Chapter 1

텍스트마이닝

텍스트 마이닝 이란

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

설치 패키지

패키지 설치 및 로드(기존 방식에 설치 이슈 문제가 있습니다)

기존의 사용하던 일방적인 방식

```
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/")
```

기존 방식에 설치 이슈 문제가 있습니다

설치 패키지

github를 이용한 패키지 설치 및 로드

KoNLP설치 문제가 있어서 아래같은 방식으로 다운을 진행합니다.
참조사이트
https://www.facebook.com/notes/r-korea-krugkorean-r-user-group/konlp-%EC%84%A4%EC%B9%98-%EC%9D%B4%EC%8A%88-%EA%B3%B5%EC%9C%A0/1847510068715020/

```
install.packages("rJava")
install.packages("multilinguer")
library(multilinguer)

install_jdk()
# 의존성 패키지 설치
install.packages(c('stringr', 'hash', 'tau', 'Sejong', 'RSQLite', 'devtools'), type = "binary")
# github 버전 설치
install.packages("remotes")
# 64bit 에서만 동작합니다.
remotes::install_github('haven-jeon/KoNLP', upgrade = "never", INSTALL_opts=c("--no-multiarch"))

library(KoNLP) #확인 -> 없는 패키지 설치후 재 업로드

useNIADic() #사전 업로드
```

분석절차

1. 데이터 준비하기

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

head(txt)

## 결과…
```

2. 특수문자 제거하기

```
# stringr패키지로 특수문자제거 (W는 특수문자를 의미)
library(stringr)
hiphop <- str_replace_all(hiphop, "\\W", " ") # R정규표현식 특수문자를 의미
hiphop
```

3. 명사추출 함수 사용하기

```
# extractNoun은 명사를 추출해서 리스트 형태로 반환합니다.
hip_list <- extractNoun(hiphop)
hip_list
```

3. unlist함수로 vector형변환

```
# unlist함수는 list의 인자값을 vector의 형태로 반환하는 자주사용되는 함수
hip_vec <- unlist(hip_list)
hip_vec
```

분석절차

4. 데이터프레임으로 변환

```
# vetor를 데이터 프레임으로 (변수에 문자가 있을때 factor로 자동변환되는데 이를 방지함)
hip_df <- as.data.frame(hip_count, stringsAsFactors = F)
hip_df
```

5. 컬럼변경

```
# 이름변경 rename(data, 변경컬럼=기존컬럼)
hip_df <- rename(hip_df, word = hip_vec, freq = Freq )
head(hip_df)
```

6. 특정 글자수 빈도 출력

```
# 2글자 이상 자주사용된 단어 출력
library(dplyr)
hip_df <- hip_df %>%
  filter( nchar(word) >= 2 ) %>%
  arrange( desc(freq) )

hip_df
```

워드클라우드 만들기

워드클라우드 생성하기

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

# 색상목록을 추출하는 기능
color <- brewer.pal(8, "Accent")
color

# 난수 고정
set.seed(1234)

# 워드클라우드 함수
wordcloud(words = hip_df$word, #단어
          freq = hip_df$freq, #빈도
          min.freq = 2, #최소 단어 빈도
          max.words = 200, #표현 단어 수
          random.order = F, #고빈도 단어 중앙배치(F는 중앙배치)
          rot.per = .1, #회전 단어 비율
          scale = c(4, 0.3), #단어 크기 범위 (가운데, 끝)
          colors = color) #색상목록록
```

Chapter 1

수고하셨습니다