R통계 과제3 트위터 API를 활용한 SMA

조성우_201823869

2020년 10월 24일

SMA

Twitter의 Open API를 통한 데이터 수집과 같은 각종기법을 사용하여 데이터를 수집하여 이를 통해 Social Media 및 Network에서 발생하는 현상들을 분석하여 Social Network에 대한 insight를 도출하는 분석법이다.

*1.*개요

목적:

Twitter API를 활용하여 #Tottenham 해시태그를 사용하는 트윗에서 수집할 수 있는 데이터를 얻고 이를 분석하여 EPL(England premier league)에 소속된 축구 구단인 Tottenham에 대한 Twitter 이용자들의 Social Media활동에서 드러나는 특징 등의 insight를 도출해보고자 한다.

활용 데이터셋:

'Twitter API의 #Tottenham 해시태그를 포함한 트윗 수집 데이터(2020.10.19-2020.10.25)'

상세:

트윗게시 시간, 본문내용, 게시자ID 등의 세부정보를 포함한 데이터셋

2.본문

2.1 데이터 수집

2.1.1 Twitter API를 통한 계정연결

```
#관련패키지 불러오기
library(tm)
library(ggplot2)
library(rtweet)
library(wordcloud)

#인증정보 세팅
APP_NAME <- "twitterianism"
API_KEY <- "EqVe1lnnDpbiBNw7VFMdQBfXX"
API_KEY_SECRET <- "UgRjuJVimMuTUxvNFp4mjGOEvrYTZ3GXCtJQkqpLepbbmpYVob"
ACCESS_TOKEN <- "1307868993311498240-aP3PxC1T66ZEfhQbPXBWCamUlm8XlU"
ACCESS_TOKEN_SECRET <- "16fcsGE6hbk60nXz1ZBWZV8PUSz1WujBi532giClaAoX2"

# 트위터앱에 연결
twitter_token <- create_token(app = APP_NAME, consumer_key = API_KEY, consumer_secret = API_KEY_SE
CRET,
                              access_token = ACCESS_TOKEN, access_secret = ACCESS_TOKEN_SECRET)
```

2.1.2데이터 수집실행

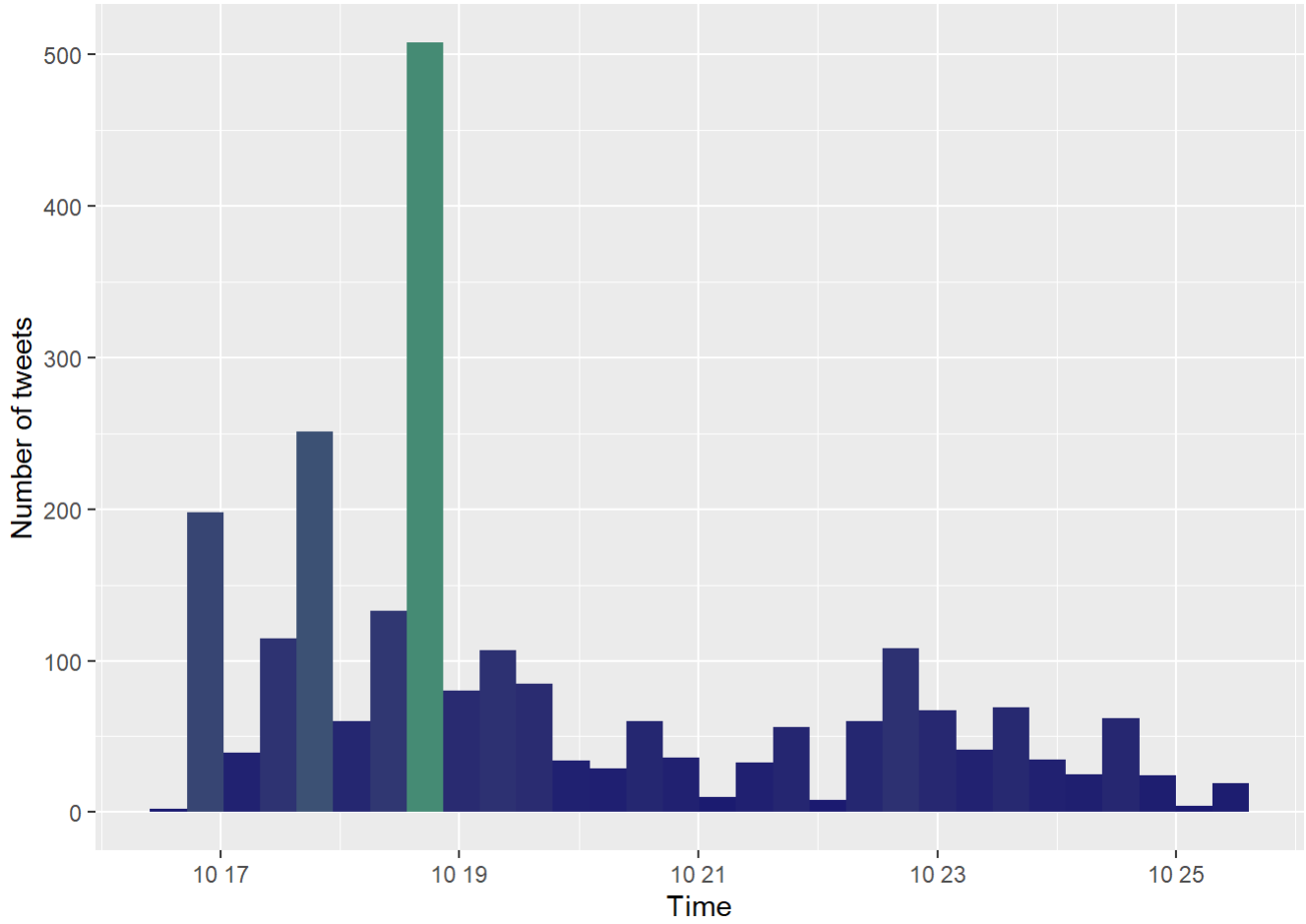
```
#API 검색/추출
searchTerm =  '#Tottenham' #해당 보고서에서 다룰 Tottenham 해시태그를 색인 키워드로 지정
trendingTweets = search_tweets(searchTerm, n=10000, lang = "en")  #영어 트윗 10000개를 수집

head(trendingTweets)
```

```
## # A tibble: 6 x 90
##   user_id status_id created_at          screen_name text  source
##   <chr>   <chr>     <dttm>              <chr>       <chr> <chr>
## 1 821857~ 13203733~ 2020-10-25 14:35:20 BeatsonFoo~ "Jus~ Twitt~
## 2 142153~ 13203731~ 2020-10-25 14:34:14 dandydon19~ "Jus~ Twitt~
## 3 108514~ 13203640~ 2020-10-25 13:58:08 jamesne602~ "Spu~ Twitt~
## 4 122946~ 13203633~ 2020-10-25 13:55:30 thfcollie03 "@_T~ Twitt~
## 5 124725~ 13203630~ 2020-10-25 13:54:12 DeeTombsGe~ "#Fr~ Twitt~
## 6 243728~ 13203613~ 2020-10-25 13:47:41 clairerobe~ "Hav~ Twitt~
## # ... with 84 more variables: display_text_width <dbl>,
## #   reply_to_status_id <chr>, reply_to_user_id <chr>,
## #   reply_to_screen_name <chr>, is_quote <lgl>, is_retweet <lgl>,
## #   favorite_count <int>, retweet_count <int>, quote_count <int>,
## #   reply_count <int>, hashtags <list>, symbols <list>, urls_url <list>,
## #   urls_t.co <list>, urls_expanded_url <list>, media_url <list>,
## #   media_t.co <list>, media_expanded_url <list>, media_type <list>,
## #   ext_media_url <list>, ext_media_t.co <list>, ext_media_expanded_url <list>,
## #   ext_media_type <chr>, mentions_user_id <list>, mentions_screen_name <list>,
## #   lang <chr>, quoted_status_id <chr>, quoted_text <chr>,
## #   quoted_created_at <dttm>, quoted_source <chr>, quoted_favorite_count <int>,
## #   quoted_retweet_count <int>, quoted_user_id <chr>, quoted_screen_name <chr>,
## #   quoted_name <chr>, quoted_followers_count <int>,
## #   quoted_friends_count <int>, quoted_statuses_count <int>,
## #   quoted_location <chr>, quoted_description <chr>, quoted_verified <lgl>,
## #   retweet_status_id <chr>, retweet_text <chr>, retweet_created_at <dttm>,
## #   retweet_source <chr>, retweet_favorite_count <int>,
## #   retweet_retweet_count <int>, retweet_user_id <chr>,
## #   retweet_screen_name <chr>, retweet_name <chr>,
## #   retweet_followers_count <int>, retweet_friends_count <int>,
## #   retweet_statuses_count <int>, retweet_location <chr>,
## #   retweet_description <chr>, retweet_verified <lgl>, place_url <chr>,
## #   place_name <chr>, place_full_name <chr>, place_type <chr>, country <chr>,
## #   country_code <chr>, geo_coords <list>, coords_coords <list>,
## #   bbox_coords <list>, status_url <chr>, name <chr>, location <chr>,
## #   description <chr>, url <chr>, protected <lgl>, followers_count <int>,
## #   friends_count <int>, listed_count <int>, statuses_count <int>,
## #   favourites_count <int>, account_created_at <dttm>, verified <lgl>,
## #   profile_url <chr>, profile_expanded_url <chr>, account_lang <lgl>,
## #   profile_banner_url <chr>, profile_background_url <chr>,
## #   profile_image_url <chr>
```

2.2 일별 트윗빈도 출력 및 해석(GGPLOT2)

gmodels 패키지의 함수를 사용하여 교차표를 출력하고 결과자료를 분석하여 가설을 검증합니다.

```
############################<1>    일별 트윗빈도 ##################################################
########
# plot on tweets by time
ggplot(data = trendingTweets, aes(x = created_at)) +
  geom_histogram(aes(fill = ..count..)) +
  theme(legend.position = "none") +
  xlab("Time") + ylab("Number of tweets") +
  scale_fill_gradient(low = "midnightblue", high = "aquamarine4")
```



***#Tottenham* 해시태그가 포함된 트윗이 게시된 일별 빈도를 바차트로 표현한 모습**

해석

1. **토트넘이 해시태그된 트윗은 *10*월 *19*일에 압도적인 게시수를 보여주고있다**
   EPL 경기일정을 조사해본 결과 10월 19일(월)에 토트넘VS웨스트햄전 경기가 치뤄졌는데, 해당 경기가 토트넘의 압도적인 우위속 경기를 리드하던중, 후반 마지막 10분경 3골을 내주며 극적인 무승부를 맞이하여서 팬들의 성화를 불러일으킨것으로 보인다.

2. ***10*월 *23*일 경기당시의 트윗은 매우 적은 게시빈도를 보여준다**
   10월 23일 토트넘은 LASK린츠와의 유로파리그 경기도 가졌는데 10월 19일의 EPL 웨스트햄전과는 상반되게 매우 적은 tweet수를 보여주고있다, 해당 경기는 웨스트햄과 비교하면 비교적 네임밸류가 적은상대와 치른 경기이며 매경기가 우승레이스에 직결되는 EPL과 다르게 유로파리그 초반일정에 해당하는 조별리그로 팬들에게 큰 기대를 사지 못했다고 생각된다.

일별 트윗빈도 해석결론 :

축구클럽에 대한 게시빈도는 해당하는 기간에 예정된 경기의 기대감요인에 반응하는것으로 짐작된다.

후속 연구를 통해 클럽이 치르는 경기에 따른 팬들의 기대감 요인(상대의 네임밸류/경기의중요성/리그중요성:해당 클럽리그순위와 다음순위로의 점수차이 등/더비여부)을 파악하고, 이를 검증해보고자 한다.

2.3 Tottenham 트윗의 본문의 빈출단어 (WordCloud)

2.3.1 트윗의 Text를 Corpus(말뭉치)로 변환 및 전처리

table()함수 등 내장함수를 활용하여 빈도표 및 교차표를 출력합니다.

```
library(stringr)
#전처리
trendingTweets$text <- sapply(trendingTweets$text,function(x) iconv(enc2utf8(x), sub="byte"))

head(trendingTweets$text, 2)
```

```
## Just 2 teams left on the football card for @Beatson_Charity <U+0001F49B><U+26BD> #NottsForest #
Tottenham https://t.co/npm4vnRX5Q
## "Just 2 teams left on the football card for @Beatson_Charity <f0>웋쎻슷 #NottsForest #Tottenham
https://t.co/npm4vnRX5Q"
## Just 2 teams left on the football card for @Beatson_Charity <U+0001F49B><U+26BD> #NottsForest #
Tottenham https://t.co/npm4vnRX5Q
## "Just 2 teams left on the football card for @Beatson_Charity <f0>웋쎻슷 #NottsForest #Tottenham
https://t.co/npm4vnRX5Q"
```

<전처리*1*> 데이터의 텍스트를 추출하여 인코딩한 모습

```
nohandles <- str_replace_all(trendingTweets$text, "@WWw+", "")
nohandles[15]
```

```
## [1] "Tottenham Hotspur v West Ham United https://t.co/4JNjFPZ9qg #tottenham #coys #thfc"
```

<전처리*2*> 특수문자를 제거한 모습

```
wordCorpus <- Corpus(VectorSource(nohandles))
wordCorpus[[15]]$content
```

```
## [1] "Tottenham Hotspur v West Ham United https://t.co/4JNjFPZ9qg #tottenham #coys #thfc"
```

<전처리*3*> 말뭉치로 변환한 모습

```
wordCorpus <- tm_map(wordCorpus, removePunctuation)
wordCorpus[[15]]$content
```

```
## [1] "Tottenham Hotspur v West Ham United httpstco4JNjFPZ9qg tottenham coys thfc"
```

<전처리*4*> *Punctuation*을 제거한 모습

```
wordCorpus <- tm_map(wordCorpus, content_transformer(tolower))
wordCorpus[[15]]$content
```

```
## [1] "tottenham hotspur v west ham united httpstco4jnjfpz9qg tottenham coys thfc"
```

<전처리*5*> 소문자로 변환한 모습

```
wordCorpus <- tm_map(wordCorpus, removeWords, stopwords("english"))
wordCorpus[[15]]$content
```

```
## [1] "tottenham hotspur v west ham united httpstco4jnjfpz9qg tottenham coys thfc"
```

<전처리*6*> 불용어사전을 통해 불용어*(*영어*)*를 제거한 모습

```
wordCorpus <- tm_map(wordCorpus, removeWords, c("amp","https","tottenham"))  #manual assignment
wordCorpus[[15]]$content
```

```
## [1] " hotspur v west ham united httpstco4jnjfpz9qg  coys thfc"
```

<전처리*7*> 특정 문자를 제거한 모습

```
wordCorpus <- tm_map(wordCorpus, removeNumbers)  #manual assignment
wordCorpus[[15]]$content
```

```
## [1] " hotspur v west ham united httpstcojnjfpzqg  coys thfc"
```

<전처리*8*> 숫자를 제거한 모습

```
wordCorpus <- tm_map(wordCorpus, stripWhitespace)
wordCorpus[[15]]$content
```

```
## [1] " hotspur v west ham united httpstcojnjfpzqg coys thfc"
```

<전처리*9*> 여백공간을 제거한 모습
전처리가 완료된 최종 데이터의 모습

2.3.2 WordCloud를 통해 빈출단어 출력

Tottenham이 태그된 게시물들이 포함한 단어의 빈출수를 기준으로 크기와 출력중심도를 반영하여 출력합니다.

```
pal = brewer.pal(8,'Dark2')

wordcloud(words = wordCorpus, scale=c(5,1), min.freq=5, max.words=200,
          random.order=F, rot.per=0.5, use.r.layout=TRUE, colors=pal)
```

해석

***1. EPL*내 경쟁구단에 대한 언급**

(1) 빈출단어로 현시점 리그 선두를 달리고 있는 Everton FC(1위)와 Liverpool FC(2위) 언급이 포함되어 있다.

(2) 북런던의 지역거점 라이벌구단인 Arsenal FC에 대한 언급도 빈번히 등장한것으로 보인다.

***2.* 최근 *Tottenham*경기에서 활약한 주축선수에 대한 언급**
최근 토트넘의 공격진에서 K-B-S조합으로 많은 기대를 받으며 대단한 활약을 보여주고있는 케인,손 그리고 베일에 대한 언급이 높은 빈도를 보여주고있다.

2. 무리뉴 감독에 대한 언급
'2년차의 무리뉴', 세계적 Top명장의 반열에 속해있는 무리뉴는 부임 2년차에 반드시 우승컵을 들어올린다는 징크스가 축구팬 사이에서 유명하다.

무리뉴에 대한 높은 언급빈도는 토트넘의 지휘봉을 잡은지 2년차에 돌입한 무리뉴에 대한 기대감이 반영된 결과로 보인다.

2.4 Tottenham 트윗의 감정분포 (감성사전)

2.4.1 임의의 감성함수 활용(1~5)

```
library(syuzhet)



## Sentiment analysis using syuzhet
tweetSentiments <- get_sentiment(trendingTweets$text,method = "syuzhet")
```

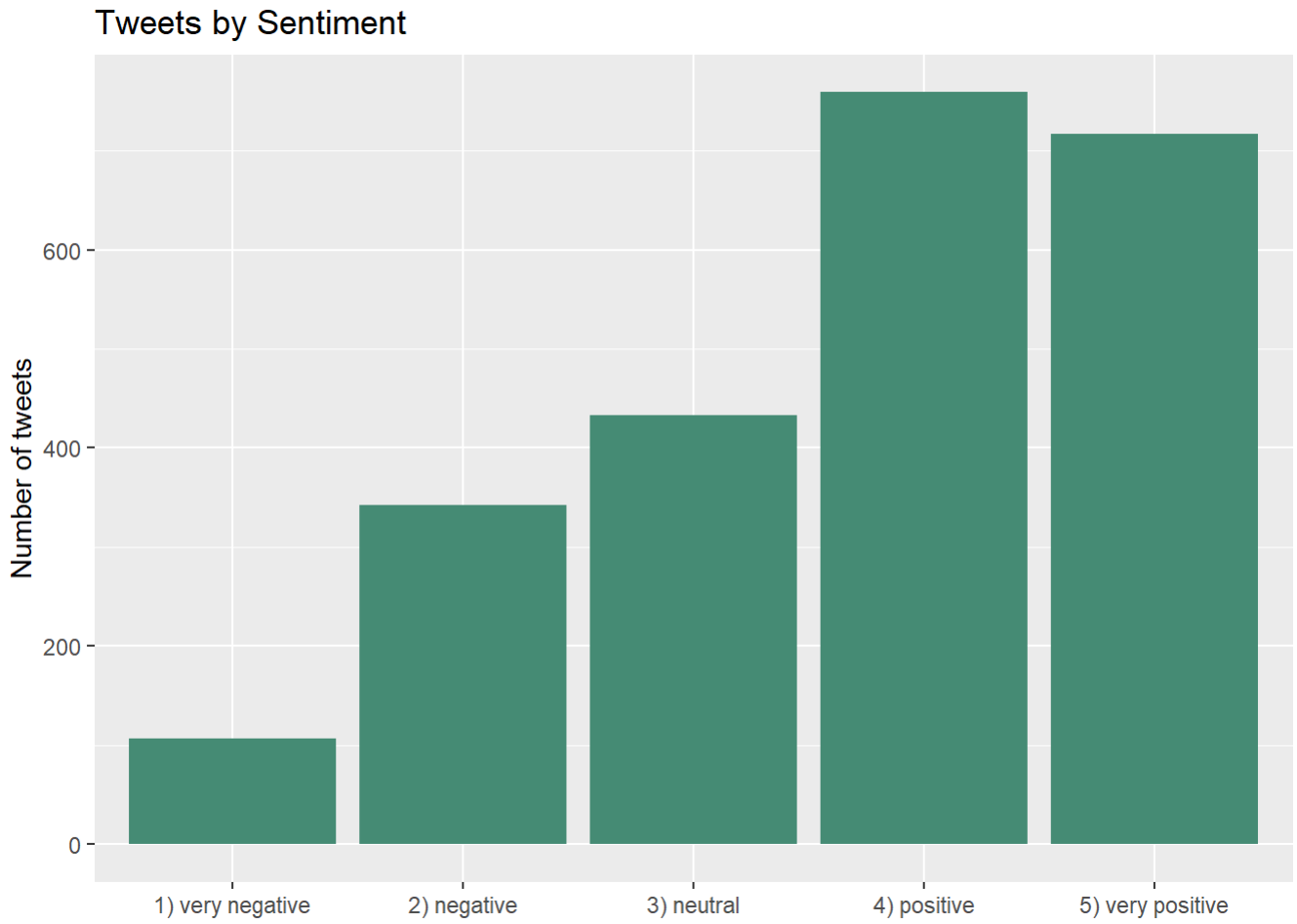
***Very Positive~Very negative*까지 *5*개의 범주로 감성을 분류하는 *Function*함수를 적용**

```
# 1~5의 정도로 감성값 메기기 위한 함수 정의
encodeSentiment <- function(x) {
  if(x <= -1){
    "1) very negative"
  }else if(x > -1 & x < 0){
    "2) negative"
  }else if(x > 0 & x < 1){
    "4) positive"
  }else if(x >= 1){
    "5) very positive"
  }else {
    "3) neutral"
  }
}
tweets <- cbind(trendingTweets, tweetSentiments)
tweets$sentiment <- sapply(tweets$tweetSentiments,encodeSentiment)
```

*ggplot*을 활용한 감성정도*(1~5)* 그래프 출력

table()함수 등 내장함수를 활용하여 빈도표 및 교차표를 출력합니다.

```
#plot by sentiment category
ggplot(tweets, aes(sentiment)) +
  geom_bar(fill = "aquamarine4") +
  theme(legend.position="none", axis.title.x = element_blank()) +
  ylab("Number of tweets") +
  ggtitle("Tweets by Sentiment")
```



2.4.2 syuzhet 감성사전 적용

syuzet을 적용한 감성분포 및 이에 해당하는 감성값을 적용한다.

```
## Sentiment analysis using syuzhet
tweetSentiments <- get_sentiment(trendingTweets$text,method = "syuzhet")
head(get_sentiment_dictionary(dictionary = 'syuzhet'),10)
```

```
##           word value
## 1      abandon -0.75
## 2    abandoned -0.50
## 3    abandoner -0.25
## 4  abandonment -0.25
## 5     abandons -1.00
## 6     abducted -1.00
## 7    abduction -0.50
## 8   abductions -1.00
## 9     aberrant -0.60
## 10  aberration -0.80
```

syuzhet* 감성사전 내의 감정과 이에 할당된 감성값 분포의 모습*(*요약출력*)

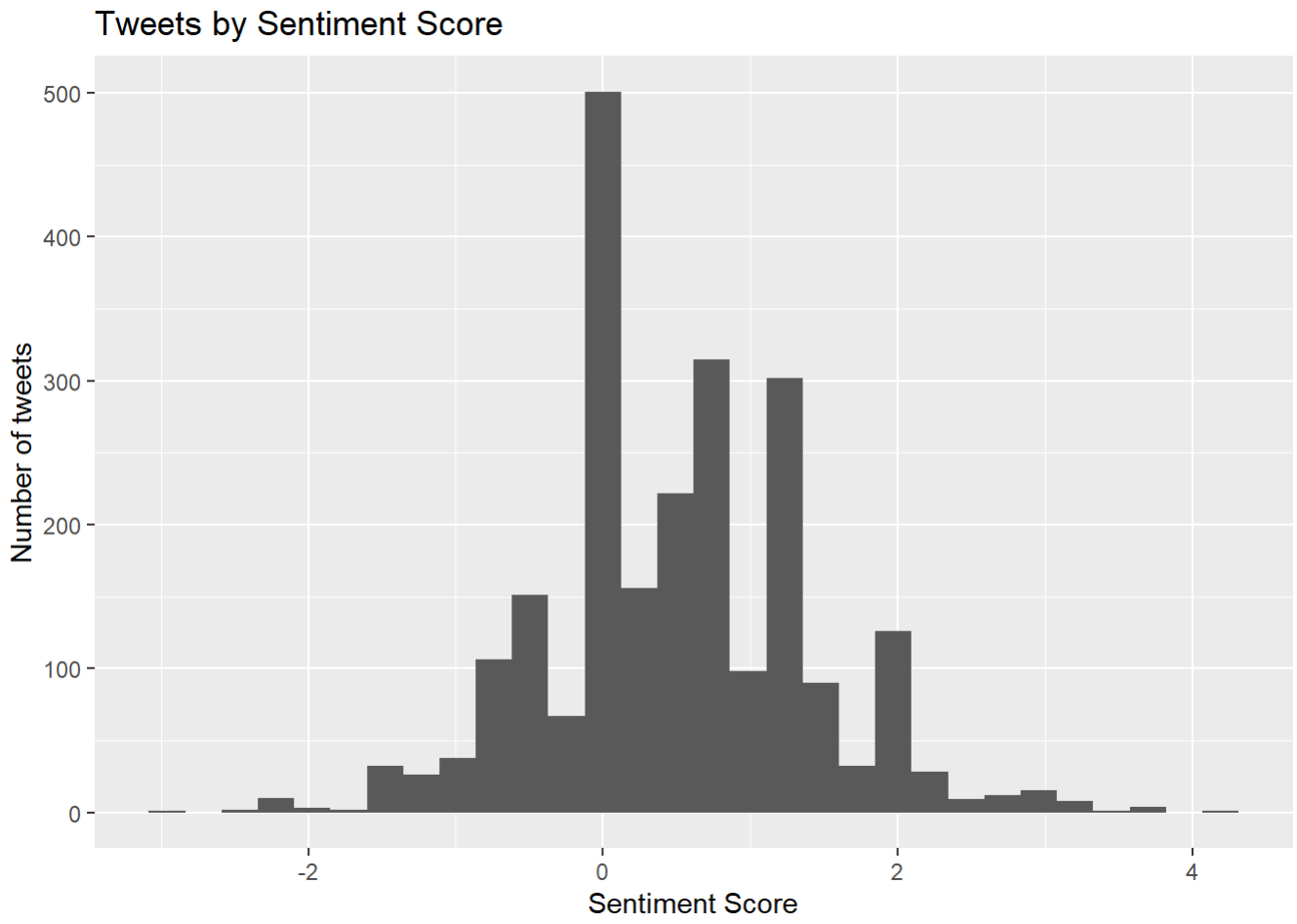
***Tottenham*데이터에 *Syuzhet*적용**

```
tweets <- cbind(trendingTweets, tweetSentiments)
tweets$sentiment <- sapply(tweets$tweetSentiments,encodeSentiment)
```

*Qplot*을 이용한 *Shuzhet*감성분포 그래프 출력

Tottenham이 태그된 게시물들이 포함한 단어의 빈출수를 기준으로 크기와 출력중심도를 반영하여 출력한다.

```
#plot by sentiment score
qplot(tweets$tweetSentiments) + theme(legend.position="none")+
  xlab("Sentiment Score") +
  ylab("Number of tweets") +
  ggtitle("Tweets by Sentiment Score")
```



2.4.3 NRC 감성사전 적용

NRC 감성사전을 적용하여 감성 범주를 할당합니다.

```
# NRC Sample (various emotions such as anger, fear, joy, ...)
library(stringi)
trendingTweets$text <- stri_trans_general(trendingTweets$text, "latin-ascii") #to remove non-Engli
sh text
tweetSentiments <- get_nrc_sentiment(trendingTweets$text)

tweets <- cbind(trendingTweets, tweetSentiments)
tweets[1, c(5, 91:100)]
```

```
##
text
## 1 Just 2 teams left on the football card for @Beatson_Charity <f0>웓쎻슺 #NottsForest #Tottenha
m https://t.co/npm4vnRX5Q
##   anger anticipation disgust fear joy sadness surprise trust negative positive
## 1     0            1       0    0   2       0        0     0        0        2
```

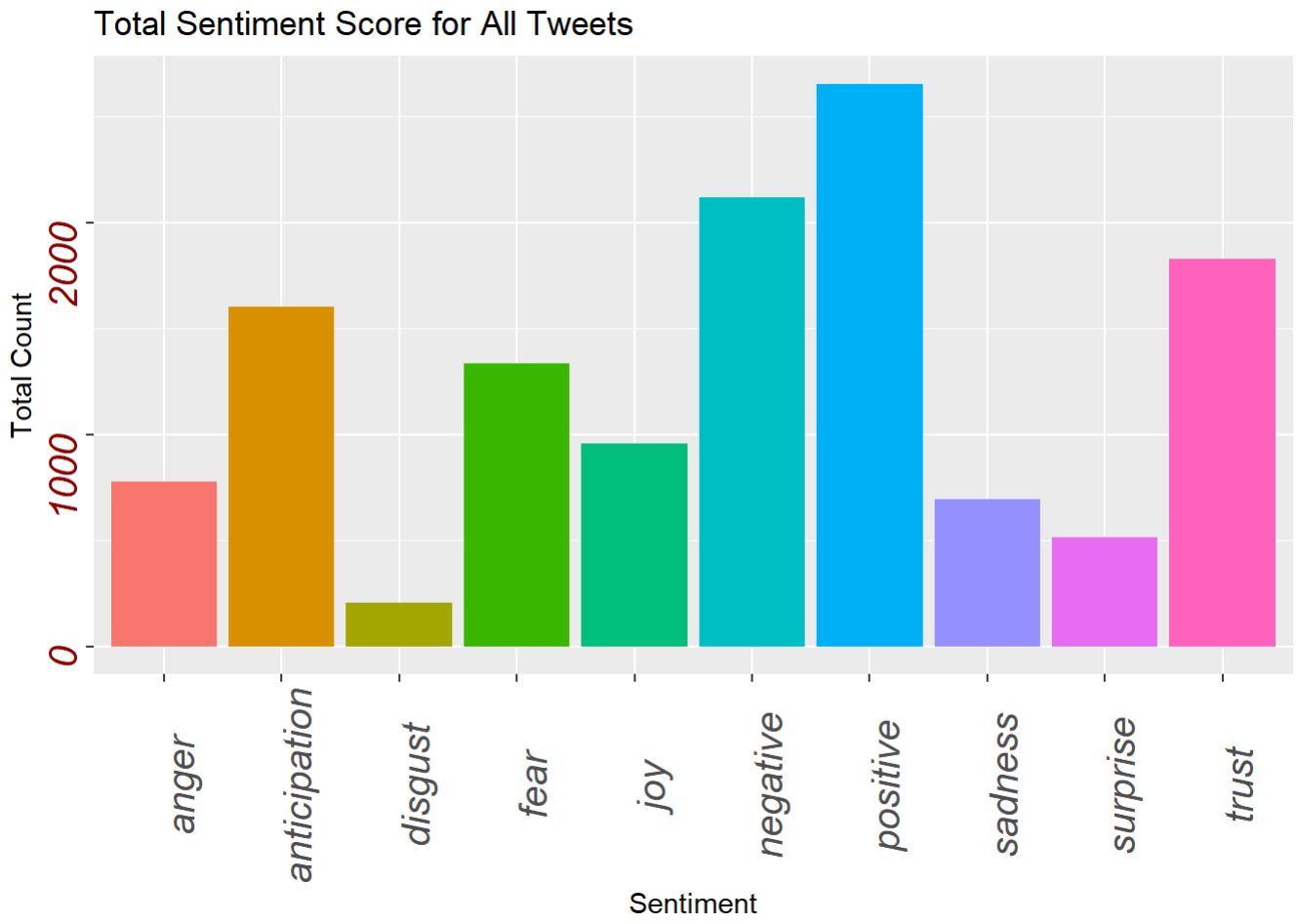
```
sentimentTotals <- data.frame(colSums(tweets[,c(91:100)]))

names(sentimentTotals) <- "count"
sentimentTotals <- cbind("sentiment" = rownames(sentimentTotals), sentimentTotals)
rownames(sentimentTotals) <- NULL
sentimentTotals
```

```
##       sentiment count
## 1         anger   779
## 2  anticipation  1604
## 3       disgust   207
## 4          fear  1335
## 5           joy   957
## 6       sadness   696
## 7      surprise   513
## 8         trust  1827
## 9      negative  2120
## 10     positive  2655
```

***NRC* 감성사전을 활용해 감성의 *Category*가 분류된 감성 빈도표의 모습**

*ggplot*을 이용한 *NRC*감성 *Category* 그래프 출력

```
ggplot(data = sentimentTotals, aes(x = sentiment, y = count)) +
  geom_bar(aes(fill = sentiment), stat = "identity") +
  theme(legend.position = "none",
        axis.text.y = element_text(size =15,face = 'italic',angle = 90,color='darkred'),
        axis.text.x = element_text(size =15,
                                   face = 'italic',
                                   angle = 90)) +
  xlab("Sentiment") + ylab("Total Count") + ggtitle("Total Sentiment Score for All Tweets")
```



해석

(1) Positive와,Negative,trust와 anticipation 순의 상위빈도 감정분포

해당하는 상위빈도 네개의 감정은 축구클럽에 관련한 트윗에서 전형적으로 보일 수 있는 상위빈도의 감정인것으로 짐작된다.
이는 추후 후속연구를 통해서 많은 축구클럽의 트윗을 표본으로 검증해보고자 한다.

(2) Positive & negative

negative와 비교하면 높은 positive 점수를 보이고 있는 Tottenham의 트윗분석결과는 현재 구단에 대한 긍정적인 여론이 더 크다는 사실을 알려준다.
다만,이를 감안하고서라도 높은 negative점수는 최근 westham전에서의 무승부에 대한 팬들의 충격이 유의미하게 반영된 결과로 보인다.

(3) anticipation & trust

최근 팬들의 무리뉴호에 대한 높은 기대감과 신뢰가 반영되었음으로 짐작된다.

2.5 Tottenham의 연관계정 Network분석

#Tottenham을 해시태그로 가장 많은 트윗을 올린 계정의 Follower Network를 구성하여 시각화한다.

2.5.1 Tottenham게시트윗 연관계정 데이터 수집과정

해당 과정은 Markdown상 오류의 문제로 script로 실행 및 저장한 후 데이터를 불러오는 방법으로,실질적인 코드실행은 생략됩니다.

```
############################<5> 해당 해쉬태그로 가장 많은 트윗을 올린 계정을 대상으로 Follower network을 구성하여 시각화(성능에 따라 filtering 필요) ##################################################

library(data.table)  #for rbindlist
library(igraph)
library(dplyr)

# append rows to dataframe
#append_to_df<-function(dt, elems)
#{
#  return(rbindlist(list(dt,  elems),use.names = TRUE))
#}




# Begin with a certain username
#coreUserName <- "jack"  #more than 4M followers... too big for an exercise
coreUserName <- "#Tottenham"

#twitterUser <- lookup_users(coreUserName)
#names(twitterUser)

#Extract Followers for the core user
#twitterUser_follower_IDs <- get_followers(twitterUser$user_id, retryonratelimit = 10)
#str(twitterUser_follower_IDs)
#head(twitterUser_follower_IDs)
#twitterUser_followers_df <- lookup_users(twitterUser_follower_IDs$user_id)
#head(twitterUser_followers_df)
#names(twitterUser_followers_df)

# filter dummy accounts (and reduce the number of followers for performance)
#filtered_df <- filter(twitterUser_followers_df,
#                      followers_count < 100 &
#                        followers_count > 50 &
                       #statuses_count > 10000 & #to reduce number of followers
                       # statuses_count > 100 &
                       # statuses_count < 5000 & #too many tweets from bots?
#                        protected==FALSE)
#statusesCount: number of tweets
#follwersCount: number of follwers (who follows this user)
#favoritesCount
#friendsCount: number of followees (whom this user follows)
#name: profile name that you can change
#protected: public or not
#verified: authentic (for celeb or organization)
#screenName: twitter ID (handle)
#location:
#language:
#id: integers (system-purpose key values?)
#listedCount: number of lists
#followRequestSent: ?
#profileImageUrl: profile image
```

```
#filtered_follower_IDs <- filtered_df$screen_name
#length(filtered_follower_IDs)

# prepare edge data frame (edges to coreUserName)
#edge_df<-data.frame(from=filtered_follower_IDs,
#                    to=rep(coreUserName,
#                           length(filtered_follower_IDs)),
#                    stringsAsFactors=FALSE)
#head(edge_df)
#tail(edge_df)
# edge_df <- append_to_df(edge_df,list(from=filtered_follower_IDs,
#                                      to=rep(coreUserName,
#                                             length(filtered_follower_IDs))))
# above lines were used to add edges to coreUserName later

# Iterate and extract list of followers of followers
#counter = 1



#for(follower in filtered_follower_IDs){  #filtered_follower_IDs * 60x seconds -> stop it and use
 saved file!
  # fetch follower list for current user
#  followerScreenNameList <- lookup_users(get_followers(follower)$user_id)$screen_name
#  Sys.sleep(10)  #twitter API limit is 15 for 15 mins
#  print(paste("Processing completed for:",
#              follower,
#              "(",counter,"/",
#              length(filtered_follower_IDs),")"
#  ))
  # append to edge list
#  edge_df <- append_to_df(edge_df,list(from=followerScreenNameList,
#                                       to=rep(follower,
#                                              length(followerScreenNameList))))
#  counter <- counter + 1
#}
#save(edge_df, file = "edge_df20200917.Rda")
```

2.5.2 Tottenham게시트윗 연관계정 데이터셋 빈도표 출력

불러온 데이터로부터 연관계정의 빈도표를 출력한다.

```
load("edge_df20200917.Rda")                                    #######앞서 네트워크로 저장한 데이터 불러오기

# prepare network object


#net <- graph.data.frame(edge_df, directed=T)  #same with graph_from_data_frame()
net <- graph_from_data_frame(edge_df, directed=T)
class(net)
```

```
## [1] "igraph"
```

```
table(edge_df$to)
```

```
##
##     #Tottenham     AmanyireeD   authurmsipa5   AyomideJohn8   BrenoQuental
##             83             78             67             74             94
##      BwoyElvis Fredhutchings7  GEORGEMBITHI7      mbacke_el    MusaChibale
##             61             62             52             51             66
##  njugunaJimmy1   only1tuggers       osbakken      PMukhwana        siq1013
##             92             82             77             91             94
##  tunji_oyeleye
##             86
```

```
head(table(edge_df$from),5)
```

```
##
##        __3lvis        _CL_123   _Full__Time_ _members_Only_    _rafahuesca
##              1              1              1              1              1
```

```
#edge_df[to=="#Tottenham"]$from
#edge_df[from=="#Tottenham"]$to
```

***Tottenham*과 연관된 계정들의 *to/from*의 방향에 따른 빈도표의 모습**

2.5.3 Tottenham게시트윗 연관계정 Network 출력 및 시각화(igraph)

불러온 데이터로부터 연관계정의 빈도표를 출력한다.

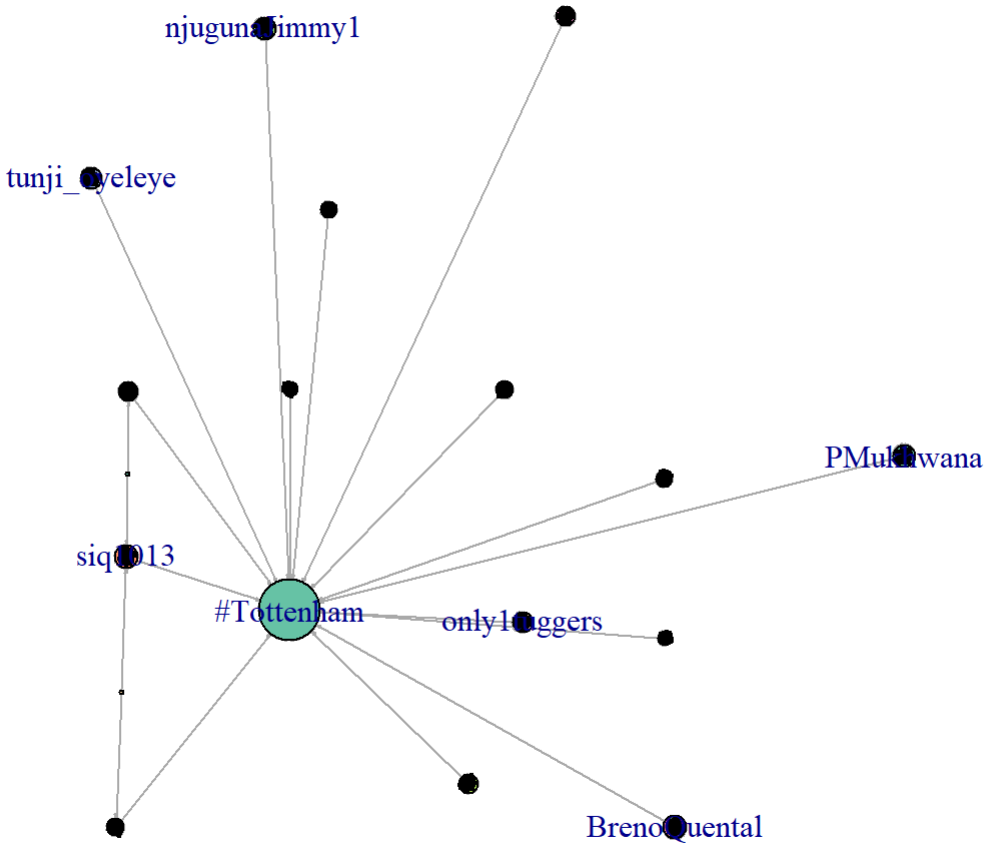
```
# simplify network
net <- simplify(net, remove.multiple = F, remove.loops = T)
# temp -> temp is deleted later using remove.loops option

# adjust the size of nodes based on in and out degrees
deg <- degree(net, mode="all")
V(net)$size <- deg*0.05 + 1
V(net)[name == coreUserName]$size <- 15
V(net)[size >= 5]$name
```

```
## [1] "PMukhwana"     "siq1013"       "njugunaJimmy1" "BrenoQuental"
## [5] "only1tuggers"  "tunji_oyeleye" "#Tottenham"
```

```
V(net)[name == coreUserName]$size
```

```
## [1] 15
```

```
# node coloring
pal3 <- brewer.pal(10, "Set2") #그래프에 사용될 색설정

# overall follower graph
op <- par(mar = c(0, 0, 0, 0))
plot(net, edge.arrow.size=0.1,
     #vertex.label = ifelse(V(net)$size >= 15, V(net)$name, NA),
     vertex.label = ifelse(V(net)$size >= 5, V(net)$name, NA),
     vertex.color = pal3)
```



토트넘 해시태그를 게시한 계정의 전체적인 *follower Network* 그래프의 모습

```
par(op)

################################################################################
#             Friends Among Follwers (optional)
################################################################################


# Plot to highlight Followers with large number of followers
deg <- degree(net, mode="out")
V(net)$size <- deg*0.05+2
V(net)[size==max(V(net)$size)]  #the most ties
```

```
## + 2/1209 vertices, named, from 6f28691:
## [1] JamesSpurs1882 COYS_com
```

토트넘을 해시태그한 계정중 사이즈*(edge*의 연결수*)*가장 큰 *vertex(*계정*)*를 보여준다

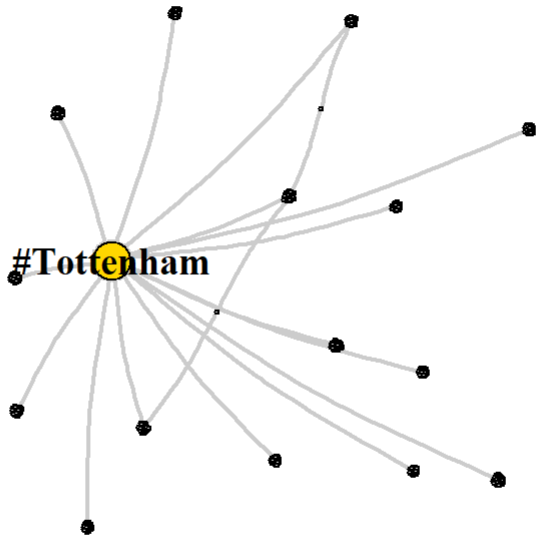
```
# Highlight the coreUser
V(net)[coreUserName]$size <- 15

# identify friend vertices (the vertices coreUserName is also following)
friendVertices <- ends(net, es=E(net)[from(coreUserName)])[,2]   #ends finds vertices at the ends
 of edges
#ends(net, es=E(net)[from(coreUserName)])[,2]
```

Vertex 및 *edge*의 *Attributes*를 조정하여 출력될 그래프의 모습을 설정한다

```
# Generate edge color variable: (normal: grey80, friend: red)
ecol <- rep("grey80", ecount(net))
ecol[which (V(net)$name %in% friendVertices)] <- 'red'

# Generate edge width variable: (normal: 2, friend: 4)
ew <- rep(2, ecount(net))
ew[which (V(net)$name %in% friendVertices)] <- 4

# add core_user for vertex coloring
friendVertices <- append(friendVertices,coreUserName)

# Generate node color variable: (normal: grey80, friend & coreUser: gold)
vcol <- rep("grey80", vcount(net))
vcol[which (V(net)$name %in% friendVertices)] <- "gold"

# vertex label size
V(net)$label.cex <- 1.2
```

Tottenham 연관계정 네트워크의 모습

```
plot(net,
     vertex.color=vcol,
     edge.color=ecol,
     edge.width=ew,
     edge.arrow.mode=0,
     vertex.label = ifelse(V(net)$name %in% friendVertices, V(net)$name, NA),
     vertex.label.color="black",
     vertex.label.font=2,
     edge.curved=0.1
)
```



***igraph*로 시각화된 *Tottenham* 관련계정의 네트워크 그래프 모습**

*3.*결론

해당 보고서에선 #Tottenham 키워드의 해시태그를 중심으로,
수집된 트윗의 일별 게시빈도,감성분석을 통한 감정분포,워드클라우드를 이용한 트윗중 빈출등장 단어의 시각화 및 igraph를 이용한 Tottenham을 둘러싼 계정 Network를 분석하고 알아보았다.

각 분석파트마다 해석을 명시해놨으며 이를통해 Social Network상 트윗 게시의 특징적 징후를 도출해냈고, 이를 검증하기위한 후속연구의 필요성도 제기할 수 있었다.

해당 보고서에서 도출한 특징적 징조및 결론 정리

<1.일별 트윗게시 빈도분석>

해석

1. **토트넘이 해시태그된 트윗은 *10*월 *19*일에 압도적인 게시수를 보여주고있다**
   EPL 경기일정을 조사해본 결과 10월 19일(월)에 토트넘VS웨스트햄전 경기가 치뤄졌는데, 해당 경기가 토트넘의 압도적인 우위속 경기를 리드하던중, 후반 마지막 10분경 3골을 내주며 극적인 무승부를 맞이하여서 팬들의 성화를 불러일으킨것으로 보인다.

2. ***10*월 *23*일 경기당시의 트윗은 매우 적은 게시빈도를 보여준다**
   10월 23일 토트넘은 LASK린츠와의 유로파리그 경기도 가졌는데 10월 19일의 EPL 웨스트햄전과는 상반되게 매우 적은 tweet수를 보여주고있다, 해당 경기는 웨스트햄과 비교하면 비교적 네임밸류가 적은상대와 치른 경기이며 매경기가 우승레이스에 직결되는 EPL과 다르게 유로파리그 초반일정에 해당하는 조별리그로 팬들에게 큰 기대를 사지 못했다고 생각된다.

일별 트윗빈도 해석결론 :

축구클럽에 대한 게시빈도는 해당하는 기간에 예정된 경기의 기대감요인에 반응하는것으로 짐작된다.

후속 연구를 통해 클럽이 치르는 경기에 따른 팬들의 기대감 요인(상대의 네임밸류/경기의중요성/리그중요성:해당 클럽리그순위와 다음순위로의 점수차이 등/더비여부)을 파악하고, 이를 검증해보고자 한다.

<2. Word colud>

해석

***1. EPL*내 경쟁구단에 대한 언급**

(1) 빈출단어로 현시점 리그 선두를 달리고 있는 Everton FC(1위)와 Liverpool FC(2위) 언급이 포함되어 있다.

(2) 북런던의 지역거점 라이벌구단인 Arsenal FC에 대한 언급도 빈번히 등장한것으로 보인다.

***2.* 최근 *Tottenham*경기에서 활약한 주축선수에 대한 언급**
최근 토트넘의 공격진에서 K-B-S조합으로 많은 기대를 받으며 대단한 활약을 보여주고있는 케인,손 그리고 베일에 대한 언급이 높은 빈도를 보여주고있다.

***2.* 무리뉴 감독에 대한 언급**
'2년차의 무리뉴', 세계적 Top명장의 반열에 속해있는 무리뉴는 부임 2년차에 반드시 우승컵을 들어올린다는 징크스가 축구팬 사이에서 유명하다.

무리뉴에 대한 높은 언급빈도는 토트넘의 지휘봉을 잡은지 2년차에 돌입한 무리뉴에 대한 기대감이 반영된 결과로 보인다.

<3. 감성분석>

해석

***(1) Positive*와*,Negative,trust*와 *anticipation* 순의 상위빈도 감정분포**

해당하는 상위빈도 네개의 감정은 축구클럽에 관련한 트윗에서 전형적으로 보일 수 있는 상위빈도의 감정인것으로 짐작된다.
이는 추후 후속연구를 통해서 많은 축구클럽의 트윗을 표본으로 검증해보고자 한다.

(2) Positive & negative

negative와 비교하면 높은 positive 점수를 보이고 있는 Tottenham의 트윗분석결과는 현재 구단에 대한 긍정적인 여론이 더 크다는 사실을 알려준다.
다만,이를 감안하고서라도 높은 negative점수는 최근 westham전에서의 무승부에 대한 팬들의 충격이 유의미하게 반영된 결과로 보인다.

(3) anticipation & trust

최근 팬들의 무리뉴호에 대한 높은 기대감과 신뢰가 반영되었음으로 짐작된다.