#Step 1: Load the required packages (including rtweet) in RStudio

library(rtweet)

library(textdata)

library(ngram)

library(janeaustenr)

library(reshape2)

library(devtools)

library(widyr)

library(lubridate) # Date & Time

library(ggplot2)# Data Visualisation

library(dplyr) #Data Wrangling

library(tidytext) #Text Mining

library(tm) #Text Mining

library(wordcloud)

library(readr)

library(tidyr) #Tidy Text

library(RColorBrewer) #Data Visualisation

library(stringr) #String Manipulation

library(RSentiment) #Sentiment Analysis

library(cowplot) #Plot Arrange

library(ggthemes) #Data Visualisation

library(knitr)

library(kableExtra)

#Step 2: Authenticate using your credentials to Twitter's API by creating an access token. Steps on getting Twitter access tokens:

create\_token(app\_name <- "Thinkkade",

api\_key <- "0H4vX3CkqWnoijZ2oisoaa7zo",

api\_secret <- "1t0Mcnu8mZX3FrAoygYWYwfNXJOj09DrDGjVjtHuryyCVQNY39",

access\_token <- "415803231-zy660bncbZR5RoJSsgnkkGeV4aMILpyJtOEyyupi",

access\_token\_secret <- "Nn0vYP6hmtcD0oXjzSbvQsFhwdzEyAWDKgT9nckCBXn2p")

#Step 3: search tweets

virus <- search\_tweets('#COVID-19 + #Coronavirus', n = 20000, since = '2020-01-01', retryOnRateLimit = 1e3, en = 'lang')

virus

#Step 4: Process each set of tweets into tidy text or corpus objects.

tweets.virus = virus %>% select(screen\_name,text)

tweets.virus

#Step 5: Use pre-processing text transformations to clean up the tweets; this includes stemming words. An example of stemming is rolling the words "computer", "computational" and "computation" to the root "comput".

head(tweets.virus$text)

#Remove http elements manually

tweets.virus$stripped\_text1 <- gsub("http\\s+","",tweets.virus$text)

#vuse the unnest\_tokens() function to convert to lowercase,

#remove punctuation, and add id for each tweet

tweets.virus\_stem <- tweets.virus %>%

select(stripped\_text1) %>%

unnest\_tokens(word, stripped\_text1)

head(tweets.virus\_stem)

# remove stop words from your list of words

cleaned\_tweets.virus <- tweets.virus\_stem %>%

anti\_join(stop\_words)

head(cleaned\_tweets.virus)

#Step6: The top 30 commonly used words in the set of tweets for Covid-19; this gives an overall picture of what people are most concerned about, and the extent to which they are engaged on the topic.#Top 10 words in #Covid\_19 tweets

cleaned\_tweets.virus %>%

count(word, sort = TRUE) %>%

top\_n(30) %>%

mutate(word = reorder(word, n)) %>%

ggplot(aes(x = word, y = n)) +

geom\_col() +

xlab(NULL) +

coord\_flip() +

theme\_classic() +

labs(x = "count",

y = "Unique words",

title = "Unique word counts found in #Covid-19tweets")

#Step 7A: Perform sentiment analysis using the Bing lexicon and get\_sentiments function from the tidytext package.

#bing sentiment analysis

bing\_virus = cleaned\_tweets.virus %>%

inner\_join(get\_sentiments("bing")) %>%

count(word, sentiment, sort = TRUE) %>%

ungroup()

bing\_virus

#Step 7B: Then to visually depict the word counts, you can filter and plot the words side-by-side to compare the positive vs negative emotion.

bing\_virus %>%

group\_by(sentiment) %>%

top\_n(30) %>%

ungroup() %>%

mutate(word = reorder(word, n)) %>%

ggplot(aes(word, n, fill = sentiment)) +

geom\_col(show.legend = FALSE) +

facet\_wrap(~sentiment, scales = "free\_y") +

labs(title = "Tweets containing '#Covid-19'",

y = "contribution to sentiment",

x = NULL) +

coord\_flip() + theme\_bw()

#Step 8: Apply the function to the sets of tweets:

#Apply the function

#Apply the lapply function returns a list of all the sentiment scores, types, and tables of tweets

virus\_sent = lapply(virus$text,function(x){sentiment\_bing(x)})

virus\_sent

#Step 9: Then to visually depict the word counts, you can filter and plot the words side-by-side to compare the positive vs negative emotion(7B Contd)

bing\_virus %>%

group\_by(sentiment) %>%

# Take the top 10 for each sentiment

top\_n(30) %>%

ungroup() %>%

# Make word a factor in order of n

mutate(word = reorder(word, n)) %>%

# Use aes() to put words on the x-axis and n on the y-axis

ggplot(aes(word, n, fill = sentiment)) +

# Make a bar chart with geom\_col()

geom\_col(show.legend = FALSE) +

geom\_text (aes(label = n, hjust=1), size = 3.5, color = "black")+

facet\_wrap(~sentiment, scales = "free") +

coord\_flip() +

ggtitle("Most common positive and negative words#Covid-19")

#Step 10: Bing overall sentiment.

bing\_virus %>%

group\_by(sentiment) %>%

summarise(word\_count = n()) %>%

ungroup() %>%

mutate(sentiment = reorder(sentiment, word\_count)) %>%

ggplot(aes(sentiment, word\_count, fill = sentiment)) +

geom\_col() +

guides(fill = FALSE) +

labs(x = NULL, y = "Word Count") +

scale\_y\_continuous(limits = c(0, 1000)) +

ggtitle("Covid-19 Bing Sentiment") +

coord\_flip()

#Step 11: wordcloud plot the 50 most common words

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

tidy\_virus = cleaned\_tweets.virus %>%

anti\_join(stop\_words) %>%

count(word) %>%

with(wordcloud(word, n, random.order = FALSE, max.words = 50, colors=pal))

#Sentiment word cloud: Classifying the words into different types of emotions also helps us understand how people are feeling towards a subject

#Wordcloud:the most common positive and negative words

tidy\_virus = cleaned\_tweets.virus %>%

inner\_join(get\_sentiments("bing")) %>%

count(word, sentiment, sort = TRUE) %>%

acast(word ~ sentiment, value.var = "n", fill = 0) %>%

comparison.cloud(colors = c("gray20", "gray80"),

max.words = 50)

covidvirus=subset(virus,user\_id=="virus")

covidvirus=covidvirus %>% filter(!(is\_retweet=="True"))

clean=function(x){

x$text=str\_replace\_all(x$text,'[^[:alnum:]]',' ')

x$text=gsub("[[:digit:]]"," ",x$text)

x$text=gsub("[\t]{2,}"," ",x$text)

x$text=gsub("https.\*"," ",x$text)

return(x)

}

temp=subset(virus,!(is.na(virus$text)))

temp=as.data.frame(clean(temp))

temp=temp %>% unnest\_tokens(word,text) %>% count(user\_id ,word,sort=TRUE) %>% ungroup()

total\_words=temp %>% group\_by(user\_id) %>% summarise(count=n())

temp=left\_join(temp,total\_words)

kable(head(temp,20),"html") %>% kable\_styling(bootstrap\_options="condensed",position="center")

ggplot(temp,aes(n/count,fill=user\_id))+

geom\_histogram(show.legend = FALSE)+

facet\_wrap(~user\_id,scales="free")+

xlim(NA,0.030)+

labs(x="TF",y="Count",title="Term Frequency for Covid-19")

#Bigram

temp=clean(virs)

temp=temp %>% select(user\_id,text) %>% unnest\_tokens(bigram,text,token="ngrams",n=2)

kable(head(temp,10),"html") %>% kable\_styling(bootstrap\_options="condensed",position="center")

tempseperated=temp %>% separate(bigram,c("word1","word2"),sep=" ")

tempfiltered=tempseperated %>% filter(!(word1 %in% stop\_words$word)) %>%

filter(!(word2 %in% stop\_words$word))

temp=tempfiltered %>%

unite(bigramwords,word1,word2,sep=" ") %>%

group\_by(bigramwords,user\_id) %>% tally()%>%

ungroup() %>% arrange(desc(n)) %>%

mutate(bigramwords=factor(bigramwords,levels=rev(unique(bigramwords))))

temp %>% group\_by(user\_id) %>%

top\_n(20) %>% ggplot(aes(bigramwords,n,fill=user\_id))+

geom\_col(show.legend=FALSE)+

labs(x="",y="Count")+

facet\_wrap(~user\_id,ncol=2,scales="free")+coord\_flip()