

제8장 텍스트 마이닝과 워드 클라우드 활용

제8장 텍스트 마이닝과 워드 클라우드 활용

2. 지역별 인구수의 변화에 대한 클라우드 출력
 - 2-1. 지역별 순이동에 따른 워드 클라우드
 - 2-2. 단어들의 색 변환
 - 2-3. 다양한 단어 색 출력을 위한 팔레트 패키지의 활용
 - 2-4. 페이지 로딩 및 데이터 파일 열기
 - 2-5. 데이터 정제: 불필요 지역 제외 ('전국' 지역 제외)
 - 2-6. '구' 단위 지역 통계 삭제
 - 2-7. 전입자 수가 많은 지역
 - 2-8. 전출자 수가 많은 지역
3. 연설문의 워드 클라우드 만들기
 - 3-1. 패키지 설치
 - 3-2. 세종 사전 업로드
 - 3-3. 연설문(data/speech.txt 또는 data/speech2.txt) 파일 불러오기
 - 3-4. Keyword와 사용횟수 추출하기
 - 3-5. word_count의 차트 작성 (그래픽 시각화)
 - 3-6. wordcloud 작성 (그래픽 시각화)
 - 3-7. 3-4.의 보완 사전에 단어 추가 및 추출된 명사의 삭제
 - 1) 사전에 새로운 단어 추가 : mergeUserDic() 함수 이용
 - 2) 불필요한 단어 사전에서 제거하기 : gsub() 함수 이용
 - 3-8. word_count의 차트 작성
 - 3-9. wordcloud 작성 (그래픽 시각화)
 - 3-10. 출력 결과의 이미지 저장

[연습문제]

2. 지역별 인구수의 변화에 대한 클라우드 출력

2-1. 지역별 순이동에 따른 워드 클라우드

```
install.packages("wordcloud")
library(wordcloud)

word<- c("서울특별시", "부산광역시", "대구광역시", "광주광역시", "대전광역시", "인천광역시")
# Keywords
frequency <- c(351, 285, 199, 161, 148, 125) # frequencies of
Keywords

wordcloud(word, frequency, colors="blue") # wordcloud
wordcloud(word, frequency, colors=rainbow(length(word)))

# wordcloud
```

2-2. 단어들의 색 변환

```
wordcloud(word, frequency, random.order=F, random.color=F,  
colors=rainbow(length(word)))
```

2-3. 다양한 단어 색 출력을 위한 팔레트 패키지의 활용

```
install.packages("RColorBrewer")  
library(RColorBrewer)  
  
display.brewer.all() # display all palettes  
display.brewer.pal(n = 8, name = 'Dark2') # Dark2 팔레트  
  
pal2 <- brewer.pal(8, "Dark2") # (참고) https://statklee.github.io/viz/viz-r-  
colors.html  
pal2 # 16진수 문자열 parsing  
  
word<- c("서울특별시", "부산광역시", "대구광역시", "광주광역시", "대전광역시", "인천광역시")  
# Keywords  
frequency <- c(351, 285, 199, 161, 148, 125) # frequencies of  
Keywords  
wordcloud(word, frequency, colors=pal2)
```

2-4. 페이지 로딩 및 데이터 파일 열기

```
library(wordcloud)  
library(RColorBrewer)  
pal2 <- brewer.pal(8, "Dark2")  
  
# Data/101_DT_1B26001_A01_M.csv 파일 불러오기 (wide table 형식)  
data <- read.csv(file.choose(), header=T)  
head(data)  
str(data)
```

2-5. 데이터 정제: 불필요 지역 제외 ('전국' 지역 제외)

```
data2 <- data[data$행정구역.시군구.별 != "전국", ] # 전국이 아닌 데이터만 data2 값으로...  
head(data2)
```

2-6. '구' 단위 지역 통계 삭제

```
x <- grep("구$", data2$행정구역.시군구.별) # grep() : 특정 텍스트를 갖는 색인 번호 검색  
x # (참고)  
http://blog.naver.com/PostView.nhn?  
blogId=coder1252&logNo=220947332269&parentCategoryNo=&categoryNo=10&viewDate=&isShowPo  
pularPosts=true&from=search  
  
data3 <- data2[-c(x), ]  
head(data3)
```

2-7. 전입자 수가 많은 지역

```

data4 <- data3[data3$순이동.명 > 0, ]
word <- data4$행정구역.시군구.별
frequency <- data4$순이동.명

wordcloud(word, frequency, colors=pal2)

```

2-8. 전출자 수가 많은 지역

```

data5 <- data3[data3$순이동.명<0, ]
word <- data5$행정구역.시군구.별
frequency <- abs(data5$순이동.명)

wordcloud(word, frequency, colors=pal2)

```

3. 연설문의 워드 클라우드 만들기

3-1. 패키지 설치

```

install.packages("KoNLP")
install.packages("RColorBrewer")
install.packages("wordcloud")

library(KoNLP)
library(RColorBrewer)
library(wordcloud)

```

```

##### # library(KoNLP) 에러발생과 처리
방법 ##### # <에러발생> # Error:
package or namespace load failed for 'KoNLP': # .onLoad가 loadNamespace()에서 'rJava'때문에 실패했습
니다: # 호출: fun(libname, pkgname) # 에러: JAVA_HOME cannot be determined from the Registry
##### #<처리방법> # 자바 다운로드 : ht
tp://www.java.com/ko 혹은 http://java.com/ko/download/manual.jsp (32비트 / 64비트) # 설치하면 됨.
#####

```

3-2. 세종 사전 업로드

```

useSejongDic()

pal2 <- brewer.pal(8, "Dark2")

```

3-3. 연설문(data/speech.txt 또는 data/speech2.txt) 파일 불러오기

```

text <- readLines(file.choose()) # readLines()은 각 줄을 문자열로 다루므로,
# 반환값은 문자열로 이뤄진 '문자 벡터'가 된다.
text

```

3-4. Keyword와 사용횟수 추출하기

```

noun <- sapply(text, extractNoun, USE.NAMES=F) \# KoNLP의 extractNoun 을 이용해서 한글명
사만 추출.
# USE.NAMES=F : 원문장이 없이 명사만 출력

noun
typeof(noun)          \# noun은 list type임.

noun2 <- unlist(noun)  \# unlist : 리스트 구조를 벡터 구조로 변환한다.
noun2                 \# noun2 : 문자벡터

word_count <- table(noun2) \# table() : 단어별로 count...
word_count

head(sort(word_count, decreasing=TRUE), 10) \# 상위 10개 단어 출력

```

3-5. word_count의 차트 작성 (그래픽 시각화)

```

temp <- sort(word_count, decreasing=T) \# 내림차순(빈도가 가장 많은 것에서 부터
가장 작은 순)으로 단어 정렬, 상위 30개 선택
temp \# 확인

temp <- temp[-1] \# 공백단어 제거
temp <- temp[1:30] \# 상위 10개

barplot(temp, las = 2, names.arg = names(temp), \# 차트 출력
col = "lightblue", main = "Most frequent words", \# 축, 제목 입력
ylab = "Word frequencies") \# 축 입력

```

3-6. wordcloud 작성 (그래픽 시각화)

```

wordcloud(names(word_count), freq=word_count, scale=c(8,0.4), min.freq=1,
random.order=F, rot.per=.1, colors=pal2)

```

3-7. 3-4.의 보완 사전에 단어 추가 및 추출된 명사의 삭제

1) 사전에 새로운 단어 추가 : mergeUserDic() 함수 이용

```

mergeUserDic(data.frame(c("정치"), c("ncn"))) \# '정치'를 사전에 추가하기
mergeUserDic(data.frame(c("노태우"), c("ncn"))) \# '노태우'를 사전에 추가하기
mergeUserDic(data.frame(c("문민"), c("ncn"))) \# '문민'을 사전에 추가하기

noun <- sapply(text, extractNoun, USE.NAMES=F)
noun2 <- unlist(noun)

```

2) 불필요한 단어 사전에서 제거하기 : gsub() 함수 이용

```

noun2 <- gsub("여러분", "", noun2)
noun2 <- gsub("우리", "", noun2)
noun2 <- gsub("오늘", "", noun2)
noun2 <- gsub("이것", "", noun2) \# 최종 차트를 보면서 불필요한 단어 확인하고 삭제한다.
noun2 <- gsub("그것", "", noun2)
noun2 <- gsub("하계", "", noun2)

noun2 <- Filter(function(x){nchar(x) >= 2}, noun2) \# 한 글자 단어 제거하기...

word_count <- table(noun2)
word_count

```

3-8. word_count의 차트 작성

```

temp <- sort(word_count, decreasing=T)[1:30] \# 내림차순(빈도가 가장 많은 것에서 부터 가장
작은 순)으로 단어 정렬, 상위 30개 선택
temp                                     \# 확인

temp <- temp[-1]                         \# 공백단어 제거
barplot(temp, las = 2, names.arg = names(temp), \# 차트 출력
        col = "lightblue", main = "Most frequent words", \# 축, 제목 입력
        ylab = "Word frequencies") \# 축 입력

```

3-9. wordcloud 작성 (그래픽 시각화)

```

wordcloud(names(word_count), freq=word_count, scale=c(8,0.4), min.freq=1,
random.order=F, rot.per=.1, colors=pal2)

```

3-10. 출력 결과의 이미지 저장

```

setwd("/temp")
png(filename = "speech.png", width = 480, height = 480)
\# 저장 디바이스 종료
dev.off()

```

[연습문제]

1) 역대 대통령의 연설기록 사이트(<http://www.pa.go.kr/research/contents/speech/index.jsp>)에서 대통령들의 취임연설문을 비교분석하라.

- 대통령 후보수락 연설문의 비교분석

2) 목원대 총장 취임사 분석



