Intro to Data Science - HW 9

```
# Enter your name here: Victoria Haley
```

Copyright Jeffrey Stanton, Jeffrey Saltz, Christopher Dunham, and Jasmina Tacheva

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 article is about the contest held to name the new snowplows that were purchased by the city of Syr
library(quanteda)
```

```
## Package version: 3.2.3
## Unicode version: 14.0
## ICU version: 70.1
## Parallel computing: 4 of 4 threads used.
## See https://quanteda.io for tutorials and examples.
```

```
library(quanteda.textplots)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.5
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

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

```
df <- read.csv("https://intro-datascience.s3.us-east-2.amazonaws.com/snowplownames.csv")
```

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*

```r
glimpse(df)
```

```
## Rows: 1,907
## Columns: 5
## $ submission_number         <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 1~
## $ submitter_name_anonymized <chr> "kjlt9cua", "KXKaabXN", "kjlt9cua", "Rv9sODq~
## $ snowplow_name             <chr> "rudolph", "salt life", "blizzard", "butter"~
## $ meaning                   <chr> "The red nose cuts through any storm.", "We ~
## $ winning_name              <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FA~
```

```r
#$meaning contains an explanation of the meaning of each submitted snowplow name
meaningCorpus <- corpus(df$meaning) #Converts df$meaning column to a corpus
```

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

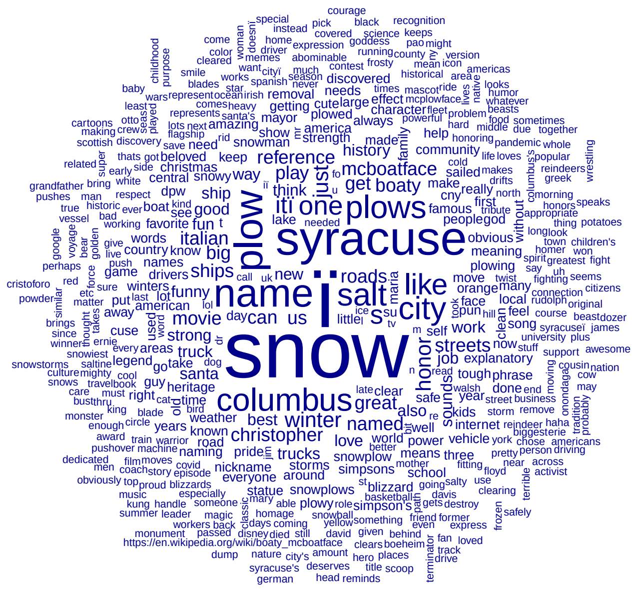
```r
meaningCorpus <- tokens(meaningCorpus, remove_punct = TRUE, remove_numbers = TRUE, remove_symbols = TRU
meaningDFM <- dfm(meaningCorpus, remove=stopwords("english")) #creates DFM
```

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

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

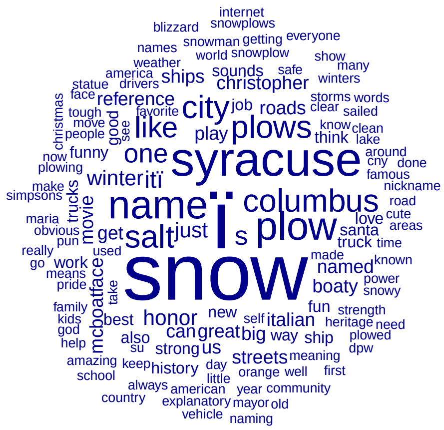
Hint: Make sure you have libraried (and installed if needed) *quanteda.textplots*

```r
textplot_wordcloud(meaningDFM, min_count=2)
```



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

```r
textplot_wordcloud(meaningDFM, min_count = 10)
```




```
#The word cloud shrinks and presents less words than with a minimum count of 2.
```

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

```
#Based on the size of the word, the top words are snow, Syracuse, salt, plow, name, columbus, and city.
```

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.

```
library(quanteda.textstats)
textstat_frequency(meaningDFM)[1:10,]
```

```
##     feature frequency rank docfreq group
## 1         ï       336    1     147   all
## 2      snow       321    2     292   all
## 3  syracuse       174    3     164   all
## 4      name       143    4     137   all
## 5      plow       140    5     130   all
## 6      salt       104    6      83   all
## 7     plows       100    7      98   all
## 8  columbus       100    7      96   all
## 9      city        96    9      94   all
## 10     like        88   10      85   all
```

```
#Here are the 10 most frequent words and their word count
```

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

```
#The most frequent words in this list are the same words that are noticeably bigger in the word cloud
#I also noticed that ï was included as a frequent "word", which is likely because it is not included as
```

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.

```r
#first, I'll read in the lists
posFile <- "https://intro-datascience.s3.us-east-2.amazonaws.com/positive-words.txt"
posWords <- scan(posFile, character(0), sep="\n")
negFile <- "https://intro-datascience.s3.us-east-2.amazonaws.com/negative-words.txt"
negWords <- scan(negFile, character(0), sep="\n")
#then, I'll clean up the lists
posWords <- posWords[-1:-34]
negWords <- negWords[-1:-34]
#and output the first 5 words in each list
head(posWords, 5) #positive words
```

```
## [1] "a+"        "abound"    "abounds"   "abundance" "abundant"
```

```r
head(negWords, 5) #negative words
```

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

```r
#the scan() functional originally read 2040 items in the posFile and 4817 items in the negFile, but aft
```

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.

```r
posDFM <- dfm_match(meaningDFM, posWords)
posDFM #shows that the 1907 entries from the meaningDFM have been expanded to cover all 2006 in the pos
```

```
## Document-feature matrix of: 1,907 documents, 2,006 features (99.98% sparse) and 0 docvars.
##        features
## docs    a+ abound abounds abundance abundant accessable accessible acclaim
##   text1  0      0       0         0        0          0          0       0
##   text2  0      0       0         0        0          0          0       0
##   text3  0      0       0         0        0          0          0       0
##   text4  0      0       0         0        0          0          0       0
##   text5  0      0       0         0        0          0          0       0
##   text6  0      0       0         0        0          0          0       0
##        features
## docs    acclaimed acclamation
##   text1         0           0
##   text2         0           0
##   text3         0           0
##   text4         0           0
##   text5         0           0
##   text6         0           0
## [ reached max_ndoc ... 1,901 more documents, reached max_nfeat ... 1,996 more features ]
```

```r
posFreq <- textstat_frequency(posDFM)
posFreq <- posFreq[posFreq$frequency>0,]
posFreq
```

```
##             feature frequency rank docfreq group
## 1              like        88    1      85   all
## 2             honor        47    2      47   all
## 3             great        43    3      43   all
## 4              good        28    4      28   all
## 5               fun        27    5      24   all
```

```
## 6           strong    25   6    25   all
## 7             best    23   7    22   all
## 8             love    21   8    21   all
## 9             work    21   8    21   all
## 10           clear    19  10    19   all
## 11          famous    16  11    16   all
## 12           pride    16  11    16   all
## 13            safe    16  11    16   all
## 14           tough    15  14    15   all
## 15            well    15  14    15   all
## 16           clean    13  16    13   all
## 17        favorite    13  16    13   all
## 18         amazing    12  18    12   all
## 19            cute    10  19    10   all
## 20         beloved     9  20     9   all
## 21           right     9  20     9   all
## 22          better     8  22     8   all
## 23        honoring     8  22     8   all
## 24        powerful     8  22     8   all
## 25            cool     7  25     7   all
## 26          homage     7  25     7   all
## 27         respect     7  25     7   all
## 28     appropriate     6  28     6   all
## 29         classic     6  28     6   all
## 30          golden     6  28     6   all
## 31          pretty     6  28     6   all
## 32          clears     5  32     5   all
## 33          enough     5  32     5   all
## 34        greatest     5  32     5   all
## 35           loves     5  32     5   all
## 36           magic     5  32     5   all
## 37          mighty     5  32     5   all
## 38           proud     5  32     5   all
## 39         support     5  32     5   all
## 40           works     5  32     5   all
## 41           award     4  41     4   all
## 42         cleared     4  41     4   all
## 43       dedicated     4  41     4   all
## 44            hero     4  41     4   all
## 45           humor     4  41     4   all
## 46           loved     4  41     4   all
## 47         popular     4  41     4   all
## 48           smile     4  41     4   all
## 49           super     4  41     4   all
## 50             top     4  41     4   all
## 51          winner     4  41     4   all
## 52             won     4  41     4   all
## 53         awesome     3  53     3   all
## 54          catchy     3  53     3   all
## 55       celebrate     3  53     3   all
## 56         courage     3  53     3   all
## 57       excellent     3  53     3   all
## 58           happy     3  53     3   all
## 59       hilarious     3  53     3   all
```

```
## 60        important        3   53        3   all
## 61             lead        3   53        3   all
## 62            liked        3   53        3   all
## 63         positive        3   53        3   all
## 64           safely        3   53        3   all
## 65            saint        3   53        3   all
## 66        uplifting        3   53        2   all
## 67              win        3   53        3   all
## 68           worked        3   53        3   all
## 69       autonomous        2   69        2   all
## 70      awesomeness        2   69        2   all
## 71        beautiful        2   69        2   all
## 72          benefit        2   69        2   all
## 73             boom        2   69        1   all
## 74           bright        2   69        2   all
## 75             easy        2   69        2   all
## 76             free        2   69        2   all
## 77          freedom        2   69        2   all
## 78             gold        2   69        2   all
## 79             holy        2   69        2   all
## 80          honored        2   69        2   all
## 81           loving        2   69        2   all
## 82             neat        2   69        2   all
## 83             nice        2   69        2   all
## 84            noble        2   69        2   all
## 85     perseverance        2   69        2   all
## 86       prosperity        2   69        2   all
## 87       protection        2   69        2   all
## 88            ready        2   69        2   all
## 89         recovery        2   69        1   all
## 90             rich        2   69        2   all
## 91           trophy        2   69        2   all
## 92            trust        2   69        2   all
## 93             warm        2   69        2   all
## 94          winning        2   69        2   all
## 95             wins        2   69        2   all
## 96          abounds        1   96        1   all
## 97        accolades        1   96        1   all
## 98       accomplish        1   96        1   all
## 99  accomplishments        1   96        1   all
## 100        accurate        1   96        1   all
## 101     achievement        1   96        1   all
## 102    achievements        1   96        1   all
## 103           angel        1   96        1   all
## 104          appeal        1   96        1   all
## 105          awards        1   96        1   all
## 106       awesomely        1   96        1   all
## 107        backbone        1   96        1   all
## 108          beauty        1   96        1   all
## 109         blossom        1   96        1   all
## 110           brave        1   96        1   all
## 111        brighten        1   96        1   all
## 112      capability        1   96        1   all
## 113         capable        1   96        1   all
```

```
## 114         cheer         1   96        1   all
## 115       clearer         1   96        1   all
## 116        clever         1   96        1   all
## 117    consistent         1   96        1   all
## 118    continuity         1   96        1   all
## 119       coolest         1   96        1   all
## 120     correctly         1   96        1   all
## 121    courageous         1   96        1   all
## 122         crisp         1   96        1   all
## 123       darling         1   96        1   all
## 124          dawn         1   96        1   all
## 125        decent         1   96        1   all
## 126       dignity         1   96        1   all
## 127        easier         1   96        1   all
## 128         elite         1   96        1   all
## 129     encourage         1   96        1   all
## 130         enjoy         1   96        1   all
## 131          envy         1   96        1   all
## 132   everlasting         1   96        1   all
## 133    excellence         1   96        1   all
## 134       excited         1   96        1   all
## 135          fair         1   96        1   all
## 136         faith         1   96        1   all
## 137          fame         1   96        1   all
## 138     fantastic         1   96        1   all
## 139       fastest         1   96        1   all
## 140           fav         1   96        1   all
## 141      fidelity         1   96        1   all
## 142        finest         1   96        1   all
## 143      freedoms         1   96        1   all
## 144         fresh         1   96        1   all
## 145      friendly         1   96        1   all
## 146        genius         1   96        1   all
## 147        gifted         1   96        1   all
## 148         glory         1   96        1   all
## 149          glow         1   96        1   all
## 150         grace         1   96        1   all
## 151      grateful         1   96        1   all
## 152          hail         1   96        1   all
## 153         hardy         1   96        1   all
## 154        helped         1   96        1   all
## 155       helping         1   96        1   all
## 156       heroine         1   96        1   all
## 157        honest         1   96        1   all
## 158      humorous         1   96        1   all
## 159    incredible         1   96        1   all
## 160    innovation         1   96        1   all
## 161     inspiring         1   96        1   all
## 162     instantly         1   96        1   all
## 163  instrumental         1   96        1   all
## 164         jolly         1   96        1   all
## 165     legendary         1   96        1   all
## 166         likes         1   96        1   all
## 167       logical         1   96        1   all
```

```
## 168          lovable         1   96        1   all
## 169            loyal         1   96        1   all
## 170            lucky         1   96        1   all
## 171          magical         1   96        1   all
## 172            merit         1   96        1   all
## 173          miracle         1   96        1   all
## 174           modern         1   96        1   all
## 175        motivated         1   96        1   all
## 176          patriot         1   96        1   all
## 177        persevere         1   96        1   all
## 178         pleasant         1   96        1   all
## 179           prefer         1   96        1   all
## 180        proactive         1   96        1   all
## 181          protect         1   96        1   all
## 182          recover         1   96        1   all
## 183       respectful         1   96        1   all
## 184     respectfully         1   96        1   all
## 185          satisfy         1   96        1   all
## 186          savings         1   96        1   all
## 187           savior         1   96        1   all
## 188        sensation         1   96        1   all
## 189            shiny         1   96        1   all
## 190      significant         1   96        1   all
## 191            smart         1   96        1   all
## 192           smiles         1   96        1   all
## 193           smooth         1   96        1   all
## 194         spirited         1   96        1   all
## 195           steady         1   96        1   all
## 196        strongest         1   96        1   all
## 197           sturdy         1   96        1   all
## 198          success         1   96        1   all
## 199        supported         1   96        1   all
## 200            sweet         1   96        1   all
## 201         talented         1   96        1   all
## 202         tenacity         1   96        1   all
## 203         thrilled         1   96        1   all
## 204         trusting         1   96        1   all
## 205    unforgettable         1   96        1   all
## 206        unlimited         1   96        1   all
## 207     unparalleled         1   96        1   all
## 208          winners         1   96        1   all
## 209        wonderful         1   96        1   all
## 210            worth         1   96        1   all
## 211              wow         1   96        1   all
```

C. Sum all the positive words

```
sum(posFreq$frequency)
```

```
## [1] 866
```

```
#there are 866 positive words in the document
```

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.

```r
negDFM <- dfm_match(meaningDFM, negWords)
negDFM #shows that the 1907 entries from the meaningDFM have been expanded to cover all 4783 in the neg
```

```
## Document-feature matrix of: 1,907 documents, 4,783 features (>99.99% sparse) and 0 docvars.
##        features
## docs    2-faced 2-faces abnormal abolish abominable abominably abominate
##   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    abomination abort aborted
##   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 ... 4,773 more features ]
```

```r
negFreq <- textstat_frequency(negDFM)
negFreq <- negFreq[negFreq$frequency>0,]
head(negFreq, 10) #shows the 10 most frequent negative words
```

```
##       feature frequency rank docfreq group
## 1       funny        25    1      25   all
## 2        cold         8    2       8   all
## 3       twist         8    2       8   all
## 4        hard         7    4       7   all
## 5  abominable         6    5       6   all
## 6     problem         6    5       6   all
## 7         bad         5    7       5   all
## 8     destroy         5    7       5   all
## 9        died         5    7       5   all
## 10       bust         4   10       4   all
```

```r
sum(negFreq$frequency)
```

```
## [1] 255
```

```r
#There are 255 negative words in the document
```

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?

```r
#After matching the positive and negative words, I found that the submitted names for the snowplows wer
#Some of the negative words like "funny" and "abominable" might not actually be used in a negative way
#It looks as though the words in the positive words list were used in a positive way in this case, but
```