R_Assignment_twitter

Laxman

10/2/2020

Q1 : 1. Extraction of live data from the relevant datasource which the candidates can choose, the reference of which should be clearly mentioned in this report.

The date used is Twitter data and is extracted from apps.twitter.com (developer account)

#loading twitteR Package

```
library(twitteR)
```

#Connecting to the app

```
api_key<-'IbtnD7jelY71oB2ZOW0MogI2Q'
api_secret<-'enFOEdlDbx9GgXrbH8s1mOyOAj9zVrZ6qtMP4nb9mkoeqNfSel'
access_token<-'987964433711300614-TTYwLNT79dbdfxMlby5LPOyiYIzkCiS'
access_token_secret<-'loBbDmoZeylVOlw9iJWC6JfhDBCcTl4Xd4ZzfCDeLM5lZ'
setup_twitter_oauth(api_key, api_secret, access_token, access_token_secret)
```

```
## [1] "Using direct authentication"
```

2. Data Preprocessing & cleaning in R

DATA PROCESSING

#Extract tweets

```
tweets<-searchTwitter("CSKvSRH",n=1000,lang = "en")
```

#converting tweets to Data Frame

```
tweets.df<-twListToDF(tweets)
```

#Extracting text from tweets

```
tweets_text <- tweets$text
str(tweets_text)
```

```
##  NULL
```

DATA CLEANING

#Cleaning the data i.e. http, emoji, punctuations, and other symbols

```
tweets.df$text=gsub("&amp", "", tweets.df$text)
tweets.df$text = gsub("&amp", "", tweets.df$text)
tweets.df$text = gsub("(RT|via)((?:\\b\\W*@\\w+)+)", "", tweets.df$text)
tweets.df$text = gsub("@\\w+", "", tweets.df$text)
tweets.df$text = gsub("[[:punct:]]", "", tweets.df$text)
tweets.df$text = gsub("[[:digit:]]", "", tweets.df$text)
tweets.df$text = gsub("http\\w+", "", tweets.df$text)
tweets.df$text = gsub("[ \t]{2,}", "", tweets.df$text)
tweets.df$text = gsub("^\\s+|\\s+$", "", tweets.df$text)
tweets.df$text = gsub("http[^[:blank:]]+","",tweets.df$text)
tweets.df$text = gsub("[^\x01-\x7F]","",tweets.df$text)
```

#loading tm Library

```
library(tm)
```

```
## Loading required package: NLP
```

```
amzn<-Corpus(VectorSource(tweets.df$text))
amzn<-tm_map(amzn,content_transformer(removeNumbers))
```

```
## Warning in tm_map.SimpleCorpus(amzn, content_transformer(removeNumbers)):
## transformation drops documents
```

```
amzn<-tm_map(amzn,removePunctuation)
```

```
## Warning in tm_map.SimpleCorpus(amzn, removePunctuation): transformation drops
## documents
```

```
amzn<-tm_map(amzn,content_transformer(tolower))
```

```
## Warning in tm_map.SimpleCorpus(amzn, content_transformer(tolower)):
## transformation drops documents
```

```
amzn<-tm_map(amzn,stripWhitespace)
```

```
## Warning in tm_map.SimpleCorpus(amzn, stripWhitespace): transformation drops
## documents
```

```
amzn<-tm_map(amzn,removeWords,stopwords("english"))
```

```
## Warning in tm_map.SimpleCorpus(amzn, removeWords, stopwords("english")):
## transformation drops documents
```

```
mystopword<-"cskvsrh"
amzn<-tm_map(amzn,removeWords,mystopword)
```

```
## Warning in tm_map.SimpleCorpus(amzn, removeWords, mystopword): transformation
## drops documents
```

3. Data Visualisation in R

#loading library for wordclod

```
library(wordcloud)
```

```
## Loading required package: RColorBrewer
```

#colour

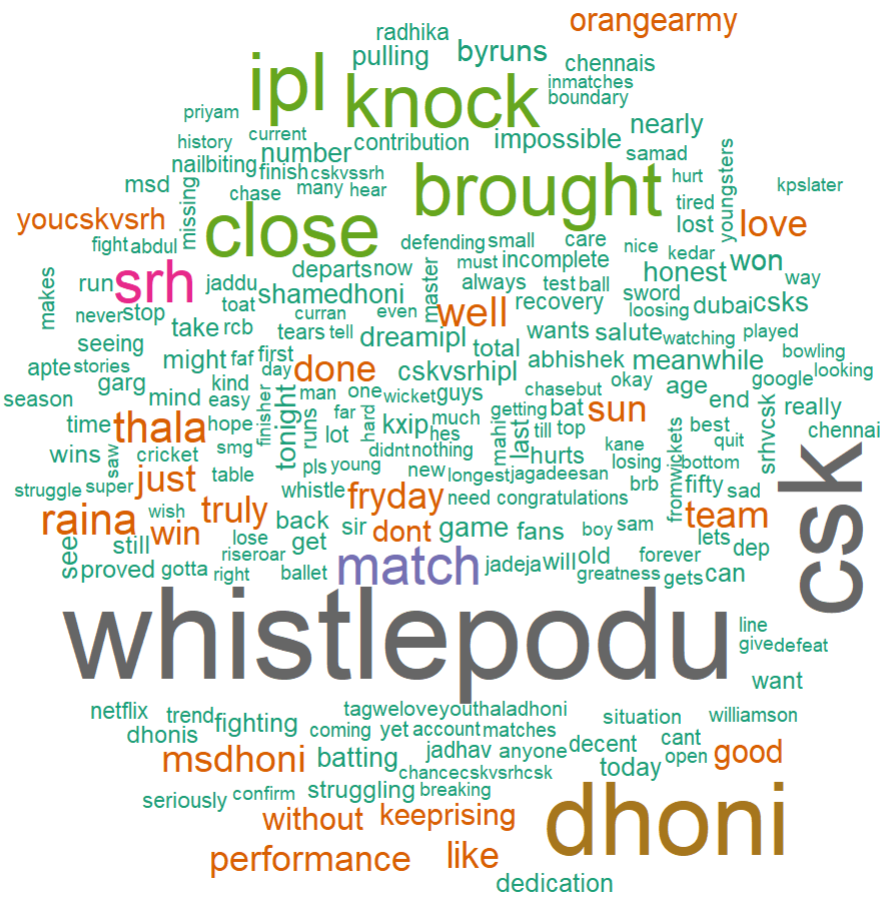
```
pal<-brewer.pal(8,"Dark2")
```

#wordcloud

```
wordcloud(amzn,min.freq = 7,max.words = Inf,colors = pal)
```

```
## Warning in wordcloud(amzn, min.freq = 7, max.words = Inf, colors = pal): yellove
## could not be fit on page. It will not be plotted.
```

```
## Warning in wordcloud(amzn, min.freq = 7, max.words = Inf, colors = pal):
## whistlefromhome could not be fit on page. It will not be plotted.
```



4. Exploratory or any other type of analysis

analysis performed

1.Number of tweets per month 2.Number of tweets per Day 3.Comparision between tweets and retweets
4.Number of tweets based on the Emotion.

```
library(ggplot2)
```

```
##
## Attaching package: 'ggplot2'
```

```
## The following object is masked from 'package:NLP':
##
##     annotate
```

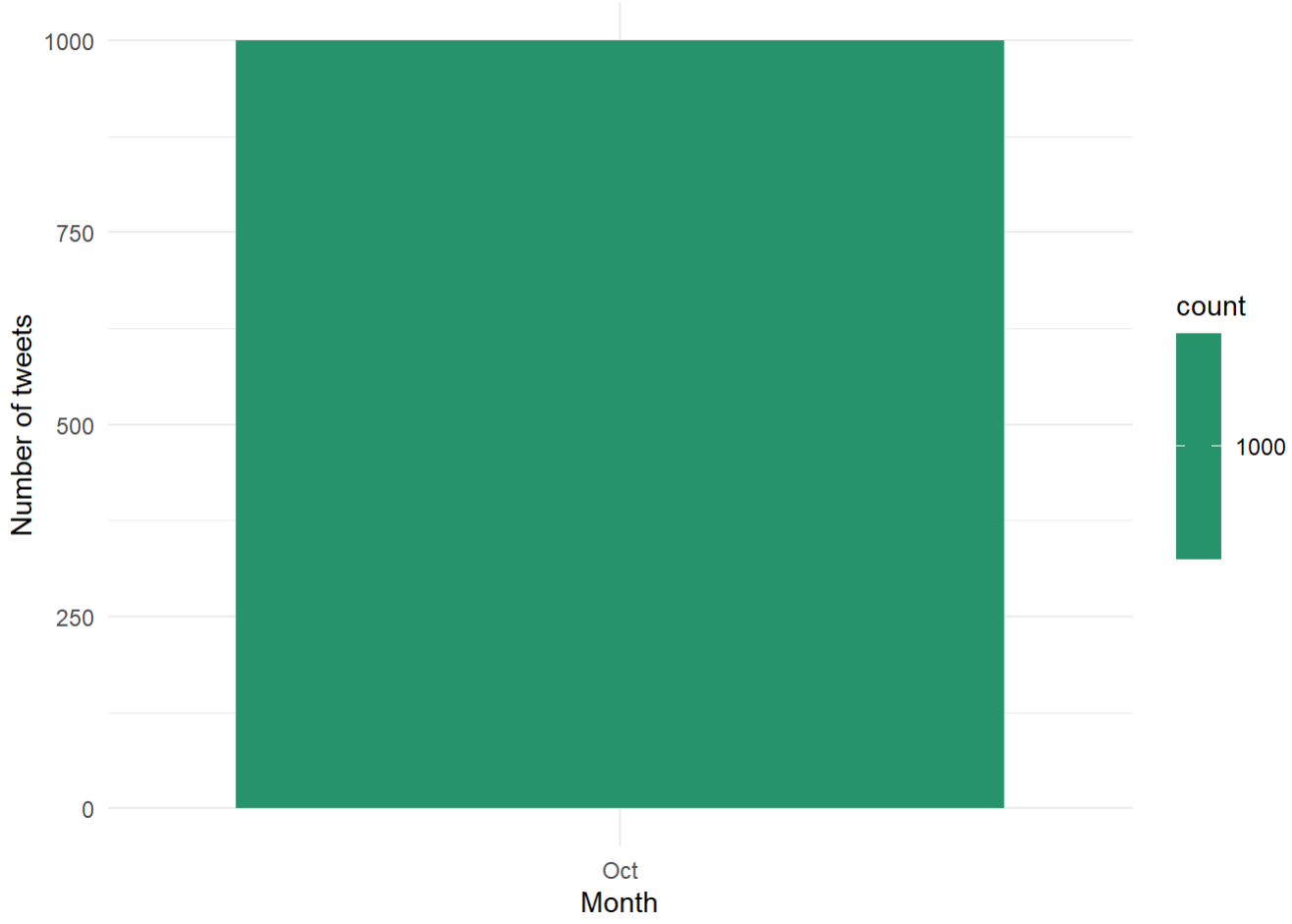
```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

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

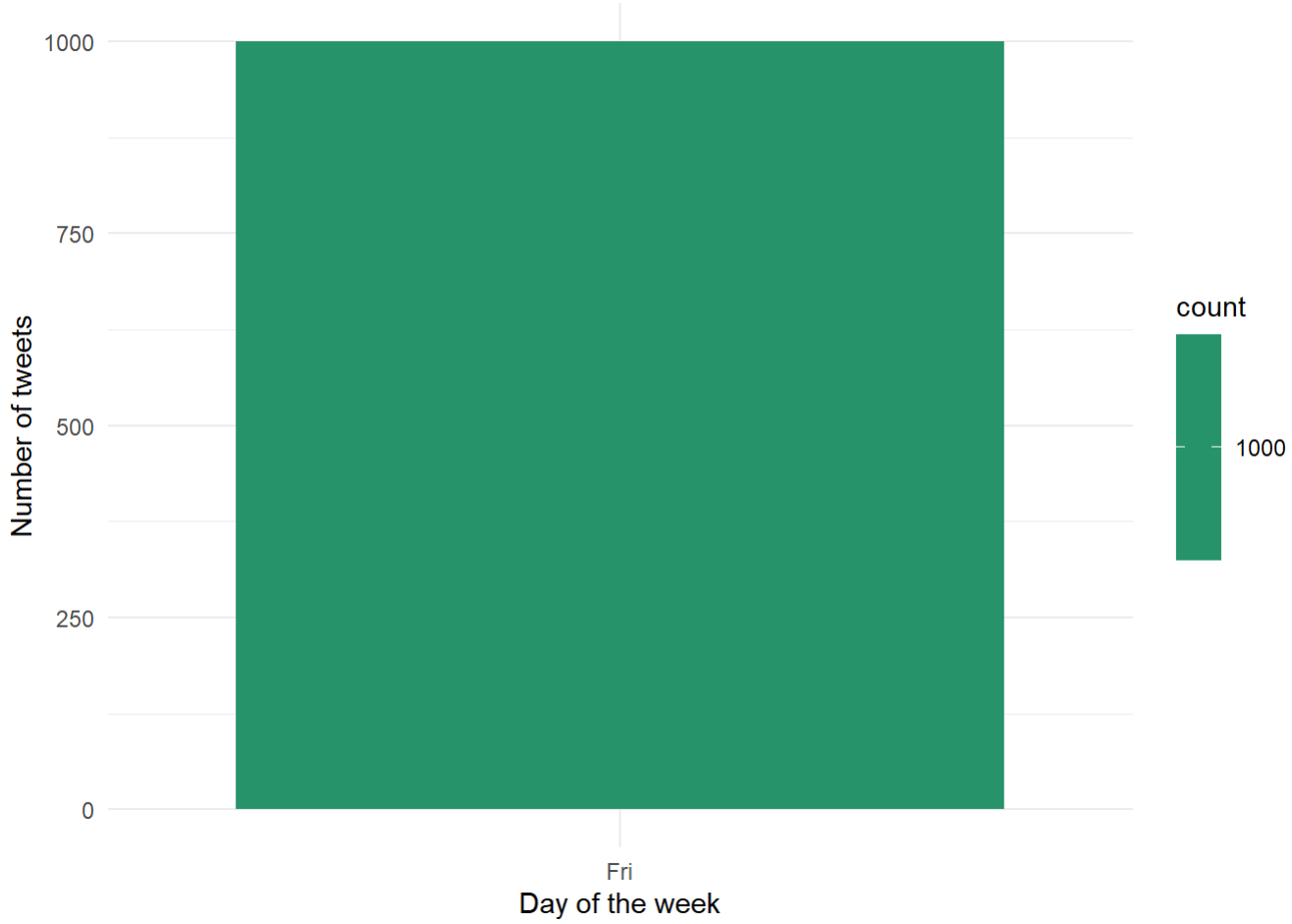
#Number of tweets per month

```
ggplot(data = tweets.df, aes(x = month(created, label = TRUE))) +
  geom_bar(aes(fill = ..count..)) +
  xlab("Month") + ylab("Number of tweets") +
  theme_minimal() +
  scale_fill_gradient(low = "turquoise3", high = "darkgreen")
```



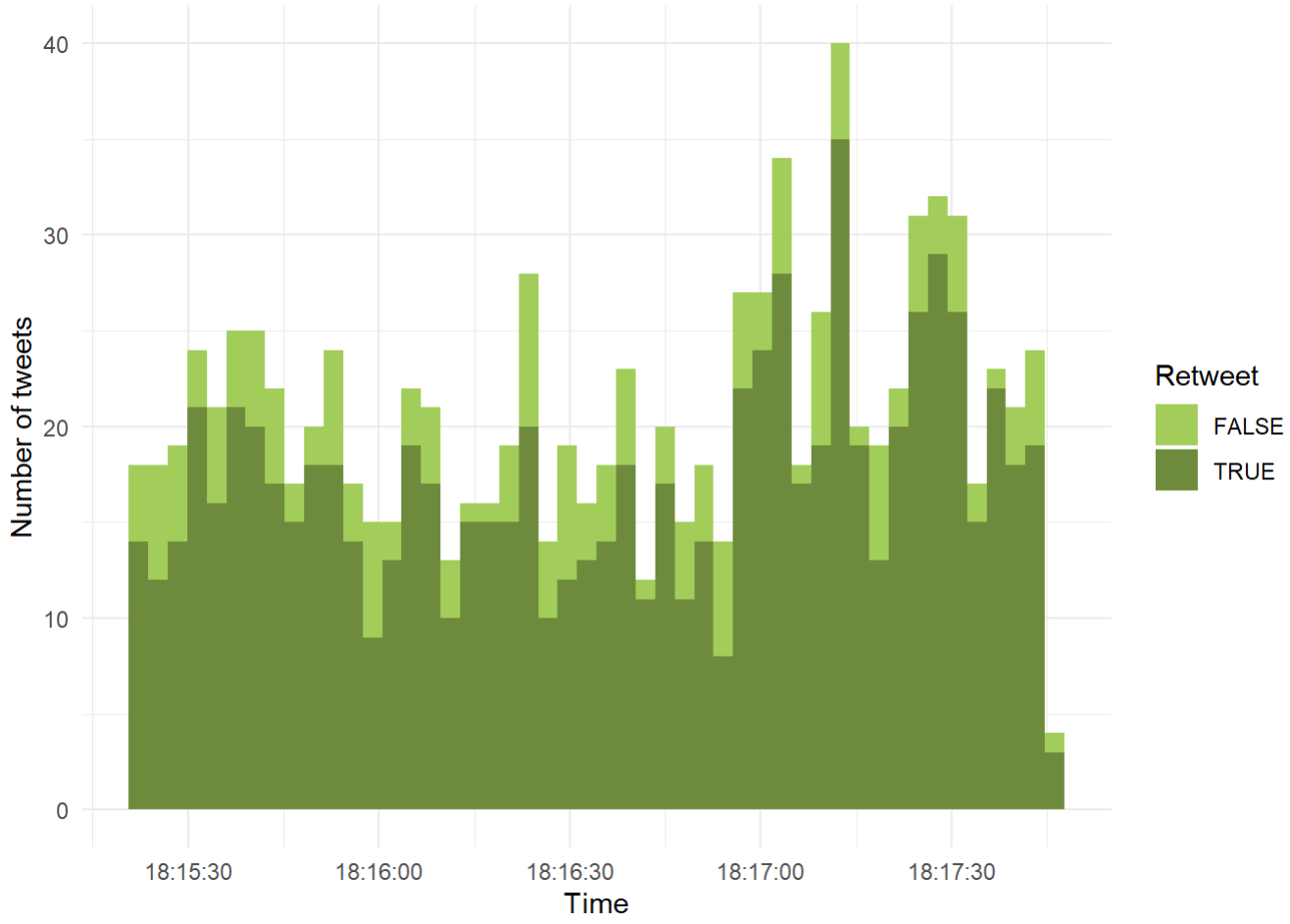
#number of tweets per Day

```
ggplot(data = tweets.df, aes(x = wday(created, label =  TRUE))) +
  geom_bar(aes(fill = ..count..)) +
  xlab("Day of the week") + ylab("Number of tweets") +
  theme_minimal() +
  scale_fill_gradient(low = "turquoise3", high = "darkgreen")
```



#comparision between tweets and retweets

```
ggplot(data = tweets.df, aes(x = created, fill = isRetweet)) +
  geom_histogram(bins=48) +
  xlab("Time") + ylab("Number of tweets") +
  theme_minimal() +
  scale_fill_manual(values = c("darkolivegreen3","darkolivegreen4"), name = "Retweet")
```



#diving tweets to emotions

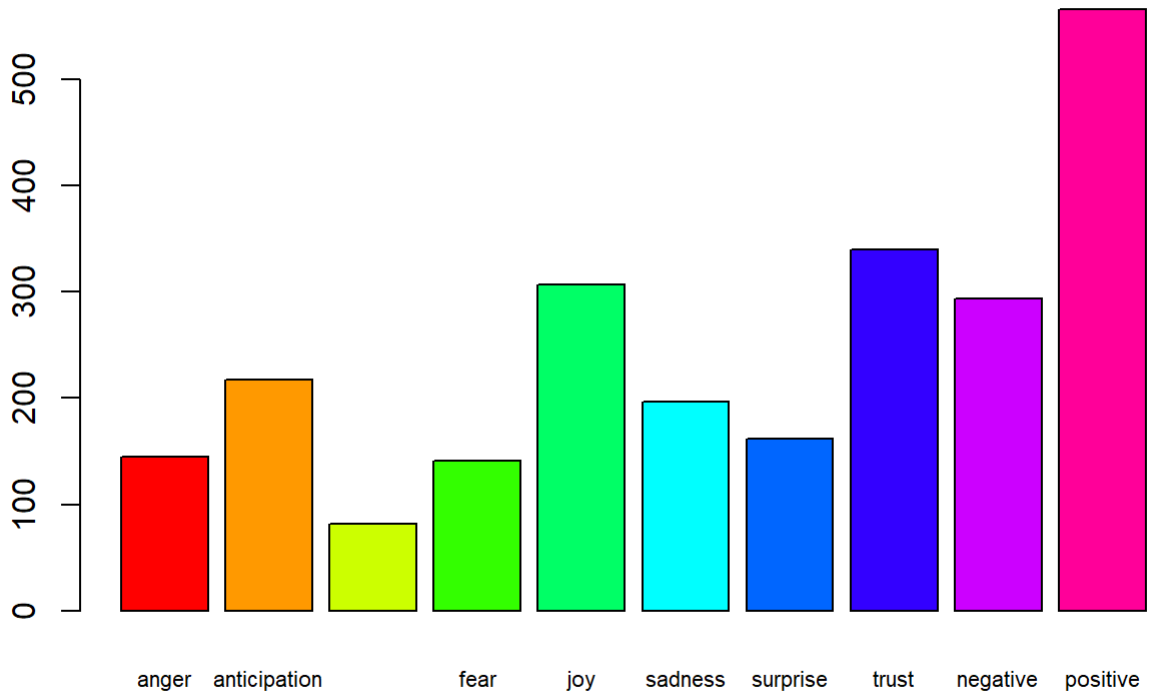
```
library(syuzhet)
```

```
Emotion_IPL<-get_nrc_sentiment(tweets.df$text)
```

```
## Warning: `data_frame()` is deprecated as of tibble 1.1.0.
## Please use `tibble()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```

#Bar plot for tweets per emotion

```
barplot(colSums(Emotion_IPL),cex.names = .7,col = rainbow(10),main = "Emotion score for SRH vs C
SK")
```


Emotion score for SRH vs CSK
0
100
200
300
400
500
anger
anticipation
fear
joy
sadness
surprise
trust
negative
positive