

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



원광대학교
WONKWANG UNIVERSITY

<힙합 가사 텍스트 마이닝>

```
1 #20173147 이명진
2 install.packages("Sejong")
3 install.packages("hash")
4 install.packages("rJava")
5 install.packages("tau")
6 install.packages("RSQLite")
7 install.packages("devtools")
8 install.packages("dplyr")
9 install.packages("KONLP")
10 install.packages("stringr")
11 install.packages("wordcloud")
12
13 library(KONLP)
14 library(dplyr)
15
16 Sys.setenv(JAVA_HOME="C:/Program Files/Java/jdk1.8.0_171/")
17
18 useNIADic()
19
20 #x<-read.csv(file.choose(),header=T) #경로 지정이 어려워 특정 파일을 열수 없을 때 사용 ~ + 데이터 저장할 장소
21
22 # 데이터 불러오기
23 txt <- readLines("hiphop.txt")
24 head(txt)
25
26 # ## Warning in readLines("hiphop.txt"): incomplete final line found on ## 'hiphop.txt'
27
28 library(stringr)
29
30 # 특수문제 제거
31 txt <- str_replace_all(txt, "\\W", " ")
32 txt
33
34 # 명사 추출하기
35 extractNoun("대한민국의 영토는 한반도와 그 부속도서로 한다")
36 # 가사에서 명사 추출
37 nouns <- extractNoun(txt)
38
39 # 추출한 명사 list를 문자열 벡터로 변환, 단어별 빈도표 생성
40 wordcount <- table(unlist(nouns))
41
42 # 데이터 프레임으로 변환
43 df_word <- as.data.frame(wordcount, stringsAsFactors = F)
44
45
46 # 변수명 수정
47 df_word <- rename(df_word,
48                   word = var1,
49                   freq = Freq)
50
51 # 두 글자 이상 단어 추출
52 df_word <- filter(df_word, nchar(word) >= 2)
53
54 top_20 <- df_word %>%
55   arrange(desc(freq)) %>%
56   head(20)
57
58 # 워드클라우드 만들기
59
60 # 패키지 로드
61 library(wordcloud)
62 ## Loading required package: RColorBrewer
63 library(RColorBrewer)
64
65 #단어 색상 목록 만들기
66 pal <- brewer.pal(8,"Dark2") # Dark2 색상 목록에서 8 개 색상 추출
67
68 #워드 클라우드 생성
69
70 set.seed(1234)
71 wordcloud(words = df_word$word, # 단어 고정
72           freq = df_word$freq, # 빈도
73           min.freq = 2, # 최소 단어 빈도
74           max.words = 200, # 표현 단어 수
75           random.order = F, # 고정된 단어 중앙 배치
76           rot.per = .1, # 회전된 단어 비율
77           scale = c(4, 0.3), # 단어 크기 범위
78           colors = pal) # 색상 목록
79
80
81 #단어 색상 바꾸기
82 pal <- brewer.pal(9,"Blues")[5:9] # 색상 목록 생성
83 set.seed(1234) # 단어 고정
84
85
86 wordcloud(words = df_word$word, # 단어
87           freq = df_word$freq, # 빈도
88           min.freq = 2, # 최소 단어 빈도
89           max.words = 200, # 표현 단어 수
90           random.order = F, # 고정된 단어 중앙 배치
91           rot.per = .1, # 회전된 단어 비율
92           scale = c(4, 0.3), # 단어 크기 범위
93           colors = pal) # 색상 목록
94
95
96
97 ##### 힙합가사 텍스트 마이닝 끝...
98
```

```

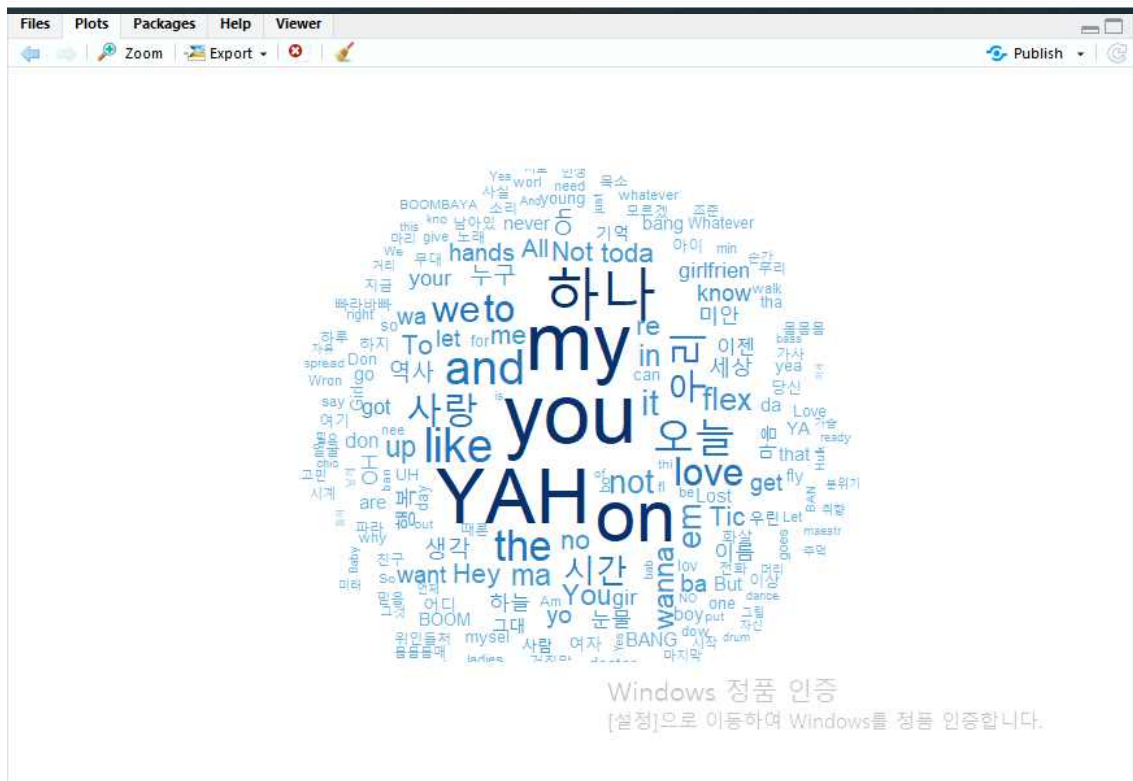
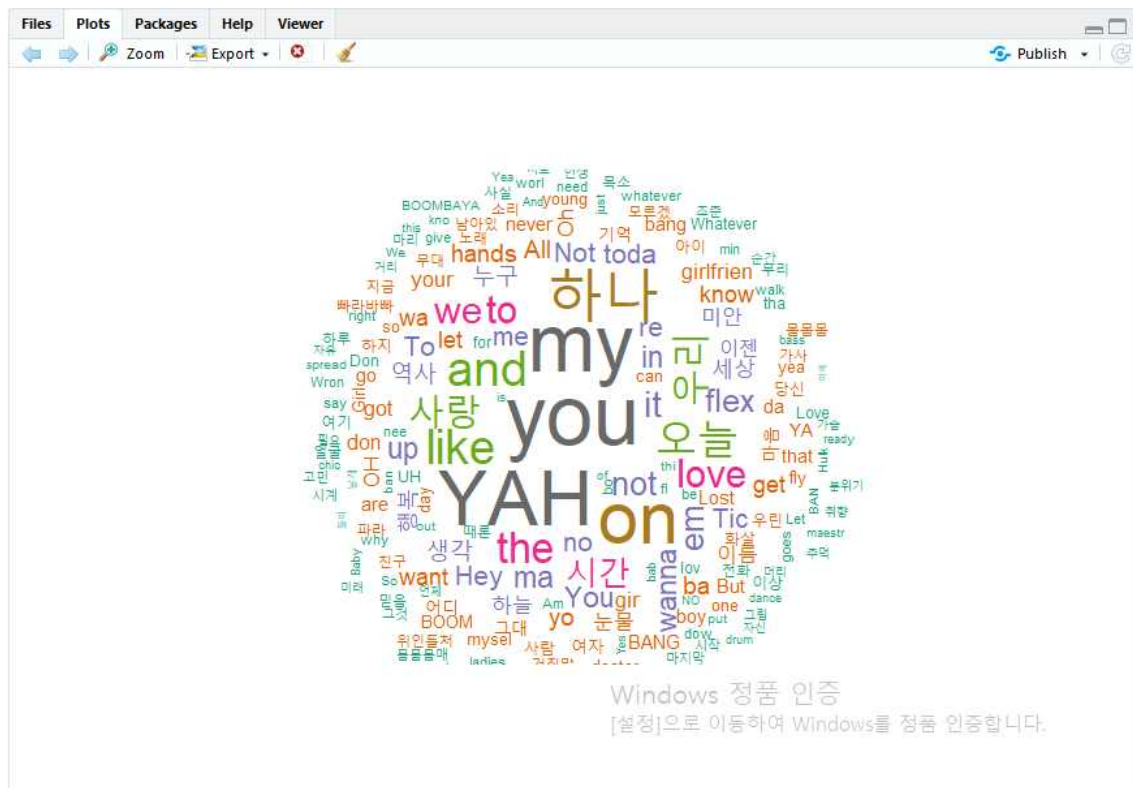
> library(koNLP)
> library(dplyr)
> Sys.setenv(JAVA_HOME="C:/Program Files/Java/jdk1.8.0_171/")
> useNIADic()
Backup was just finished!
983012 words dictionary was built.
> # 데이터 불러오기
> txt <- readLines("hiphop.txt")
warning message:
In readLines("hiphop.txt") : incomplete final line found on 'hiphop.txt'
> head(txt)
[1] "\"보고 싶다\" \"이렇게 말하니까 더 보고 싶다\" \"너희 사진을 보고 있어도\"
[4] \"보고 싶다\" \"너무 아속한 시간\" \"나는 우리가 밎다\"
> library(stringr)
> # 특수문자 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 아속한 시간" "나는 우리가 밎다"
[7] "이제 얼굴 한번 보는 것 조차" "침들어진 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 떨어져요"
[25] "보고 싶다 보고 싶다" "보고 싶다 보고 싶다"
[27] "얼마나 기다려야" "또 몇 밤을 더 새워야"

```

```

Console Terminal
C:/Users/leemyeongjin/Desktop/백대미터처리/
[ 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 을 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) # 색상 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
>
>
>

```



<국정원 트윗 텍스트 마이닝>

```
1 #20173147 이명진
2
3 # 국정원 트윗 텍스트 마이닝
4 #데이터 준비하기
5
6 install.packages("sejong")
7 install.packages("hash")
8 install.packages("rJava")
9 install.packages("tau")
10 install.packages("RSQLite")
11 install.packages("devtools")
12 install.packages("dplyr")
13 install.packages("KONLP")
14 install.packages("stringr")
15 install.packages("wordcloud")
16
17 library(KONLP)
18 library(dplyr)
19 library(stringr)
20 library(wordcloud)
21 library(RColorBrewer)
22 Sys.setenv(JAVA_HOME="C:/Program Files/Java/jdk1.8.0_171/")
23 # 데이터 로드
24 twitter <- read.csv("twitter.csv",
25                   header = T,
26                   stringsAsFactors = F,
27                   fileEncoding = "UTF-8")
28
29 # 변수명 수정
30 twitter <- rename(twitter,
31                  no = 번호,
32                  id = 계정이름,
33                  date = 작성일,
34                  tw = 내용)
35
36 # 특수문자 제거
37 twitter$tw <- str_replace_all(twitter$tw, "\\w", " ")
38
39 head(twitter$tw)
40
41 #단어 빈도표 만들기
42 # 트윗에서 명사추출
43 nouns <- extractNoun(twitter$tw)
44
45 # 추출한 명사 list 를 문자열 벡터로 변환 , 단어별 빈도표 생성
46 wordcount <- table(unlist(nouns))
47
48 # 데이터 프레임으로 변환
49 df_word <- as.data.frame(wordcount, stringsAsFactors = F)
50
51 # 변수명 수정
52 df_word <- rename(df_word, word = Var1, freq = Freq)
53
54 # 두 글자 이상으로 된 단어 추출, 빈도 상위 20개 단어 추출
55 # 두 글자 이상 단어만 추출
56 df_word <- filter(df_word, nchar(word) >= 2)
57
58 # 상위 20 개 추출
59 top20 <- df_word %>% arrange(desc(freq)) %>% head(20)
60
61 top20
62
63 #단어 빈도 막대 그래프 만들기
64 library(ggplot2)
65
66 order <- arrange(top20, freq)$word # 빈도 순서 변수 생성
67
68
69 ggplot(data = top20, aes(x = word, y = freq)) + ylim(0, 2500) + geom_col() + coord_flip() + scale_x_discrete()
70   geom_text(aes(label = freq), hjust = -0.3) # 빈도 표시
71
72 #워드 클라우드 만들기
73 pal <- brewer.pal(8,"Dark2") # 색상 목록 생성
74 set.seed(1234) # 난수 고정
75
76
77 wordcloud(words = df_word$word, # 단어
78          freq = df_word$freq, # 빈도
79          min.freq = 10, # 최소 단어 빈도
80          max.words = 200, # 표현 단어 수
81          random.order = F, # 고정된 단어 배열 배치
82          rot.per = .1, # 회전된 단어 비율
83          scale = c(6, 0.2), # 단어 크기 범위
84          colors = pal) # 색상 목록
85
86 #색깔 바꾸기
87 pal <- brewer.pal(9,"Blues")[5:9] # 색상 목록 생성
88 set.seed(1234) # 난수 고정
89
90
91 wordcloud(words = df_word$word, # 단어
92          freq = df_word$freq, # 빈도
93          min.freq = 10, # 최소 단어 빈도
94          max.words = 200, # 표현 단어 수
95          random.order = F, # 고정된 단어 배열 배치
96          rot.per = .1, # 회전된 단어 비율
97          scale = c(6, 0.2), # 단어 크기 범위
98          colors = pal) # 색상 목록
99
100 ##### 국정원 자료 워드 클라우드 끝...
```



```

Console Terminal
C:/Users/teemyeongjin/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] "탈로만 미제타도 사실은 미제환장 김정일 운구차가 링컨 컨티넨탈이던데 북한의 독재자나 우리나라 종북들이나 겉으로는 노동자 서민을 대변한다면서 고급 외제차 마이팰에 자식을 미국 유학에 환장하는 워선자들인거요"
[3] "한나라당이 보수를 버린다네요 뭔가착각하는모양인데 국민들이보수를싫어하는게 아니라빨짚거리하는분들을싫어하는겁니다당이진보여져고저쩌고한다고해서그들을조아한다고생각하면대박각"
[4] "FTA를 대하는 현명한 자세 사실 자유주의 경제의 가장 큰 수혜자는 한국이요 농어업분야 피해를 줄이는 정부대안을 최대한 보완하고 일자리 창출 등 실익을 최대화해 나가는게 현실적인 대처자세일듯"
[5] "박근혜씨 갈수록 가관입니다 뇌물질에 아들 병역 의혹까지 도대체 아이들이 뭘 보고 배우겠습니까 미래도 자리 연연하시겠습니까"
[6] "과거 집권시 한미FTA를 적극 추진하던 세력이 이제 집권하면 폐기하겠다고 주장합니다 어미없어 말도 안 나오네요 표만 얻을 수 있다면 국가 안보나 경제가 어떻게 되든 상관없다는 무책임한 행태를 우리 정치의 후진성을 드러내는 거죠"

```

```

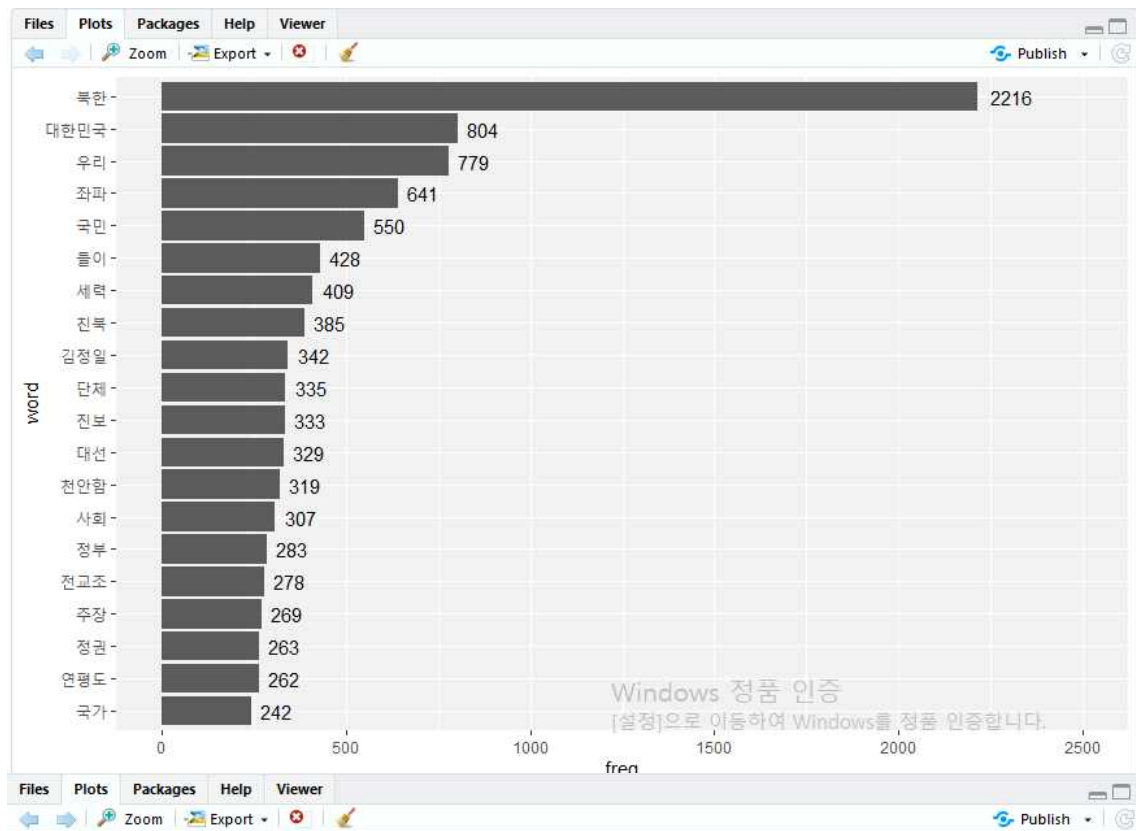
Console Terminal
C:/Users/teemyeongjin/Desktop/빅데이터처리/ <=
[6] "과거 집권시 한미FTA를 적극 추진하던 세력이 이제 집권하면 폐기하겠다고 주장합니다 어미없어 말도 안 나오네요 표만 얻을 수 있다면 국가 안보나 경제가 어떻게 되는 상관없다는 무책임한 행태를 우리 정치의 후진성을 드러내는 거죠"
> #단어 빈도표 만들기
> # 트윗에서 명사추출
> nouns <- extractNoun(twitter$tw)
> # 추출한 명사 list 를 문자열 벡터로 변환 , 단어별 빈도표 생성
> wordcount <- 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   우린 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 207

```

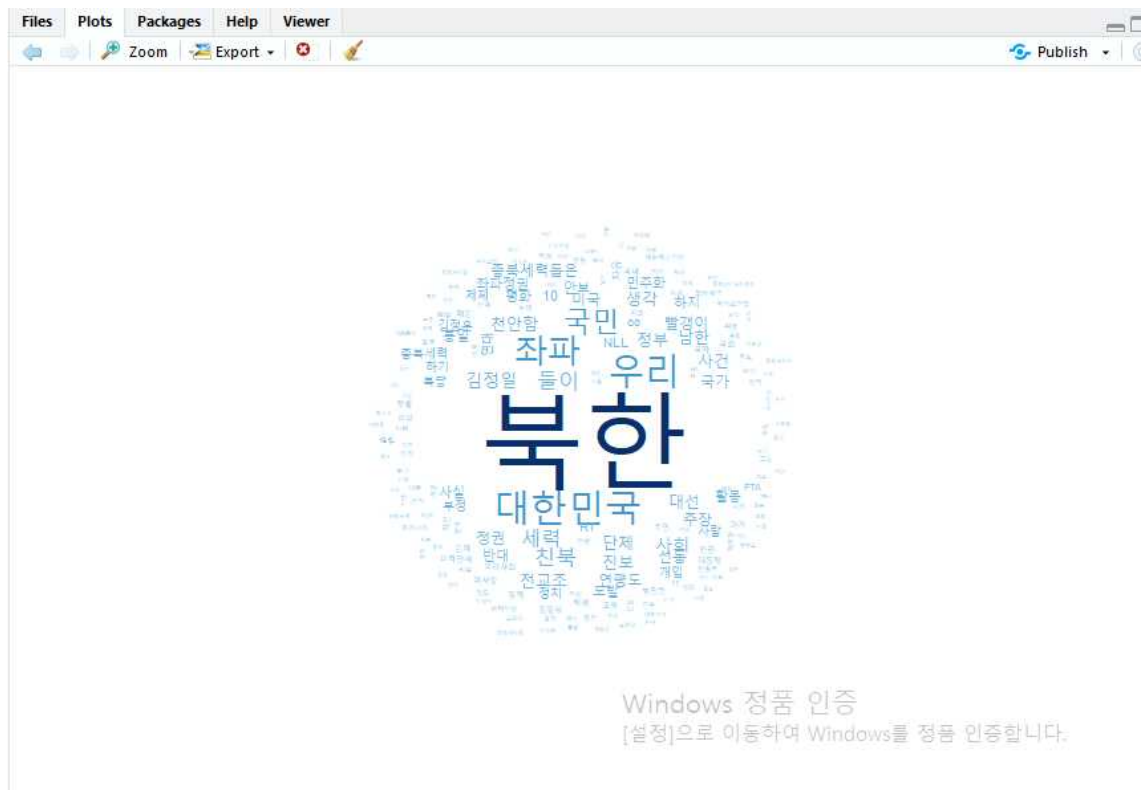
```

Console Terminal
C:/Users/teemyeongjin/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
>

```



Windows 정품 인증
[설정]으로 이동하여 Windows를 정품 인증합니다.



<무협지 21세기 무인 텍스트 마이닝>

```
1 #20173147 이명진
2 #무협지 21세기 무인 워드클라우드
3 install.packages("sejong")
4 install.packages("hash")
5 install.packages("rJava")
6 install.packages("tau")
7 install.packages("RSQLite")
8 install.packages("devtools")
9 install.packages("dplyr")
10 install.packages("KONLP")
11 install.packages("stringr")
12 install.packages("wordcloud")
13
14 library(KONLP)
15 library(dplyr)
16 Sys.setenv(JAVA_HOME="C:/Program Files/Java/jdk1.8.0_171/")
17 useNIADic()
18
19 #x<-read.csv(file.choose(),header=T) #경로 지정이 어려워 특정 파일을 열수 없을 때 사용 ~ + 데이터 저장할 장소
20
21 # 데이터 불러오기
22 txt <- readLines("(무협)+21세기+무인(완).txt")
23 head(txt)
24 # ## Warning in readLines("hiphop.txt"): incomplete final line found on ## 'hiphop.txt'
25
26
27
28 library(stringr)
29
30
31 # 특수문자 제거
32 txt <- str_replace_all(txt, "\\w", " ")
33 txt
34
35 # 가사에서 명사 추출
36 nouns <- extractNoun(txt)
37
38 # 추출한 명사 list 를 문자열 벡터로 변환 , 단어별 빈도표 생성
39 wordcount <- table(unlist(nouns))
40
41 # 데이터 프레임으로 변환
42 df_word <- as.data.frame(wordcount, stringsAsFactors = F)
43
44 # 변수명 수정
45 df_word <- rename(df_word,
46                   word = Var1,
47                   freq = Freq)
48
49 # 두 글자 이상 단어 추출
50 df_word <- filter(df_word, nchar(word) >= 2)
51
52 top_20 <- df_word %>% arrange(desc(freq)) %>% head(20)
53
54
55 # 워드클라우드 만들기
56
57
58 # 패키지 로드
59 library(wordcloud)
60 ## Loading required package: RColorBrewer
61 library(RColorBrewer)
62
63 #단어 색상 목록 만들기
64 pal <- brewer.pal(8,"dark2") # Dark2 색상 목록에서 8 개 색상 추출
65
66 #워드 클라우드 생성
67
68 set.seed(1234) # 난수 고정
69 wordcloud(words = df_word$word, # 단어
70           freq = df_word$freq, # 빈도
71           min.freq = 2, # 최소 단어 빈도
72           max.words = 200, # 표현 단어 수
73           random.order = F, # 고빈도 단어 중앙 배치
74           rot.per = .1, # 회전 단어 비율
75           scale = c(4, 0.3), # 단어 크기 범위
76           colors = pal) # 색상 목록
77
78 #단어 색상 바꾸기
79 pal <- brewer.pal(9,"blues")[5:9] # 색상 목록 생성
80 set.seed(1234) # 난수 고정
81
82
83 #단어 색상 바꾸기
84 pal <- brewer.pal(9,"blues")[5:9] # 색상 목록 생성
85 set.seed(1234) # 난수 고정
86
87
88 wordcloud(words = df_word$word, # 단어
89           freq = df_word$freq, # 빈도
90           min.freq = 2, # 최소 단어 빈도
91           max.words = 200, # 표현 단어 수
92           random.order = F, # 고빈도 단어 중앙 배치
93           rot.per = .1, # 회전 단어 비율
94           scale = c(4, 0.3), # 단어 크기 범위
95           colors = pal) # 색상 목록
96
97 ##### 내 자료 무협지 21세기 무인 텍스트 마이닝 끝 #####
98 > library(KONLP)
99 > library(dplyr)
100 > Sys.setenv(JAVA_HOME="C:/Program Files/Java/jdk1.8.0_171/")
101 > useNIADic()
102 Backup was just finished!
103 983012 words dictionary was built.
104 > # 데이터 불러오기
105 > txt <- readLines("(무협)+21세기+무인(완).txt")
106 > head(txt)
107 [1] "제 목: 21세기 무인(武人) [22 회]"
108 [2] "- 프롤로그 -"
109 [3] "이편 편은 사실상의 - 프롤로그 - 입니다. 차후에 전체공사를 할 때는 가장 앞에"
110 [4] "배치될 글"
111 [5] "입니다. 주인공이 마음속에 어떤 기억을 가지고 살아가고 있는 지를 말하고 싶어서"
112 [6] "되겠네 "
```