기말 프로젝트

데이터과학융합스쿨 20173218 김주형

2020 12-07

```
# library
library(knitr)
library(tm) # 연관성 검사 결과값을 시각화하기 위한 그래프
```

```
## Loading required package: NLP
```

```
library(qgraph) # 연관성 검사 결과값을 시각화하기 위한 그래프
```

```
## Registered S3 methods overwritten by 'huge':
##   method    from
##   plot.sim  BDgraph
##   print.sim BDgraph
```

```
library(qdapRegex) #
library(KoNLP) # 한글 형태소 분석
```

```
## Checking user defined dictionary!
```

```
library(stringr) # Regexp를 하기 위해서 한글을 문자별로, 단어별로 잘라내고 바꾸는 일을 담당
library(wordcloud) # 워드클라우드
```

```
## Loading required package: RColorBrewer
```

```
library(wordcloud2)
library(RColorBrewer) # 워드클라우드 색깔을 예쁘게 해 주는 걸로 알고 있음
library(NIADic)# KoNLP의 한글사전 최신판. SejongDic보다 단어량도 많고 정확도도 높다.
```

```
## Successfully Loaded NIADic Package.
```

```
library(googleVis) # 차트 그리기
```

```
## Creating a generic function for 'toJSON' from package 'jsonlite' in package 'googleVis'
```

```
##
## Welcome to googleVis version 0.6.9
##
## Please read Google's Terms of Use
## before you start using the package:
## https://developers.google.com/terms/
##
## Note, the plot method of googleVis will by default use
## the standard browser to display its output.
##
## See the googleVis package vignettes for more details,
## or visit https://github.com/mages/googleVis.
##
## To suppress this message use:
## suppressPackageStartupMessages(library(googleVis))
```

```
# 카카오톡 대화 불러오기
```

```
text <- file("대화.txt", encoding = 'UTF-8')
kakaotalk = readLines(text, encoding = 'UTF-8')
```

```
## Warning in readLines(text, encoding = "UTF-8"): '�□��.txt'에서 불완전한 마지막 행
## 이 발견되었습니다
```

```
head(kakaotalk)
```

```
## [1] "상대방 님과 카카오톡 대화"
## [2] "저장한 날짜 : 2020-12-07 15:14:56"
## [3] ""
## [4] "--------------- 2020년 11월 16일 월요일 ---------------"
## [5] "[상대방] [오후 11:45] 저 목요일날 될거 같아요!"
## [6] "[나] [오후 11:58] 몇시에"
```

```
kakaotalk <- kakaotalk[-1:-3] # 필요없는 부분 제거
kakaotalk <- str_replace_all(kakaotalk, '핸드폰에 저장한 상대방 이름', '본명')
```

```
# 누가 더 많이 톡을 하는가?
```

```
me <- length(kakaotalk[grep("WW[나", kakaotalk)])
partner <- length(kakaotalk[grep("WW[상대방", kakaotalk)])

# data.frame으로 변환
volume <- c(me, partner)  # 카톡량
name <- c('나', '상대방')  # 이름
kakao_df <- data.frame(name, volume)  # 카톡 data.frame으로 변환
str(kakao_df)
```

```
## 'data.frame':    2 obs. of  2 variables:
##  $ name  : chr  "나" "상대방"
##  $ volume: int  1245 1523
```

```
# 시각화
pie <- gvisPieChart(kakao_df, options = list(width = 400, height = 300))

header <- pie$html$header
header <- gsub("charset = UTF-8", "charset = EUC-KR", header)
pie$html$header <- header

plot(pie)
```

```
## starting httpd help server ... done
```

```
# 오전과 오후 중에 언제 더 많은 톡을 하는가?
```

```
am <- length(kakaotalk[grep("₩₩[오전", kakaotalk)])  # 오전
pm <- length(kakaotalk[grep("₩₩[오후", kakaotalk)])  # 오후

volume2 <- c(am, pm)
name2 <- c('오전', '오후')
time_df <- data.frame(name, volume2)
str(time_df)
```

```
## 'data.frame':    2 obs. of  2 variables:
##  $ name   : chr  "나" "상대방"
##  $ volume2: int  1105 1663
```

```
# 시각화
pie1 <- gvisPieChart(time_df, options = list(width = 400, height = 300))

header <- pie1$html$header
header <- gsub("charset = UTF-8", "charset = EUC-KR", header)
pie1$html$header <- header

plot(pie1)
```

```
# 분석을 위한 데이터 전처리
```

```
prep <- str_replace_all(kakaotalk, "이모티콘", "") %>%   # 이모티콘 없애기
  str_replace_all("ㅠㅠ[오후", "") %>%   # 오후 지우기
  str_replace_all("ㅠㅠ[오전", "") %>%   # 오전 지우기
  str_replace_all("[ㄱ-ㅎ]+", "") %>%   # 자음 없애기
  str_replace_all("ㅠㅠ[나", "") %>%   # 대화방 사람 이름 없애기(나)
  str_replace_all("ㅠㅠ[상대방", "") %>%   # 대화방 사람 이름 없애기(사촌형)
  str_replace_all("ㅠㅠ[|ㅠㅠ]", "") %>%   # 카톡 텍스트 데이터의 대괄호 지우기
  str_replace_all("[0-9]+:[0-9]+ㅠㅠ]", "") %>%   # 모든 시간 없애기
  str_replace_all("사진", "") %>%   # 사진 없애기
  str_replace_all("[가-힣]요", "") %>%   # txt데이터가 요일별로 나눠져 있기 때문
  str_replace_all("년|월|일", "") %>%   # 연월일 지우기
  str_replace_all("[0-9]+", "") %>%   # 숫자 지우기
  str_replace_all("(http).+(ㅠㅠw)", "") %>%   # 링크 지우기
  str_replace_all(",+", "") %>%   # 문장부호 지우기
  as.character()
head(prep)
```

```
## [1] "---------------     ---------------" "  : 저 날 될거 같!"
## [3] "  : 몇시에"                          "  : 볼"
## [5] "  : ?"                               "  : 주형님 시간 언제 가능하!?"
```

```
# 명사 추출
noun1 <- sapply(prep, extractNoun, USE.NAMES = F) %>% unlist()
head(noun1)
```

```
## [1] "저"   "날"   "될거" "같"   "몇"   "시"
```

```
noun <- Filter(function(prep){nchar(prep) >= 2}, noun1)   # 두음절 이상의 단어만 추출
head(noun)
```

```
## [1] "될거"         "형님"         "시간"         "언제"         "가능"
## [6] "수업들으시러"
```

```
nouns <- sort(table(noun), decreasing = T)

# 명사 빈도
wordFreq <- table(noun)
head(wordFreq)
```

```
## noun
##     #스타벅스         alpha         anova Anova써도ㅠ될           api
##             1             1             1             1             1
##   ChildHeight
##             1
```

```
# 명사 빈도 50순위
wordFreq_top <- head(sort(wordFreq, decreasing = T), 50)
head(wordFreq_top)
```

```
## noun
## 진짜 시간 과제 그거 이거 분석
##   30   29   21   18   18   17
```

```
print(wordFreq_top)
```

```
## noun
##       진짜       시간       과제       그거       이거       분석       코드
##         30         29         21         18         18         17         17
##     파이썬       저거   정규분포       우리       하나 히스토그램       금주
##         15         14         14         13         13         12         10
##     산점도       생각       아들       학기       거기       사람       상관
##          9          9          9          9          8          8          8
##       정규       제출       통사       회귀     교수님       버스       설명
##          8          8          8          8          7          7          7
##       언제       오늘       이상       자식       평균        rmd       강의
##          7          7          7          7          7          6          6
##       관계     노가다       복전     부모님       세상       스벅     시각화
##          6          6          6          6          6          6          6
##       아빠     아이들       알겠       얘기       전화       친구       코딩
##          6          6          6          6          6          6          6
##       학점
##          6
```

```
# 워드클라우드
```

```
wordcloud2(nouns)
```


```
pal2 <- brewer.pal(8, "Dark2")
pal <- brewer.pal(12, 'Set3')
pal <- pal[-c(1:2)]

png("wordcloud.png", width = 400, height = 300)
wordcloud(names(wordFreq_top),
          freq = wordFreq_top,
          random.order = F,
          rot.per = 0,
          col= pal)
```

```
# qgraph
```

```
# 명사만 가져오기
tt <- paste(unlist(SimplePos22(prep)))
allnoun <- str_match_all(tt, "[가-힣]+/[N][C]|[가-힣]+/[N][Q]+") %>% unlist()
N <- str_replace_all(allnoun, "/[N][C]", "") %>%
  str_replace_all("/[N][Q]", "") %>% unlist()   # 명사로 추출된 단어들의 분류표인 /NC, /NQ 등을 제거한다.

CorpusNC <- Corpus(VectorSource(N))
myDtm <- TermDocumentMatrix(CorpusNC, control = list(wordLengths = c(4, 10),
                                                     removePunctuation = T,
                                                     removeNumbers = T,
                                                     weighting = weightBin))
```

```
## Warning in TermDocumentMatrix.SimpleCorpus(CorpusNC, control = list(wordLengths
## = c(4, : custom functions are ignored
```

```
Encoding(myDtm$dimnames$Terms) = "UTF-8"

# 확인
findFreqTerms(myDtm, lowfreq = 10)
```

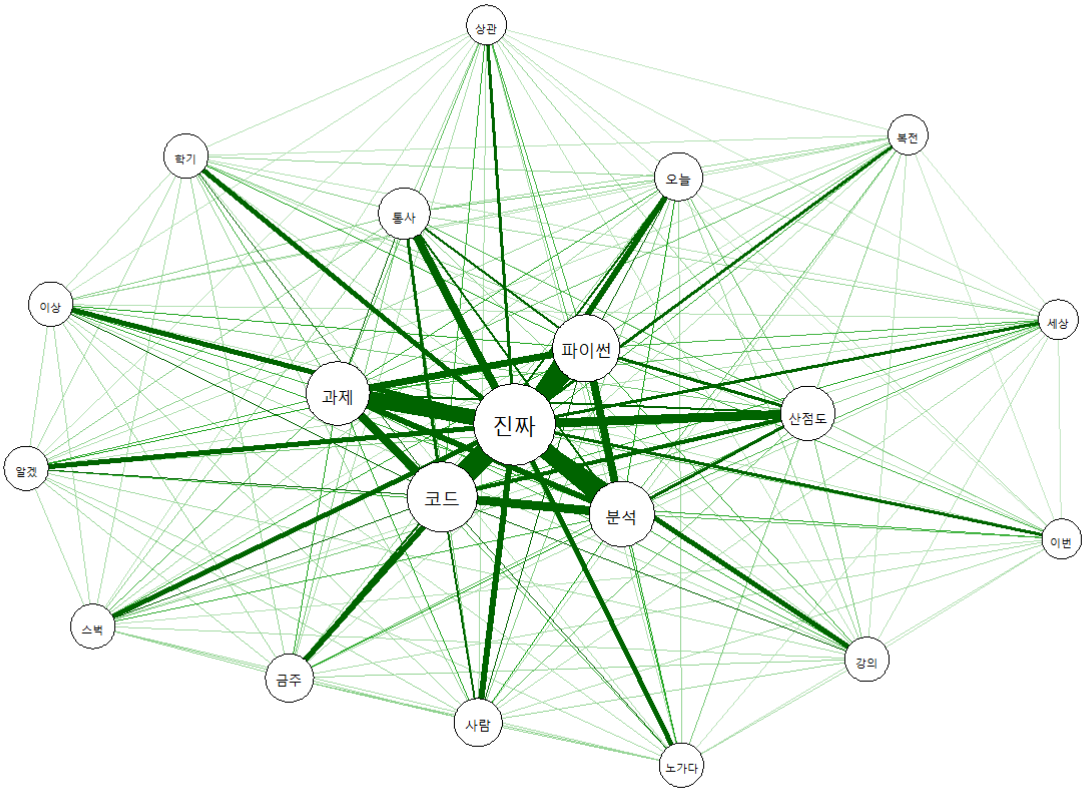
```
## [1] "파이썬" "진짜"   "과제"   "코드"   "분석"
```

```
myDtmM <- as.matrix(myDtm) # 행렬로 변환
myrowDtmM <- rowSums(myDtmM)
myDtmM.order <- myrowDtmM[order(myrowDtmM, decreasing = T)]
freq.wordsNC <- myDtmM.order[1:20] ##sample(myDtmM.order[myDtmM.order > 5], 20,replace=F)인걸
 이거로 바꿈
freq.wordsNC <- as.matrix(freq.wordsNC)
freq.wordsNC
```

```
##        [,1]
## 진짜     27
## 코드     17
## 파이썬   15
## 분석     14
## 과제     13
## 산점도    9
## 통사      8
## 오늘      7
## 금주      7
## 사람      7
## 스벅      6
## 강의      6
## 노가다    6
## 학기      6
## 알겠      6
## 이상      6
## 상관      5
## 이번      5
## 복전      5
## 세상      5
```

```
co.matrix <- freq.wordsNC %*% t(freq.wordsNC)

# qgraph
qgraph(co.matrix,
       labels = rownames(co.matrix),
       diag = F,
       layout = 'spring',
       vsize = log(diag(co.matrix)))
```


상관
학기
통사
오늘
복전
이상
세상
파이썬
과제
진짜
산점도
알겠
코드
분석
이번
스벅
금주
강의
사람
노가다