```
HW4_201521466_김남현
HW4_201521466_김남현
2019 10 5
갤럭시 노트10과 아이폰11의 최근 일주일 간 트윗을 수집 및 일별 빈도 비교
 library(syuzhet)
 library(twitteR)
 library(lubridate)
 ## Attaching package: 'lubridate'
 ## The following object is masked from 'package:base':
 ##
 ##
        date
 library(dplyr)
 ## Attaching package: 'dplyr'
 ## The following objects are masked from 'package:lubridate':
 ##
 ##
       intersect, setdiff, union
 ## The following objects are masked from 'package:twitteR':
       id, location
 ##
 ## The following objects are masked from 'package:stats':
 ##
 ##
        filter, lag
 ## The following objects are masked from 'package:base':
 ##
       intersect, setdiff, setequal, union
 library(tm)
 ## Loading required package: NLP
 library(stringr)
 library(RColorBrewer)
 library(igraph)
 ## Attaching package: 'igraph'
 ## The following objects are masked from 'package:dplyr':
        as_data_frame, groups, union
 ## The following objects are masked from 'package:lubridate':
 ##
       %--%, union
 \#\# The following objects are masked from 'package:stats':
 ##
       decompose, spectrum
 ##
 \#\# The following object is masked from 'package:base':
 ##
       union
 library(data.table)
 ## Attaching package: 'data.table'
 ## The following objects are masked from 'package:dplyr':
 ##
        between, first, last
 ## The following objects are masked from 'package:lubridate':
 ##
       hour, isoweek, mday, minute, month, quarter, second, wday,
 ##
       week, yday, year
 library(ggplot2)
 ## Attaching package: 'ggplot2'
 ## The following object is masked from 'package:NLP':
 ##
 ##
        annotate
 library(tinytex)
 setwd("C:/Users/Namlister/Desktop/Assignments/R_A")
 CONSUMER_SECRET <- "E8u5PZvaNvCNXIi34yEE8Efs8AEz2TxJbvyjpjWi2OpE3HENDQ"
 CONSUMER KEY <- "hvHT3aR6Reugq40MSEUG0sums"</pre>
 ACCESS_SECRET <- "BbKb4BFRfzyVppA380bh5FHxEjgV2s0PefdNy7Avthmsy"
 ACCESS_TOKEN <- "701807611465740288-g0Dp8IPrCv8VTEJfKRStR1dGHONeoxN"
 setup_twitter_oauth(consumer_key = CONSUMER_KEY,
                    consumer_secret = CONSUMER_SECRET,
                    access_token = ACCESS_TOKEN,
                    access_secret = ACCESS_SECRET)
 ## [1] "Using direct authentication"
 searchTerm1 <- "#GalaxyNote10"</pre>
 searchTerm2 <- "#iPhone11"</pre>
 # 갤럭시와 아이폰의 최근 일주일 데이터 수집.
 tweetsG = searchTwitter(searchTerm1, since = "2019-09-28", n=11000,lang = "en") \# 24 \%.
 tweetsI = searchTwitter(searchTerm2, since = "2019-09-28", n=12000,lang = "en") \#0/0/\Xi.
 ## [1] "Rate limited .... blocking for a minute and retrying up to 119 times ..."
 ## [1] "Rate limited .... blocking for a minute and retrying up to 118 times ..."
 ## [1] "Rate limited .... blocking for a minute and retrying up to 117 times ..."
 ## [1] "Rate limited .... blocking for a minute and retrying up to 116 times ..."
 ## [1] "Rate limited .... blocking for a minute and retrying up to 115 times ..."
 ## [1] "Rate limited .... blocking for a minute and retrying up to 114 times ..."
 ## [1] "Rate limited .... blocking for a minute and retrying up to 113 times ..."
 ## [1] "Rate limited .... blocking for a minute and retrying up to 112 times ..."
 ## [1] "Rate limited .... blocking for a minute and retrying up to 111 times ..."
 ## [1] "Rate limited .... blocking for a minute and retrying up to 110 times ..."
 ## [1] "Rate limited .... blocking for a minute and retrying up to 109 times ..."
 tweetsG.df = twListToDF(tweetsG) #데이터프레임화.
 tweetsI.df = twListToDF(tweetsI)
 #View(tweetsI.df)
 #생성된 날짜에서 연월일만 추출해서 새로운 칼럼 생성한 후 따로 데이터프레임화.
 G_ymd <- transmute(tweetsG.df, G_ymd = date(tweetsG.df$created))</pre>
 I_ymd <- transmute(tweetsI.df, I_ymd = date(tweetsI.df$created))</pre>
 # 갤럭시와 아이폰 빈도 비교.
 table(G_ymd$G_ymd)
 ## 2019-09-28 2019-09-29 2019-09-30 2019-10-01 2019-10-02 2019-10-03
                                     1279 271
      1403
                  1957
                             5430
 ## 2019-10-04 2019-10-05
         257
 table(I_ymd$I_ymd)
 ## 2019-09-29 2019-09-30 2019-10-01 2019-10-02 2019-10-03 2019-10-04
 ## 1147
                 1676 1434 1394 2279 2973
 ## 2019-10-05
      1097
갤럭시보다 아이폰에 대한 언급이 훨씬 많다는 것을 알 수 있다. 갤럭시의 경우, 30일날 언급되는 경우가 갑자기 늘어났다가 그 이후로 점차 언급이
점점 줄어드는 것을 관찰할 수 있다.
수집된 트윗의 감정분포를 비교
 # 갤럭시와 아이폰에 관한 트윗에서 감정적인 텍스트를 추출한다.
 G_Sentiments <- get_nrc_sentiment(tweetsG.df$text)</pre>
 I_Sentiments <- get_nrc_sentiment(tweetsI.df$text)</pre>
 tweetsG <- cbind(tweetsG.df, G_Sentiments)</pre>
 tweetsI <- cbind(tweetsI.df, I Sentiments)</pre>
 # 데이터프레임에서 감정에 관한 칼럼들만 따로 추출한다.
 sentimentTotalsG <- data.frame(colSums(tweetsG[,c(17:26)]))</pre>
 sentimentTotalsI <- data.frame(colSums(tweetsI[,c(17:26)]))</pre>
 # 감정의 종류별로 등장하는 총 횟수를 정리한다.
 names(sentimentTotalsG) <- "count"</pre>
 names(sentimentTotalsI) <- "count"</pre>
 sentimentTotalsG <- cbind("sentiment" = rownames(sentimentTotalsG), sentimentTotalsG)</pre>
 sentimentTotalsI <- cbind("sentiment" = rownames(sentimentTotalsI), sentimentTotalsI)</pre>
 rownames(sentimentTotalsG) <- NULL</pre>
 rownames(sentimentTotalsI) <- NULL</pre>
 # 시각화
 ggplot(data = sentimentTotalsG, aes(x = sentiment, y = count)) +
  geom_bar(aes(fill = sentiment), stat = "identity") +
   theme(legend.position = "none", axis.text.x = element_text(angle = 45)) +
   xlab("Sentiment") + ylab("Total Count") + ggtitle("GalaxyNote10")
        GalaxyNote10
   10000 -
   7500 -
 Total Count
    5000 -
   2500 -
                                        Sentiment
 ggplot(data = sentimentTotalsI, aes(x = sentiment, y = count)) +
   geom bar(aes(fill = sentiment), stat = "identity") +
   theme(legend.position = "none", axis.text.x = element_text(angle = 45)) +
   xlab("Sentiment") + ylab("Total Count") + ggtitle("iPhone11")
       iPhone11
   6000
   4000 -
 Total Count
                                                                                  감정 종류에 따른 분포는 대략적으로
   2000
                                       Sentiment
비슷해보인다. 다만 갤럭시의 경우 아이폰보다 negative 의견이 많으나 positive 의견이 압도적으로 많은 것을 볼 수 있다.
각각의 주제에서 가장 많은 트윗을 올린 계정 2개를 대상으로 Follower network(1차만)
을 구성하여 시각화
 # 갤럭시와 아이폰에 관한 트윗을 가장 많이 올린 계정을 각각 찾는다.
 head(sort(table(tweetsG.df$screenName), decreasing = T))
 ##
 ## whitestonedome whitestone_UK naijagym karan875
                    174 109 31 31
              281
       djkirimino
 ##
               23
 head(sort(table(tweetsI.df$screenName), decreasing = T))
 ##
                    vasara740 whitestonedome yadhell dj_sunspot
76 63 57 47
 ##
       sshanthan8
 ##
        ld_vova
 ##
               31
 # 특정 계정의 팔로워들을 추출하는 함수
 get follower list <- function(userName) {</pre>
   twitterUser <- getUser(userName)</pre>
   twitterUserFollowerIDs<-twitterUser$getFollowers(retryOnRateLimit=1)</pre>
   return (sapply(twitterUserFollowerIDs, screenName))
 append_to_df<-function(dt, elems)</pre>
   return(rbindlist(list(dt, elems), use.names = TRUE))
 # 갤럭시와 아이폰에 관한 트윗을 가장 많이 올린 계정이다.
 coreUserName_G <- "whitestonedome"</pre>
 coreUserName_I <- "sshanthan8"</pre>
 whitestonedome <- getUser(coreUserName_G)</pre>
 sshanthan8 <- getUser(coreUserName_I)</pre>
 # 팔로워 리스트를 추출한다.
 sshanthan8_follower_IDs <- sshanthan8$getFollowers(retryOnRateLimit=10, n = 100)
 whitestonedome followers df = rbindlist(lapply(
  whitestonedome_follower_IDs,as.data.frame
 sshanthan8_followers_df = rbindlist(lapply(
  sshanthan8_follower_IDs,as.data.frame
 # 팔로워 목록 중에서 규모가 어느 정도 있는 계정만 걸러낸다.
 G filtered <- subset(whitestonedome followers df,</pre>
                      followersCount < 100 &
                        followersCount > 50 &
                        #statusesCount > 10000 & #to reduce number of followers
                        # statusesCount > 100 &
                        # statusesCount < 5000 & #too many tweets from bots?</pre>
                        protected==FALSE)
 G_filtered_follower <- G_filtered$screenName</pre>
 I_filtered <- subset(sshanthan8_followers_df,</pre>
                     followersCount < 100 &
                       followersCount > 50 &
                       #statusesCount > 10000 & #to reduce number of followers
                       # statusesCount > 100 &
                       # statusesCount < 5000 & #too many tweets from bots?</pre>
                       protected==FALSE)
 I_filtered_follower <- I_filtered$screenName</pre>
 # 데이터프레임으로 만들어서 정리한다.
 edge_df_G <- data.frame(from=G_filtered_follower,</pre>
                    to=rep(coreUserName_G,
                           length(G_filtered_follower)),
                    stringsAsFactors=FALSE)
 edge_df_I <- data.frame(from=I_filtered_follower,</pre>
                        to=rep(coreUserName_I,
                               length(I_filtered_follower)),
                        stringsAsFactors=FALSE)
 counter = 1
 for(follower in G_filtered_follower) {
   followerScreenNameList_G <- get_follower_list(follower)</pre>
   Sys.sleep(30)
   print(paste("Processing completed for:",
              follower,
              "(",counter,"/",
              length(G_filtered_follower),")"
   ) )
   edge_df_G <- append_to_df(edge_df_G, list(from=followerScreenNameList_G,</pre>
                                      to=rep(follower,
                                             length(followerScreenNameList_G))))
   counter <- counter + 1</pre>
 \#\# [1] "Processing completed for: VazValbert ( 1 / 12 )"
 ## [1] "Processing completed for: CarolinaBoy828 ( 2\ /\ 12 )"
 \#\# [1] "Processing completed for: Tamerdrg ( 3 / 12 )"
 ## [1] "Processing completed for: HarisSljivo ( 4 / 12 )"
 \#\# [1] "Processing completed for: oestrada579 ( 5 / 12 )"
 ## [1] "Processing completed for: gamereric22 ( 6 / 12 )"
 ## [1] "Processing completed for: jlhigh02 ( 7 / 12 )"
 ## [1] "Processing completed for: Zamantut ( 8 \ / \ 12 )"
 \#\# [1] "Processing completed for: Kiwibornaussie ( 9 / 12 )"
 ## [1] "Processing completed for: johncmcgee2 ( 10 / 12 )"
 \#\# [1] "Processing completed for: MyTekReview ( 11 / 12 )"
 ## [1] "Processing completed for: LongTres ( 12 / 12 )"
 save(edge_df_G, file = "edge_df_G.Rda")
 load("edge_df_G.Rda")
 counter I = 1
 for(follower in I_filtered_follower) {
   followerScreenNameList_I <- get_follower_list(follower)</pre>
   Sys.sleep(30)
   print(paste("Processing completed for:",
              follower,
              "(",counter_I,"/",
              length(I_filtered_follower),")"
  ) )
   edge_df_I <- append_to_df(edge_df_I,list(from=followerScreenNameList_I,</pre>
                                          to=rep(follower,
                                                 length(followerScreenNameList_I))))
   counter_I <- counter_I + 1</pre>
 \mbox{\#\#} [1] "Processing completed for: veeryemula ( 1 / 1 )"
 save(edge_df_I, file = "edge_df_I.Rda")
 load("edge_df_I.Rda")
 # 시각화 준비
 net_G <- graph_from_data_frame(edge_df_G, directed=T)</pre>
 net_I <- graph_from_data_frame(edge_df_I, directed=T)</pre>
 table(edge_df_G$to)
 ## CarolinaBoy828 gamereric22 HarisSljivo jlhigh02 johncmcgee2
 ## 72 54 74 63 47
## Kiwibornaussie LongTres MyTekReview oestrada579 Tamerdrg
       74 85 66 57 62
      VazValbert whitestonedome Zamantut
 ##
 table(edge_df_I$to)
 ## sshanthan8 veeryemula
 edge_df_G[to=="whitestonedome"]$from
 ## [1] "VazValbert"
                         "CarolinaBoy828" "Tamerdrg"
                                                          "HarisSljivo"
 ## [5] "oestrada579" "gamereric22" "jlhigh02"
                                                          "Zamantut"
 ## [9] "Kiwibornaussie" "johncmcgee2" "MyTekReview"
                                                          "LongTres"
 edge_df_G[from=="whitestonedome"]$to
 ## character(0)
 edge_df_I[to=="sshanthan8"]$from
 ## [1] "veeryemula"
 edge_df_I[from=="sshanthan8"]$to
 ## character(0)
 net_G <- simplify(net_G, remove.multiple = F, remove.loops = T)</pre>
 net_I <- simplify(net_I, remove.multiple = F, remove.loops = T)</pre>
 # 각각의 차트의 사이즈와 글자 크기를 조정한다.
 deg_G <- degree(net_G, mode="all")</pre>
 V(net_G)$size <- deg_G*0.05 + 1</pre>
 V(net_G)[name == coreUserName_G]$size <- 15</pre>
 V(net_G)[size >= 15]$name
 ## [1] "whitestonedome"
 V(net_G)[name == coreUserName_G]$size
 ## [1] 15
 deg I <- degree(net I, mode="all")</pre>
 V(net_I)$size <- deg_I*0.05 + 1</pre>
 V(net I)[name == coreUserName I]$size <- 15</pre>
 V(net_I)[size >= 15]$name
 ## [1] "sshanthan8"
 V(net_I)[name == coreUserName_I]$size
 ## [1] 15
 # 차트에 사용할 색을 지정한다.
 pal3 <- brewer.pal(10, "Set3")</pre>
 plot(net_G, edge.arrow.size=0.1,
      vertex.label = ifelse(V(net G)$size >= 5, V(net G)$name, NA),
      vertex.color = pal3)
```

plot(net_I, edge.arrow.size=0.1,
 vertex.label = ifelse(V(net_I)\$size >= 5, V(net_I)\$name, NA),
 vertex.color = pal3)