

## 10. 텍스트 마이닝



## 텍스트 마이닝(Text mining)

- 문자로 된 데이터에서 가치 있는 정보를 얻어 내는 분석 기법
- SNS 나 웹 사이트에 올라온 글을 분석해 사람들이 어떤 이야기를 나누고 있는지 파악할 때 활용
- 형태소 분석(Morphology Analysis) : 문장을 구성하는 어절들이 어떤 품사로 되어 있는지 분석
- 분석 절차
  - 형태소 분석
  - 명사, 동사 형용사 등 의미를 지닌 품사 단어 추출
  - 빈도표 만들기
  - 시각화

# 10-1. 힙합 가사 텍스트 마이닝

## 텍스트 마이닝 준비하기

Java 다운로드 및 설치

<https://www.java.com/ko/download/manual.jsp>

패키지 설치 및 로드 : <https://mrchypark.github.io/textR/print.html>

*# 패키지 설치*

**R.version.string**

*1. 버전에 맞는 Rtools 설치* : <https://cran.r-project.org/bin/windows/Rtools/>

*2. package 설치*

```
install.packages("rJava")  
install.packages("memoise")  
install.packages("KoNLP")
```

*# 패키지 로드*

```
library(KoNLP)
```

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

```
library(dplyr)
```

패키지 로드 에러 발생할 경우 - java 설치 경로 확인 후 경로 설정

*# java 폴더 경로 설정*

```
Sys.setenv(JAVA_HOME="C:/Program Files/Java/jre1.8.0_111/")
```

## 사전 설정하기

```
useNIADic()
```

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

## 데이터 준비

```
# 데이터 불러오기
```

```
txt <- readLines("hiphop.txt")
```

```
## Warning in readLines("hiphop.txt"): incomplete final line found on  
## 'hiphop.txt'
```

```
head(txt)
```

```
## [1] "\"보고 싶다"           "이렇게 말하니까 더 보고 싶다"  
## [3] "너희 사진을 보고 있어도" "보고 싶다"  
## [5] "너무 야속한 시간"      "나는 우리가 밉다"
```

## 특수문자 제거

```
install.packages("stringr") # 문자처리 패키지
```

```
library(stringr)
```

*# 특수문제 제거*

```
txt <- str_replace_all(txt, "\\W", " ") # \\W 특수문자를 나타내는 정규표현식
```

참고: \\w : 단어를 만들 수 있는 글자, not \\w == \\W

## 가장 많이 사용된 단어 알아보기

```
# 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))
```

## 자주 사용된 단어 빈도표 만들기

*# 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)
```



top\_20

##		word	freq
## 1		you	89
## 2		my	86
## 3		YAH	80
## 4		on	76
## 5		하나	75
## 6		오늘	51
## 7		and	49
## 8		사랑	49
## 9		like	48
## 10		우리	48
## 11		the	43
## 12		시간	39
## 13		love	38
## 14		to	38
## 15		we	36
## 16		it	33
## 17		em	32
## 18		not	32
## 19		역사	31
## 20		flex	30

# 워드 클라우드 만들기

## 패키지 준비하기

```
# 패키지 설치
install.packages("wordcloud")

# 패키지 로드
library(wordcloud)

## Loading required package: RColorBrewer
library(RColorBrewer)
```

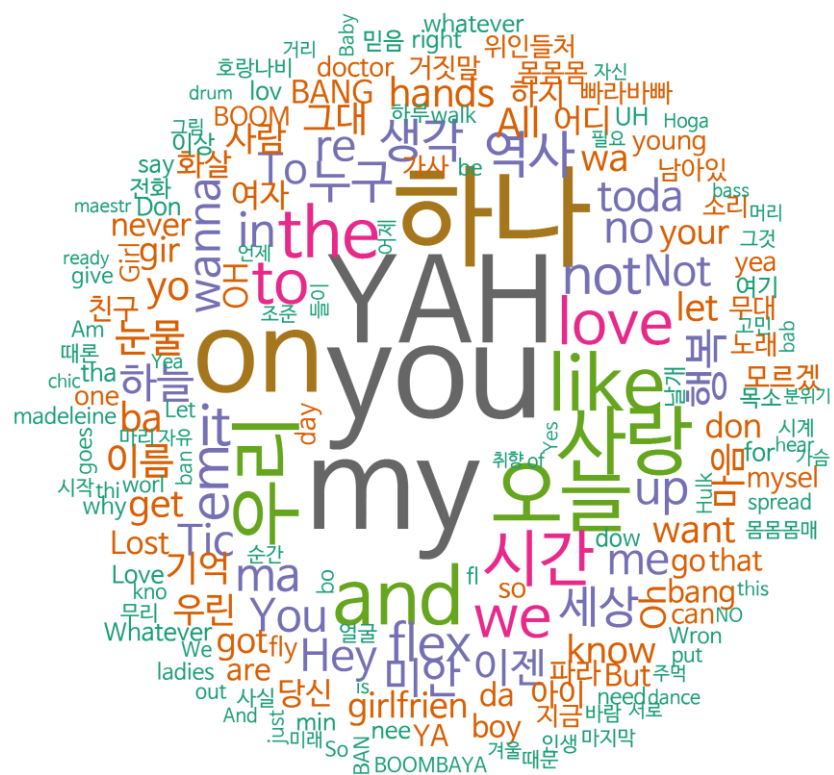
# 워드 클라우드 만들기

## 단어 색상 목록 만들기

```
pal <- brewer.pal(8, "Dark2") # Dark2 색상 목록에서 8개 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
```



## 10-2. 국정원 트윗 텍스트 마이닝

- 국정원 계정 트윗 데이터
  - 국정원 대선 개입 사실이 밝혀져 논란이 됐던 2013년 6월, 독립 언론 뉴스타파가 인터넷을 통해 공개
  - 국정원 계정으로 작성된 3,744개 트윗

# 국정원 트윗 텍스트 마이닝

## 데이터 준비하기

*# 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 ISD 관련 주장이 전부 거짓으로 속속 드러나고있다 미국이 ISD 를 장악하고 있다고 주장하지만 중재인 123 명 가운데 미국인은 10 명뿐이라고 한다 "

## [2] "말로만 미제타도 사실은 미제환장 김정일 운구차가 링컨 컨티넨탈이던데 북한의 독재자나 우리나라 종북들이나 겉으로는 노동자 서민을 대변한다면서 고급 외제차 아이팟에 자식들 미국 유학에 환장하는 위선자들인거죠"

## [3] "한나라당이 보수를 버린다네요 뭔가착각하는모양인에 국민들이보수를싫어하는게 아니라빨짓거리하는분들을싫어하는겁니다야당이진보어쩌고저쩌고한다고해서그들을조아한다고생각하면대착각"

## [4] "FTA 를 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 아들 병역 의혹까지 도대체 아이들이 뭘 보고 배우겠습니까 이래도 자리 연연하시겠습니까 "

## [6] "과거 집권시 한미 FTA 를 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 20개 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
## 15		정부	283
## 16		전교조	278
## 17		주장	269

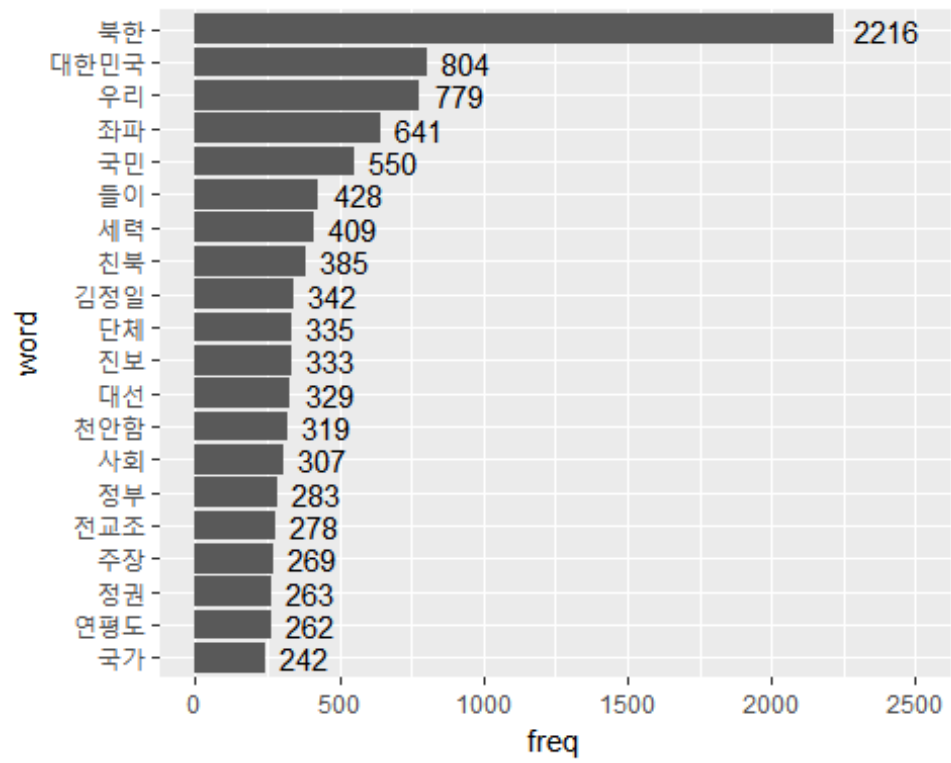
## 18	정권	263
## 19	연평도	262
## 20	국가	242

## 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
```

