Text Analytics (Sentiment Analysis) of a New Agent 'Skye' in Valorant Game

Loading necessary libraries to execute text mining process from Twitter

```{r}
library(twitteR)
library(ROAuth)
library(RCurl)
library(httr)
library(stringr)
library(plyr)
library(dplyr)
library(tm)
```

Establishing a connection between R and Twitter server to collect data. This can be done by different methods but 'Direct Method' is used for its simplicity. Every method has some pros and cons. IDs for key and token have been deleted here for privacy.

```{r}
key=""
secret=""

atoken =  ""
asecret = ""

setup_twitter_oauth(key, secret, atoken, asecret)
```

Creating a function that takes four parameters. Here tweettext = original tweets, pos = list of positive words as vector, neg = list of negative sentiments as vector

```{r}
sentimentfun = function(tweettext, pos, neg, .progress='non')
```

Within function mentioned above there is another variable defined that takes another function within.

```{r}
{
scores = laply(tweettext,
        function(singletweet, pos, neg)
        {
```

Cleaning the tweets by removing irregularities like punctuation and digits. They are useless for text analytics and make the analysis complicated

```{r}
singletweet = gsub("[[:punct:]]", "", singletweet)
singletweet = gsub("[[:cntrl:]]", "", singletweet)
singletweet = gsub("\\d+", "", singletweet)
```

Finding errors and converting texts into lower case. Converting texts into lowercase is required to maintain throughout consistency

```{r}
tryTolower = function(x)
        {
         y = NA
         try_error = tryCatch(tolower(x), error=function(e) e)
```

```
                if (!inherits(try_error, "error"))
                  y = tolower(x)
                return(y)
               }
```

Converting each tweet into words to assign score to each tweet

```{r}
singletweet = sapply(singletweet, tryTolower)
word.list = str_split(singletweet, "\\s+")
words = unlist(word.list)
```

Comparing words from tweets from list of positive and negative words

```{r}
pos.matches = match(words, pos)
neg.matches = match(words, neg)
```

Only tweets that matches list of words would be used

```{r}
pos.matches = !is.na(pos.matches)
neg.matches = !is.na(neg.matches)
```

Final score is calculated and assigned to required variables

```{r}
 score = sum(pos.matches) - sum(neg.matches)
 return(score)
```

}, pos, neg, .progress=.progress)

```

Creating data frame containing tweet texts and scores

```{r}

sentiment.df = data.frame(text=tweettext, score=scores)

return(sentiment.df)

}

```

Scraping 2500 tweets that contain keywords 'valorant' and 'skye' in 'english' language starting from 11-Nov-2020 and ends on 09-Nov-2020

```{r}

tweets = searchTwitter("valorant+skye", n=2500,

lang="en", since = '2020-11-02', until = '2020-11-09'

)

```

Extracting texts while cleaning by removing links and other irregularities

```{r}

tweettext = sapply(tweets, function(x) x$getText())

## first cleaning stage

tweettext=lapply(tweettext, function(x) iconv(x, "latin1",

"ASCII", sub=""))

tweettext=lapply(tweettext, function(x) gsub("htt.*",' ',x))

tweettext=lapply(tweettext, function(x) gsub("#",'',x))

tweettext=unlist(tweettext)

```

List of words in the form of text file that has to be present in working directory

```{r}

pos = readLines("positive-words.txt")

neg = readLines("negative-words.txt")

```

Extracting other elements to the data frame other than tweets and scores, removing duplicates, creating data frame and exporting it as a 'csv' file

```{r}

scores = sentimentfun(tweettext, pos, neg, .progress='text')


tweetdate=lapply(tweets, function(x) x$getCreated())

tweetdate=sapply(tweetdate,function(x) strftime(x, format="%Y-%m-%d %H:%M:%S",tz = "UTC"))


isretweet=sapply(tweets, function(x) x$getIsRetweet())


retweetcount=sapply(tweets, function(x) x$getRetweetCount())


favoritecount=sapply(tweets, function(x) x$getFavoriteCount())


data=as.data.frame(cbind(ttext=tweettext,
                         date=tweetdate,
                         isretweet=isretweet,
                         retweetcount=retweetcount,
                         favoritecount=favoritecount,
                         score = scores$score
                        ))
```
```

```
data2 = duplicated(data[,1])


data$duplicate = data2


write.csv(data, file= "Valorant_Skye.csv")
```