**Assignment 3**

Data link: <https://app.box.com/s/vjbv27r081a2d8doy3o1ar27j3zfnjpc>

1) (20 points) The data file contains tweets that have been pulled from Twitter. In this dataset use the text data in the “OriginalTweet” column and perform the following:

Step1) First we need to read the csv file and assign it to data.

Code:

data = read.csv("~/Downloads/Corona\_NLP\_test.csv")

data

dim(data)

View(data)

Graphical user interface, application

Description automatically generated

Step2) Convert the text corpus into tokens.

We need to install the required packages text2vec, dplyr and tidytext. Then we need to convert the text corpus into tokens using tibble and unnest\_tokens functions .

Code:

install.packages("text2vec")

library(text2vec)

install.packages("tidytext")

library(tidytext)

library(dplyr)

tokens <- tibble(text = data\_tweet) %>%

unnest\_tokens(word, text)

print(tokens)

dim(tokens)

View(tokens)

Graphical user interface, application

Description automatically generated

Step3) Perform stop word removal.

Here stop words are the list contains words like "a", "an", "the", "and", "or", "but", "is", "are", "was", "were", "of", "in", "on", "to", "with", and so on.

We define the stop words list using the stop\_words$word function from the tidytext package. We then remove the stop words from the tokenized corpus by filtering out any rows where the word column matches a stop word using the filter() function.

Code:

stop\_words <- stop\_words$word

tokens\_no\_stop <- tokens %>%

filter(!word %in% stop\_words)

print(tokens\_no\_stop)

dim(tokens\_no\_stop)

View(tokens\_no\_stop)

Graphical user interface, application

Description automatically generated

Step4) Count Word frequencies

To count the word frequencies, we use the count() function from dplyr to count the number of occurrences of each word in the tokens data frame. We set the sort argument to TRUE to sort the results by frequency.

Code:

word\_counts <- tokens\_no\_stop %>%

count(word, sort = TRUE)

print(word\_counts)

dim(word\_counts)

View(word\_counts)

Graphical user interface, text, application

Description automatically generated

Step5) Create word clouds.

Finally, we create a word cloud using the wordcloud() function from the wordcloud package. We pass in the word column and n column from word\_counts as the words and freq arguments, respectively. We also set the maximum number of words to show using max.words, and set random.order to FALSE to show the most frequent words first.

Graphical user interface, text, application

Description automatically generated

Text

Description automatically generated