I have chosen the first data set which is about COVID 19 Tweets. I have analyzed on thedata set in the form of word cloud and by counting on the frequency of the words. Covid 19 tweets wereall around the world and in different sources. Text mining techniques helps us identify the key words which will lead us to business insights. This data set has huge number of rows and 18 columns namely name, address, hashtag, followers, userid, text etc.…

Analysis:

I have initially installed the packages;

```
# Install the packages
install.packages("tm")
install.packages("SnowballC")
install.packages("wordcloud")
install.packages("RColorBrewer")
```

Here tm is for text mining which will be helpful for mining the text data, snowball is used for text stemming, wordcloud for word cloud formation and RColorBrewer for adding the color pallete.

I have imported the dataset as covid19tweets from the paths specified from my workspace in the computer to explore the data.

covid19tweets <- read.csv ("c:/Users/Harish Bodasinghi/ Desktop/ covid19_tweets.csv")

I have explored the data, using summary command to find the maximum, minimum, 1st and 3rd quartile.

```
> summary(covid19tweets)
  user_name         user_location      user_description   user_created       user_followers      user_friends
 Length:179108      Length:179108      Length:179108      Length:179108      Min.   :       0   Min.   :     0
 Class :character   Class :character   Class :character   Class :character   1st Qu.:     172   1st Qu.:   148
 Mode  :character   Mode  :character   Mode  :character   Mode  :character   Median :     992   Median :   542
                                                                            Mean   :  109056   Mean   :  2122
                                                                            3rd Qu.:    5284   3rd Qu.:  1725
                                                                            Max.   :49442559   Max.   :497363
 user_favourites   user_verified          date               text              hashtags            source
 Min.   :      0   Length:179108      Length:179108      Length:179108      Length:179108      Length:179108
 1st Qu.:    206   Class :character   Class :character   Class :character   Class :character   Class :character
 Median :   1791   Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character
 Mean   :  14444
 3rd Qu.:   9388
 Max.   :2047197
  is_retweet
 Length:179108
 Class :character
 Mode  :character
```

Here the data set has 179108 rows and 13 rows in it.

```
> nrow(covid19tweets)
[1] 179108
> ncol(covid19tweets)
[1] 13
>
```

To convert the dataset, into text file we use readLines and once converted we further pursue the analysis;

filepath <- (“c:/ Users/ Harish Bodasinghi/ Desktop/ covid19_tweets.csv”)
text <- readLines (filepath)

The data is loaded as corpus and inspected as docs.

```
> toSpace <- content_transformer(function (x , pattern ) gsub(pattern, " ",
+                                                             x))

> docs <- tm_map(docs, toSpace, "/")

> docs <- tm_map(docs, toSpace, "@")

> docs <- tm_map(docs, toSpace, "\\|")
```

The above code executes, and they eliminate the /, @ and \\ from the text file. While doing text mining it is essential to remove all such blocks and clean the data.

Text before removing the /, @ and \\

```
   [3] Tom Basile us,"New York, NY","Husband, Father, Columnist & Commentator. Author of Tough Sell: Fig
r in Iraq. Bush Admin Alum. Newsmax Contributor. Fmr Exec Dir NYSGOP",2009-04-16 20:06:23,2253,1677,24,
2:27:17,Hey @Yankees @YankeesPR and @MLB - wouldn't it have made more sense to have the players pay the
 A… https://t.co/1QvW0zgyPu,,Twitter for Android,False
```

Text after removing the /, @ and \\

```
   [3] tom basile 🎈🎈,"new york, ny","husband, father, columnist & commentator. author of tough sell: fighting the media
 war in iraq. bush admin alum. newsmax contributor. fmr exec dir nysgop",2009-04-16 20:06:23,2253,1677,24,true,2020-07-25
 12:27:17,hey  yankees  yankeespr and  mlb - wouldn't it have made more sense to have the players pay their respects to t
he a… https:  t.co 1qvw0zgypu,,twitter for android,false
```

From the above, we can find that the symbols are removed.

```
> # Convert the text to lower case
> docs <- tm_map(docs, content_transformer(tolower))
```

Before converting the text from upper case to lower case:

```
   [3] Tom Basile us,"New York, NY","Husband, Father, Columnist & Commentator. Author of Tough Sell: Fighting the Media Wa
r in Iraq. Bush Admin Alum. Newsmax Contributor. Fmr Exec Dir NYSGOP",2009-04-16 20:06:23,2253,1677,24,True,2020-07-25 1
2:27:17,Hey @Yankees @YankeesPR and @MLB - wouldn't it have made more sense to have the players pay their respects to the
 A… https://t.co/1QvW0zgyPu,,Twitter for Android,False
```

After converting text from upper case to lower case:

```
   [3] tom basile 🎈🎈,"new york, ny","husband, father, columnist & commentator. author of tough sell: fighting the media
 war in iraq. bush admin alum. newsmax contributor. fmr exec dir nysgop",2009-04-16 20:06:23,2253,1677,24,true,2020-07-25
 12:27:17,hey  yankees  yankeespr and  mlb - wouldn't it have made more sense to have the players pay their respects to t
he a… https:  t.co 1qvw0zgypu,,twitter for android,false
```

We can find that the capital letter was changed to lower case letters.

Similarly, I have checked for the below code, and it removes numbers, punctuations and stop words. If there any particular stop words we can mention them and can remove them from the docs.

```
> # Remove numbers
> docs <- tm_map(docs, removeNumbers)
```

```
> # Remove english common stopwords
> docs <- tm_map(docs, removeWords, stopwords("english"))

> # Remove your own stop word
> # specify your stopwords as a character vector
> docs <- tm_map(docs, removeWords, c("blabla1", "blabla2"))

> # Remove punctuations
> docs <- tm_map(docs, removePunctuation)

> # Eliminate extra white spaces
> docs <- tm_map(docs, stripWhitespace)
```

I have done the text stemming which is finding the root word and maintaining the document with root words but not their forms with different tenses.

```
> # Text stemming
> # docs <- tm_map(docs, stemDocument)
> dtm <- TermDocumentMatrix(docs)
```

Convert into matrix, by using term document matrix as dtm and further converting into matrix with limitation of 3000, initially while working with the whole data, sorting the rows in decreasing order we then form a word cloud.

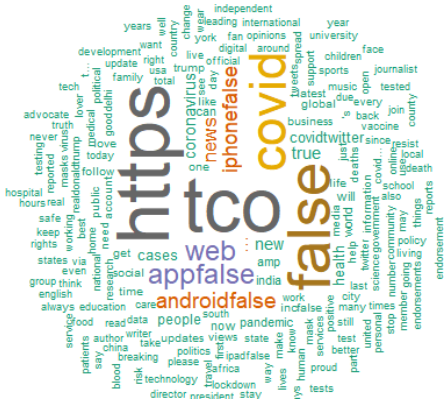
The head of the resultant data is mentioned below:

```
> head(d, 10)
                     word   freq
tco                   tco 199251
https               https 196720
false               false 156610
covid               covid 128012
web                   web  58202
appfalse         appfalse  57363
androidfalse androidfalse  40522
iphonefalse   iphonefalse  35473
news                 news  30651
…                       …  29377
```

tco, https, false, covid are the top words we can find from the above words which are mentioned.

For forming the wordcloud, we can use the below the below code to get a visually interactive graph,

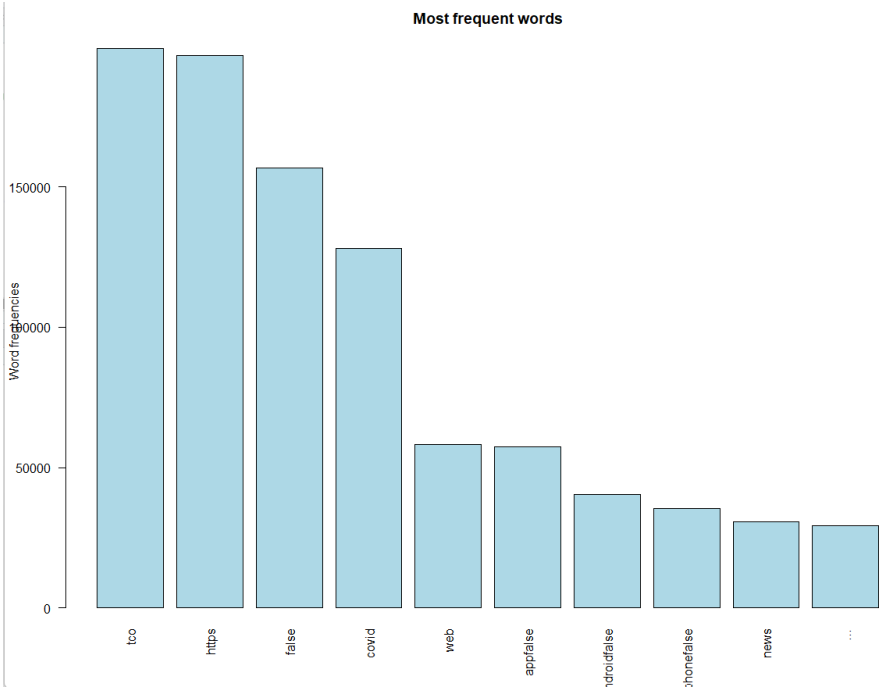
```
> wordcloud(words = d$word, freq = d$freq, min.freq = 1,
+           max.words=200, random.order=FALSE, rot.per=0.35,
+           colors=brewer.pal(8, "Dark2"))
```



After identifying the top words, we then try to do a bar plot with the frequency along with words.

Here is the barplot, with the most frequent words and their frequency, with light blue color.

```
> barplot(d[1:10,]$freq, las = 2, names.arg = d[1:10,]$word,
+         col ="lightblue", main ="Most frequent words",
+         ylab = "Word frequencies")
```



Tco and https, false and covid are top words with more frequency than all other top words. It is obvious that false and covid words pop up often as this dataset was collected during the covid time in 2020. This analysis gives us insight on the top frequent words, hashtags which can be useful for business growth and determining the trends.