ISM 4403 Homework Week 6

### **Tasks:**

Watch the video Text mining the Tidy Way <https://www.rstudio.com/resources/videos/text-mining-the-tidy-way/>.

Using the information provided in the video perform a similar analysis upon the works by Edgar Rice Burrows that are available on Project Gutenberg. (<https://www.gutenberg.org/ebooks/author/48>)

Please provide the following data making sure to remove stop words and clean up punctuation etc…

What is the most common word used in these works?  
 Attempt to use inverse document frequency to determine the people and places in these books. If this results in alternative data please describe. I tried to couldn’t get the inverse  
 Attempt to determine the most common names using the Tidy framework. What are they?

PASTE YOUR CODE HERE

# lab 6 text mining

#install.packages("tidyverse")

#install.packages("gutenbergr")

#install.packages("tidytext")

#install.packages("reshape2")

#install.packages("wordcloud")

library(gutenbergr)

library(tidyverse)

library(tidytext)

library(dplyr)

library(wordcloud)

#looking in the books

edgar\_df <- gutenberg\_download(c(62, 78, 64, 1154, 551, 106, 106, 90, 605, 92, 123, 85, 1401, 58874, 3475, 59752, 3475, 59752, 363, 331, 552, 149, 369, 96, 554, 2020, 29405, 58904, 20802), meta\_fields = "title")

# normalize text and turn untidy text into tidy text

# turning text df into tidy data

edgar\_tidy <- edgar\_df %>%

unnest\_tokens(word, text)

edgar\_tidy

# remove stop words

edgar\_clean <- edgar\_tidy %>%

anti\_join(get\_stopwords())

# word count

edgar\_count <- edgar\_clean %>%

count(word, sort = TRUE)

# sentiments

bing <- get\_sentiments("bing")

# positive sentiments

edgar\_pos<- edgar\_count <- get\_sentiments("bing") %>%

filter(sentiment == "positive")

# getting the positive and negitive sentiments

edgar\_bing <- edgar\_tidy %>%

inner\_join(bing) %>%

count(word, sentiment, sort = TRUE)

# graph of word count and sentiment from all of edgar's books

edgar\_bing %>%

filter(n > 150) %>%

mutate(n = ifelse(sentiment == "negative", -n, n)) %>%

mutate(word = reorder(word, n)) %>%

ggplot(aes(word, n, fill = sentiment)) +

geom\_col() +

coord\_flip() +

labs(y = "Contribution to sentiment")

# getting the idf and tf

edgar\_idf <- edgar\_tidy %>%

bind\_tf\_idf(word, story, n)

edgar\_idf %>%

select(-total) %>%

arrange(desc(tf\_idf))

END PASTE

PASTE YOUR RESULTS HERE

R version 3.6.1 (2019-07-05) -- "Action of the Toes"

Copyright (C) 2019 The R Foundation for Statistical Computing

Platform: x86\_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.

You are welcome to redistribute it under certain conditions.

Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.

Type 'contributors()' for more information and

'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or

'help.start()' for an HTML browser interface to help.

Type 'q()' to quit R.

> # lab 6 text mining

>

> #install.packages("tidyverse")

> #install.packages("gutenbergr")

> #install.packages("tidytext")

> #install.packages("reshape2")

> #install.packages("wordcloud")

> library(gutenbergr)

Warning message:

package ‘gutenbergr’ was built under R version 3.6.2

> library(tidyverse)

-- **Attaching packages** --------------------------------------- tidyverse 1.3.0 --

v ggplot2 3.2.1 v purrr 0.3.3

v tibble 2.1.3 v dplyr 0.8.4

v tidyr 1.0.2 v stringr 1.4.0

v readr 1.3.1 v forcats 0.4.0

-- **Conflicts** ------------------------------------------ tidyverse\_conflicts() --

x dplyr::filter() masks stats::filter()

x dplyr::lag() masks stats::lag()

Warning messages:

1: package ‘tidyverse’ was built under R version 3.6.2

2: package ‘tidyr’ was built under R version 3.6.2

3: package ‘purrr’ was built under R version 3.6.2

4: package ‘dplyr’ was built under R version 3.6.2

5: package ‘forcats’ was built under R version 3.6.2

> library(tidytext)

Warning message:

package ‘tidytext’ was built under R version 3.6.2

> library(dplyr)

> library(wordcloud)

Loading required package: RColorBrewer

Warning message:

package ‘wordcloud’ was built under R version 3.6.2

>

> #looking in the books

> edgar\_df <- gutenberg\_download(c(62, 78, 64, 1154, 551, 106, 106, 90, 605, 92, 123, 85, 1401, 58874, 3475, 59752, 3475, 59752, 363, 331, 552, 149, 369, 96, 554, 2020, 29405, 58904, 20802), meta\_fields = "title")

Determining mirror for Project Gutenberg from http://www.gutenberg.org/robot/harvest

Using mirror http://aleph.gutenberg.org

>

> # normalize text and turn untidy text into tidy text

>

> # turning text df into tidy data

> edgar\_tidy <- edgar\_df %>%

+ unnest\_tokens(word, text)

>

> edgar\_tidy

# A tibble: 1,534,007 x 3

gutenberg\_id title word

*<int>* *<chr>* *<chr>*

1 62 A Princess of Mars frontispiece

2 62 A Princess of Mars with

3 62 A Princess of Mars my

4 62 A Princess of Mars back

5 62 A Princess of Mars against

6 62 A Princess of Mars a

7 62 A Princess of Mars golden

8 62 A Princess of Mars throne

9 62 A Princess of Mars i

10 62 A Princess of Mars fought

# ... with 1,533,997 more rows

>

> # remove stop words

> edgar\_clean <- edgar\_tidy %>%

+ anti\_join(get\_stopwords())

Joining, by = "word"

>

> # word count

> edgar\_count <- edgar\_clean %>%

+ count(word, sort = TRUE)

> # sentiments

> bing <- get\_sentiments("bing")

>

>

> # positive sentiments

> edgar\_pos<- edgar\_count <- get\_sentiments("bing") %>%

+ filter(sentiment == "positive")

>

> # getting the positive and negitive sentiments

> edgar\_bing <- edgar\_tidy %>%

+ inner\_join(bing) %>%

+ count(word, sentiment, sort = TRUE)

Joining, by = "word"

>

> # graph of word count and sentiment from all of edgar's books

> edgar\_bing %>%

+ filter(n > 150) %>%

+ mutate(n = ifelse(sentiment == "negative", -n, n)) %>%

+ mutate(word = reorder(word, n)) %>%

+ ggplot(aes(word, n, fill = sentiment)) +

+ geom\_col() +

+ coord\_flip() +

+ labs(y = "Contribution to sentiment")

>

> # getting the idf and tf

> edgar\_idf <- edgar\_tidy %>%

+ bind\_tf\_idf(word, story, n)

Error in split.default(X, group) : first argument must be a vector

>

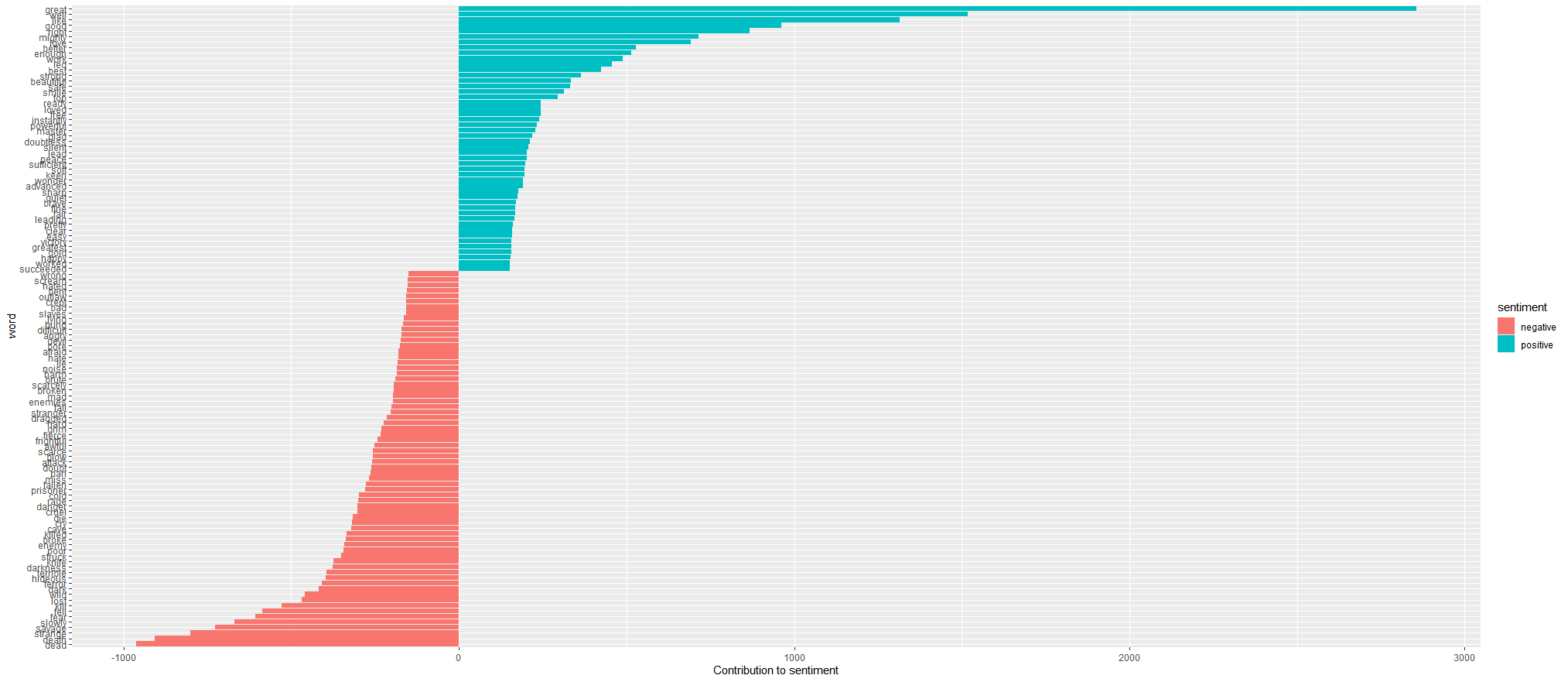
> edgar\_idf %>%

+ select(-total) %>%

+ arrange(desc(tf\_idf))

Error in eval(lhs, parent, parent) : object 'edgar\_idf' not found

END PASTE



20% Installed Tidy framework and imported the library, or loaded a similar library

20% Load the words of each book into a data structure, normalized text to lowercase, remove punctuation, and spaces.

20% Calculated the frequency of each word for each book.

20% Remove common words from the collections of each book to determine unique words

20% Calculate the frequency of each unique word and sort by frequency to determine important words and items.