R을 이용한 텍스트마이닝

Jinseog Kim
Dongguk University
jskim1986@gmail.com

2018-03-14

Contents

1 텍스트마이닝

1.1 텍스트마이닝 (text mining)이란

- 웹페이지, 블로그, 전자저널, 이메일 등 전자문서로 된 텍스트자료로 부터 유용한 정보를 추출하여 분석하기 위한 도구
- 텍스트 데이터로 부터 새로운 고급 정보를 이끌어 내는 과정 (Wikipedia, 2011b).
- 텍스트마이닝은 데이터마이닝, 자연어처리, 정보검색등의 이론 및 실무까지 다양한 분야가 융합되어 있는 영역

1.2 텍스트마이닝의 주제

- 유사 단어들(또는 문서들)간의 군집 분석(커뮤니티 분석)
- 연관 단어의 추출 또는 단어 네트워크 분석
- 주요 키워드의 추출
- 토픽모델
- 토픽 트렌드 분석, 이상치 분석 (anormality)
- 분류(classification): 텍스트문서를 미리 정해놓은 범주에 할당하는 것으로, 예를 들면, 이메일의 내용이 스팸인지 아닌지를 구분하는 것

1.3 텍스트 마이닝의 절차

1. 분석 문서자료의 준비
2. 코퍼스의 구성
3. 전처리: 전 과정의 70~80% 정도 차지
4. 분석

1.4 코퍼스란

- 코퍼스(corpus; 말뭉치): 언어학에서 구조를 이루고 있는 텍스트 집합으로 통계 분석 및 가설 검증, 언어 규칙의 검사등에 사용됨, 코퍼스는 보통 전자문서들로 구성됨
  - 예) 신문사의 기사, 세익스피어의 소설, …
- 코퍼스의 저장 형식
  - volatile corpus : 메모리에 저장되는 코퍼스
  - permanent corpus: DB 혹은 디렉토리에 저장
  - distributed corpus: 분산저장장치(예, 하둡파일시스템)을 이용
- 전자문서의 형식
  - text, PDF, Microsoft Word, XML, HTML등 형식의 문서

1.5 텍스트 마이닝을 위한 R패키지

```
library(tm) # text mining 기본 툴
library(KoNLP) # 한글처리
```

2 데이터 분석 1 - 진로 계획서

2.1 예제 문서

- 응용통계2학년 학생들의 개인 진로 계획서
- Example data의 plans.zip

```
dir("data/plan")[1:10]
```

```
##  [1] "1.txt"  "10.txt" "11.txt" "12.txt" "13.txt" "14.txt" "15.txt"
##  [8] "16.txt" "17.txt" "18.txt"
```

2.2 디렉토리의 텍스트문서들을 tm 코퍼스로 가져오기

- tm 패키지의 아래 함수를 이용
  - VCorpus : Volatile(메모리에 저장되는) 코퍼스(생성 함수)
  - DirSource(): 디렉토리(data/plan)에 있는 확장명이 txt인 파일들을 코퍼스를 생성하기 위한 소스로 지정

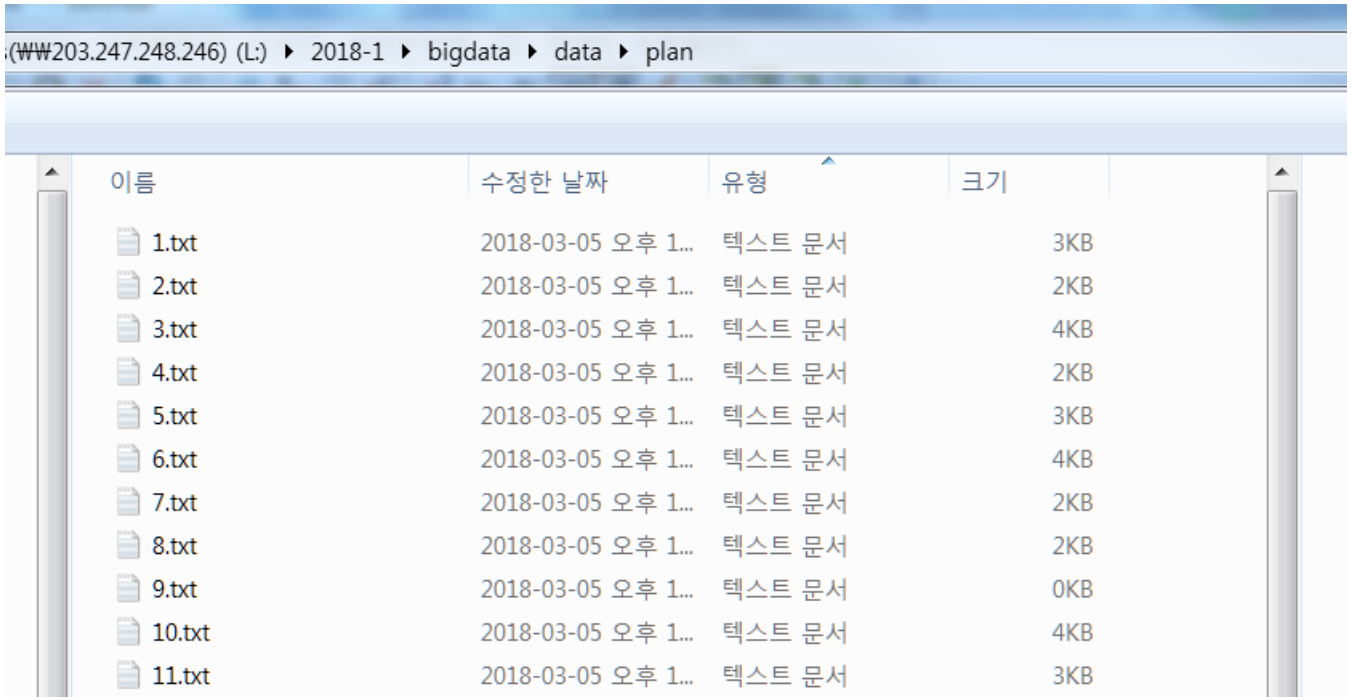


Figure 1: 디렉토리 구조

```
library(tm)
plans <- VCorpus(DirSource("data/plan", pattern="txt"))
plans
```

```
## <<VCorpus>>
## Metadata:  corpus specific: 0, document level (indexed): 0
## Content:  documents: 32
```

- tm 코퍼스의 구조 확인
  - 문서별로 content(문서의 내용), meta(메타정보)요소를 가지는 리스트 구조(PlainTextDocument)

```
str(plans[[1]])
```

```
## List of 2
##  $ content: chr [1:3] "이번 학기에는 아직 배운 것이 없기 때문에, 학점 4점 이상으로 만드
##  $ meta   :List of 7
##   ..$ author       : chr(0)
##   ..$ datetimestamp: POSIXlt[1:1], format: "2018-03-13 17:19:53"
##   ..$ description  : chr(0)
##   ..$ heading      : chr(0)
##   ..$ id           : chr "1.txt"
##   ..$ language     : chr "en"
```

```
##   ..$ origin       : chr(0)
##   ..- attr(*, "class")= chr "TextDocumentMeta"
##  - attr(*, "class")= chr [1:2] "PlainTextDocument" "TextDocument"
```

 문서내용 추출 함수: content()

```
content(plans[[1]]) # lapply(plans, content)
```

```
## [1] "이번 학기에는 아직 배운 것이 없기 때문에, 학점 4점 이상으로 만드는 것이 최고목표입
## [2] "그리하여, 3학년에는 평균학점 4점과 토익 850점, 토익 스피킹 레벨6, 컴퓨터 활용능력 
## [3] "4학년 1학기 때에도 마찬가지로 학점 4점을 만들어 볼 것이며, 학과에 기초지식을 바탕으
```

 문서메타정보 추출/추가 함수: meta()

```
meta(plans[[1]])
```

```
##   author       : character(0)
##   datetimestamp: 2018-03-13 17:19:53
##   description  : character(0)
##   heading      : character(0)
##   id           : 1.txt
##   language     : en
##   origin       : character(0)
```

```
meta(plans[[1]], tag="etc") <- "추가 정보입니다"
meta(plans[[1]])
```

```
##   author       : character(0)
##   datetimestamp: 2018-03-13 17:19:53
##   description  : character(0)
##   heading      : character(0)
##   id           : 1.txt
##   language     : en
##   origin       : character(0)
##   etc          : 추가 정보입니다
```

2.3 Text 전처리 (pre-processing)

 방법

1. stringr::str_...함수, 또는
2. tm::tm_map함수를 이용

1. 여러 줄로 구성된 content를 한 줄로 변환

```
for(i in seq_along(plans)){
  plans[[i]]$content <- paste(plans[[i]]$content, collapse=" ")
}
plans[[1]]$content
```

```
## [1] "이번 학기에는 아직 배운 것이 없기 때문에, 학점 4점 이상으로 만드는 것이 최고목표입니다
```

2. 문장부호 제거: removePunctuation

```
plans <- tm_map(plans, removePunctuation)
plans[[1]]$content
```

```
## [1] "이번 학기에는 아직 배운 것이 없기 때문에 학점 4점 이상으로 만드는 것이 최고목표입니다
```

3. 숫자 제거: 숫자가 불필요한 경우

```
plans = tm_map(plans, removeNumbers)
plans[[1]]$content
```

4. 공백문자 제거

```
plans = tm_map(plans, stripWhitespace)
plans[[1]]$content
```

```
## [1] "이번 학기에는 아직 배운 것이 없기 때문에 학점 4점 이상으로 만드는 것이 최고목표입니다
```

5. 불용어 제거하기

```
# 불용어 추가 등록 가능
plans <- tm_map(plans,removeWords,stopwords('english')) # 불용어 제거하기 (전치사 , 관사 등)
plans[[1]]$content
```

```
## [1] "이번 학기에는 아직 배운 것이 없기 때문에 학점 4점 이상으로 만드는 것이 최고목표입니다
```

6. 어근 추출(영문) - 한글은 KoNLP를 이용하여 별도로 처리

```
library(SnowballC)
plans <- tm_map(plans,stemDocument)
plans[[1]]$content
```

```
## [1] "이번 학기에는 아직 배운 것이 없기 때문에 학점 4점 이상으로 만드는 것이 최고목표입니다
```

2.4 기타 한글처리

▯ 불필요 단어의 삭제

```
kst <- c("궁극적", "그다음", "기본적", "나머지","중학교","고등학교", "겠습니", "능수능란", "마
         "마찬가지", "싶습니", "아이들", "안녕하십니까", "같습니", "있습니", "좋겠습니", "부터
         "것입니다", "하나씩", "생각", "그렇게", "학기", "제가", "좋은", "열심히", "무엇",
         "재밌", "입니다", "있을", "다른", "습니다", "곳에")
rm_words <- paste(kst, collapse = '|')
for(i in seq_along(plans)){
  plans[[i]]$content <- stringr::str_replace_all(plans[[i]]$content, rm_words, " ")
}
```

▯ 단어 띄어쓰기

```
library(KoNLP)
library(NIADic)
useNIADic()
```

```
## Backup was just finished!
## 983012 words dictionary was built.
```

```
x <- "성적이 제일 좋아 동국 대학교 경주 캠퍼스에 응용 통계학과에 입학을 하게 되었습니다."
extractNoun(x)
```

```
## [1] "성적"     "제"       "동국"     "대학교"   "경주"     "캠퍼스"
## [7] "응용"     "통계학과" "입학"
```

 띄어쓰기 오류 확인 후 교정

  코퍼스 전체 문서에 적용 후 단어들을 다시 문서 단위로 결합

```
for(i in seq_along(plans)){
  plans[[i]]$content <- gsub("동국 대학교", "동국대학교", plans[[i]]$content);
  plans[[i]]$content <- gsub("경주 캠퍼스", "경주캠퍼스", plans[[i]]$content);
  plans[[i]]$content <- gsub("응용 통계학과", "응용통계학과", plans[[i]]$content);
}
```

 사전에 새로운 단어를 추가 / 아래의 어근 추출과 함께 반복작업이 필요함

```
new_term <- c("응용통계학과", "경주캠퍼스", "빅데이터", "동국대학교")
new_dic <- data.frame(new_term , "ncn")
buildDictionary(ext_dic=c('sejong', 'woorimalsam', 'insighter'), user_dic=new_dic)
```

```
## 1070052 words dictionary was built.
```

 어근추출(명사추출)

1. extractNoun: 명사추출
2. paste: 추출된 단어들을 재결합

```r
#extractNoun(plans[[1]]$content)
for(i in seq_along(plans)){
  nouns <- extractNoun(plans[[i]]$content)
  nouns <- nouns[nchar(nouns) > 2]
  plans[[i]]$content <- paste(nouns, collapse=" ")
}
```

2.5 분석 방법

- 워드클라우드
- 군집분석
  - 키워드/문서(학생)

2.6 코퍼스에서 TermDocumentMatrix의 생성

- TermDocumentMatrix의 생성
  - 여기서, control의 wordLengths는 문자의 길이에 대한 옵션(최소, 최대)
  - tokenize="scan"은 공백으로 문서의 단어들을 분리시킴

```
plan_tdm <- TermDocumentMatrix(plans, control=list(tokenize="scan", wordLengths=c(2, 7)))
#plan_tdm_1 <- TermDocumentMatrix(plans, control=list(wordLengths=c(2, 7)))
inspect(plan_tdm)
```

```
## <<TermDocumentMatrix (terms: 369, documents: 32)>>
## Non-/sparse entries: 598/11210
## Sparsity           : 95%
## Maximal term length: 7
```

```
## Weighting          : term frequency (tf)
## Sample             :
##               Docs
## Terms          10.txt 11.txt 20.txt 3.txt 32.txt 4.txt 5.txt 6.txt 7.txt
##   교수님            0      0      0     2      2     0     1     0     0
##   데이터            0      7      0     3      9     7     0     0     4
##   빅데이터          0      0      0     1      0     1     4     0     5
##   스포츠            0     17      0     0      0     0     0     0     0
##   응용통계학과      1      3      2     1      0     4     1     1     1
##   자격증            7      0      8     5      2     1     6     0     0
##   전문가            0      0      0     0      9     2     1     0     1
##   컴퓨터            3      0      1     6      0     0     2     0     0
##   통계학            1      1      2     2      3     0     2     0     2
##   통계학과          0      0      1     1      0     0     2     1     4
##               Docs
## Terms          9.txt
##   교수님            0
##   데이터            0
##   빅데이터          5
##   스포츠            0
##   응용통계학과      0
##   자격증            2
```

```
##   전문가          1
##   컴퓨터          0
##   통계학          1
##   통계학과        0
```

- TermDocumentMatrix관련함수
  - 문서의 수, 단어의 수 추출

```
nTerms(plan_tdm)
```

```
## [1] 369
```

```
nDocs(plan_tdm)
```

```
## [1] 32
```

  - 단어의 길이

```
Terms(plan_tdm)[nchar(Terms(plan_tdm)) == 5]
```

```
##  [1] "1차적으로"  "ispss"      "toefl"      "toeic"      "경주캠퍼스"
##  [6] "경찰공무원" "그와중에도" "늦은나이로" "도와주실꺼" "동국대학교"
## [11] "되었다내가" "보험계리사" "분석보고서" "빅데이터였" "소프트웨어"
```

```
## [16] "수치데이터" "시험치는게" "아르바이트" "알아봐야겠" "인문학교수"
## [21] "자기소개서" "잘부탁드리" "전자계산기" "정보시스템" "제정됫다고"
## [26] "조사해보았" "조언을듣고" "지켜보았었" "질문을하여" "칼럼니스트"
## [31] "통계라는걸" "포털사이트" "프로그래머" "프로그래밍" "하는것보단"
## [36] "흥미가잇엇"
```

 빈번 단어의 추출

```
findFreqTerms(plan_tdm, lowfreq = 5, highfreq = Inf)
```

```
##  [1] "16학번"       "겨울방학"     "경주캠퍼스"   "공무원"
##  [5] "교수님"       "대학교"       "대학원"       "데이터"
##  [9] "동국대학교"   "마케팅"       "보험계리사"   "빅데이터"
## [13] "사람들"       "사이버"       "사회조사"     "세무사"
## [17] "수업시간"     "스포츠"       "아르바이트"   "어려움"
## [21] "여름방학"     "응용통계학과" "인터넷"       "자격증"
## [25] "전문가"       "컴퓨터"       "통계직"       "통계학"
## [29] "통계학과"     "파일럿"       "프로그래밍"   "프로그램"
```

2.7 단어간 연관성

 findAssocs함수

```r
findAssocs(plan_tdm, c("취업", "학업"), c(0.75))
```

```
## $취업
## numeric(0)
##
## $학업
## numeric(0)
```

2.8 워드클라우드 분석

- Term-Document 희소단어의 제거
  - 희소단어

$$sparsity(t) = \frac{\text{단어 t가 포함되지 않은 문서수}}{\text{전체 문서의 수}}$$

  - removeSparseTerms함수이용
    * sparse=0.90: 10%이상의 문서에서 출현하는 단어들만 이용하여 TDM구성

```r
plan_tdm <- removeSparseTerms(plan_tdm, sparse=0.95)
```

- 단어 빈도계산 및 워드클라우드

```r
wordFreq <- slam::row_sums(plan_tdm)
wordFreq <- sort(wordFreq, decreasing=TRUE)
```

```r
library(wordcloud)
pal <- brewer.pal(8,"Dark2")
w <- names(wordFreq)
wordcloud(words=w, freq=wordFreq,
          min.freq=3, random.order=F,
            random.color=T, colors=pal)
```


겨울방학
선생님
16학번
여름방학
동국대학교
전문가
캠퍼스
수업시간
스피킹
스포츠
통계학과
통계학
시
데이터마이닝
컴퓨터
adsp
자격증
프로그램
마케팅
금융관련
공무원
아르바이트
대학교
사회조사
빅데이터
정보시스템
응용통계학과
2016
교수님
사람들
보험계리사
금융계
프로그래밍
부모님
정보처리기사

▯ **불용어의 처리**

 통계학과 학생의 진로계획이므로 “통계”가 포함된 단어 제거

```
rm.idx <- grep("[통계|대학]", names(wordFreq))
wordFreq1 <- wordFreq[-rm.idx]
```

 분석목적과 관계없는 단어

* “생각” “공부”, “학년”, “계획”, “진로”, “하게”, “분야”

```
stopwords <- c("생각", "공부", "학년", "계획", "진로", "하게", "분야",
   "관련", "관심", "이번", "해서", "무엇", "가지", "정도",
   "포함", "필요", "때문", "안녕", "동안", "이상", "최선",
   "사실", "나중", "이유", "이해", "구체", "기회", "중요",
   "마음", "사용", "자신", "어려움", "포함", "막막", "예전",
   "우선", "취득", "전공", "이것")
wordFreq1 <- wordFreq1[(!(names(wordFreq1) %in% stopwords))]
```

 불용어 제거 후 재분석

```
w1 <- names(wordFreq1)
wordcloud(words=w1, freq=wordFreq1,
          min.freq=2, random.order=F, random.color=T,   colors=pal)
```


영어공부
경쟁력
토익시험
소프트웨어
sas
인터넷
스피킹
한국사
사회조사
선생님
빅데이터
긍정적
정보시스템
금융관련
시스템
데이터
중소기업
스포츠
sas
adsp
자격증
컴퓨터
교수님
많아졌
기업체
전문가
700
수업시간
데이터마이닝
캠퍼스
마케팅
경주캠퍼스
부모님
2016
아르바이트
본격적
동아리
강의실
해외여행
정보처리기사
취득자
마음가짐

▯ 가중치 변경

- TF-IDF : $\text{tfidf}(t, d, D) = tf(t, d) \times idf(t, D)$
  - 단어 빈도, tf(t,d): 문서 내에 나타나는 해당 단어의 빈도
  - 역문서빈도 :$idf(t, D) = \log \frac{|D|}{|\{d \in D : t \in d\}|}$
    - $|D|$: 전체 문서의 수
    - $|\{d \in D : t \in d\}|$: 단어 t가 포함된 문서의 수
  - 특정 문서 내에서 단어 빈도가 높고, 전체 문서들 중 그 단어를 포함한 문서가 적을 수록 TF-IDF값이 높아짐
- binary: 문서에 중복 출현하는 단어를 한번 출현으로 간주, 즉 출현 여부만으로 가중치를 계산

```
plan_tdm <- plan_tdm[na.omit(match(Terms(plan_tdm), w1)),]
tds1 <- weightBin(plan_tdm) # binary weighting
wordFreq1 <- slam::row_sums(tds1)
wordFreq1 <- round(sort(wordFreq1, decreasing=TRUE))
```

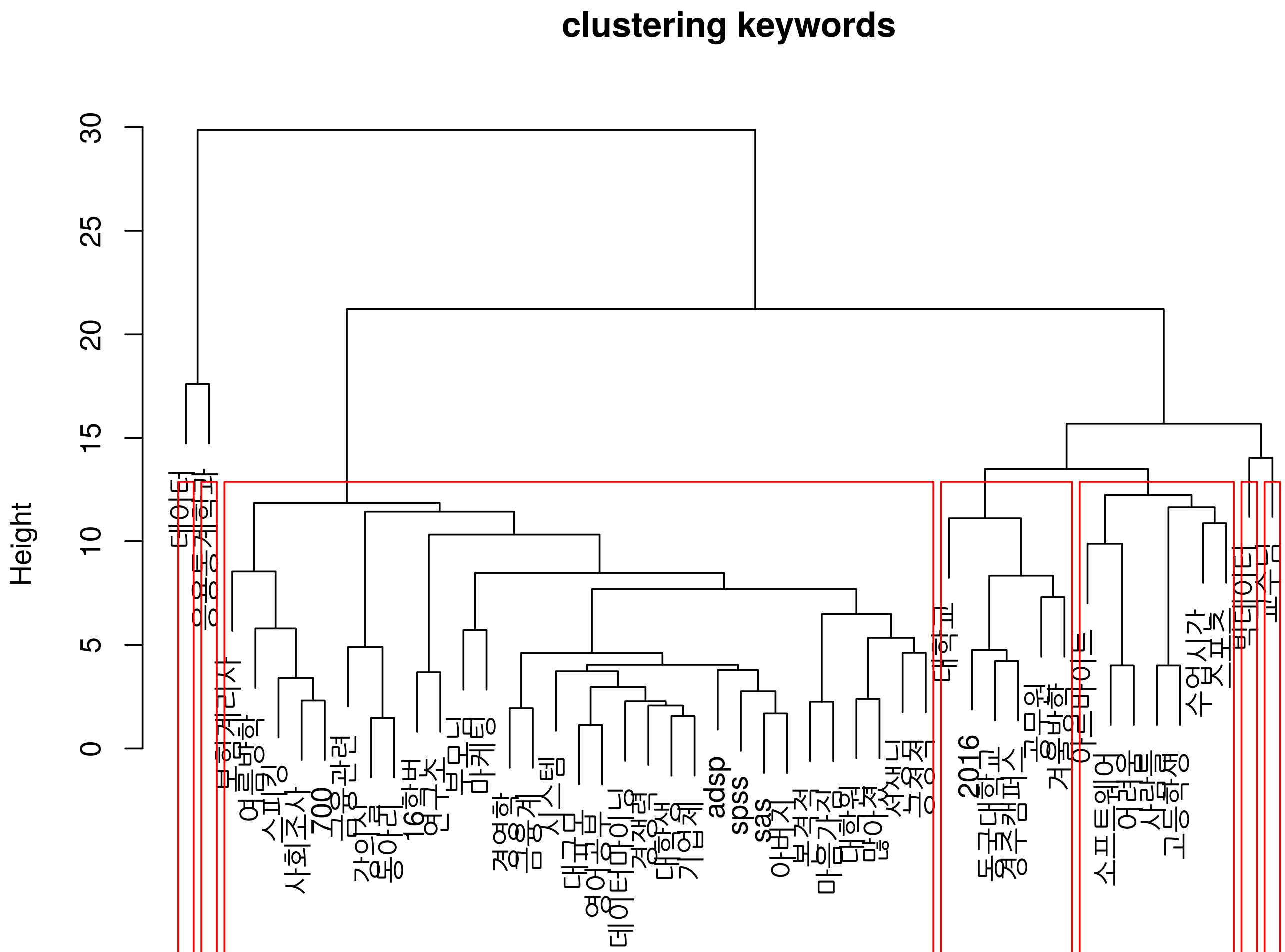
```
w1 <- names(wordFreq1)
wordcloud(words=w1, freq=wordFreq1,
          min.freq=2, random.order=F, random.color=T, colors=pal)
```


응용통계학과
교수님
경주캠퍼스
동국대학교
데이터
빅데이터
16학번
대학원
부모님
사람들
공무원
어려움
겨울방학
2016
여름방학
데이터마이닝
기업체
adsp
고등학생
동아리
대학생
sas
spss
700
본격적
아르바이트
금융계
강의실
많아졌
사회조사
아버지
금융관련
보험계리사
경쟁력
소프트웨어
영어공부
연구소
마음가짐
대규모
스포츠
경영학
선생님
마케팅

2.9 군집분석

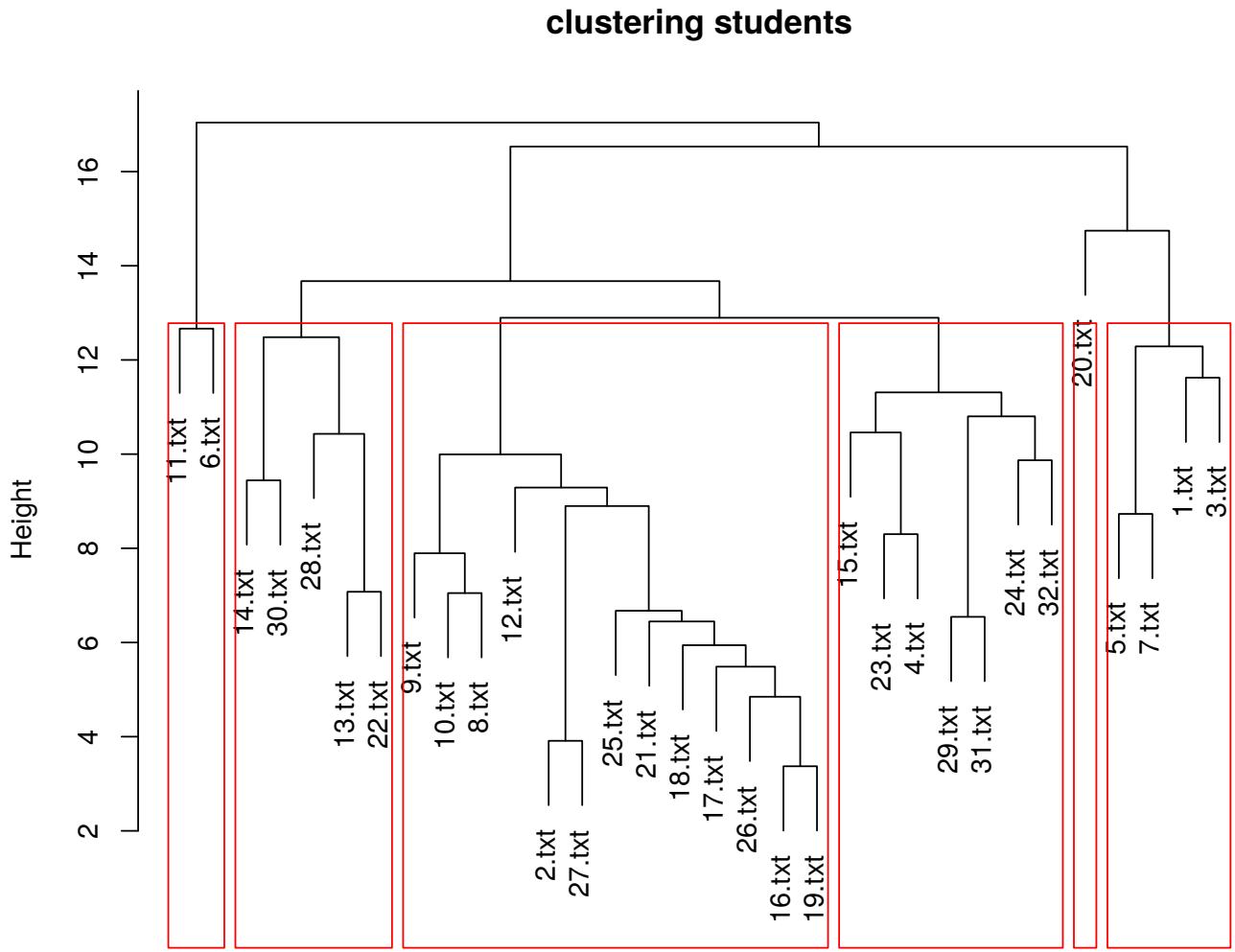
 키워드 클러스터

```
# cluster terms
tds <- plan_tdm[Terms(plan_tdm) %in% w1,]
m2 <- as.matrix(tds)
colnames(m2) <- gsub(".txt", "", colnames(m2))
distMatrix <- dist(scale(m2))
fit <- hclust(distMatrix, method = "ward.D")
plot(fit, xlab="", sub="", main="clustering keywords")
rect.hclust(fit, k = 7)
```


clustering keywords
Height
0
5
10
15
20
25
30
adsp
spss
sas
2016

▯ 문서(학생) 클러스터

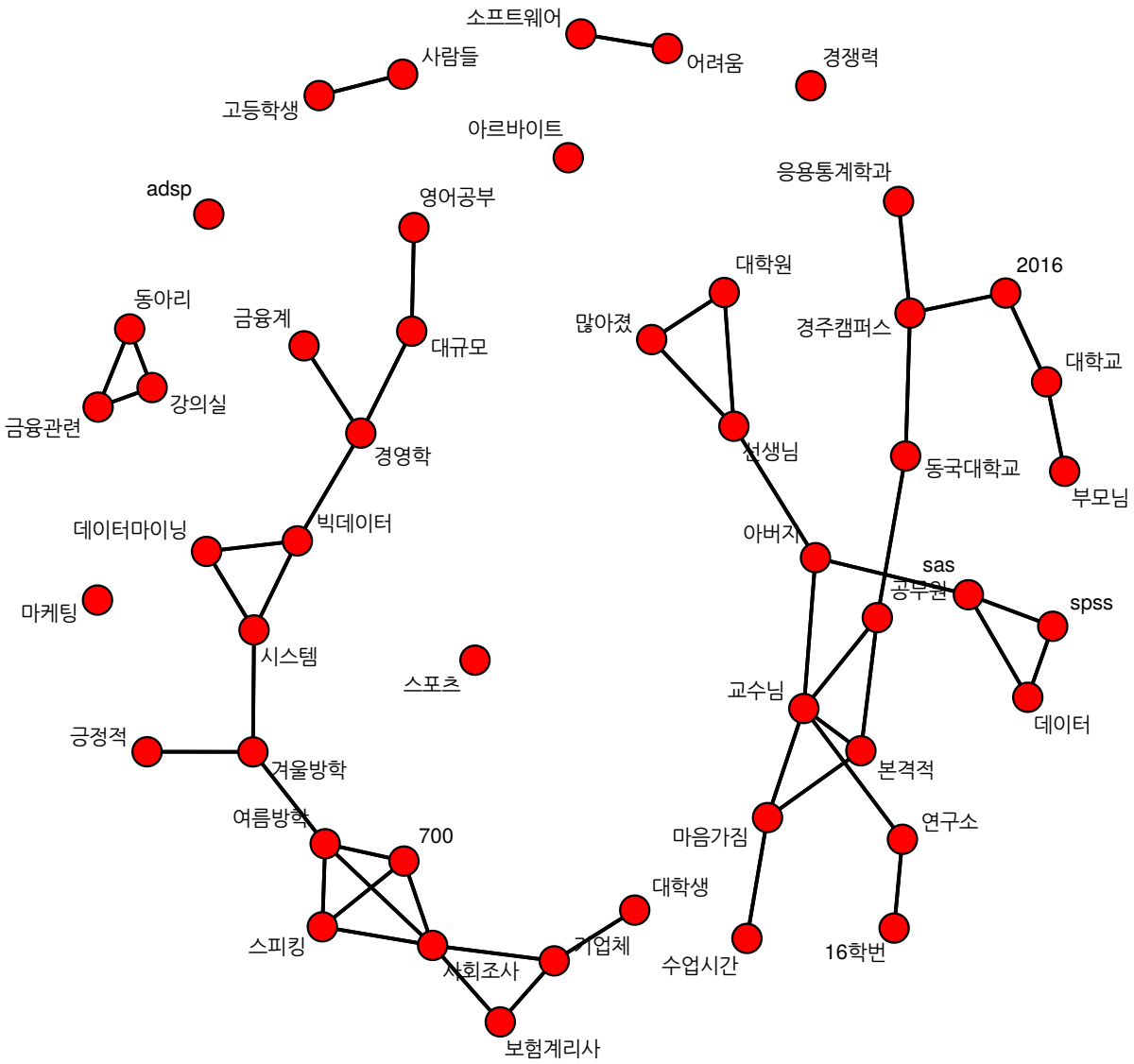
```r
# cluster students
tds <- plan_tdm[Terms(plan_tdm) %in% w1,]
m2 <- as.matrix(tds)
tm2 <- t(m2)
distMatrix <- dist(scale(tm2))
fit <- hclust(distMatrix, method = "ward.D")
# cut tree into 4 clusters
plot(fit, xlab="", sub="", main="clustering students")
rect.hclust(fit, k = 6)
```



2.10 키워드 연관 네트워크

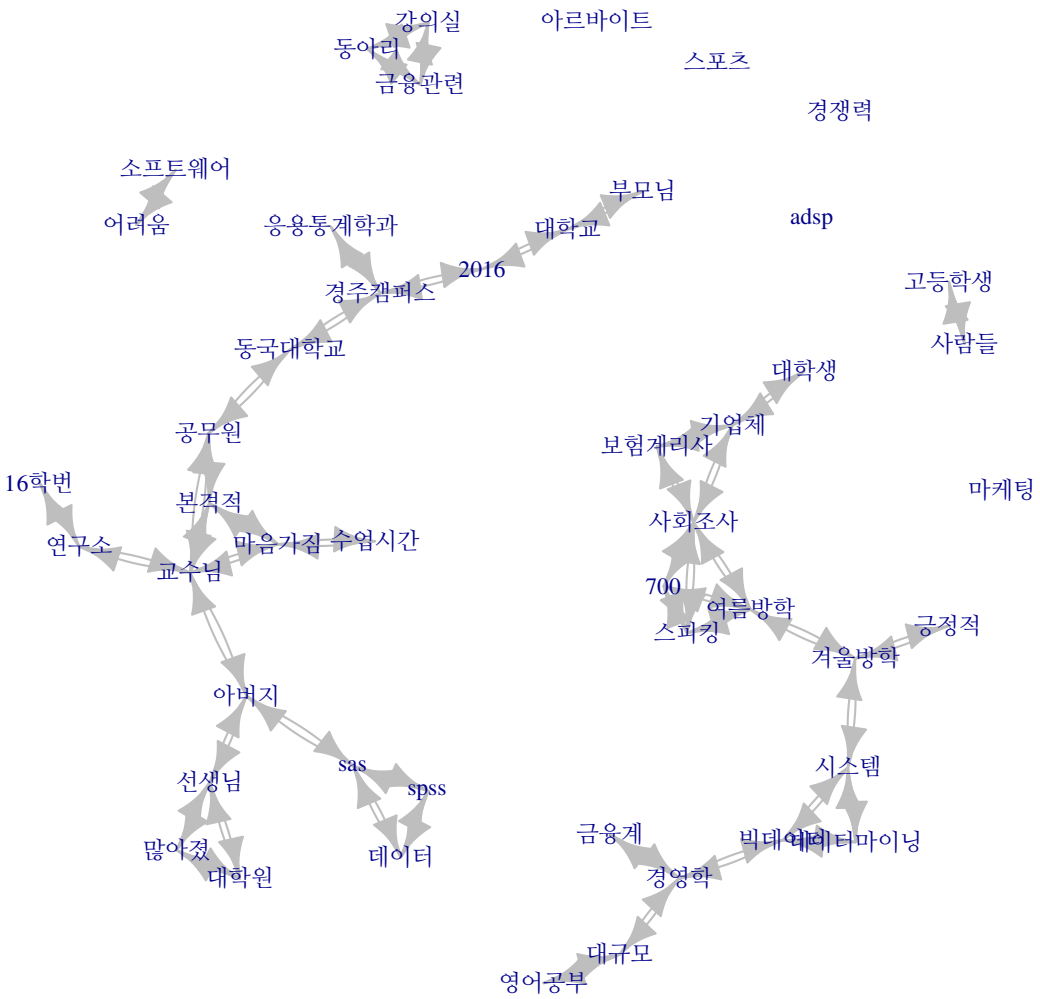
1. 키워드간 연관성을 이용한 네트워크 분석

```r
tds1 <- weightTfIdf(plan_tdm)
M <- t(as.matrix(tds1))
  g <- cor(M)
  diag(g) <- 0
  g[is.na(g)] <- 0
  g[g < 0.4 ] <- 0
  rownames(g) <- colnames(g) <- Terms(tds1)
library(sna)
library(igraph)
  sna::gplot(g, label=colnames(g), gmode="graph",
  label.cex=0.6, vertex.cex=1)
```


소프트웨어
어려움
사람들
고등학생
경쟁력
아르바이트
adsp
영어공부
응용통계학과
대학원
2016
동아리
금융계
많아졌
경주캠퍼스
대규모
대학교
강의실
금융관련
선생님
동국대학교
부모님
경영학
데이터마이닝
빅데이터
아버지
sas
공무원
spss
마케팅
시스템
스포츠
교수님
데이터
긍정적
본격적
겨울방학
여름방학
700
마음가짐
연구소
대학생
스피킹
기업체
수업시간
16학번
사회조사
보험계리사

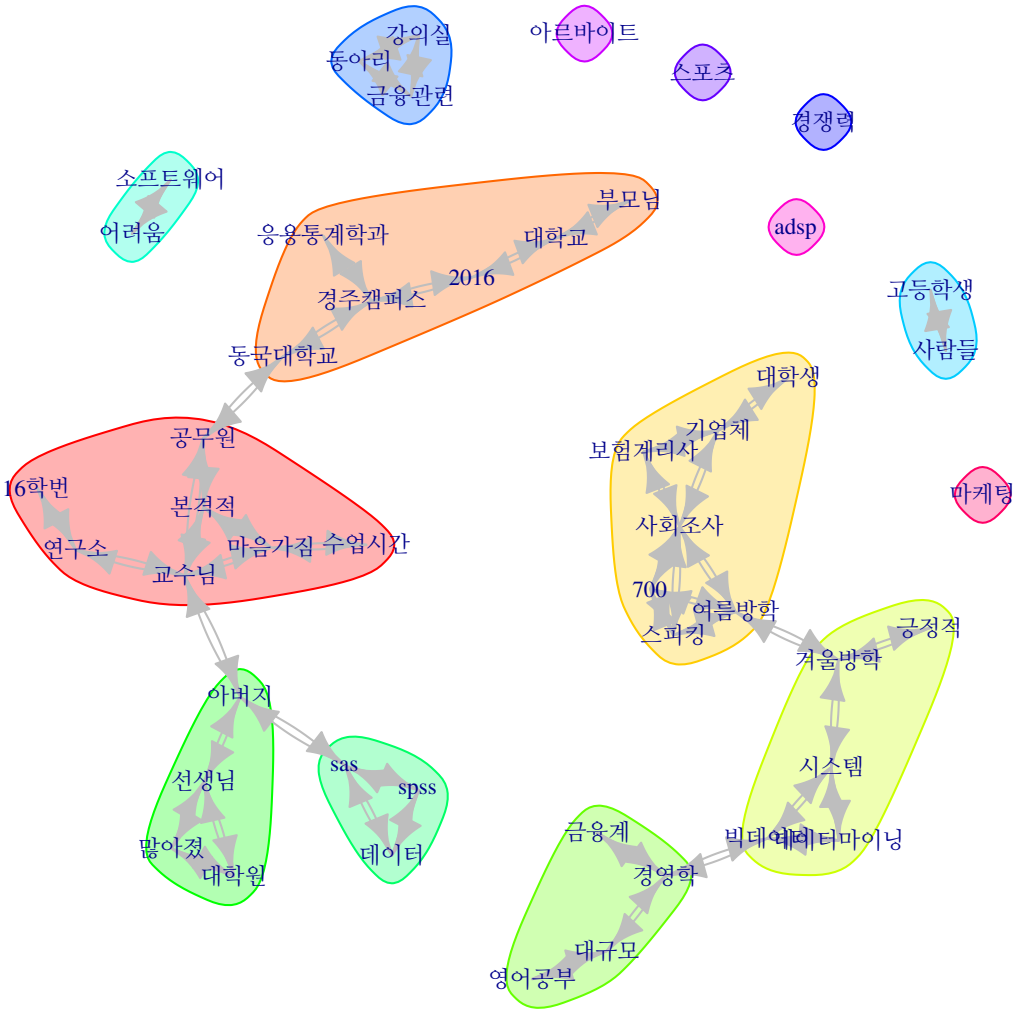
```r
  g1 <- graph_from_adjacency_matrix(g, weighted=TRUE)
  set.seed(12345)
plot(g1, edge.curved=.1, vertex.label.cex=0.7, edge.color="grey",
   vertex.frame.color="grey", vertex.size=0, vertex.shape="none", vertex.color="white",
   main=paste0("진로 계획 연관-키워드 네트워크"))
```

진로 계획 연관-키워드 네트워크



```r
wc <- cluster_walktrap(g1)
set.seed(12345)
plot(wc, g1, edge.curved=.1, vertex.label.cex=0.7, edge.color="grey",
 vertex.frame.color="grey", vertex.size=0, vertex.shape="none", vertex.color="white",
 main=paste0("진로 계획 연관-키워드 네트워크 커뮤니티"))
```

진로 계획 연관-키워드 네트워크 커뮤니티



2.11 토픽 분석

- latent Dirichlet allocation

1. Blei et al. (2003)의 LDA
   - 토픽은 단어들의 출현 분포에 따라 구별됨
   - 문서는 여러개의 토픽으로 구성됨
   - 문서집합이 있으면, 각 문서의 토픽 분포와 토픽에서 단어들의 출현분포를 추정
2. 토픽별 keyword의 분포 : $P(keyword = t|topic)$
   - posterior(lda)$terms
   - lda@beta: log-scale
3. Document에 대한, 토픽의 사후확률($P(topic = k|doc_i)$)
   - lda@gamma
   - posterior(lda)$topics

- R을 이용한 LDA 토픽분석

1. LDA 토픽 모델링
   - 5개의 유형으로 분류 (잠정적으로)

```
plan_tdm_1 <- plan_tdm[,slam::col_sums(plan_tdm)>0]
dtm <- as.DocumentTermMatrix(plan_tdm_1)
library(topicmodels)
```

```r
lda <- LDA(dtm, k = 5, control=list(seed=123456)) # find 10 topics
#plot(lda@loglikelihood, type="l")
(lda@loglikelihood[length(lda@loglikelihood)])
```

```
## [1] -14.87384
```

2. 토픽별 키워드 및 사후 분포

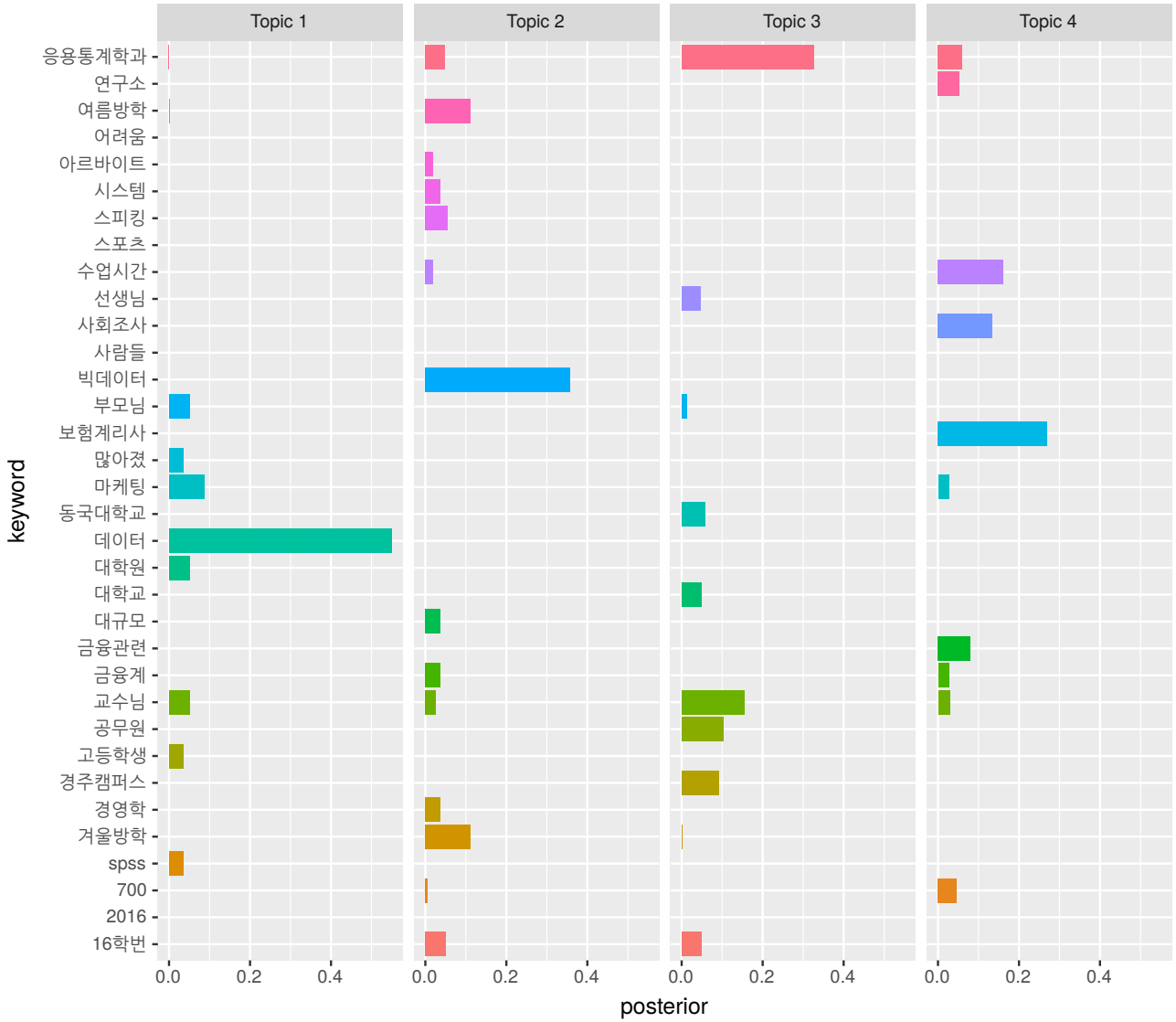
```r
(term <- terms(lda, 10))
```

```
##       Topic 1    Topic 2        Topic 3        Topic 4
##  [1,] "데이터"   "빅데이터"     "응용통계학과" "보험계리사"
##  [2,] "마케팅"   "겨울방학"     "교수님"       "수업시간"
##  [3,] "교수님"   "여름방학"     "공무원"       "사회조사"
##  [4,] "대학원"   "스피킹"       "경주캠퍼스"   "금융관련"
##  [5,] "부모님"   "16학번"       "동국대학교"   "응용통계학과"
##  [6,] "spss"     "응용통계학과" "16학번"       "연구소"
##  [7,] "고등학생" "경영학"       "대학교"       "700"
##  [8,] "많아졌"   "시스템"       "선생님"       "교수님"
##  [9,] "adsp"     "대규모"       "본격적"       "마음가짐"
## [10,] "긍정적"   "금융계"       "아버지"       "기업체"
##       Topic 5
##  [1,] "스포츠"
```

```
##  [2,] "데이터"
##  [3,] "대학교"
##  [4,] "동국대학교"
##  [5,] "응용통계학과"
##  [6,] "사람들"
##  [7,] "아르바이트"
##  [8,] "어려움"
##  [9,] "2016"
## [10,] "대학원"
```

```
x <- posterior(lda)$terms
library(ggplot2)
y <- data.frame(t(x[, apply(x, 2, max) > 0.03]))
z <- data.frame(type=paste("Topic", 1),
        keyword=rownames(y), posterior=y[,1])
for(i in 2:4){
    z <- rbind(z, data.frame(type=paste("Topic", i),
            keyword=rownames(y), posterior=y[,i]))
}
ggplot(z, aes(keyword, posterior, fill=as.factor(keyword)))+
   geom_bar(position="dodge",stat="identity")+
   coord_flip() +
```

```
  facet_wrap(~type,nrow=1) +
  theme(legend.position="none")
```



3. 학생들의 주요 토픽(관심사: 사후확률이 최대인 토픽을 주요토픽으로 간주)

```r
tt <- apply(posterior(lda)$topics, 1, which.max)
head(data.frame(topic=tt))
```

```
##        topic
## 1.txt      2
## 10.txt     2
## 11.txt     5
## 12.txt     5
## 13.txt     3
## 14.txt     3
```

3 예제 데이터 분석 2 - 학생 민원데이터

3.1 데이터 전처리

```r
load(file="campus_qna.RData")
oo <- do.call("rbind", out)
library(KoNLP)
```

```
library(NIADic)
useNIADic()
```

```
## Backup was just finished!
## 983012 words dictionary was built.
```

```
new_term <- c("취업지원센터", "학생cs센터", "시약장", "이클래스", "국가장학", "유드림스",
              "피라미타칼리지","교양필수", "학사운영실", "원효관", "에어컨", "100주년기념관",
              "열람실", "근로장학", "계절학기", "의학관", "사회과학대학", "졸업앨범",
              "통학버스", "경주캠퍼스", "자연과학대학", "핵심교양", "자연과학관", "생화학연구실",
              "진흥관", "원효관", "동국대학교", "사범대")
new_dic <- data.frame(term=new_term , tag="ncn", stringsAsFactors=F)
buildDictionary(ext_dic=c('woorimalsam', 'insighter'), user_dic=new_dic, replace_usr_dic = T)
```

```
## 983040 words dictionary was built.
```

```
library(tm)
text <- oo$제목
text <- paste(text, "관련")
text <- str_replace_all(text, "UDRIMS", "유드림스")
```

```r
text <- str_replace_all(text, "와이파이", "무선인터넷")
text <- str_replace_all(text, "wifi", "무선인터넷")
text <- str_replace_all(text, "시약장", " 시약장 ")
text <- str_replace_all(text, "사람", " 사람 ")
text <- str_replace_all(text, "취업지원센터", " 취업지원센터 ")
text <- str_replace_all(text, "도서관", " 도서관 ")
text <- str_replace_all(text, "학생cs센터", " 학생cs센터 ")
text <- str_replace_all(text, "편입생", " 편입생 ")
text <- str_replace_all(text, "된다", " ")
text <- str_replace_all(text, "민원", " 민원 ")
text <- str_replace_all(text, "안내", " 안내 ")
text <- str_replace_all(text, "개선", " 개선 ")
text <- str_replace_all(text, "기숙사", "기숙사 ")
text <- str_replace_all(text, "통학버스", " 통학버스 ")
text <- str_replace_all(text, "해서", " ")
text <- str_replace_all(text, "드립", " ")

text <- text[!grepl("분할등록", text)]
text <- str_replace_all(text, "——", " ")
text <- str_replace_all(text, "Q&A", "QA")
text <- tolower(text)
```

```
text <- str_replace_all(text, "[[:punct:]]", " ")
text1 <- sapply(text, function(tt){
  x <- extractNoun(tt, autoSpacing = F)
  paste(x[nchar(x)>1], collapse=" ")
}, USE.NAMES = F)



text <- VCorpus(VectorSource(text1))
#text = tm_map(text, removeNumbers)
#text = tm_map(text, stripWhitespace)
#text <- tm_map(text,stemDocument)
```

3.2 연관 키워드 네트워크

 워드클라우드

```
library(wordcloud)
txt_tdm <- TermDocumentMatrix(text, control=list(tokenize="scan", wordLengths=c(2, Inf)))
txt_tdm <- txt_tdm[is.na(match(Terms(txt_tdm), c("민원","학교","관련", "해주", "안녕하십니까", "이
txt_tdm <- weightBin(txt_tdm)
```

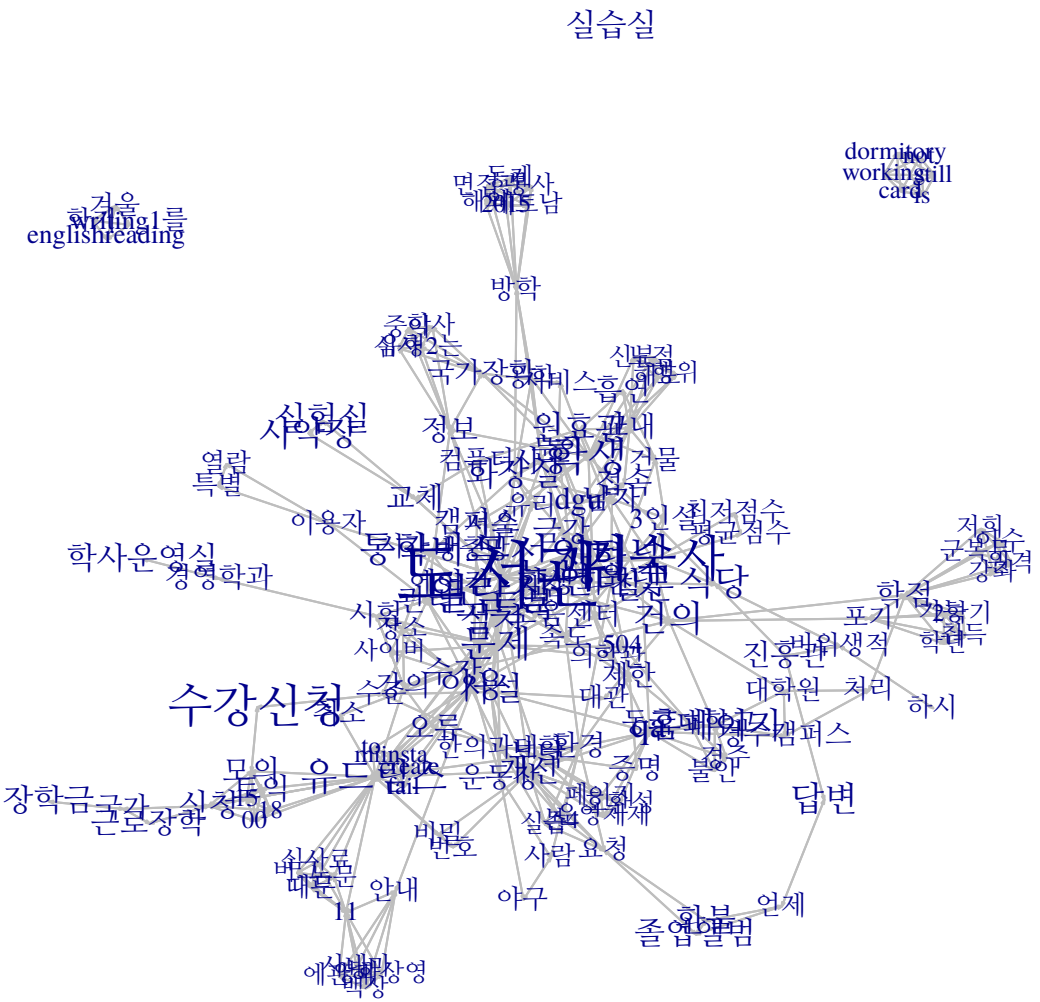
```r
txt_tdm <- removeSparseTerms(txt_tdm, sparse=0.997)
txt_tdm_1 <- txt_tdm[slam::row_sums(txt_tdm)>=1,]
txt_tdm_2 <- txt_tdm_1[,slam::col_sums(txt_tdm_1)>=1]

pal <- brewer.pal(8,"Dark2")
wordFreq1 <- slam::row_sums(txt_tdm_2)
wordFreq1 <- round(sort(wordFreq1, decreasing=TRUE))
w1 <- names(wordFreq1)
wordcloud(words=w1, freq=wordFreq1,
  min.freq=2, random.order=F, random.color=T, colors=pal)
```


도서관
기숙사
수강신청
무선인터넷
유드림스
학생
열람실
홈페이지
통학버스
시설
노트북
문제
환불
시약장
건의
이용
실험실
qa
인터넷
dgu
englishreading
핵심교양
포기
한의과대학
동아리
대관
과목
사회과학대학
때문
이클래스
공학
국가
학사운영실
writing1를
석장동
문의
수정
교체
사용
장학금
유아교육
유리
발급
수강
화장실
수준
건물
운동장
방학
원효관
에어컨
불안
전속
504
환경
변경
경주캠퍼스
요청
모의
남자
겨울
3인실
대학원
답변
시간
관리
학생회관
특별
금강
마에
경주
이수
안나
캠퍼스
제한
야구
시험
방충망
컴퓨터
사람
흡연
진흥관
학기
학점
11
경영학과
취소
증명
신청
강의
휴게실
센터
장소
서비스
학생증
교양필수
교내
하시
속도
실습실
토익
서울
이상
근로장학
최저점수
열람
국가장학
오류
동국대학교
처리
참사랑
번호
평균점수
비위생적
체육관
언제
자연과학대학
의학관
취업지원센터

 키워드 연관 네트워크

```r
g <- slam::tcrossprod_simple_triplet_matrix(txt_tdm_2)
wf <- diag(g)
diag(g) <- 0
rownames(g) <- colnames(g) <- Terms(txt_tdm_2)
sidx <- apply(g, 2, sum) >= 5
g1 <- g[sidx, sidx]
library(igraph)
g1 <- graph_from_adjacency_matrix(g1, weighted=T)
set.seed(22345)
plot(g1, edge.arrow.size=0.1, edge.color="grey", vertex.label.cex=0.7+wf[sidx]/20,
  vertex.frame.color="grey", vertex.size=0, vertex.shape="none", vertex.color="white",
  main=paste0("학생 민원 연관-키워드 네트워크"))
```

학생 민원 연관-키워드 네트워크



- 간단한 그림

```r
sidx <- apply(g, 2, sum) >= 8
g1 <- g[sidx, sidx]
library(igraph)
g1 <- graph_from_adjacency_matrix(g1, weighted=T)
set.seed(22345)
plot(g1, edge.arrow.size=0.1, edge.color="grey", vertex.label.cex=0.7+wf[sidx]/20,
  vertex.frame.color="grey", vertex.size=0, vertex.shape="none", vertex.color="white",
  main=paste0("학생 민원 연관-키워드 네트워크"))
```

학생 민원 연관-키워드 네트워크