Report

빅데이터 입문 과제

(아프리카 돼지열병에 관한 빅데이터 분석)

■ 과　　목 : 빅데이터 입문
■ 담당교수 : 임동훈 교수님
■ 학　　과 : 정보통계학과
■ 학　　번 : 2015013166
■ 이　　름 : 송 연 석
■ 제 출 일 : 2019년 11월 14일

경상대학교
GYEONGSANG NATIONAL UNIVERSITY

목차

1. 서론

아프리카 돼지열병(African Swine Fever, ASF)이란 바이러스에 의해 발생하는 돼지 전염병이다. 치료제나 백신이 없어 급성형인 경우에 치사율이 최대 100%에 이르는 전염병이다. 현재 유럽 15개국, 아프리카 29개국, 아시아 9개국 등 전 세계 53개국에서 발생중이다. 가까운 나라로는 첫 번째 발생국인 중국을 포함하여 몽골, 베트남, 캄보디아, 라오스, 미얀마, 필리핀, 북한 등이 포함되어 있다. 우리나라에선 2019년 9월 16일 경기도 파주시 소재 양돈농장을 시작으로 10월 9일까지 14개 농장에서 발생하였다. 아프리카돼지열병 확산을 막기 위해 일시이동중지명령을 발령을 내렸었고, 현재 강도 높은 방역이 진행 중이다.

아프리카돼지열병 바이러스는 낮은 온도에서도 안정적으로 생존한다고 알려져 있고, 심지어 냉동육, 소금에 절인 고기, 부패된 혈액 등에서도 장기간 생존이 가능한 것으로 과학계에서 파악하고 있다. 돼지에게는 감염이 되지만, 사람에게는 감염이 되지 않는다. 농림축산식품부에 의하면 기본적으로 아프리카돼지열병에 감염된 돼지는 전량 살처분·매몰 처리되며, 이상이 있는 축산물의 경우 국내로 유통되지 않는 만큼 안심하고 돼지고기를 소비해도 된다고 하였다.

사람에게는 전염이 되지 않음에도 불구하고 아프리카돼지열병 발생에 따른 돼지고기 소비침체 및 가격하락으로 양돈농가의 어려움이 가중되고 있어 정부 및 기관 등에서 돼지고기 소비촉진 행사와 피해농가 성금을 전달하고 있다.

이러한 소식을 듣고, 아프리카 돼지열병에 관한 생각과 아프리카 돼지열병으로 인한 돼지고기 소비에 대한 사람들의 생각이 어떤지 확인하기 위해 워드클라우드와 감성분석을 진행해 보려고 한다.

2. R코드 전문

```
# 트위터로 데이터 가지고 오기

setwd('C:/Users/XPS/Desktop/대학교 자료/공부/대학교/정보통계학과/3-2학기/빅데이터입문/기말과제') # 작업 디렉토리 지정

install.packages("twitteR")
install.packages("KoNLP")
install.packages("wordcloud")
install.packages("stringr")
install.packages("digest") # KoNLP 사용 위해 설치

library(twitteR)
library(stringr)
library(KoNLP)
library(wordcloud)

source("authenticate.R") # 토큰 인증

# 인코딩
keyword1 <- enc2utf8("돼지열병")

# 키워드 : 돼지열병
disease <- searchTwitter(keyword1, n = 3200, lang="ko")
                                        # 돼지열병에 대해 표본 3200개를 한글로 뽑기

df_disease <- do.call("rbind", lapply(disease, as.data.frame))
                                        # 돼지열병 데이터프레임

# 데이터 저장
write(df_disease$text, '돼지열병 검색어 데이터.txt')
                                        # 돼지열병 데이터프레임 txt파일로 저장
write(df_disease$text, '돼지열병 검색어 데이터.csv')
                                        # 돼지열병 데이터프레임 csv파일로 저장
```

```
disease.text <- gsub("(RT|via)((?:₩₩₩*@₩₩+)+)", "", df_disease$text)
                                                  # 리트윗 제거
disease.text <- gsub("http₩₩+", "", disease.text) # 링크 제거
disease.text <- gsub("@[a-z]*", "", disease.text)
                                          # @로 시작하는 영어소문자 0개 이상을 제거
disease.text <- gsub("&[a-z]*", "", disease.text)
                                          # &로 시작하는 영어소문자 0개 이상을 제거
disease.text <- gsub("#[a-z]*", "", disease.text)
                                          # #로 시작하는 영어소문자 0개 이상을 제거
disease.text <- gsub("RT", "", disease.text)      # RT 제거



useNIADic()  # 사전 불러오기

# 전처리 과정
dis_words <- sapply(disease.text, extractNoun, USE.NAMES=F)  # 명사만 추출
dis_words <- unlist(dis_words)
dis_words <- gsub("[[:punct:]]", "", dis_words)                  # 구두점 지우기
dis_words <- gsub("[^ [:alnum:][:blank:]?&/₩ ]", "", dis_words) # 유니코드 제거
dis_words <- gsub(keyword1, "", dis_words) # 키워드 지우기
dis_words <- gsub("₩d+", "", dis_words)     # 숫자 지우기
dis_words <- gsub("[A-z]", "", dis_words)    # 모든 영문자 지우기
dis_words <- gsub("▶*", "", dis_words)      # ▶로 시작되는 것들 지우기
dis_words <- gsub("ð*", "", dis_words)       # ð로 시작되는 거들 지우기
dis_words <- gsub("아프리카+|열병", "", dis_words)
dis_words <- gsub("돼지", "", dis_words)
dis_words <- gsub("가고|가능|가에", "", dis_words)
dis_words <- gsub("감역", "감염", dis_words)
dis_words <- gsub("강춘혁탈북래", "", dis_words)
dis_words <- gsub("개더러웠음|개쓰레기", "개지랄", dis_words)
dis_words <- gsub("거기", "", dis_words)
dis_words <- gsub("갱기", "", dis_words)
dis_words <- gsub("걸려뒈질|걸렸냐니|걸렸냐|걸린거", "걸렸나", dis_words)
dis_words <- gsub("검색", "검역", dis_words)
```

```
dis_words <- gsub("결과", "결국", dis_words)
dis_words <- gsub("경기지사이해찬|경기지사·이해찬", "이해찬", dis_words)
dis_words <- gsub("계엄령", "", dis_words)
dis_words <- gsub("고생많으셨습니다", "고생", dis_words)
dis_words <- gsub("관심을", "관심", dis_words)
dis_words <- gsub("구닌들", "군인들", dis_words)
dis_words <- gsub("국립", "국내", dis_words)
dis_words <- gsub("군데", "군부대", dis_words)
dis_words <- gsub("굿모닝|굿모닝하우스|굿모닝하우스나", "", dis_words)
dis_words <- gsub("기레", "기레기", dis_words)
dis_words <- gsub("기적이닷", "기적", dis_words)
dis_words <- gsub("꼬락서니봐라|꼴값", "꼬라지", dis_words)
dis_words <- gsub("꼴배", "", dis_words)
dis_words <- gsub("난리났을때는", "난리", dis_words)
dis_words <- gsub("났을때는경기도만", "경기도", dis_words)
dis_words <- gsub("농가돕기|농림", "농가", dis_words)
dis_words <- gsub("누구|니네|ㄷㅈㅇ|다하겠습니", "", dis_words)
dis_words <- gsub("대국민", "국민", dis_words)
dis_words <- gsub("더부", "더불어민주당", dis_words)
dis_words <- gsub("동안|들이|마리", "", dis_words)
dis_words <- gsub("못됐음|못한새끼야", "못됐고", dis_words)
dis_words <- gsub("문푸정국이라|문푸정부가", "문푸정부", dis_words)
dis_words <- gsub("뭐길래|뭐쩌라는거야|뭔지|뭘까|뭣때문에", "뭔데", dis_words)
dis_words <- gsub("민원서비스", "민원", dis_words)
dis_words <- gsub("발생지역", "발생", dis_words)
div_words <- gsub("부전프라임뉴스이재명", "이재명", dis_words)

#
dis_words <- dis_words[nchar(dis_words) >= 2]
dis_words_table <- table(dis_words)
dis_words_table <- head(sort(dis_words_table, decreasing=T), 300)
```

```
# 최다 빈출되는 15개의 단어에 대한 빈도별 그래프 그리기
dis_copy <- dis_words
dis_copy <- table(dis_words)
dis_copy <- head(sort(dis_copy, decreasing=T), 15)
dis_copy <- as.data.frame(dis_copy)

library(ggplot2)
ggplot(dis_copy, aes(dis_copy$dis_words, dis_copy$Freq)) +
        ggtitle("최다 빈출 단어 빈도별 그래프") +
        theme(plot.title = element_text(colour = "blue", face = "bold", size = 20,
hjust = 0.5))+
        labs(x = "단어", y = "빈도수") +
        geom_bar(color = "black", fill = 'skyblue', stat = "identity")

# 빈도별 그래프 저장하기
ggsave("bar graph.jpg", dpi = 300)

#display.brewer.all() # 색상 확인
pair <- brewer.pal(5, "Paired")
windowsFonts(namsan=windowsFont("서울남산체 M")) # 서울남산체 M으로 지정함.
set.seed(1234)
wordcloud(words = names(dis_words_table), freq=dis_words_table, scale=c(5, 0.5),
                              colors = pair, min.freq=20, random.order=F,
                              family='namsan')

# 워드클라우드2 만들기
install.packages("wordcloud2")
library(wordcloud2)
dis_cloud = wordcloud2(dis_words_table, size = 1, backgroundColor = "white")
dis_cloud
```

```
# 워드클라우드2 저장
install.packages("htmlwidgets")
library(htmlwidgets)
saveWidget(dis_cloud, "아프리카 돼지열병.html", selfcontained = F)
                                                    # html파일로 저장

# 감성분석하기

library(twitteR)
library(plyr)
library(stringr)

source("authenticate.R") # 토큰 인증

score.sentiment = function(sentences, pos.words, neg.words)
{
   scores = laply(sentences,
   function(sentence, pos.words, neg.words)
   {
      sentence = gsub("[[:punct:]]", "", sentence) # 문장부호 제거
      sentence = gsub("[[:cntrl:]]", "", sentence)  # 특수문자 제거
      sentence = gsub('₩d+', '', sentence)          # 숫자 제거
      word.list = strsplit(sentence, "₩s+")    # 문장을 '빈칸'으로 나눔
                                          # ₩s+ : 빈칸 1칸 이상을 의미함.
      words = unlist(word.list)
      pos.matches = match(words, pos.words)
                                       # words의 단어를 positive에서 맞춘다.
      neg.matches = match(words, neg.words)
                                       # words의 단어를 negative에서 맞춘다.
      pos.matches = !is.na(pos.matches)     # NA 제거함. 위치(숫자)만 추출함.
      neg.matches = !is.na(neg.matches)
      score = sum(pos.matches) - sum(neg.matches)
                                             # score = 긍정점수 - 부정점수
      return(score)                          # score값 반환
```

```
    }, pos.words, neg.words)
  scores.df = data.frame(text=sentences, score=scores)
                                  # 각각의 문장과 점수를 데이터프레임으로 변환
  return(scores.df)
}                                  # 문장을 감성 점수를 측정하는 함수 생성

# 군산대에서 만든 감성사전에 있는 긍정 사전과 부정 사전을 변수에다 저장한다.
pos.words <- readLines("pos_pol_word.txt", encoding = "UTF-8")  # 긍정 사전
neg.words <- readLines("neg_pol_word.txt", encoding = "UTF-8")  # 부정 사전

# 아까 분석했었던 트위터 데이터를 사용한다.
disease_txt <- sapply(disease, function(x) x$getText(), USE.NAMES=F)
write.csv(disease_txt, "돼지열병 트위터 내용.txt")

disease.score <- score.sentiment(disease_txt, pos.words, neg.words)
table(disease.score$score)
mean(disease.score$score)

# qplot 만들기
library(ggplot2)
qplot(disease.score$score, xlab = "감성 점수", ylab = "개수")+
      geom_bar(color = 'black', fill = "skyblue")

# qplot 저장하기
ggsave(file = "C:/Users/XPS/Desktop/대학교 자료/공부/대학교/정보통계학과/3-2학기/빅
데이터입문/기말과제/histogram.jpg",
             width = 3.5, height = 5)

## 돼지고기 가격에 대하여 감성분석 실시하기
# rbind로 합친다.

word1 <- enc2utf8("돼지고기 가격")
word2 <- enc2utf8("삼겹살 가격")
```

```
price1 <- searchTwitter(word1, n = 500, lang="ko")
price2 <- searchTwitter(word2, n = 500, lang="ko")

df_price1 <- do.call("rbind", lapply(price1, as.data.frame))
df_price2 <- do.call("rbind", lapply(price2, as.data.frame))
price <- rbind(df_price1, df_price2)



# 데이터 저장
write.csv(price$text, "돼지고기 가격 검색어 데이터.txt")

# 감성분석 실시
price.score <- score.sentiment(price$text, pos.words, neg.words)

table(price.score$score)
mean(price.score$score)

# qqplot 쓰기
library(ggplot2)
qplot(price.score$score, xlab = "감성 점수", ylab = "개수")+
        geom_bar(color = 'black', fill = "skyblue")

# xlab, ylab으로 x축, y축 이름 정해줌
# 그래프는 테두리가 검은색인 파란색 히스토그램을 그렸다.

# qplot 저장하기
ggsave(file = "C:/Users/XPS/Desktop/대학교 자료/공부/대학교/정보통계학과/3-2학기/빅
데이터입문/기말과제/bargraph3.jpg",
               width = 3.5, height = 5)
```

3. 본론

1) 아프리카 돼지열병에 관한 워드클라우드 작성

- 먼저 작업 디렉토리를 설정한 후, 이용할 패키지를 설치 및 실행시킨다.

```
setwd('C:/Users/XPS/Desktop/대학교 자료/공부/대학교/정보통계학과/3-2학기/빅데이터입문/기말과제') # 작업 디렉토리 지정

install.packages("twitteR")
install.packages("KoNLP")
install.packages("wordcloud")
install.packages("stringr")
install.packages("digest") # KoNLP 사용 위해 설치

library(twitteR)
library(stringr)
library(KoNLP)
library(wordcloud)
```

- 트위터 api 토큰을 인증받은 후, searchTwitter로 키워드를 검색 후 데이터를 수집한다. 한글은 따로 인코딩하여 UTF-8로 변환 시켜야 한다.

```
source("authenticate.R") # 토큰 인증

keyword1 <- enc2utf8("돼지열병") # 키워드를 인코딩 시킨다.
# 키워드 : 돼지열병
disease <- searchTwitter(keyword1, n = 3200, lang="ko")
                                        # 돼지열병에 대해 표본 3200개를 한글로 뽑기

df_disease <- do.call("rbind", lapply(disease, as.data.frame))
                                        # 돼지열병 데이터프레임으로 변환
# 데이터 저장
write(df_disease$text, '돼지열병 검색어 데이터.txt')
                                        # 돼지열병 데이터프레임 txt파일로 저장
write(df_disease$text, '돼지열병 검색어 데이터.csv')
                                        # 돼지열병 데이터프레임 csv파일로 저장
```

- 데이터를 수집 후에 필요없는 부분을 제거하기 위해 gsub()와 extractNoun을 이용하여 전처리 과정을 진행해야 한다.

```
disease.text <- gsub("(RT|via)((?:₩₩₩₩*@₩₩+)+)", "", df_disease$text)
                                                  # 리트윗 제거
disease.text <- gsub("http₩₩+", "", disease.text) # 링크 제거
disease.text <- gsub("@[a-z]*", "", disease.text)
                                    # @로 시작하는 영어소문자 0개 이상을 제거
disease.text <- gsub("&[a-z]*", "", disease.text)
                                    # &로 시작하는 영어소문자 0개 이상을 제거
disease.text <- gsub("#[a-z]*", "", disease.text)
                                    # #로 시작하는 영어소문자 0개 이상을 제거
disease.text <- gsub("RT", "", disease.text)      # RT 제거


useNIADic()  # 사전 불러오기

# 전처리 과정
dis_words <- sapply(disease.text, extractNoun, USE.NAMES=F)  # 명사만 추출
dis_words <- unlist(dis_words)
dis_words <- gsub("[[:punct:]]", "", dis_words)                  # 구두점 지우기
dis_words <- gsub("[^ [:alnum:][:blank:]?&/₩₩]", "", dis_words) # 유니코드 제거
dis_words <- gsub(keyword1, "", dis_words) # 키워드 지우기
dis_words <- gsub("₩₩+", "", dis_words)     # 숫자 지우기
dis_words <- gsub("[A-z]", "", dis_words)    # 모든 영문자 지우기
dis_words <- gsub("▶*", "", dis_words)      # ▶로 시작되는 것들 지우기
dis_words <- gsub("ð*", "", dis_words)      # ð로 시작되는 것들 지우기
dis_words <- gsub("아프리카+|열병", "", dis_words)
dis_words <- gsub("돼지", "", dis_words)
dis_words <- gsub("가고|가능|가에", "", dis_words)
dis_words <- gsub("감역", "감염", dis_words)
dis_words <- gsub("강춘혁탈북래", "", dis_words)
dis_words <- gsub("개더러웠음|개쓰레기", "개지랄", dis_words)
dis_words <- gsub("거기", "", dis_words)
```

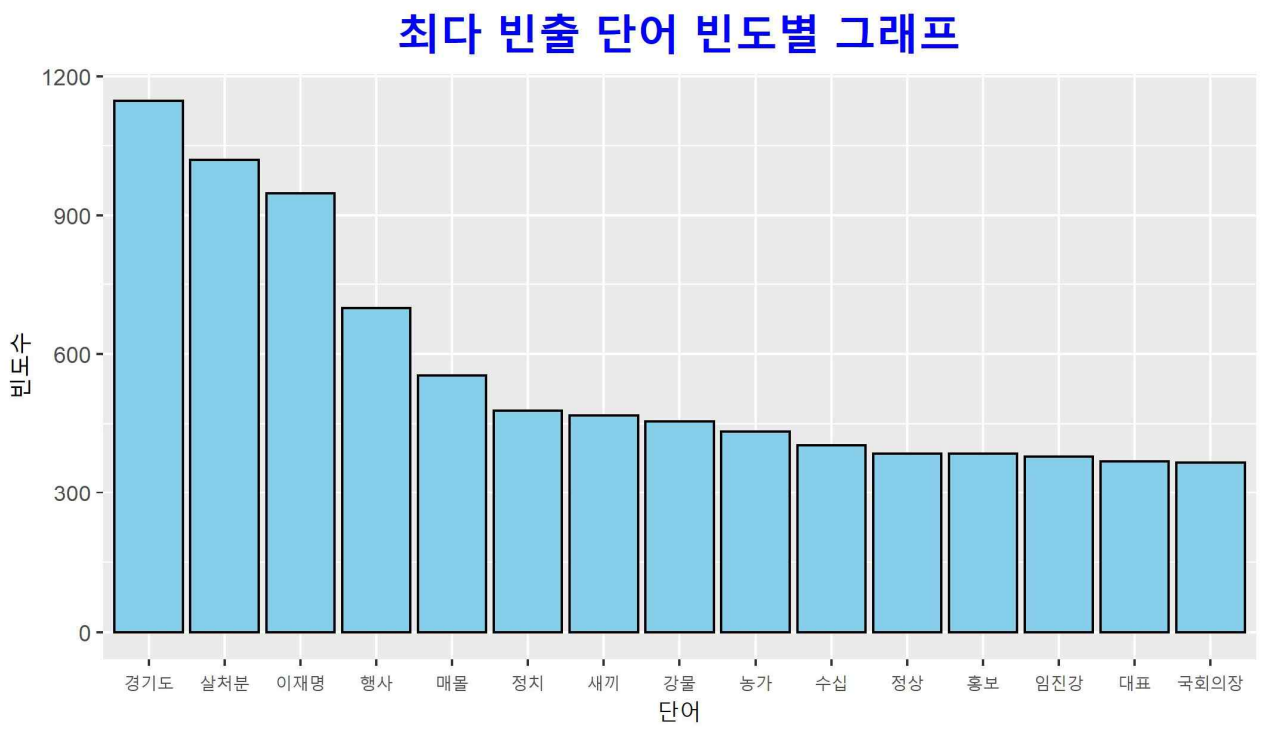
```
dis_words <- gsub("갱기", "", dis_words)
dis_words <- gsub("걸려뭬질|걸렸냐니|걸렸냐|걸린거", "걸렸나", dis_words)
dis_words <- gsub("검색", "검역", dis_words)
dis_words <- gsub("결과", "결국", dis_words)
dis_words <- gsub("경기지사이해찬|경기지사·이해찬", "이해찬", dis_words)
dis_words <- gsub("계엄령", "", dis_words)
dis_words <- gsub("고생많으셨습니다", "고생", dis_words)
dis_words <- gsub("관심을", "관심", dis_words)
dis_words <- gsub("구닌들", "군인들", dis_words)
dis_words <- gsub("국립", "국내", dis_words)
dis_words <- gsub("군데", "군부대", dis_words)
dis_words <- gsub("굿모닝|굿모닝하우스|굿모닝하우스나", "", dis_words)
dis_words <- gsub("기레", "기레기", dis_words)
dis_words <- gsub("기적이닷", "기적", dis_words)
dis_words <- gsub("꼬락서니봐라|꼴값", "꼬라지", dis_words)
dis_words <- gsub("꼴배", "", dis_words)
dis_words <- gsub("난리났을때는", "난리", dis_words)
dis_words <- gsub("났을때는경기도만", "경기도", dis_words)
dis_words <- gsub("농가돕기|농림", "농가", dis_words)
dis_words <- gsub("누구|니네|ㄷㅈㅇ|다하겠습니", "", dis_words)
dis_words <- gsub("대국민", "국민", dis_words)
dis_words <- gsub("더부", "더불어민주당", dis_words)
dis_words <- gsub("동안|들이|마리", "", dis_words)
dis_words <- gsub("못됐음|못한새끼야", "못됐고", dis_words)
dis_words <- gsub("문푸정국이라|문푸정부가", "문푸정부", dis_words)
dis_words <- gsub("뭐길래|뭐쩌라는거야|뭔지|뭘까|뭣때문에", "뭔데", dis_words)
dis_words <- gsub("민원서비스", "민원", dis_words)
dis_words <- gsub("발생지역", "발생", dis_words)
div_words <- gsub("부전프라임뉴스이재명", "이재명", dis_words)
```

- 그 후에 명사 중에서 2개 글자이상만 검색하도록 지정한 후 table()함수를 이용하여 단어 빈도분석을 하고, sort()함수를 사용하여 단어의 사용빈도를 내림차순으로 정렬한다. 단어들이 많으므로 단어 중 head()를 이용하여 제일 많이 사용한 300개를 뽑아서 변수에 저장한다.

```
dis_words <- dis_words[nchar(dis_words) >= 2] # 2개 글자 이상만 검색
dis_words_table <- table(dis_words)            # 단어 빈도분석
dis_words_table <- head(sort(dis_words_table, decreasing=T), 300)
                                # 사용빈도를 내림차순으로 정렬 후 제일 많이 사용한
                                # 300개를 뽑는다.
```

- 300개 중에서 제일 많이 사용하는 단어의 순위를 시각적으로 표시하기 위해 최다빈출 단어 15개에 대하여 빈도별 그래프를 그린다. 그 후에 저장한다.

```
# 최다 빈출되는 15개의 단어에 대한 빈도별 그래프 그리기
dis_copy <- dis_words                          # 변수를 복사함
dis_copy <- table(dis_words)
dis_copy <- head(sort(dis_copy, decreasing=T), 15) # 내림차순으로 정렬 후
                                               # 최다빈출 단어 15개 뽑는다.
dis_copy <- as.data.frame(dis_copy)             # 데이터프레임 형태로 변환

library(ggplot2)
ggplot(dis_copy, aes(dis_copy$dis_words, dis_copy$Freq)) +
        ggtitle("최다 빈출 단어 빈도별 그래프") +
        theme(plot.title = element_text(colour = "blue", face = "bold", size = 20,
hjust = 0.5))+
        labs(x = "단어", y = "빈도수") +
        geom_bar(color = "black", fill = 'skyblue', stat = "identity")

# ggplot()의 aes에 x축(단어)과 y축(빈도)을 지정하고 제목을 지정하였다.
# 그래프 제목을 ggtitle()으로 지정하고, theme()를 이용하여 제목을 가운데 정렬, 글씨
진하게, 크기를 20, 파란색 글씨로 지정하였다.
# geom_bar()을 이용하여 검은색 테두리의 하늘색 그래프를 그리게 하고, stat요소에
identity를 적용함으로써 데이터프레임 값을 그대로 사용해 그래프를 그리게 하였다.

# 빈도별 그래프 저장하기
ggsave("bar graph.jpg", dpi = 300)
```



- 워드클라우드를 작성한다. 서울남산체를 다운받은 후, 서울남산체 M 글씨체를 이용하였고, display.brewer.all()에서 찾은 'Paired'색상을 사용하였다.

```
#display.brewer.all() # 색상 확인

pair <- brewer.pal(5, "Paired")  # 색상 저장
windowsFonts(namsan=windowsFont("서울남산체 M")) # 서울남산체 M으로 지정함.
set.seed(1234) # 난수 고정 -> 그래프 형태 고정시킨다.
wordcloud(words = names(dis_words_table), freq=dis_words_table, scale=c(5, 0.5),
                              colors = pair, min.freq=20, random.order=F,
                              family='namsan')

# 워드클라우드2 만들기
install.packages("wordcloud2")
library(wordcloud2)
dis_cloud = wordcloud2(dis_words_table, size = 1, backgroundColor = "white")
dis_cloud
```

```
# 워드클라우드2 저장
install.packages("htmlwidgets")
library(htmlwidgets)
saveWidget(dis_cloud, "아프리카 돼지열병.html", selfcontained = F)
                                                    # html파일로 저장
```

* wordcloud 패키지를 이용한 워드클라우드 결과

* wordcloud2를 이용한 워드클라우드 결과

2) 아프리카 돼지열병에 관한 감성분석

- 이용할 패키지를 설치 및 실행시킨다. 그리고 감성분석할 때 트위터 api 인증을 받아야 하지만, 위에서 이미 인증 받았으므로 과정은 생략한다.

```
install.packages("plyr")

library(twitteR)
library(plyr)
library(stringr)
```

- 문장에 대한 전처리 과정을 실시하고 긍정사전과 부정사전에서 매칭시켜서 나온 감성점수를 반환하고, 각각의 문장과 점수를 데이터프레임으로 변환시키는 함수를 실행한다.

```
score.sentiment = function(sentences, pos.words, neg.words)
{
   scores = laply(sentences,
   function(sentence, pos.words, neg.words)
   {
      sentence = gsub("[[:punct:]]", "", sentence) # 문장부호 제거
      sentence = gsub("[[:cntrl:]]", "", sentence)  # 특수문자 제거
      sentence = gsub('₩d+', '', sentence)          # 숫자 제거
      word.list = strsplit(sentence, "₩s+")    # 문장을 '빈칸'으로 나눔
                                              # ₩s+ : 빈칸 1칸 이상을 의미함.
      words = unlist(word.list)
      pos.matches = match(words, pos.words)
                                           # words의 단어를 positive에서 맞춘다.
      neg.matches = match(words, neg.words)
                                           # words의 단어를 negative에서 맞춘다.
      pos.matches = !is.na(pos.matches)     # NA 제거함. 위치(숫자)만 추출함.
      neg.matches = !is.na(neg.matches)
      score = sum(pos.matches) - sum(neg.matches)
                                                  # score = 긍정점수 - 부정점수
      return(score)                               # score값 반환
    }, pos.words, neg.words)
   scores.df = data.frame(text=sentences, score=scores)
                                  # 각각의 문장과 점수를 데이터프레임으로 변환
   return(scores.df)
}                                  # 문장의 감성점수를 측정하는 함수 생성
```

- 군산대학교에서 만든 감성사전에 있는 긍정 사전과 부정 사전을 변수에 저장한다.

```
pos.words <- readLines("pos_pol_word.txt", encoding = "UTF-8")  # 긍정 사전
neg.words <- readLines("neg_pol_word.txt", encoding = "UTF-8")  # 부정 사전
```

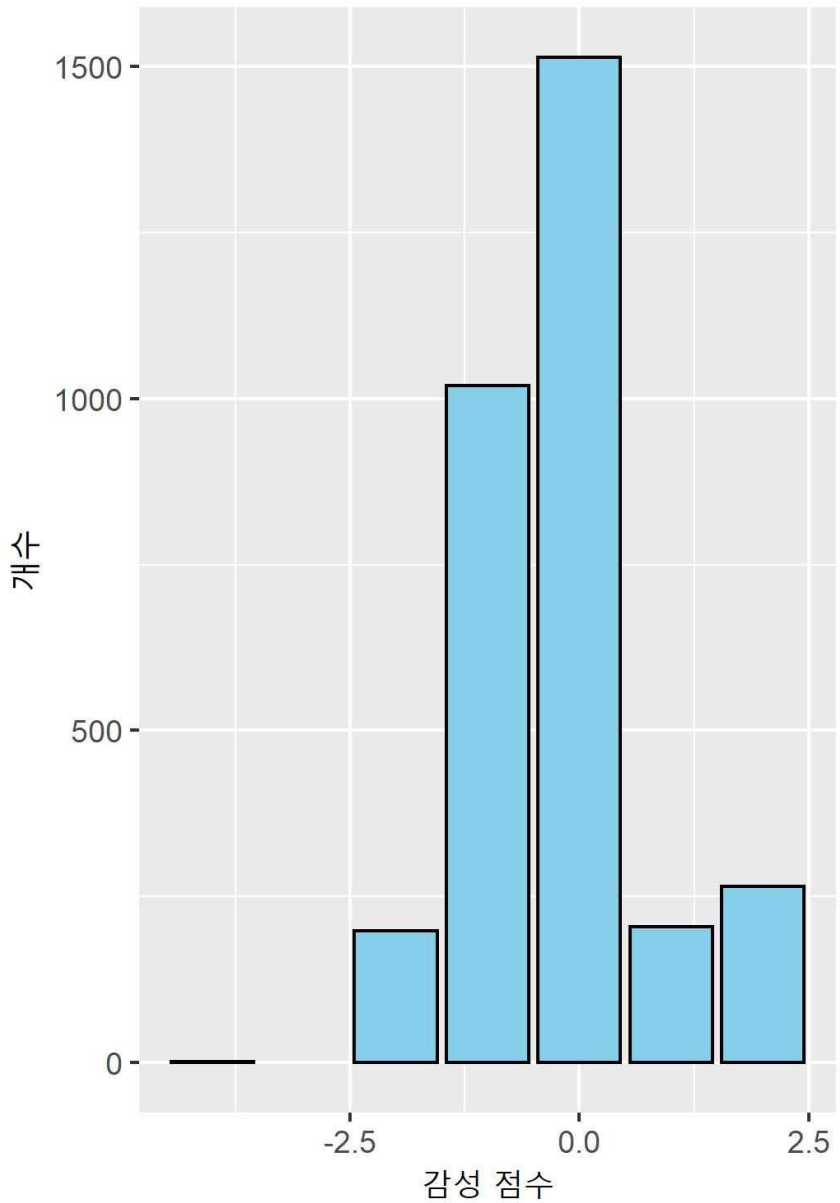
- 아까 분석했었던 트위터 데이터를 사용한다.

```
disease_txt <- sapply(disease, function(x) x$getText(), USE.NAMES=F)
write.csv(disease_txt, "돼지열병 트위터 내용.txt")    # 저장하기

disease.score <- score.sentiment(disease_txt, pos.words, neg.words)
table(disease.score$score)
mean(disease.score$score)
```

```
> table(disease.score$score)

  -4   -2   -1    0    1    2
   1  197 1020 1513  204  265
> mean(disease.score$score)
[1] -0.21375
```

- 그 후, qqplot을 그려서 시각적으로 표시한다.

```
# qplot 만들기
library(ggplot2)
qplot(disease.score$score, xlab = "감성 점수", ylab = "개수")+
        geom_bar(color = 'black', fill = "skyblue")

# xlab와 ylab으로 x축, y축 이름 정해준다.
# 그래프는 테두리가 검은색인 파란색 막대그래프를 그렸다.

# qplot 저장하기
ggsave(file = "C:/Users/XPS/Desktop/대학교 자료/공부/대학교/정보통계학과/3-2학기/빅데이터입문/기말과제/bargraph2.jpg", width = 3.5, height = 5)
```



3) 돼지고기 가격에 관한 감성분석

- 돼지고기라고 흔히 말하면 삼겹살을 많이 생각하게 되는데, 그래서 검색어를 “돼지고기 가격”과 “삼겹살 가격”로 지정한 후, 데이터프레임 형태이므로 rbind로 합치기로 하였다.

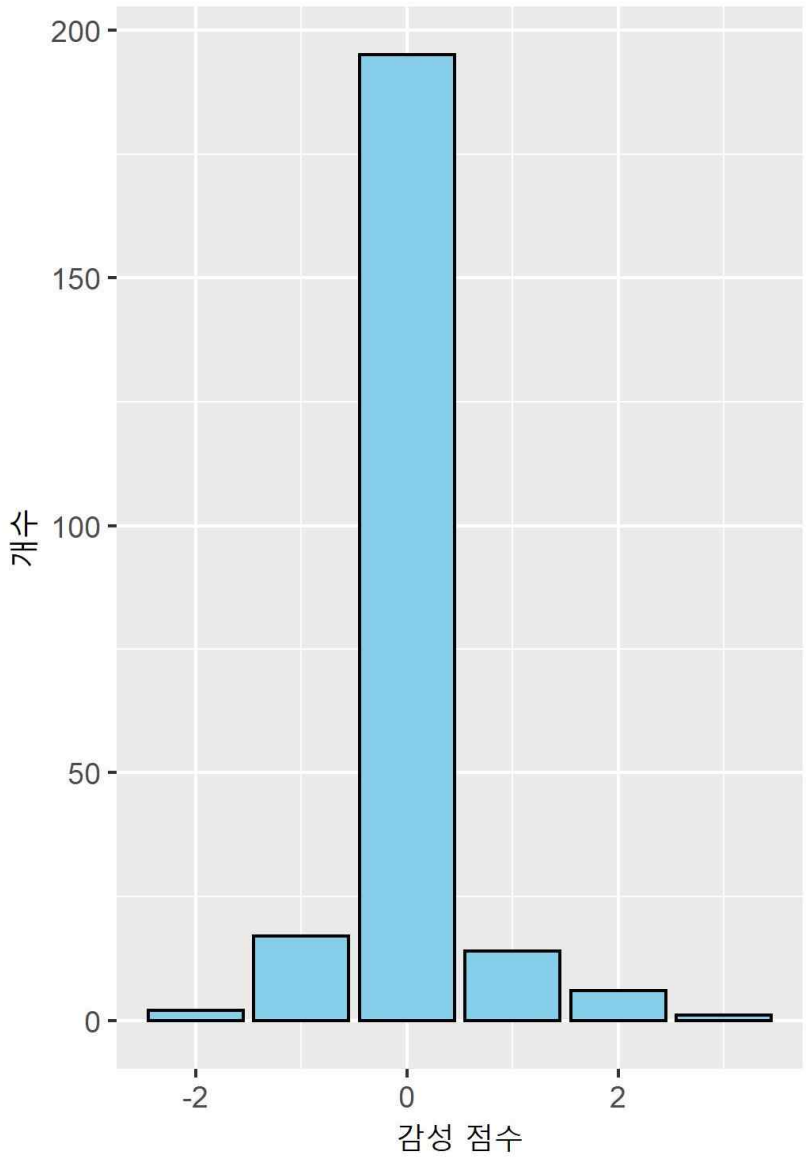
```
word1 <- enc2utf8("돼지고기 가격")
word2 <- enc2utf8("삼겹살 가격")

price1 <- searchTwitter(word1, n = 500, lang="ko")
price2 <- searchTwitter(word2, n = 500, lang="ko")
```

```
df_price1 <- do.call("rbind", lapply(price1, as.data.frame))
df_price2 <- do.call("rbind", lapply(price2, as.data.frame))
price <- rbind(df_price1, df_price2)



# 데이터 저장
write.csv(price$text, "돼지고기 가격 검색어 데이터.txt")
```

- 그 후, 감성분석을 실시하고, qqplot을 그렸다.

```
# 감성분석 실시
price.score <- score.sentiment(price$text, pos.words, neg.words)

table(price.score$score)
mean(price.score$score)

# qqplot 쓰기
library(ggplot2)
qplot(price.score$score, xlab = "감성 점수", ylab = "개수")+
        geom_bar(color = 'black', fill = "skyblue")

# xlab, ylab으로 x축, y축 이름 정해줌
# 그래프는 테두리가 검은색인 파란색 히스토그램을 그렸다.

# qplot 저장하기
ggsave(file = "C:/Users/XPS/Desktop/대학교 자료/공부/대학교/정보통계학과/3-2학기/빅
데이터입문/기말과제/bargraph3.jpg",
                width = 3.5, height = 5)
```

```
> table(price.score$score)

 -2  -1   0   1   2   3
  2  17 195  14   6   1
>
> mean(price.score$score)
[1] 0.03404255
```


200
150
100
50
0
개수
-2
0
2
감성 점수

4. 결론

- 아프리카 돼지열병에 관한 최다빈출 15개에 대한 빈도별 그래프와 워드클라우드를 보았을 때, '경기도', '살처분', '이재명', '행사', '매몰', '정치', '새끼', '강물', '농가', '수십', '정상', '홍보', '임진강', '대표', '국회의장'이라는 말이 많이 나오는 것을 알 수 있다.

- 아프리카 돼지열병에 관한 감성분석에 대해서는 -4점이 1개, -2점이 197개, -1점이 1020개, 0점이 1513개, 1점이 204개, 2점이 265개로, 감성점수 평균은 -0.21375로 측정이 되었다. 감성점수가 0보다 작으므로 대체로 부정적인 의견(negative opinion)을 나타내는 것으로 간주할 수 있다.

- 돼지고기 가격에 관한 감성분석에 대해서는 -2점이 2개, -1점이 17개, 0점이 195개, 1점이 14개, 2점이 6개, 3점이 1개로, 감성점수 평균이 0.03404255로 측정이 되었다. 감성점수가 0보다 크므로 대체로 긍정적 의견(positive opinion)을 나타내는 것으로 간주할 수 있다.

5. 참고자료

- 서론 부분

http://goodnews1.com/news/news_view.asp?seq=91750 - 돼지열병 관련 뉴스기사

http://www.mafra.go.kr/FMD-AI/1511/subview.do - 농림축산식품부 아프리카돼지열병 관련 자료

- R코드 참고

http://127.0.0.1:20482/library/twitteR/html/search.html - searchTwitter()에 관한 내용

https://github.com/Lchiffon/wordcloud2/issues/8 - wordcloud2 저장하는 방법

http://www.dodomira.com/2016/03/18/ggplot2-%EA%B8%B0%EC%B4%88/ - qqplot()에 대한 내용

빅데이터입문 교재