

230214

CJA

2023-02-12

```
library(dplyr) #통계 데이터 분석을 위한 라이브러리. 효율적인 데이터 전처리와 처리 가능.
```

```
##  
## 다음의 패키지를 부착합니다: 'dplyr'
```

```
## The following objects are masked from 'package:stats':  
##  
## filter, lag
```

```
## The following objects are masked from 'package:base':  
##  
## intersect, setdiff, setequal, union
```

```
library(KoNLP) #한국어 NLP(자연어 처리)를 위한 라이브러리. 형태소 분석, 감정 분석 등 한국어 NLP 작업에 사용 가능.
```

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

```
library(wordcloud) #텍스트 데이터에서 단어의 빈도수를 바탕으로 워드 클라우드 시각화 가능.
```

```
## 필요한 패키지를 로딩중입니다: RColorBrewer
```

```
library(wordcloud2) #wordcloud 패키지의 개선판. 좀 더 다양한 옵션으로 워드 클라우드 시각화 가능.  
library(rvest) #웹 크롤링을 위한 라이브러리. HTML, XML 등의 데이터를 가져오는데 사용 가능.  
library(stringr) #문자열 처리를 위한 라이브러리. 문자열 검색, 제거, 치환 등 간편하게 수행 가능.  
library(foreign) #SAS, SPSS, Stata 등의 파일 형식을 R에서 사용할 수 있도록 변환해주는 라이브러리.  
library(readxl) #엑셀 파일을 R에서 사용할 수 있도록 변환해주는 라이브러리.  
library(xts) #시계열 데이터를 다루기 위한 라이브러리. 시계열 데이터를 관리하고 분석하는데 편리함
```

```
## 필요한 패키지를 로딩중입니다: zoo
```

```
##  
## 다음의 패키지를 부착합니다: 'zoo'
```

```
## The following objects are masked from 'package:base':  
##  
##   as.Date, as.Date.numeric
```

```
##  
## 다음의 패키지를 부착합니다: 'xts'
```

```
## The following objects are masked from 'package:dplyr':  
##  
##   first, last
```

```
library(dygraphs) #시계열 데이터 시각화를 위한 라이브러리. 데이터의 시간 경과에 따른 변화를 직  
관적으로 시각화 가능.  
library(plotly) #시각화 라이브러리. 다양한 차트 타입과 인터랙티브한 그래프 제공.
```

```
## 필요한 패키지를 로딩중입니다: ggplot2
```

```
##  
## 다음의 패키지를 부착합니다: 'plotly'
```

```
## The following object is masked from 'package:ggplot2':  
##  
##   last_plot
```

```
## The following object is masked from 'package:stats':  
##  
##   filter
```

```
## The following object is masked from 'package:graphics':  
##  
##   layout
```

```
library(ggplot2) #고급 시각화 라이브러리. 다양한 그래프 타입과 커스텀 옵션을 제공하여 높은 정  
밀도로 시각화 가능.  
library(multilangr)  
library(ggmap) #ggmap 활성화
```

```
## ⓘ Google's Terms of Service: <-]8::https://mapsplatform.google.com▶https://mapsplatform.goog  
le.com<-]8;;▶>
```

```
## ⓘ Please cite ggmap if you use it! Use `citation("ggmap")` for details.
```

```
##  
## 다음의 패키지를 부착합니다: 'ggmap'
```

```
## The following object is masked from 'package:plotly':  
##  
##      wind
```

```
require(animation)
```

```
## 필요한 패키지를 로딩중입니다: animation
```

```
library(animation)  
library(jpeg)  
library(reshape)
```

```
##  
## 다음의 패키지를 부착합니다: 'reshape'
```

```
## The following object is masked from 'package:plotly':  
##  
##      rename
```

```
## The following object is masked from 'package:dplyr':  
##  
##      rename
```

```
attach(tips)  
detach(tips)  
attach(tips)
```

```

title <- c()
press <- c()
time <- c()
body <- c()
url <- c()
p_url <- 'https://search.naver.com/search.naver?sm=tab_hyt.top&where=news&query=%EA%B8%88%EB%A
6%AC='
library(stringr)
library(rvest)
library(KoNLP)
for(i in 1:50){
  c_url <- paste0(p_url,10*i+1)
  title_css <- '.news_tit'
  time_css <- "span.info"
  press_css <- ".info.press"
  body_css <- ".dsc_txt_wrap"

  html_doc <- read_html(p_url)
  title_node <- html_nodes(html_doc,title_css)
  press_node <- html_nodes(html_doc,press_css)
  time_node <- html_nodes(html_doc,time_css)
  body_node <- html_nodes(html_doc,body_css)

  title_part <- html_text(title_node)
  press_part <- html_text(press_node)
  time_part <- html_text(time_node)
  body_part <- html_text(body_node)
  body_part <- str_trim(body_part,side = "both")
  url_part <- html_attr(title_node,"href")

  title <- c(title,title_part)
  time <- c(time,time_part)
  press <- c(press,press_part)
  body <- c(body,body_part)
  url <- c(url,url_part)
}

n_news <- data.frame(cbind(title,press,time,body))

```

```

## Warning in cbind(title, press, time, body): number of rows of result is not a
## multiple of vector length (arg 1)

```

```

write.csv(n_news,"naver_news.csv",fileEncoding = "CP949")

```

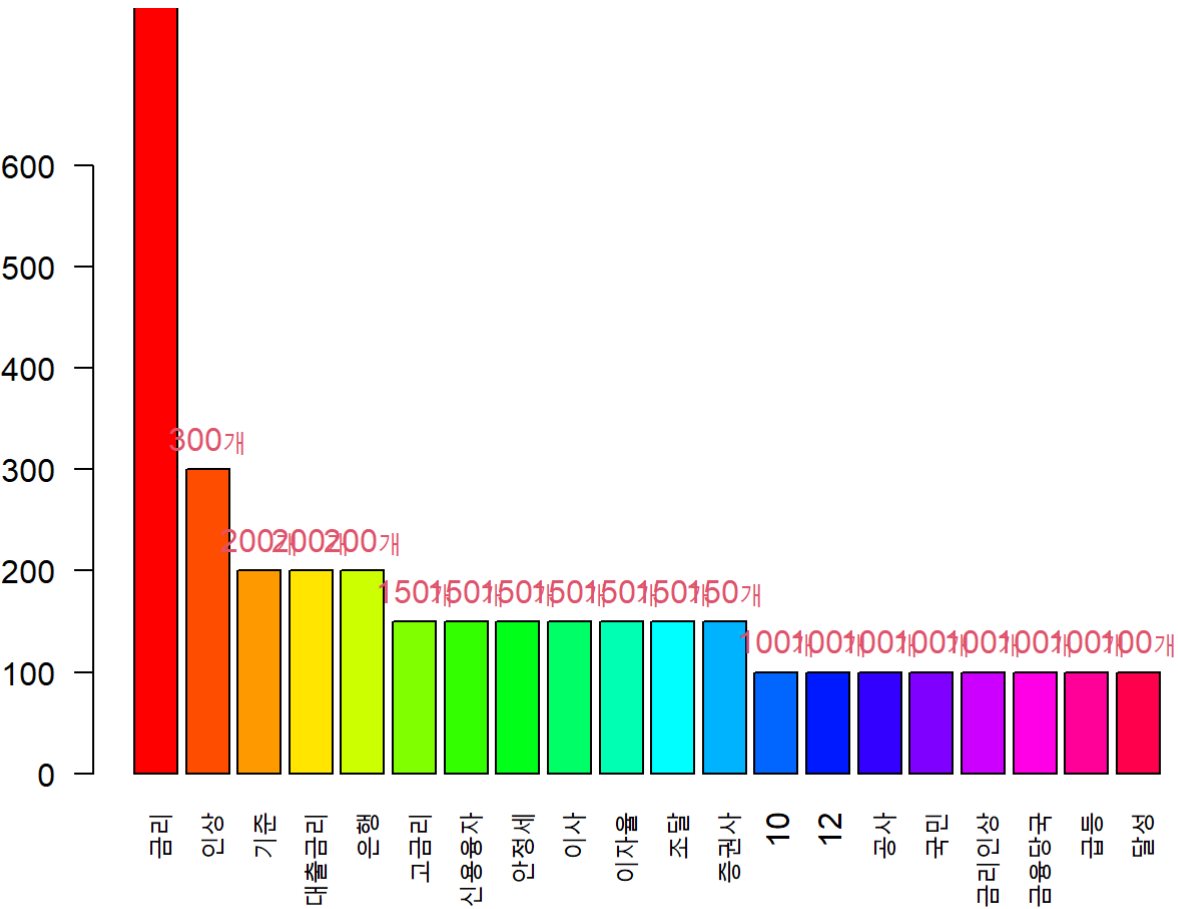
```

# 도수분포표
naver_news <- data.frame(cbind(title,body))
text <- sapply(naver_news,extractNoun,USE.NAMES = F)
text <- unlist(text)
n_count <- Filter(function(x){nchar(x)>=2},text)
word <- table(n_count)
head(sort(word,decreasing = T),20)

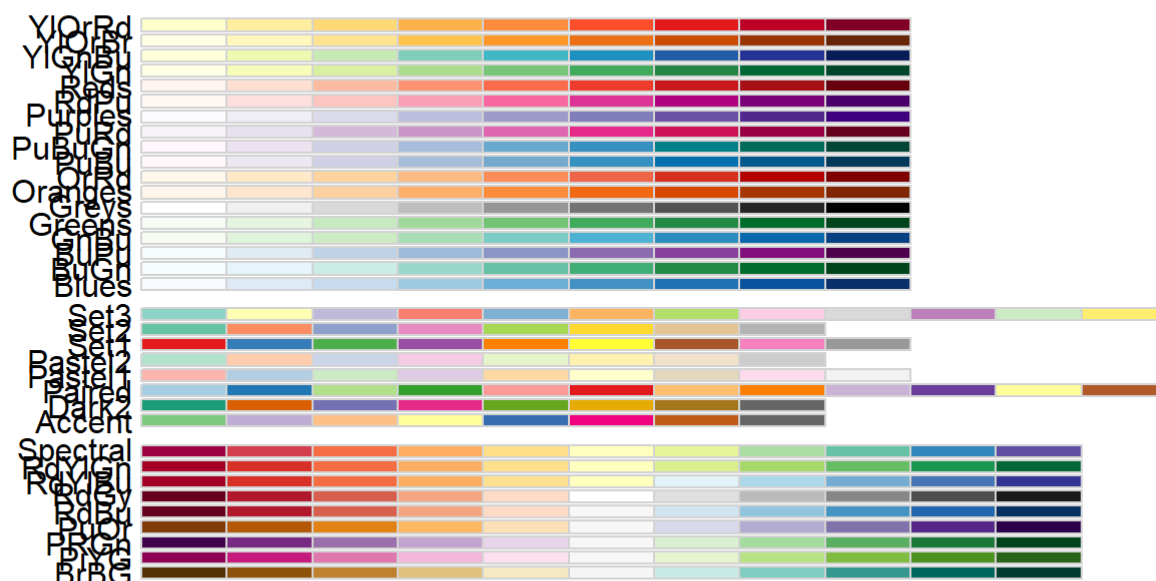
```

```
## n_count
##      금리      인상      기준 대출금리      은행      고금리      신용융자      안정세
##      1050      300      200      200      200      150      150      150
##      이 사      이 자 율      조 달      증 권 사      10      12      공 사      국 민
##      150      150      150      150      100      100      100      100
## 금리인상 금융당국 급등 달성
##      100      100      100      100
```

```
# barplot 시각화
hw <- head(sort(word,decreasing = T),20)
bar <- barplot(hw,col=rainbow(20),ylim=c(0,600),las=2)
text(bar,hw,paste0(hw,"개"),pos=3,col=2,cex=1)
```



```
# wordcloud
display.brewer.all()
```



```

palate <- brewer.pal(9,"Set1")
wordcloud(names(word),
          freq=word,
          min.freq = 2,
          scale = c(2,0.2),
          random.order = F,
          random.color = T,
          rot.per = 0.2,
          colors = palate)

```



```
# wordcloud2
wordcloud2(data = word,
            size = 0.2,
            shape = 'diamond')
```

