과제명	빅데이터 처리 증간고사 과제

- 과 목 명 빅데이터 처리
- 담당교수 김 마 루
- 학부(과) 컴퓨터소프트웨어공학과
- 학 년 4
- 분반번호 01
- 학 번 20173147
- 이 름 이 명 진
- 연 락 처 010-2999-7748
- 제 출 일 2020 년 05 월 01 일

<힙합 가사 텍스트 마이닝>

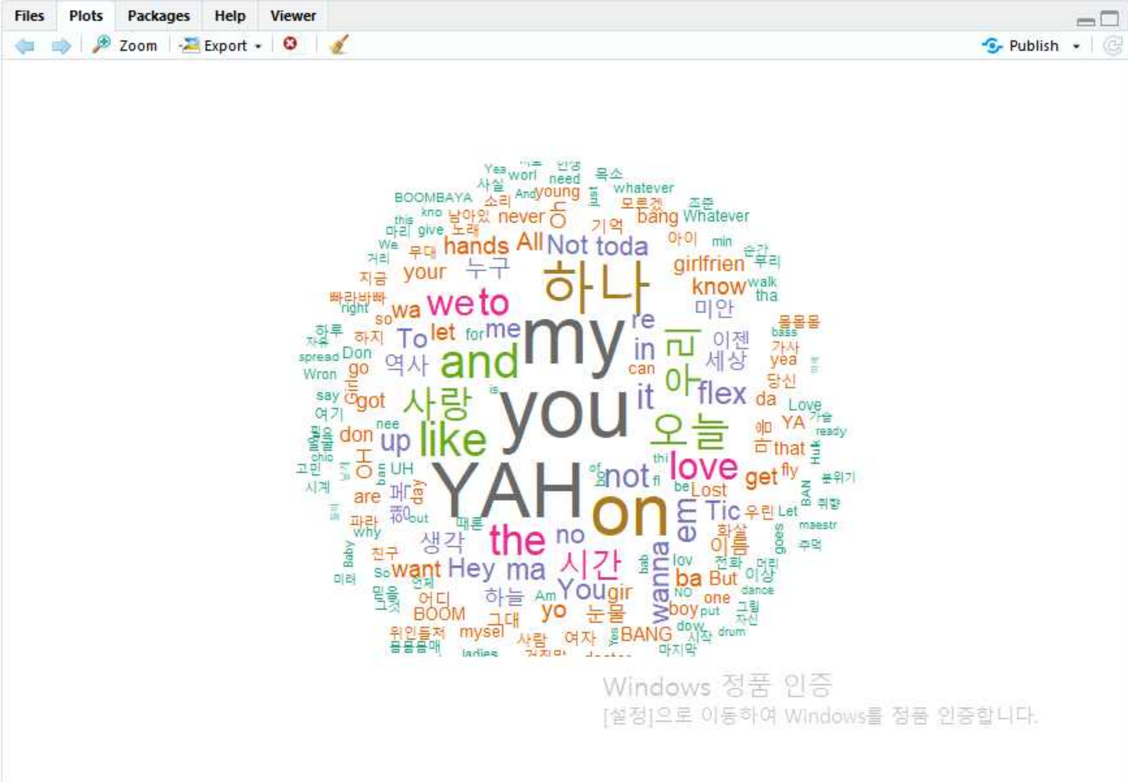
```r
#20173147 이명진
install.packages("Sejong")
install.packages("hash")
install.packages("rJava")
install.packages("tau")
install.packages("RSQLite")
install.packages("devtools")
install.packages("dplyr")
install.packages("KONLP")
install.packages("stringr")
install.packages("wordcloud")

library(KoNLP)
library(dplyr)

Sys.setenv(JAVA_HOME="C:/Program Files/Java/jdk1.8.0_171/")

useNIADic()

#x<-read.csv(file.choose(),header=T) #경로 지정이 어려워 특정 파일을 열수없을 때 사용 ㄱ + 데이터 저장할 장ㅅ

# 데이터 불러오기
txt <- readLines("hiphop.txt")
head(txt)
# ## Warning in readLines("hiphop.txt"): incomplete final line found on ## 'hiphop.txt'


library(stringr)


# 특수문제 제거
txt <- str_replace_all(txt, "\\W", " ")
txt

# 명사 추출하기
extractNoun("대한민국의 영토는 한반도와 그 부속도서로 한다")
# 가사에서 명사추출
nouns <- extractNoun(txt)

# 추출한 명사 list 를 문자열 벡터로 변환 , 단어별 빈도표 생성
wordcount <- table(unlist(nouns))

# 데이터 프레임으로 변환
df_word <- as.data.frame(wordcount, stringsAsFactors = F)


# 변수명 수정
df_word <- rename(df_word,
                  word = Var1,
                  freq = Freq)

# 두 글자 이상 단어 추출
df_word <- filter(df_word, nchar(word) >= 2)

top_20 <- df_word %>%
  arrange(desc(freq)) %>%
  head(20)


# 워드클라우드 만들기


# 패키지 로드
library(wordcloud)
## Loading required package: RColorBrewer
library(RColorBrewer)

#단어 색상 목록 만들기
pal <- brewer.pal(8,"Dark2")  # Dark2 색상 목록에서 8 개 색상 추출

#워드 클라우드 생성

set.seed(1234)                    # 난수 고정
wordcloud(words = df_word$word,   # 단어
          freq = df_word$freq,    # 빈도
          min.freq = 2,           # 최소 단어 빈도
          max.words = 200,        # 표현 단어 수
          random.order = F,       # 고빈도 단어 중앙 배치
          rot.per = .1,           # 회전 단어 비율
          scale = c(4, 0.3),      # 단어 크기 범위
          colors = pal)           # 색깔 목록



#단어 색상 바꾸기
pal <- brewer.pal(9,"Blues")[5:9]  # 색상 목록 생성
set.seed(1234)                     # 난수 고정


wordcloud(words = df_word$word,    # 단어
          freq = df_word$freq,     # 빈도
          min.freq = 2,            # 최소 단어 빈도
          max.words = 200,         # 표현 단어 수
          random.order = F,        # 고빈도 단어 중앙 배치
          rot.per = .1,            # 회전 단어 비율
          scale = c(4, 0.3),       # 단어 크기 범위
          colors = pal)            # 색상 목록


########################################################## 힙합가사 텍스트 마이닝 끝...
```

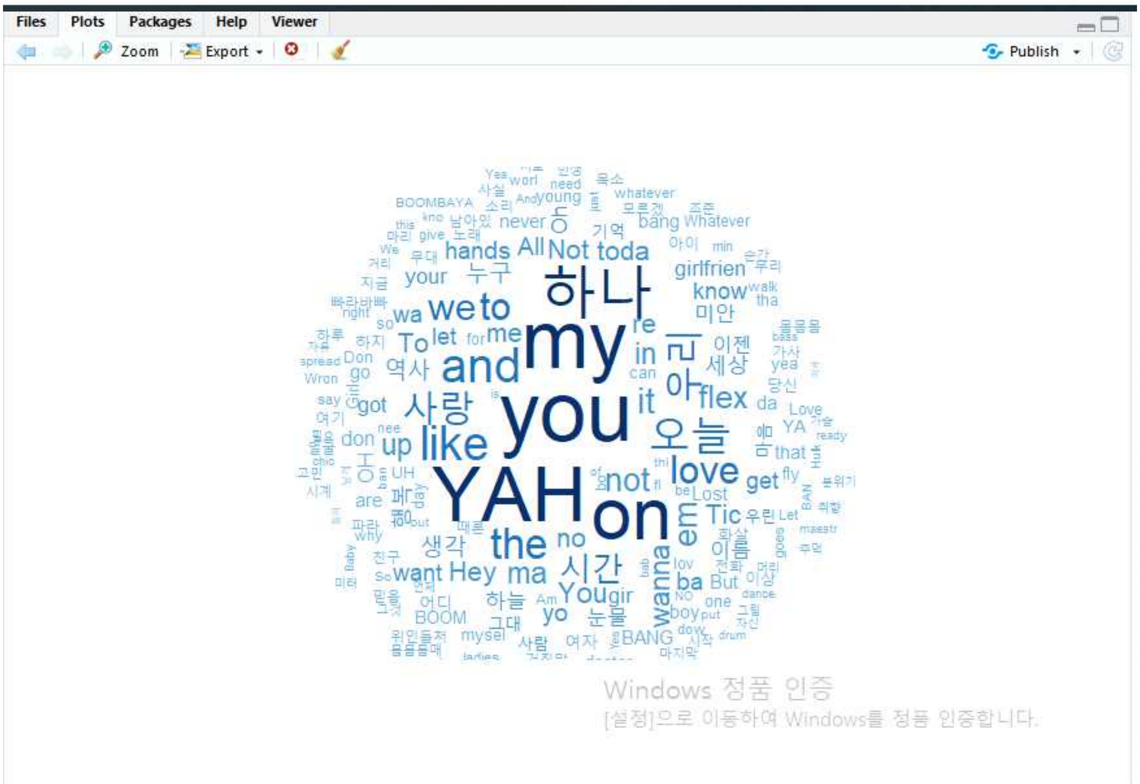
```
> library(KoNLP)
> library(dplyr)
> Sys.setenv(JAVA_HOME="C:/Program Files/Java/jdk1.8.0_171/")
> useNIADic()
Backup was just finished!
983012 words dictionary was built.
> # 데이터 불러오기
> txt <- readLines("hiphop.txt")
Warning message:
In readLines("hiphop.txt") : incomplete final line found on 'hiphop.txt'
> head(txt)
[1] "\"보고 싶다"                    "이렇게 말하니까 더 보고 싶다" "너희 사진을 보고 있어도"
[4] "보고 싶다"                      "너무 야속한 시간"             "나는 우리가 밉다"
> library(stringr)
> # 특수문제 제거
> txt <- str_replace_all(txt, "\\W", " ")
> txt
   [1] " 보고 싶다"                          "이렇게 말하니까 더 보고 싶다"
   [3] "너희 사진을 보고 있어도"             "보고 싶다"
   [5] "너무 야속한 시간"                    "나는 우리가 밉다"
   [7] "이젠 얼굴 한번 보는 것 조차"         "힘들어진 우리가"
   [9] "여긴 온통 겨울 뿐이야"               "8월에도 겨울이 와"
  [11] "마음은 시간을 달려가네"              "홀로 남은 설국열차"
  [13] "니 손 잡고 지구 반대편까지 가"       "겨울을 끝내고파"
  [15] "그리움들이 얼마나"                   "눈처럼 내려야 그 봄날이 올까"
  [17] "Friend"                              "허공을 떠도는"
  [19] "작은 먼지처럼 작은 먼지처럼"         "날리는 눈이 나라면"
  [21] "조금 더 빨리"                        "네게 닿을 수 있을 텐데"
  [23] "눈꽃이 떨어져요"                     "또 조금씩 멀어져요"
  [25] "보고 싶다 보고 싶다"                 "보고 싶다 보고 싶다"
  [27] "얼마나 기다려야"                     "또 몇 밤을 더 새워야"
```

Console Terminal

C:/Users/leemyeongjin/Desktop/빅데이터처리/

```
 [ reached getOption("max.print") -- omitted 3261 entries ]
> # 명사 추출하기
> extractNoun("대한민국의 영토는 한반도와 그 부속도서로 한다")
[1] "대한민국" "영토"     "한반도"   "부속도서" "한"
> # 가사에서 명사추출
> nouns <- extractNoun(txt)
> # 추출한 명사 list 를 문자열 벡터로 변환 , 단어별 빈도표 생성
> wordcount <- table(unlist(nouns))
> # 데이터 프레임으로 변환
> df_word <- as.data.frame(wordcount, stringsAsFactors = F)
> # 변수명 수정
> df_word <- rename(df_word,
+                   word = Var1,
+                   freq = Freq)
> # 두 글자 이상 단어 추출
> df_word <- filter(df_word, nchar(word) >= 2)
> top_20 <- df_word %>%
+   arrange(desc(freq)) %>%
+   head(20)
> # 패키지 로드
> library(wordcloud)
> ## Loading required package: RColorBrewer
> library(RColorBrewer)
> #단어 색상 목록 만들기
> pal <- brewer.pal(8,"Dark2")  # Dark2 색상 목록에서 8 개 색상 추출
> set.seed(1234)                # 난수 고정
> wordcloud(words = df_word$word,  # 단어
+           freq = df_word$freq,   # 빈도
+           min.freq = 2,          # 최소 단어 빈도
+           max.words = 200,       # 표현 단어 수
+           random.order = F,      # 고빈도 단어 중앙 배치
+           rot.per = .1,          # 회전 단어 비율
+           scale = c(4, 0.3),     # 단어 크기 범위
+           colors = pal)          # 색깔 목록
> #단어 색상 바꾸기
> pal <- brewer.pal(9,"Blues")[5:9]  # 색상 목록 생성
> set.seed(1234)                     # 난수 고정
> wordcloud(words = df_word$word,    # 단어
+           freq = df_word$freq,     # 빈도
+           min.freq = 2,            # 최소 단어 빈도
+           max.words = 200,         # 표현 단어 수
+           random.order = F,        # 고빈도 단어 중앙 배치
+           rot.per = .1,            # 회전 단어 비율
+           scale = c(4, 0.3),       # 단어 크기 범위
+           colors = pal)            # 색상 목록
>
>
>
```


Files
Plots
Packages
Help
Viewer
Zoom
Export
Publish
YAH
you
my
하나
on
and
like
the
사랑
오늘
love
not
시간
wanna
em
flex
Windows 정품 인증
[설정]으로 이동하여 Windows를 정품 인증합니다.


Files
Plots
Packages
Help
Viewer
Zoom
Export
Publish
YAH
you
my
하나
on
and
like
the
사랑
오늘
love
not
시간
wanna
em
flex
Windows 정품 인증
[설정]으로 이동하여 Windows를 정품 인증합니다.

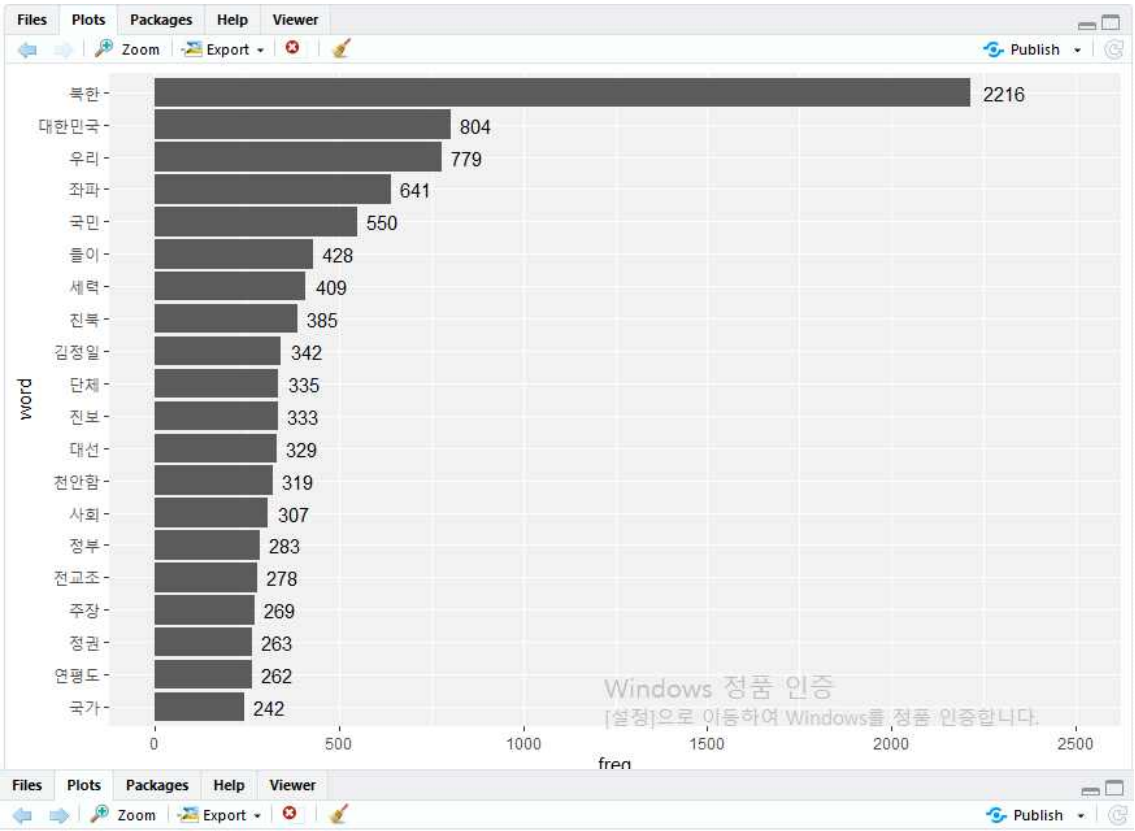
<국정원 트윗 텍스트 마이닝>

```r
#20173147 이명진

# 국정원 트윗 텍스트 마이닝
#데이터 준비하기

install.packages("Sejong")
install.packages("hash")
install.packages("rJava")
install.packages("tau")
install.packages("RSQLite")
install.packages("devtools")
install.packages("dplyr")
install.packages("KONLP")
install.packages("stringr")
install.packages("wordcloud")

library(KoNLP)
library(dplyr)
library(stringr)
library(wordcloud)
library(RColorBrewer)
Sys.setenv(JAVA_HOME="C:/Program Files/Java/jdk1.8.0_171/")
# 데이터 로드
twitter <- read.csv("twitter.csv",
                    header = T,
                    stringsAsFactors = F,
                    fileEncoding = "UTF-8")

# 변수명 수정
twitter <- rename(twitter,
                  no = 번호,
                  id = 계정이름,
                  date = 작성일,
                  tw = 내용)

# 특수문자 제거
twitter$tw <- str_replace_all(twitter$tw, "\\W", " ")

head(twitter$tw)

#단어 빈도표 만들기
# 트윗에서 명사추출
nouns <- extractNoun(twitter$tw)

# 추출한 명사 list 를 문자열 벡터로 변환 , 단어별 빈도표 생성
wordcount <- table(unlist(nouns))

# 데이터 프레임으로 변환
df_word <- as.data.frame(wordcount, stringsAsFactors = F)

# 변수명 수정
df_word <- rename(df_word,                    word = Var1,                    freq = Freq)

#두 글자 이상으로 된 단어 추출, 빈도 상위 20개 단어 추출
# 두 글자 이상 단어만 추출
df_word <- filter(df_word, nchar(word) >= 2)

# 상위 20 개 추출
top20 <- df_word %>%   arrange(desc(freq)) %>%   head(20)

top20

#단어 빈도 막대 그래프 만들기
library(ggplot2)

order <- arrange(top20, freq)$word               # 빈도 순서 변수 생성


ggplot(data = top20, aes(x = word, y = freq)) +   ylim(0, 2500) +   geom_col() +   coord_flip() +   scale_x_discrete(l
  geom_text(aes(label = freq), hjust = -0.3)     # 빈도 표시

#워드 클라우드 만들기
pal <- brewer.pal(8,"Dark2")       # 색상 목록 생성
set.seed(1234)                     # 난수 고정


wordcloud(words = df_word$word,    # 단어
          freq = df_word$freq,     # 빈도
          min.freq = 10,           # 최소 단어 빈도
          max.words = 200,         # 표현 단어 수
          random.order = F,        # 고빈도 단어 중앙 배치
          rot.per = .1,            # 회전 단어 비율
          scale = c(6, 0.2),       # 단어 크기 범위
          colors = pal)            # 색상 목록


#색깔 바꾸기
pal <- brewer.pal(9,"Blues")[5:9]  # 색상 목록 생성
set.seed(1234)                     # 난수 고정


wordcloud(words = df_word$word,    # 단어
          freq = df_word$freq,     # 빈도
          min.freq = 10,           # 최소 단어 빈도
          max.words = 200,         # 표현 단어 수
          random.order = F,        # 고빈도 단어 중앙 배치
          rot.per = .1,            # 회전 단어 비율
          scale = c(6, 0.2),       # 단어 크기 범위
          colors = pal)            # 색상 목록

############################################################ 국정원 자료 워드 클라우드 끝...
```

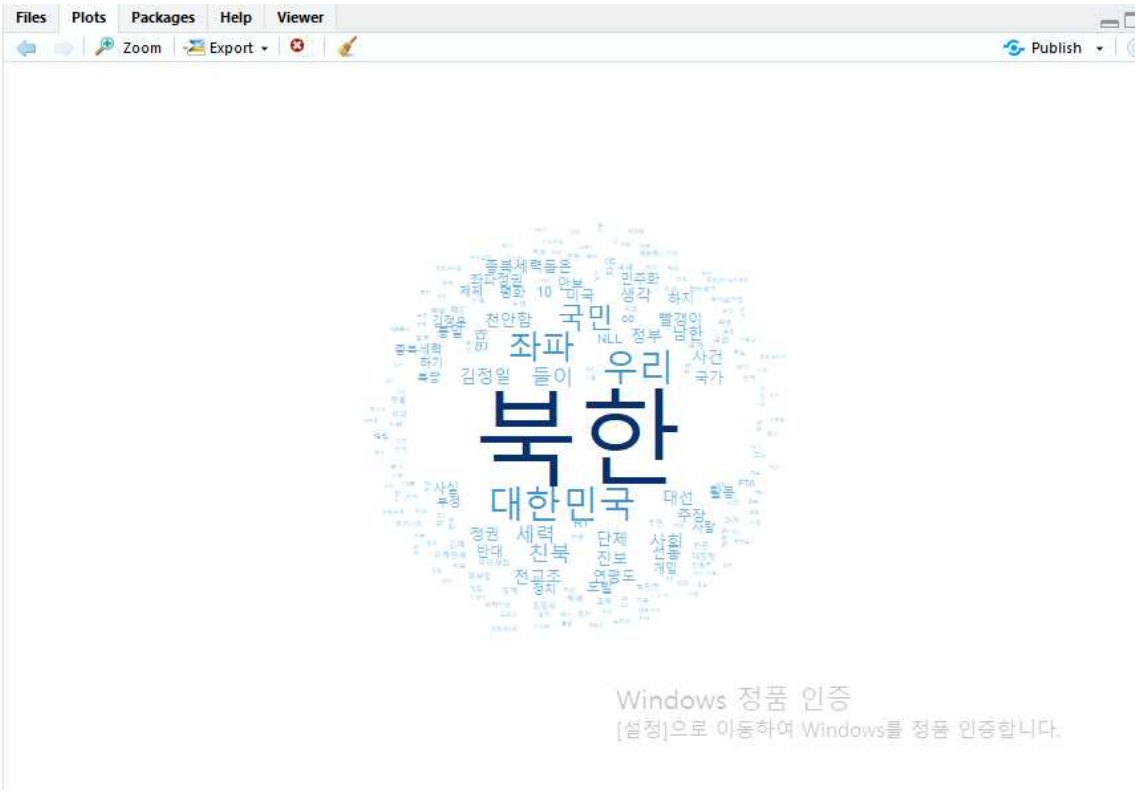
```
Console  Terminal
C:/Users/leemyeongjin/Desktop/빅데이터처리/
> library(KoNLP)
> library(dplyr)
> library(stringr)
> library(wordcloud)
> library(RColorBrewer)
> Sys.setenv(JAVA_HOME="C:/Program Files/Java/jdk1.8.0_171/")
> # 데이터 로드
> twitter <- read.csv("twitter.csv",
+                     header = T,
+                     stringsAsFactors = F,
+                     fileEncoding = "UTF-8")
> # 변수명 수정
> twitter <- rename(twitter,
+                   no = 번호,
+                   id = 계정이름,
+                   date = 작성일,
+                   tw = 내용)
> # 특수문자 제거
> twitter$tw <- str_replace_all(twitter$tw, "\\W", " ")
> head(twitter$tw)
[1] "민주당의 ISD관련 주장이 전부 거짓으로 속속 드러나고있다  미국이 ISD를 장악하고 있다고 주장하지만 중재인 123명 가운데 미국인은 10명뿐이라고 한다 "
[2] "말로만  미제타도   사실은  미제환장   김정일 운구차가 링컨 컨티넬탈이던데 북한의 독재자나 우리나라 종북들이나 걸으로는 노동자  서민을 대변한다면서 고급 외제차  아이팟에 자식을 미국 유학에 환장하는 위선자들인거죠"
[3] "한나라당이 보수를 버린다네요 뭔가착각하는모냥인에 국민들이보수를싫어하는게 아니라뻘짓거리하는분들을싫어하는겁니다야당이진보어쩌고저쩌고한다고해서그들을조아한다고생각하면대착각"
[4] "FTA를 대하는 현명한 자세  사실 자유주의 경제의 가장 큰 수해자는 한국이죠  농어업분야 피해를 줄이는 정부대안을 최대한  보완하고 일자리 창출 등 실익을 최대화해 나가는게 현실적인 대처자세일듯 "
[5] "곽노현씨 갈수록 가관입니다  뇌물질에 마늘 병역 의혹까지  도대체 아이들이 뭘 보고 배우겠습니까  이래도 자리 연연하시겠습니까 "

[6] "과거 집권시 한미FTA를 적극 추진하던 세력이 이제 집권하면 폐기하겠다고 주장합니다  어이없어 말도 안 나오네요  표만 얻을 수 있다면 국가 안보나 경제가 어떻게 되든 상관없다는 무책임한 행태들  우리 정치의 후진성을 드러내는 거죠 "
```

```
Console  Terminal
C:/Users/leemyeongjin/Desktop/빅데이터처리/
[6] "과거 집권시 한미FTA를 적극 추진하던 세력이 이제 집권하면 폐기하겠다고 주장합니다  어이없어 말도 안 나오네요  표만 얻을 수 있다면 국가 안보나 경제가 어떻게 되든 상관없다는 무책임한 행태들  우리 정치의 후진성을 드러내는 거죠 "
> #단어 빈도표 만들기
> # 트윗에서 명사추출
> nouns <- extractNoun(twitter$tw)
> # 추출한 명사 list 를 문자열 벡터로 변환 , 단어별 빈도표 생성
> wordcount <- table(unlist(nouns))
> # 데이터 프레임으로 변환
> df_word <- as.data.frame(wordcount, stringsAsFactors = F)
> # 변수명 수정
> df_word <- rename(df_word,                    word = Var1,                    freq = Freq)
> #두 글자 이상으로 된 단어 추출, 빈도 상위 20개 단어 추출
> # 두 글자 이상 단어만 추출
> df_word <- filter(df_word, nchar(word) >= 2)
> # 상위 20 개 추출
> top20 <- df_word %>%   arrange(desc(freq)) %>%   head(20)
> top20
      word freq
1     북한 2216
2  대한민국  804
3     우리  779
4     좌파  641
5     국민  550
6     들이  428
7     세력  409
8     친북  385
9    김정일  342
10    단체  335
11    진보  333
12    대선  329
13   천안함  319
14     사회  307
```

```
Console  Terminal
C:/Users/leemyeongjin/Desktop/빅데이터처리/
18     성권  263
19   연평도  262
20     국가  242
> #단어 빈도 막대 그래프 만들기
> library(ggplot2)
> order <- arrange(top20, freq)$word              # 빈도 순서 변수 생성
> ggplot(data = top20, aes(x = word, y = freq)) +   ylim(0, 2500) +   geom_col() +   coord_flip() +   scale_x_discrete(limit
= order) +                 # 빈도 순서 변수 기준 막대 정렬
+   geom_text(aes(label = freq), hjust = -0.3)     # 빈도 표시
> #워드 클라우드 만들기
> pal <- brewer.pal(8,"Dark2")       # 색상 목록 생성
> set.seed(1234)                     # 난수 고정
> wordcloud(words = df_word$word,    # 단어
+           freq = df_word$freq,     # 빈도
+           min.freq = 10,           # 최소 단어 빈도
+           max.words = 200,         # 표현 단어 수
+           random.order = F,        # 고빈도 단어 중앙 배치
+           rot.per = .1,            # 회전 단어 비율
+           scale = c(6, 0.2),       # 단어 크기 범위
+           colors = pal)            # 색상 목록
> #색깔 바꾸기
> pal <- brewer.pal(9,"Blues")[5:9]  # 색상 목록 생성
> set.seed(1234)                     # 난수 고정
> wordcloud(words = df_word$word,    # 단어
+           freq = df_word$freq,     # 빈도
+           min.freq = 10,           # 최소 단어 빈도
+           max.words = 200,         # 표현 단어 수
+           random.order = F,        # 고빈도 단어 중앙 배치
+           rot.per = .1,            # 회전 단어 비율
+           scale = c(6, 0.2),       # 단어 크기 범위
+           colors = pal)            # 색상 목록
>
```


Files
Plots
Packages
Help
Viewer
Zoom
Export
Publish
북한 2216
대한민국 804
우리 779
좌파 641
국민 550
들이 428
세력 409
친북 385
김정일 342
단체 335
진보 333
대선 329
전안함 319
사회 307
정부 283
전교조 278
주장 269
정권 263
연평도 262
국가 242
word
0
500
1000
1500
2000
2500
freq
Windows 정품 인증
[설정]으로 이동하여 Windows를 정품 인증합니다.


북한
대한민국
국민
좌파
우리
들이
김정일
전안함
세력
친북
단체
진보
대선
사회
정권
전교조
정부
사건
국가
Windows 정품 인증
[설정]으로 이동하여 Windows를 정품 인증합니다.


Files
Plots
Packages
Help
Viewer
Zoom
Export
Publish
북한
대한민국
좌파
우리
국민
들이
김정일
세력
친북
Windows 정품 인증
[설정]으로 이동하여 Windows를 정품 인증합니다.

<무협지 21세기 무인 텍스트 마이닝>

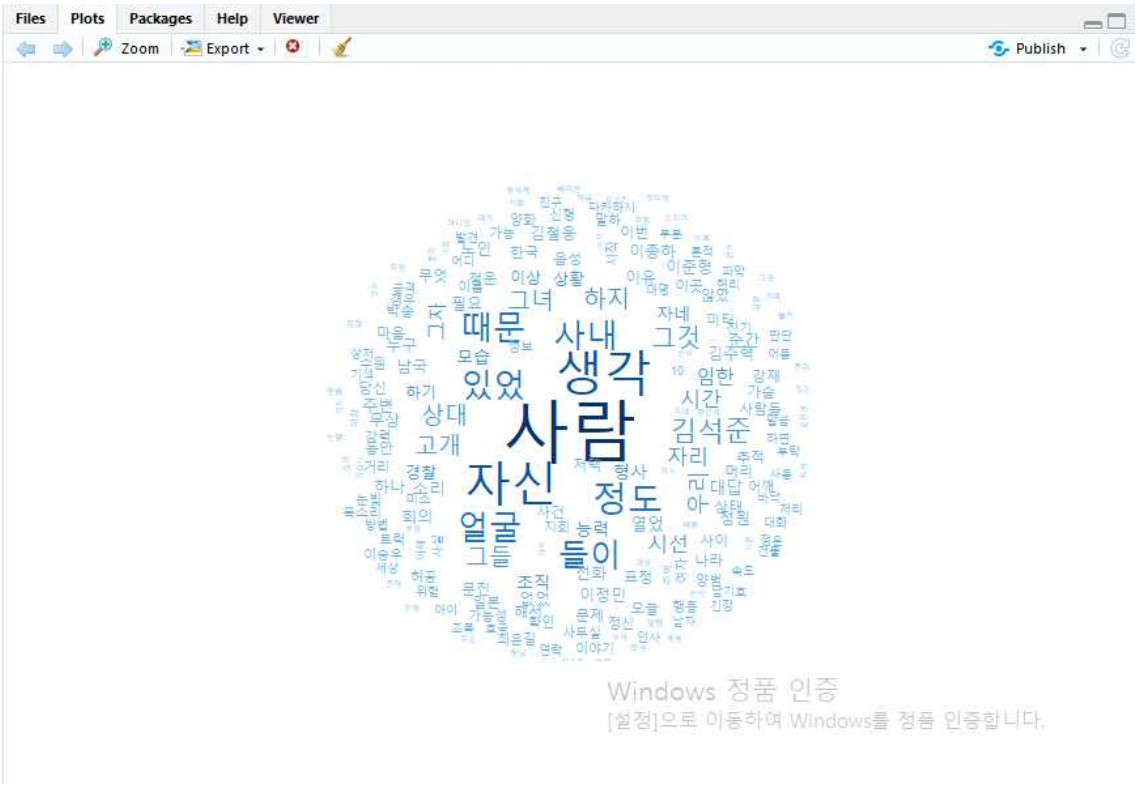
```
#20173147 이명진
#무협지 21세기 무인 워드클라우드
install.packages("Sejong")
install.packages("hash")
install.packages("rJava")
install.packages("tau")
install.packages("RSQLite")
install.packages("devtools")
install.packages("dplyr")
install.packages("KONLP")
install.packages("stringr")
install.packages("wordcloud")

library(KoNLP)
library(dplyr)
Sys.setenv(JAVA_HOME="C:/Program Files/Java/jdk1.8.0_171/")
useNIADic()

#x<-read.csv(file.choose(),header=T) #경로 지정이 어려워 특정 파일을 열수없을 때 사용 ㄱ + 데이터 저장할 장소

# 데이터 불러오기
txt <- readLines("(무협)+21세기+무인(완).txt")
head(txt)
# ## Warning in readLines("hiphop.txt"): incomplete final line found on ## 'hiphop.txt'



library(stringr)


# 특수문제 제거
txt <- str_replace_all(txt, "\\W", " ")
txt

# 가사에서 명사추출
nouns <- extractNoun(txt)

# 추출한 명사 list 를 문자열 벡터로 변환 , 단어별 빈도표 생성
wordcount <- table(unlist(nouns))

# 데이터 프레임으로 변환
df_word <- as.data.frame(wordcount, stringsAsFactors = F)
# 변수명 수정
df_word <- rename(df_word,
                  word = Var1,
                  freq = Freq)

# 두 글자 이상 단어 추출
df_word <- filter(df_word, nchar(word) >= 2)

top_20 <- df_word %>%   arrange(desc(freq)) %>%   head(20)


# 워드클라우드 만들기


# 패키지 로드
library(wordcloud)
## Loading required package: RColorBrewer
library(RColorBrewer)

#단어 색상 목록 만들기
pal <- brewer.pal(8,"Dark2")  # Dark2 색상 목록에서 8 개 색상 추출

#워드 클라우드 생성

set.seed(1234)                    # 난수 고정
wordcloud(words = df_word$word,  # 단어
          freq = df_word$freq,   # 빈도
          min.freq = 2,          # 최소 단어 빈도
          max.words = 200,       # 표현 단어 수
          random.order = F,      # 고빈도 단어 중앙 배치
          rot.per = .1,          # 회전 단어 비율
          scale = c(4, 0.3),     # 단어 크기 범위
          colors = pal)          # 색깔 목록

#단어 색상 바꾸기
pal <- brewer.pal(9,"Blues")[5:9]  # 색상 목록 생성
set.seed(1234)                     # 난수 고정


#단어 색상 바꾸기
pal <- brewer.pal(9,"Blues")[5:9]  # 색상 목록 생성
set.seed(1234)                     # 난수 고정


wordcloud(words = df_word$word,    # 단어
          freq = df_word$freq,     # 빈도
          min.freq = 2,            # 최소 단어 빈도
          max.words = 200,         # 표현 단어 수
          random.order = F,        # 고빈도 단어 중앙 배치
          rot.per = .1,            # 회전 단어 비율
          scale = c(4, 0.3),       # 단어 크기 범위
          colors = pal)            # 색상 목록

####################################### 내 자료 무협지 21세기 무인 텍스트 마이닝 끝
```

```
> library(KoNLP)
> library(dplyr)
> Sys.setenv(JAVA_HOME="C:/Program Files/Java/jdk1.8.0_171/")
> useNIADic()
Backup was just finished!
983012 words dictionary was built.
> # 데이터 불러오기
> txt <- readLines("(무협)+21세기+무인(완).txt")
> head(txt)
[1] "제  목: 21세기 무인(武人) [22 회]"
[2] "- 프롤로그 -"
[3] "이번 편은 사실상의 - 프롤로그 - 입니다. 차후에 전체공사를 할 때는 가장 앞에"
[4] "배치될 글"
[5] "입니다. 주인공이 마음속에 어떤 기억을 가지고 살아가고 있는  지를 말하고 싶어서"
[6] "뒤늦게 "
>
```

```
Console  Terminal
C:/Users/leemyeongjin/Desktop/빅데이터처리/
 [998] "볼 터였나  시댐까시든 상단미었시만 너 미상든 아니었나 "
 [999] "그만큼 점잖게 말을 했는데 영은은 자신의 마음을 받아 줄  생각미 전혀 없는"
[1000] "듯했다  간만"
 [ reached getOption("max.print") -- omitted 83944 entries ]
> # 가사에서 명사추출
> nouns <- extractNoun(txt)
> # 추출한 명사 list 를 문자열 벡터로 변환 , 단어별 빈도표 생성
> wordcount <- table(unlist(nouns))
> # 데이터 프레임으로 변환
> df_word <- as.data.frame(wordcount, stringsAsFactors = F)
> # 변수명 수정
> df_word <- rename(df_word,
+                   word = Var1,
+                   freq = Freq)
> # 두 글자 이상 단어 추출
> df_word <- filter(df_word, nchar(word) >= 2)
> top_20 <- df_word %>%   arrange(desc(freq)) %>%   head(20)
> # 패키지 로드
> library(wordcloud)
> ## Loading required package: RColorBrewer
> library(RColorBrewer)
> #단어 색상 목록 만들기
> pal <- brewer.pal(8,"Dark2")  # Dark2 색상 목록에서 8 개 색상 추출
> set.seed(1234)                # 난수 고정
> wordcloud(words = df_word$word,  # 단어
+           freq = df_word$freq,   # 빈도
+           min.freq = 2,          # 최소 단어 빈도
+           max.words = 200,       # 표현 단어 수
+           random.order = F,      # 고빈도 단어 중앙 배치
+           rot.per = .1,          # 회전 단어 비율
+           scale = c(4, 0.3),     # 단어 크기 범위
+           colors = pal)          # 색깔 목록
```

Files Plots Packages Help Viewer

Zoom Export Publish


Files
Plots
Packages
Help
Viewer
Zoom
Export
Publish
사람
생각
때문
사내
그것
있었
자신
정도
얼굴
들이
김석준
자리
시선
상대
고개
그녀
하지
조직
Windows 정품 인증
[설정]으로 이동하여 Windows를 정품 인증합니다.