Intro to Data Science - HW 9


```
# Enter your name here:  Ryan Tervo
# Course Number:         IST 687
# Assignment Name:       Homework #9
# Due Date:              12 Dec 2022
# Received Extension:
# Submitted Date:        20 Dec 2022
```

Attribution statement: (choose only one and delete the rest)

```
# 1. I did this homework by myself, with help from the book and the professor.
```

Text mining plays an important role in many industries because of the prevalence of text in the interactions between customers and company representatives. Even when the customer interaction is by speech, rather than by chat or email, speech to text algorithms have gotten so good that transcriptions of these spoken word interactions are often available. To an increasing extent, a data scientist needs to be able to wield tools that turn a body of text into actionable insights. In this homework, we explore a real **City of Syracuse dataset** using the **quanteda** and **quanteda.textplots** packages. Make sure to install the **quanteda** and **quanteda.textplots** packages before following the steps below:

Part 1: Load and visualize the data file

A. Take a look at this article: https://samedelstein.medium.com/snowplow-naming-contest-data-2dcd38272caf and write a comment in your R script, briefly describing what it is about.

```
# The city of Syracuse purchased 10 new snowplows and they allowed people to vote on their nam
es.  A person was interested in the the raw data so they got used a freedom of information act
 to collect the data.  Once they received the data they analyzed it to see what kind of submis
sions were made.
```

B. Read the data from the following URL into a dataframe called **df**: https://intro-datascience.s3.us-east-2.amazonaws.com/snowplownames.csv

```
#   LOAD LIBRARY:
#install.packages('quanteda', dependencies = TRUE)
library(quanteda)
```

```
## Package version: 3.2.4
## Unicode version: 13.0
## ICU version: 69.1
```

```
## Parallel computing: 32 of 32 threads used.
```

```
## See https://quanteda.io for tutorials and examples.
```

```
library(tidyverse)
```

```
## ── Attaching packages
## ─────────────────────────────────────── tidyverse 1.3.2 ──
```

```
##    ggplot2 3.4.0         purrr   0.3.5
##    tibble  3.1.8         dplyr   1.0.10
##    tidyr   1.2.1         stringr 1.5.0
##    readr   2.1.3         forcats 0.5.2
## ── Conflicts ──────────────────────────────────────── tidyverse_conflicts() ──
##    dplyr::filter() masks stats::filter()
##    dplyr::lag()    masks stats::lag()
```

```
library(quanteda.textplots)
library(quanteda.textstats)

#install.packages('librarien')
#library(librarien)
#   DEFINE VARIABLES:
fileName <- "https://intro-datascience.s3.us-east-2.amazonaws.com/snowplownames.csv"

#   READ IN DATA:
df   <- data.frame(read_csv(fileName, show_col_types = FALSE))
```

C. Inspect the **df** dataframe – which column contains an explanation of the meaning of each submitted snowplow name? Transform that column into a **document-feature matrix**, using the **corpus()**, **tokens()**, **tokens_select(), and** dfm()** functions. Do not forget to **remove stop words**.

Hint: Make sure you have libraried *quanteda*

```
 #   INSPECT DATAFRAME:
head(df)
```

```
##   submission_number submitter_name_anonymized       snowplow_name
## 1                 1                   kjlt9cua             rudolph
## 2                 2                   KXKaabXN           salt life
## 3                 3                   kjlt9cua            blizzard
## 4                 4                   Rv9sODqp              butter
## 5                 5                   zzcc5FDn santa's 10 reindeer
## 6                 6                   wOrKO7XI    plowy mcplowface
##
                  meaning
## 1                                                                The red nose
 cuts through any storm.
## 2 We may not be near the ocean like everyone else with the stickers that say Salt Life, but
 we have plenty of salt!
## 3                                                                     This pl
ow can handle any storm.
## 4                                                 It's amazing how the snow plows t
```

```
hrough snow like butter!
## 5                                                          They can deliver through t
he bad weather and snow.
## 6
It would be a great name
##   winning_name
## 1        FALSE
## 2        FALSE
## 3        FALSE
## 4        FALSE
## 5        FALSE
## 6        FALSE
```

```
#   DETERMINE EXPLANATION MEANING:
print('The column that indicates the meaning of the snow plow name is labeled is meaning', quo
te = FALSE)
```

```
## [1] The column that indicates the meaning of the snow plow name is labeled is meaning
```

```
#   CREATE DOCUMENT - FEATURE MATRIX:
### 'remove stop words':
intcorpus   <- corpus(df$meaning)
```

```
## Warning: NA is replaced by empty string
```

```
paras       <- corpus_reshape(intcorpus, to='paragraphs')
paras       <- tokens(paras)
webfile_dtm <- dfm(paras, stem = TRUE, remove_punct = TRUE, remove = stopwords("english"))
```

```
## Warning: '...' should not be used for tokens() arguments; use 'tokens()' first.
```

```
## Warning: 'remove' is deprecated; use dfm_remove() instead
```

```
## Warning: 'stem' is deprecated; use dfm_wordstem() instead
```

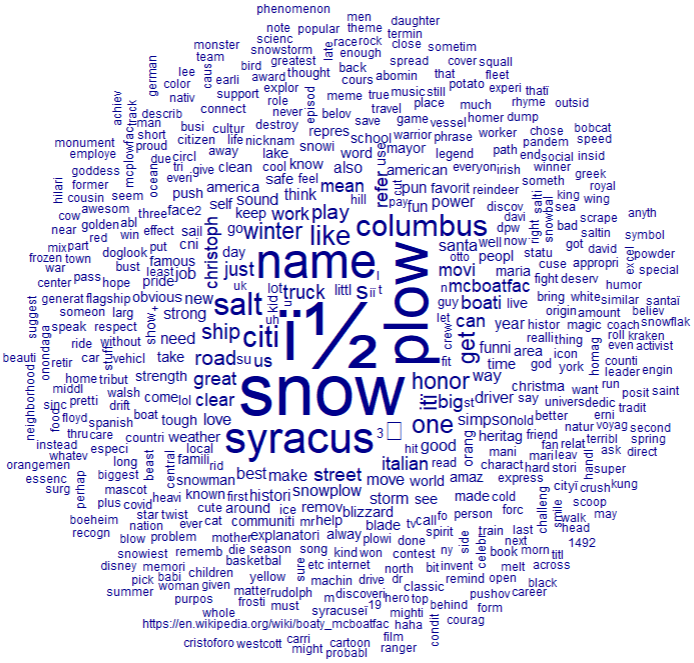
```
#mainTextcorpus()
#tokens()
#tokens_select()
#dfm()
```

D. Plot a **word cloud**, where a word is only represented if it appears **at least 2 times** . **Hint:** use **textplot_wordcloud()**:

Hint: Make sure you have libraried (and installed if needed) *quanteda.textplots*

```
#   TRIM DFM:
webfile_dtm <- dfm_trim(webfile_dtm, min_termfreq = 2)
```

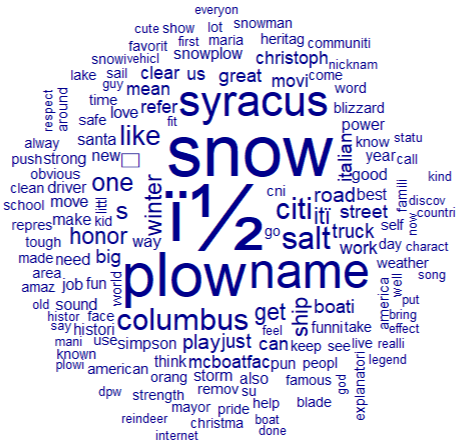
```
#   CREATE WORD CLOUD:
wordC <-textplot_wordcloud(webfile_dtm)
```



E. Next, **increase the minimum count to 10**. What happens to the word cloud? **Explain in a comment**.

```
#   TRIM DFM:
webfile_dtm <- dfm_trim(webfile_dtm, min_termfreq = 10)

#   CREATE WORD CLOUD:
wordC <-textplot_wordcloud(webfile_dtm)
```




```
#   WHAT HAPPENED WHEN THE MIN TERM FREQUENCY WAS INCREASED FROM 2 TO 10?
#-------------------------------------------------------------------------
# The size of the word cloud decreased.  However, Since the most frequent words were in the mi
ddle those words were in both of the word clouds.
```

F. What are the top words in the word cloud? Explain in a brief comment.

```
#   THE TOP WORDS IN THE WORD CLOUD:
#---------------------------------------------------------
#Plowd, snow, syracuse, name, columbus, salt, ship, 1/2.
```

Part 2: Analyze the sentiment of the descriptions

A. Create a **named list of word counts by frequency**.

output the 10 most frequent words (their word count and the word).

Hint: use **textstat_frequency()** from the *quanteda.textstats* package.

```
top10 <- textstat_frequency(webfile_dtm)
head(top10, 10)
```

```
##     feature frequency rank docfreq group
## 1         ½       432    1     143   all
```

```
## 2         ï       336    2     147   all
## 3      snow       326    3     297   all
## 4      plow       269    4     252   all
## 5      name       204    5     191   all
## 6   syracus       178    6     168   all
## 7      salt       110    7      86   all
## 8  columbus       106    8     102   all
## 9      citi       100    9      97   all
## 10     like        92   10      88   all
```

B. Explain in a comment what you observed in the sorted list of word counts.

```
#   The words in this list appear to be the same words seen in the word cloud.
```

Part 3: Match the words with positive and negative words

A. Read in the list of positive words, using the scan() function, and output the first 5 words in the list. Do the same for the the negative words list:

   https://intro-datascience.s3.us-east-2.amazonaws.com/positive-words.txt
   https://intro-datascience.s3.us-east-2.amazonaws.com/negative-words.txt

There should be 2006 positive words and 4783 negative words, so you may need to clean up these lists a bit.

```
#   DEFINE FILE NAMES:
filename_pos <- 'https://intro-datascience.s3.us-east-2.amazonaws.com/positive-words.txt'
filename_neg <- 'https://intro-datascience.s3.us-east-2.amazonaws.com/negative-words.txt'

#   SCAN DOCUMENTS:
wordlist_pos <- scan(filename_pos, what = 'character')
wordlist_pos <- wordlist_pos[-1:-198]
wordlist_neg <- scan(filename_neg, what = 'character')
wordlist_neg <- wordlist_neg[-1:-198]


#   DISPLAY TOP 5 POS AND NEG WORDS:
print(wordlist_pos[1:5])
```

```
## [1] "abound"     "abounds"    "abundance"  "abundant"   "accessable"
```

```
print(wordlist_neg[1:5])
```

```
## [1] "2-faces"    "abnormal"   "abolish"    "abominable" "abominably"
```

B. Use **dfm_match()** to match the words in the dfm with the words in posWords). Note that **dfm_match()** creates a new dfm.

Then pass this new dfm to the **textstat_frequency()** function to see the positive words in our corpus, and how many times each word was mentioned.

```
#   CREATE NEW DFM
webfile_dtm <- dfm_trim(webfile_dtm)#, min_termfreq = 0)
pos_match_dfm <- dfm_match(webfile_dtm, wordlist_pos)
pos_match_dfm
```

```
## Document-feature matrix of: 1,907 documents, 2,005 features (99.99% sparse) and 0 docvars.
##        features
## docs    abound abounds abundance abundant accessable accessible acclaim
##   text1      0       0         0        0          0          0       0
##   text2      0       0         0        0          0          0       0
##   text3      0       0         0        0          0          0       0
##   text4      0       0         0        0          0          0       0
##   text5      0       0         0        0          0          0       0
##   text6      0       0         0        0          0          0       0
##        features
## docs    acclaimed acclamation accolade
##   text1         0           0        0
##   text2         0           0        0
##   text3         0           0        0
##   text4         0           0        0
##   text5         0           0        0
##   text6         0           0        0
## [ reached max_ndoc ... 1,901 more documents, reached max_nfeat ... 1,995 more features ]
```

C. Sum all the positive words

```
sum_pos_word <- sum(pos_match_dfm)
print(sum_pos_word)
```

```
## [1] 517
```

D. Do a similar analysis for the negative words - show the 10 most frequent negative words and then sum the negative words in the document.

```
webfile_dtm <- dfm_trim(webfile_dtm, min_termfreq = 0)
neg_match_dfm <- dfm_match(webfile_dtm, wordlist_neg)
neg_match_dfm
```

```
## Document-feature matrix of: 1,907 documents, 4,782 features (100.00% sparse) and 0 docvars.
##        features
## docs    2-faces abnormal abolish abominable abominably abominate abomination
##   text1       0        0       0          0          0         0           0
##   text2       0        0       0          0          0         0           0
##   text3       0        0       0          0          0         0           0
##   text4       0        0       0          0          0         0           0
##   text5       0        0       0          0          0         0           0
##   text6       0        0       0          0          0         0           0
##        features
## docs    abort aborted aborts
##   text1     0       0      0
```

```
##   text2     0       0      0
##   text3     0       0      0
##   text4     0       0      0
##   text5     0       0      0
##   text6     0       0      0
## [ reached max_ndoc ... 1,901 more documents, reached max_nfeat ... 4,772 more features ]
```

```
sum_neg_word <- sum(neg_match_dfm)
print(sum_neg_word)
```

```
## [1] 0
```

E. Write a comment describing what you found after matching positive and negative words. Which group is more common in this dataset? Might some of the negative words not actually be used in a negative way? What about the positive words?

```
#  So I might be doing something wrong... I have been unsuccessful in finding any negative wor
ds in the list.  After many attempts to try different things I eventually resorted to quickly
scanning the list manually.

#  After doing a quick review in which I scanned a substantial portion of the meaning list I'v
e concluded that positive words outweigh negative words substantially.  There could be negativ
e words in there but they are few and far between as compared to the positive words.  The posi
tive words are everywhere and readily found.
```