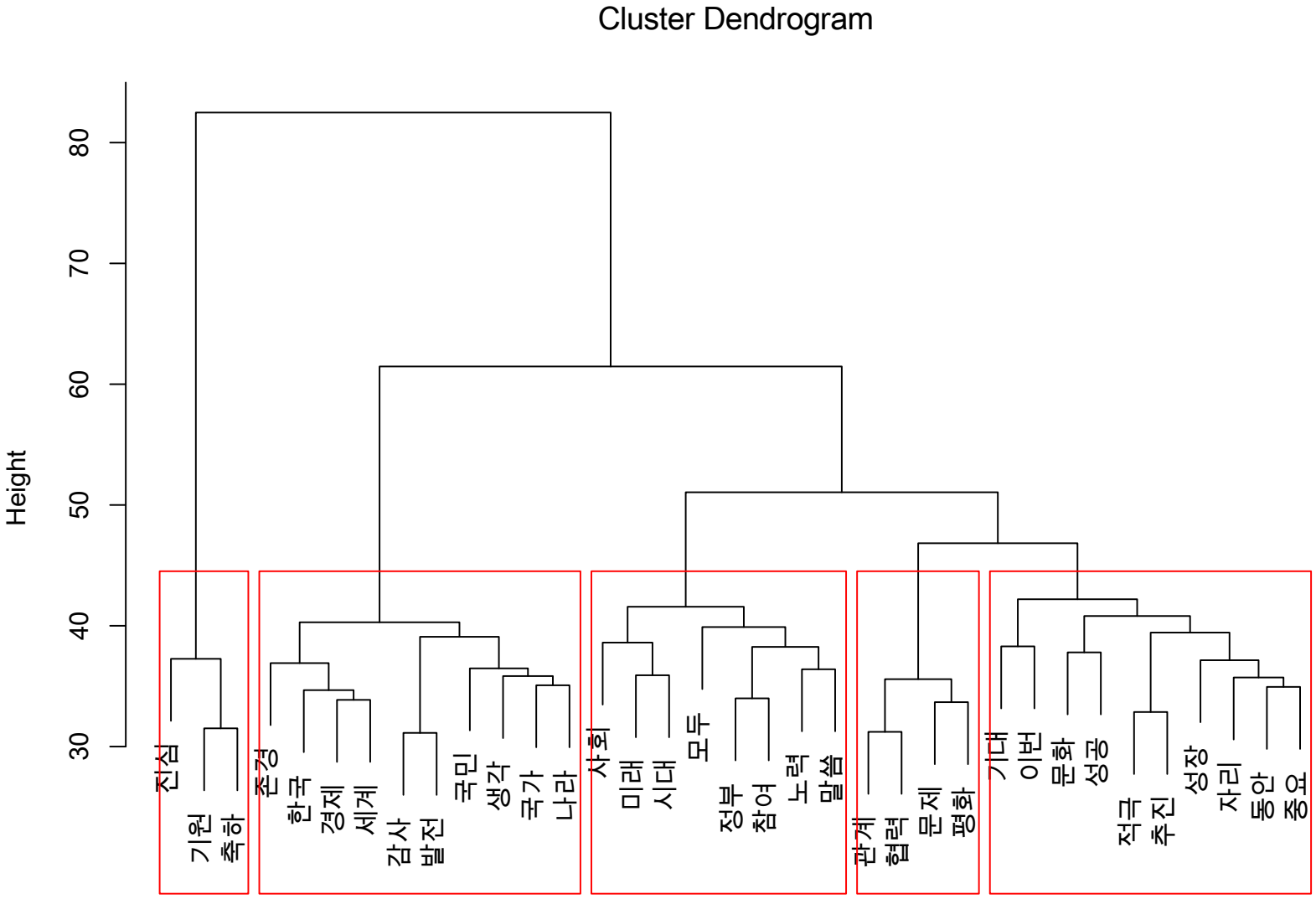
전처리 - sparse terms 제거하기

```
> compact_bin <- removeSparseTerms(tdmbtf_noh, sparse = 0.985)
> compact_bin2 <- as.matrix(compact_bin)
> dim(compact_bin2)

[1] 1903  780

> rowTotals <- apply(compact_bin2 , 1, sum)
> thres <- 300
> compact_bin3   <- compact_bin2[rowTotals > thres, ]
> dim(compact_bin3)

[1]  35 780
```

hirericy clustering

```
> distMatrix <- dist(scale(compact_bin3))
> fit <- hclust(distMatrix, method = "ward.D")
> plot(fit)
> rect.hclust(fit, k = 5)
```

hirericy clustering - result



K-Means Clustering : Analytics Process

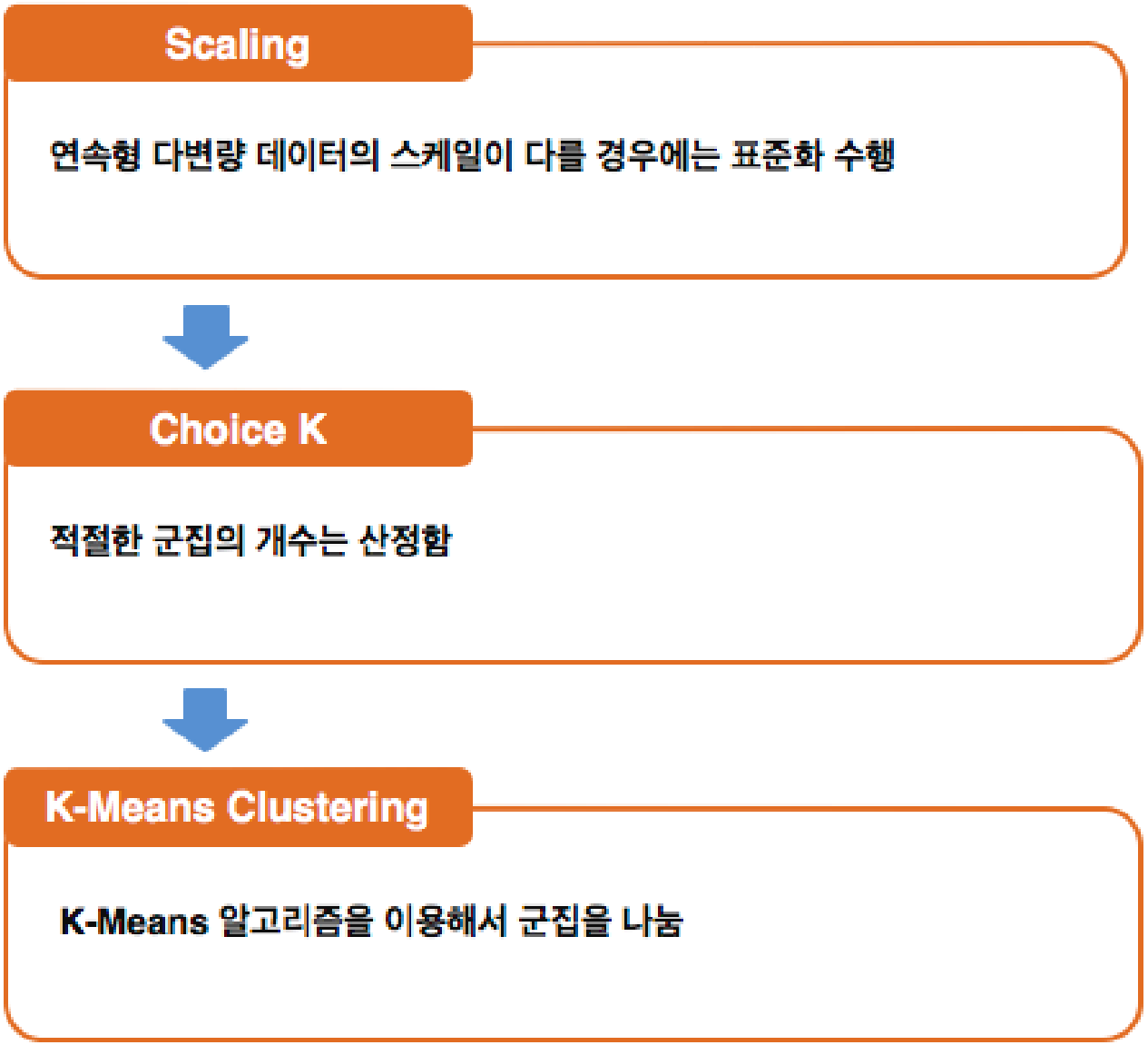
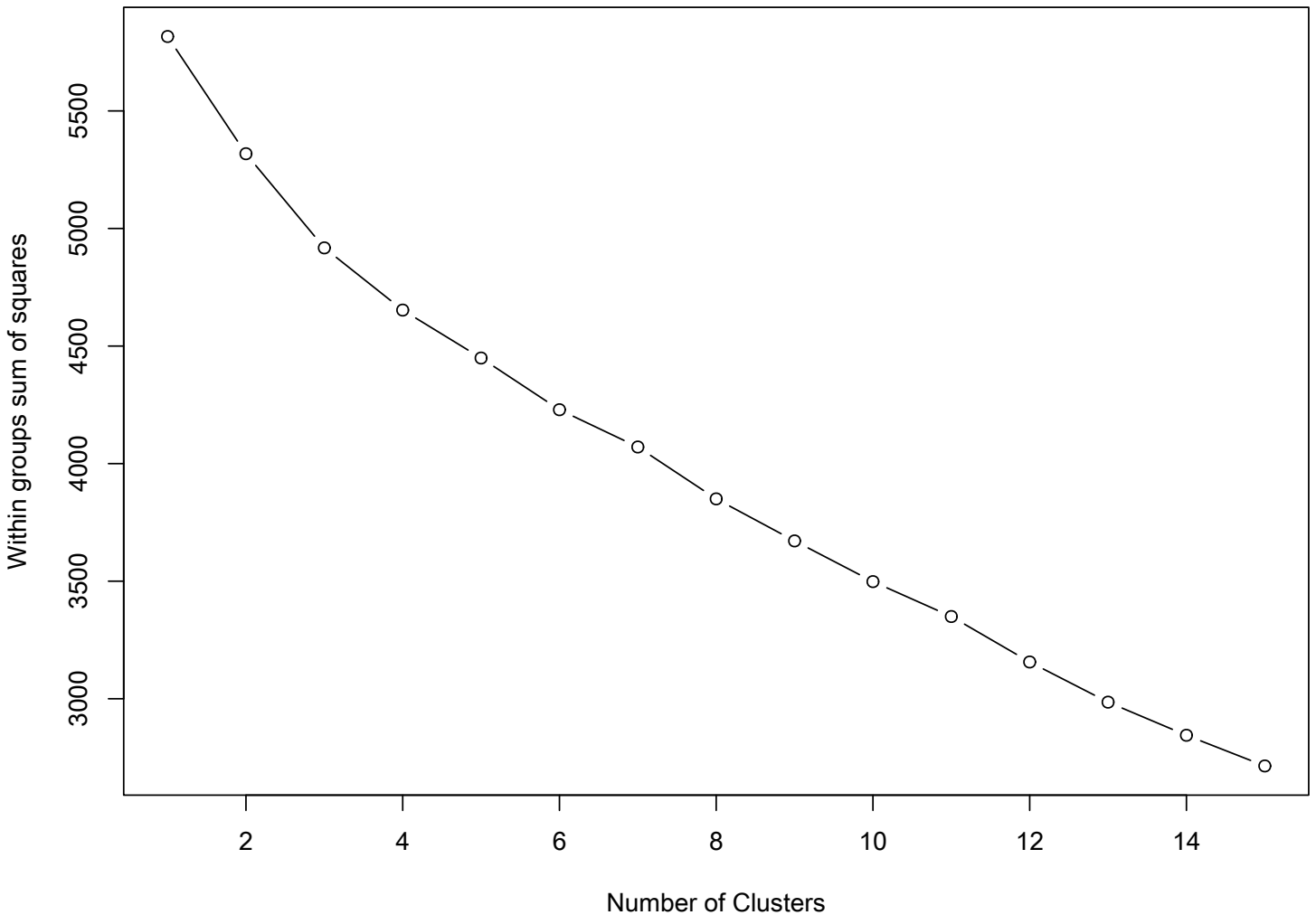


그림: K-Means Clustering : Analytics Process

군집 개수 구하기 : WSS 기반 선정

```
> compact_bin4 <- compact_bin3[, apply(compact_bin3,
+                                      2, sum) > 1]
> dim(compact_bin4)

[1]  35 778

> N <- 15
> wss <- (nrow(compact_bin4) - 1) * sum(apply(compact_bin4,
+                                             2, var))
> for (i in 2:N) wss[i] <- sum(kmeans(compact_bin4,
+                                     centers = i)$withinss)
```

군집 개수 구하기 : WSS 기반 선정 - 결과



군집 개수 구하기 : NbClust 패키지 기반 선정

```
> library(NbClust)
>
> mthds <- c("ward.D", "ward.D2", "single", "complete",
+            "average", "mcquitty", "median", "centroid",
+            "kmeans")
>
> best.nc <- lapply(mthds,
+         function(x) NbClust(compact_bin4, min.nc = 2,
+                             max.nc = 15, method = x,
+                             index = "duda")$Best.nc)
```

군집 개수 구하기 : NbClust 패키지 기반 선정 결과

```
> table(sapply(best.nc, "[", 1))


 2  3  8 10
 4  2  1  2

> best.nc[[9]]

Number_clusters     Value_Index
         2.0000         12.5248
```

K-Means Clustering

```
> k <- 3
> set.seed(123)
> kmc <- kmeans(compact_bin4, k)
>
> ## 군집으로 묶인 Term의 개수
> table(kmc$cluster)
```

```
 1  2  3
 3 11 21
```

K-Means Clustering

```
> which(kmc$cluster == 1)

기원 진심 축하
   7   29   32

> which(kmc$cluster == 2)

감사 경제 국가 국민 나라 발전 생각 세계 존경 한국 협력
   1    2    4    5    8   16   18   21   27   34   35

> which(kmc$cluster == 3)

관계 기대 노력 동안 말씀 모두 문제 문화 미래 사회 성공 성장 시
대 이번 자리
  3   6   9  10  11  12  13  14  15  17  19  20  22  23  24
```

부여

```
> ## 개별 군집 형성에 영향을 준 Document 5건 조회
> for (i in seq(k)) {
+   cat(paste("cluster ", i, ": ", sep = ""))
+   s <- sort(kmc$centers[i, ], decreasing = TRUE)
+   cat(names(s)[1:5], "\n")
+ }

cluster 1: 9 12 13 14 19
cluster 2: 7 11 15 17 37
cluster 3: 280 695 707 745 767
```

부여

```
> ## 1번째 clustering에 영향을 준 문서
> substr(fnames[fnames$docid %in% c("DOC_0280"), "doc"], 1, 300
```

[1] " 존경하는 국민 여러분, 그리고 내외신 기자 여러분, 새해 복 많이 받으십시오. 지난 한해, 좋은 일 그리고 궂은 일이 참 많았지만, 내내 경제 걱정만 한 기억밖에는 없습니다. 새해에도 여러 소망이 있겠지만 모두가 간절히 바라는 대로 우리 경제가 좀 좋아졌으면 좋겠습니다. 다행히 연초부터 많은 대기업들이 투자를 늘리겠다고 적극 나서고 있습니다. 정부도 기업들이 의욕을 가지고 투자를 확대할 수 있도록 기업하기 좋은 환경을 만드는 데 더욱 힘써나가겠습니다. 정부 재정도 상반기에 집중 투입해서 투자와 소비를 활성화해 나가"

부여

```
> ## 2번째 clustering에 영향을 준 문서
> substr(fnames[fnames$docid %in% c("DOC_0011"), "doc"], 1, 300
```

[1] "존경하는 도널드 에반스 美 상무장관, 얀 피트 헤인 도너 네덜란드 법무장관, 바오지르 피레스 브라질 감사원장, 그리고 각국의 수석대표와 내외귀빈 여러분,안녕하십니까, 이렇게 뜻깊은 자리에 초청해주신 데 대해 감사드립니다. 여러분과 자리를 함께 하게 된 것을 매우 기쁘고 영광스럽게 생각합니다. 아울러 '제3차 반부패 세계포럼'이 대한민국 서울에서 개최되었다는 사실에 무한한 자긍심과 긍지를 느낍니다. 성공적인 개최를 축하드리며, 이를 위해 애써주신 우리 법무부와 조직위원회 회원국, 그리고 국제기구 관계자 여러분께 감사의 말씀을 드립니다"

Topic Analytics : Analytics Process

그림: Topic Analytics : Analytics Process

전처리 - Sparse term 제거

```
> tdm <- removeSparseTerms(tdmtf_noh, sparse = 0.95)
> m2 <- as.matrix(tdm)
> fnames %>%
+   filter(president == "노무현") %>%
+   dplyr::select(docid) %>% .[, 1] -> colnames(m2)
> dim(m2)

[1] 708 780

> ## Find the sum of words in each Document
> colTotals <- apply(m2 , 2, sum)
> m3 <- m2[, colTotals> 0] # Remove all docs without words
> dim(m3)
```

Topic 개수 구하기 : topicmodels 패키지의 이용

```
> library(topicmodels)
> models <- lapply(2:20, function(x) LDA(t(m3), k = x,
+                                        control = list(seed = 123)))
```

```
> sapply(models, logLik)

 [1] -1038518.5 -1025660.7 -1020043.3 -1014750.9 -1009351.0 -10
 [7] -1004751.6  -999825.3  -998020.5  -996506.5  -994700.7  -99
[13]  -991509.5  -990854.4  -988717.9  -987519.9  -986319.1  -98
[19]  -984699.9
```

Topic 개수 구하기 : topicmodels 패키지의 이용

```
> ## 20개 선정 - Max Log Likelihood
> which(sapply(models, logLik) == max(sapply(models, logLik)))

[1] 19

> ## 20개 선정 - Min alpha
> alpha <- sapply(models, slot, "alpha")
> which(alpha == min(alpha))

[1] 19
```

Topic Modeling (모델 튜닝은 없음)

```
> ## find 20 topics
> k <- 20
>
> lda <- LDA(t(m3), k = k, control = list(seed = 123))
> lda

A LDA_VEM topic model with 20 topics.

> is(lda)

[1] "LDA_VEM"    "LDA"        "VEM"        "TopicModel"
```

Topic Terms 살펴보기

```
> term <- terms(lda, 10) ## first 10 terms of every topic
> term[, 1:5]
```

	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
[1,]	"혁신"	"남북"	"정부"	"국민"	"언론"
[2,]	"정부"	"문제"	"경제"	"노력"	"사람"
[3,]	"성공"	"평화"	"사회"	"감사"	"민주주의"
[4,]	"발전"	"정상"	"참여"	"건강"	"권력"
[5,]	"농업"	"북한"	"개혁"	"생활"	"생각"
[6,]	"환경"	"협력"	"국민"	"모두"	"정부"
[7,]	"축하"	"회담"	"정책"	"마음"	"정치"
[8,]	"경쟁력"	"한반도"	"혁신"	"축하"	"진보"
[9,]	"지원"	"해결"	"전략"	"안전"	"정책"
[10,]	"노력"	"합의"	"과제"	"생각"	"참여"

Topic Terms 살펴보기

```
> term[, 6:10]


      Topic 6 Topic 7  Topic 8 Topic 9    Topic 10
 [1,] "경제"  "평화"   "축하"  "시장"      "문제"
 [2,] "기업"  "동북아" "문화"  "사회"      "정책"
 [3,] "한국"  "협력"   "발전"  "민주주의" "교육"
 [4,] "투자"  "한국"   "세계"  "국가"      "사람"
 [5,] "협력"  "번영"   "감사"  "공정"      "경제"
 [6,] "시장"  "세계"   "진심"  "자유"      "기업"
 [7,] "세계"  "경제"   "기원"  "시민"      "정부"
 [8,] "무역"  "중국"   "시민"  "생각"      "사회"
 [9,] "산업"  "지역"   "시대"  "인권"      "생각"
[10,] "수출"  "문제"   "산업"  "투명"      "대통령"
```

Topic Terms 살펴보기

```
> term[, 11:15]


      Topic 11 Topic 12 Topic 13 Topic 14 Topic 15
 [1,] "기술"   "각하"   "생각"   "역사"   "국민"
 [2,] "과학"   "협력"   "사람"   "일본"   "정부"
 [3,] "한국"   "양국"   "문제"   "나라"   "국회"
 [4,] "세계"   "나라"   "대통령" "국민"   "지원"
 [5,] "개발"   "발전"   "얘기"   "동포"   "경제"
 [6,] "산업"   "국민"   "말씀"   "생각"   "추진"
 [7,] "경제"   "감사"   "국민"   "국가"   "확대"
 [8,] "기업"   "한국"   "한국"   "전쟁"   "국가"
 [9,] "연구"   "대통령" "중요"   "한국"   "의원"
[10,] "분야"   "관계"   "변화"   "세계"   "산업"
```

Topic Terms 살펴보기

```
> term[, 16:20]


      Topic 16   Topic 17 Topic 18 Topic 19 Topic 20
 [1,] "국민"     "국민"   "평화"   "지방"   "정치"
 [2,] "역사"     "역사"   "장병"   "지역"   "대통령"
 [3,] "민주주의" "시대"   "안보"   "도시"   "국민"
 [4,] "존경"     "문제"   "국민"   "발전"   "생각"
 [5,] "나라"     "나라"   "동맹"   "수도"   "선거"
 [6,] "희생"     "정치"   "국군"   "균형"   "문제"
 [7,] "감사"     "갈등"   "국방"   "정책"   "국회"
 [8,] "평화"     "극복"   "미군"   "정부"   "정당"
 [9,] "민주"     "사회"   "이라크" "행정"   "정부"
[10,] "운동"     "문화"   "주한"   "기업"   "책임"
```

Topic 내용 살펴보기

```
> noh.topic <- data.frame(topic = rep(1:k, times = NROW(m3)),
+                         term = rep(lda@terms, each = k),
+                          beta = as.vector(lda@beta))
>
> head(noh.topic, n = 3)

  topic term      beta
1     1 가능 -4.875796
2     2 가능 -5.795218
3     3 가능 -6.624963
```

Topic 내용 살펴보기

```
> noh.topic %>% ## "경제"라는 term이 각 topic에서 생성될 확률
+   filter(term == "경제") %>%
+   head

  topic term      beta
1     1 경제 -8.112521
2     2 경제 -4.137786
3     3 경제 -3.315357
4     4 경제 -6.207816
5     5 경제 -5.303881
6     6 경제 -2.541255
```

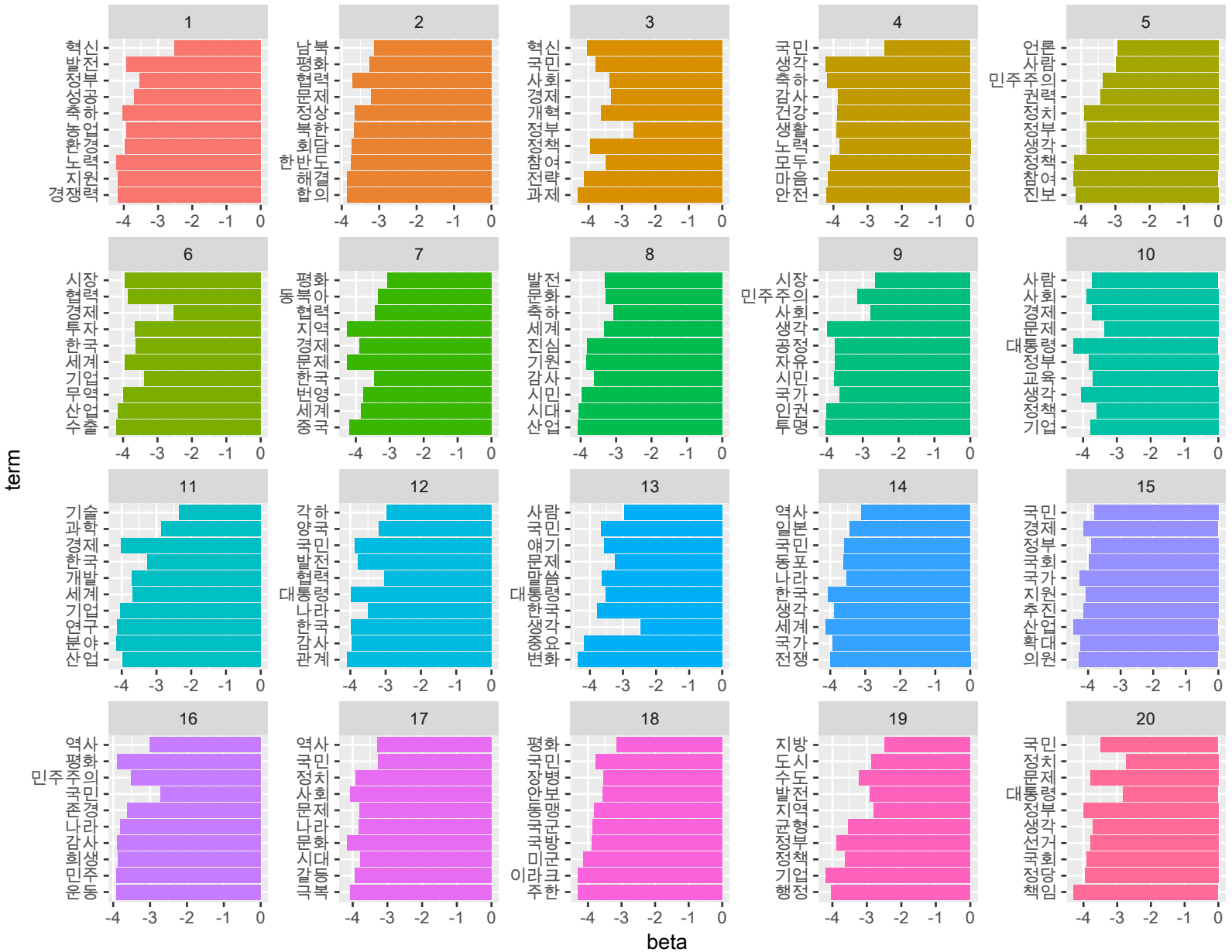
Topic별 beta를 내림차순으로 정렬하기

```
> top_terms <- noh.topic %>%
+   group_by(topic) %>%
+   top_n(10, beta) %>%
+   ungroup() %>%
+   arrange(topic, -beta)
> top_terms[1:3, ]

# A tibble: 3 x 3
  topic   term      beta
  <int> <fctr>     <dbl>
1     1   혁신 -2.515211
2     1   정부 -3.551644
3     1   성공 -3.689852
```

Top beta Term 시각화

```
> library(ggplot2)
>
> top_terms %>%
+   mutate(term = reorder(term, beta)) %>%
+   ggplot(aes(term, beta, fill = factor(topic))) +
+   geom_col(show.legend = FALSE) +
+   facet_wrap(~ topic, scales = "free") +
+   coord_flip()
```

Top beta Term 시각화



문서에서 개별 Topic이 차지하는 비중

```
> noh.document <-
+   data.frame(document = rep(lda@documents, times = k),
+              topic = rep(1:k, each = NROW(lda@gamma)),
+              gamma = as.vector(lda@gamma))
> noh.document %>%
+   filter(topic == "11") %>%
+   top_n(3, gamma) %>%
+   ungroup() %>%
+   arrange(topic, -gamma)

  document topic     gamma
1 DOC_0667    11 0.8210351
2 DOC_0098    11 0.8065945
3 DOC_0681    11 0.7769041
```

문서에서 개별 Topic이 차지하는 비중

```
> fnames %>%
+   filter(docid == "DOC_0667") %>%
+   dplyr::select(doc) %>%
+   .[, 1] %>% substr(start = 1, stop = 500)
```

[1] "안녕하십니까,오늘 우리는 '우주기술 자립'을 이루기 위한 큰 걸음을 내딛습니다. 우주로 향하는 우리의 꿈을 실현할, '우주센터'의 착공을 온 국민과 함께 기쁘게 생각합니다. 우리는 지난 90년대 초부터 우주 개척을 위한 발걸음을 한발 한발 내디뎌 왔습니다. '우리별 위성'과 '아리랑 위성', 그리고 '과학로켓'의 개발이 바로 그것입니다. 그리고 오늘 국가 우주개발의 전초기지가 될 우주센터가 역사적인 첫 삽을 뜹니다. 이제 우주시대는 먼 나라 얘기도, 머나먼 꿈도 아닙니다. 우리는 2015년까지 스무 기의 위성을 자력으로 개발하고, 세계 10위권의 우

Taxonomy - topic 7(무역경제)

```
> ## topic 7(무역경제) topic이 70% 이상 포함된 문서의 조회
> noh.document %>%
+   filter(topic == "6") %>%
+   filter(gamma >= 0.7) %>%
+   arrange(desc(gamma))

  document topic     gamma
1 DOC_0592     6 0.7474647
2 DOC_0507     6 0.7153146
```

Taxonomy - topic 7(무역경제)

```
> fnames %>%
+   filter(docid == "DOC_0592") %>%
+   dplyr::select(doc) %>%
+   .[, 1] %>% substr(start = 1, stop = 400)
```

```
[1] "  존경하는 국민 여러분,  김재철 회장을 비롯한 무역인과 근로자 여러분,  마흔두 번째 무역의 날을 진심으로 축하드립니다.  오늘은 여러분의 잔칫날입니다. 그런데 제가 더 기분이 좋습니다. 저뿐만 아니라 우리 국민 모두가 함께 축하하고 기뻐할 것입니다. 특히 오늘 수상하신 분들께 각별한 축하의 인사를 드립니다.  제가 여러분께 드리고 싶은 첫마디는 감사하다는 말씀입니다. 지구촌 구석구석을 밤낮없이 뛰고 있는 기업인 여러분, 그리고 묵묵히 땀 흘려 오신 근로자 여러분, 정말 수고 많으셨습니다. 여러분 모두에게 뜨거운 격려의 박수를 보냅
```

Taxonomy - Topic 19(국토균형발전)

```
> ## Topic 19(국토균형발전) topic이 70% 이상 포함된 문서의 조회
> noh.document %>%
+   filter(topic == "19") %>%
+   filter(gamma >= 0.7) %>%
+   arrange(desc(gamma))

  document topic     gamma
1 DOC_0718    19 0.7486320
2 DOC_0541    19 0.7474319
```

Taxonomy - Topic 19(국토균형발전)

```
> fnames %>%
+   filter(docid == "DOC_0718") %>%
+   dplyr::select(doc) %>%
+   .[, 1] %>% substr(start = 1, stop = 400)
```

[1] "존경하는 국민 여러분, 충남도민과 내외귀빈 여러분,국가균형발전의 새 역사가 열리고 있습니다. 행정중심복합도시의 기공을 온 국민과 함께 축하드립니다. 영상물을 보니 벌써부터 가슴이 설렙니다. 개방적이고 시민친화적인 정부청사, 다양한 동식물이 서식하는 금강변과 전월산, 그리고 그곳에서 문화와 여가를 즐기는 시민들의 모습이 눈앞에 선하게 그려집니다.국민 여러분이 지어주신 '세종'이라는 이름도 아주 훌륭합니다. 행복도시에 딱 맞는 이름이라고 생각합니다. 창의와 혁신으로 우리 역사의 융성기를 이뤄내신 세종대왕의 위상에 걸맞은 도시가 될 것

References

Text Mining with R

`https://tidytextmining.com`

Julia Silge and David Robinson, 2017-05-17

THE END