Week 5: Lab - Word Cloud Chapter Challenge

Alex Koener

2/14/21

# Instructions

Create a word cloud using the text column from a real twitter dataset (sample.csv).

# Add your library below.  
  
library(dplyr)  
library(tm)  
library(wordcloud)

# Step 1. Load and Clean the Data

Load in the sample.csv dataset from this project’s data folder. Please make sure you transform the document to lowercase, delete stopwords & numbers & punctuation (1 point).

# Write your code below.  
  
## Read twitter data file then examine.  
csv <- read.csv("data/sample.csv", stringsAsFactors = F)  
#str(text)  
  
## Create df for the tweets.  
text <- csv$text  
  
## Put tweets into a vector then examine.  
words\_vec <- VectorSource(text)  
#head(words\_vec)  
  
## Put tweets in vector into a corpus then examine.  
words\_corpus <- Corpus(words\_vec)  
#words\_corpus  
#words\_corpus[[1]]  
#words\_corpus[[1]][1]  
  
## Clean the corpus then examine.  
words\_corpus <- tm\_map(words\_corpus, content\_transformer(tolower))

## Warning in tm\_map.SimpleCorpus(words\_corpus, content\_transformer(tolower)):  
## transformation drops documents

words\_corpus <- tm\_map(words\_corpus, removePunctuation)

## Warning in tm\_map.SimpleCorpus(words\_corpus, removePunctuation): transformation  
## drops documents

words\_corpus <- tm\_map(words\_corpus, removeNumbers)

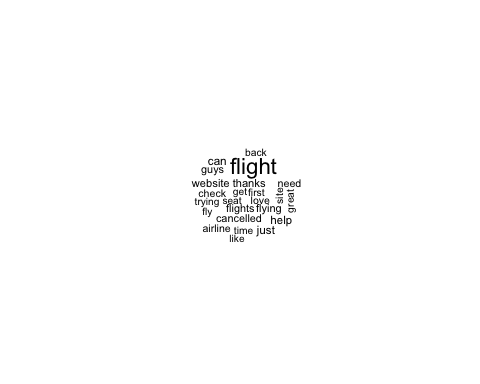
## Warning in tm\_map.SimpleCorpus(words\_corpus, removeNumbers): transformation  
## drops documents

words\_corpus <- tm\_map(words\_corpus, removeWords, stopwords("english"))

## Warning in tm\_map.SimpleCorpus(words\_corpus, removeWords, stopwords("english")):  
## transformation drops documents

#words\_corpus  
#words\_corpus[[1]]  
#words\_corpus[[1]][1]  
  
## Create TDM, matrix, word counts; then order the words in decreasing order.  
words\_tdm <- TermDocumentMatrix(words\_corpus)  
words\_m <- as.matrix(words\_tdm)  
wordCounts <- rowSums(words\_m)  
wordCounts <- sort(wordCounts, decreasing = T)  
#head(wordCounts)  
  
  
### Plot wordcloud  
wordcloud(names(wordCounts), wordCounts, max.words = 25)

## Warning in wordcloud(names(wordCounts), wordCounts, max.words = 25):  
## virginamerica could not be fit on page. It will not be plotted.



# Step 2. Adjust the Stopwords

Add “can” and “just” to the stopwords list (1 point).

# Write your code below.  
  
## Create stop vector for english stopwords and new stopwords.  
stops <- c(stopwords("en"),"can","just", "virginamerica") # Added flights twitter handle (@).  
#str(stops)  
  
## Update wordcloud data.  
words\_corpus <- tm\_map(words\_corpus, removeWords, stops)

## Warning in tm\_map.SimpleCorpus(words\_corpus, removeWords, stops): transformation  
## drops documents

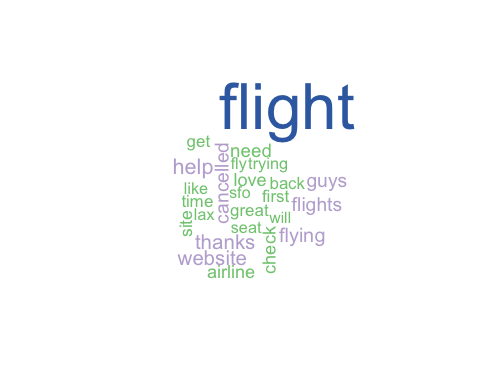
words\_tdm <- TermDocumentMatrix(words\_corpus)  
words\_m <- as.matrix(words\_tdm)  
wordCounts <- rowSums(words\_m)  
wordCounts <- sort(wordCounts, decreasing = T)  
  
## Plot wordcloud again.  
wordcloud(names(wordCounts), wordCounts, max.words = 25)



# Step 3. Adjust the Theme

Use five colors and “Accent” for color theme (1 point).

# Write your code below.  
  
## Add color to word cloud.  
wordcloud(names(wordCounts), wordCounts, max.words = 25, colors=brewer.pal(5, "Accent"))



# Step 4. Analysis

Does the word cloud convey the key points of the document? (1 point).

When completing the lab by the instructions orginially, a small wordcloud that only had 1 or 2 colors was the result. This wordcloud did not give any insight because all the words had virtually the same size and color, no way to distinguish between each.

After addding “virginamerica” to my stopword vector (~ line 80), I could eaily see the rest of the words that were over shadowed within the tweets.

Once we achieved the final word cloud, it’s no suprise that ‘flight’ was the most frequent word used, and some runner-ups being ‘thanks’, ‘website’, and ‘help’. This word cloud made it very easy for us to understand the word frequencies and how they differ.

# Step 5. Compile

Submit the compiled file. The wordcloud may prevent you from compiling. If so, do the following:

jpeg('p1.jpg')   
wordcloud(cloudFrame.t$word, cloudFrame.t$freq)   
dev.off()

This will make the graphs save in the current working directory.

# Write your code below, if necessary.