#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* # Analyzing a corpus of Science books #\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* #set working directory #setwd(“replace directory here”) #install and load the dplyr, tidytext, ggplot2, and gutenbergr library for text processing

#install.packages("dplyr")  
#install.packages("tidytext")  
#install.packages("ggplot2")  
#install.packages("gutenbergr")

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* # Creating a corpus #\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

library(dplyr)  
library(tidytext)  
library(ggplot2)  
library(gutenbergr)  
physics <- gutenberg\_download(c(37729, 14725, 13476, 30155),   
 meta\_fields = "author")

How many times was each word used in each text?

physics\_words <- physics %>%  
 unnest\_tokens(word, text) %>%  
 count(author, word, sort = TRUE)  
  
physics\_words

library(forcats)  
  
plot\_physics <- physics\_words %>%  
 bind\_tf\_idf(word, author, n) %>%  
 mutate(word = fct\_reorder(word, tf\_idf)) %>%  
 mutate(author = factor(author, levels = c("Galilei, Galileo",  
 "Huygens, Christiaan",   
 "Tesla, Nikola",  
 "Einstein, Albert")))

plot\_physics %>%   
 group\_by(author) %>%   
 top\_n(15, tf\_idf) %>%   
 ungroup() %>%  
 mutate(word = reorder(word, tf\_idf)) %>%  
 ggplot(aes(word, tf\_idf, fill = author)) +  
 geom\_col(show.legend = FALSE) +  
 labs(x = NULL, y = "tf-idf") +  
 facet\_wrap(~author, ncol = 2, scales = "free") +  
 coord\_flip()

library(stringr)  
  
physics %>%   
 filter(str\_detect(text, "\_k\_")) %>%   
 select(text)  
  
physics %>%   
 filter(str\_detect(text, "RC")) %>%   
 select(text)  
  
mystopwords <- tibble(word = c("eq", "co", "rc", "ac", "ak", "bn",   
 "fig", "file", "cg", "cb", "cm",  
 "ab", "\_k", "\_k\_", "\_x"))  
  
physics\_words <- anti\_join(physics\_words, mystopwords,   
 by = "word")

library(wordcloud)  
  
physics\_words %>%  
 count(word) %>%  
 with(wordcloud(word, n, max.words = 10))

plot\_physics <- physics\_words %>%  
 bind\_tf\_idf(word, author, n) %>%  
 mutate(word = str\_remove\_all(word, "\_")) %>%  
 group\_by(author) %>%   
 top\_n(15, tf\_idf) %>%  
 ungroup() %>%  
 mutate(word = reorder\_within(word, tf\_idf, author)) %>%  
 mutate(author = factor(author, levels = c("Galilei, Galileo",  
 "Huygens, Christiaan",  
 "Tesla, Nikola",  
 "Einstein, Albert")))  
  
ggplot(plot\_physics, aes(word, tf\_idf, fill = author)) +  
 geom\_col(show.legend = FALSE) +  
 labs(x = NULL, y = "tf-idf") +  
 facet\_wrap(~author, ncol = 2, scales = "free") +  
 coord\_flip() +  
 scale\_x\_reordered()

physics\_bigrams <- physics %>%  
 unnest\_tokens(bigram, text, token = "ngrams", n = 2)  
  
physics\_bigrams

physics\_bigrams%>%  
 count(bigram, sort = TRUE)

library(tidyr)  
  
bigrams\_separated <- physics\_bigrams %>%  
 separate(bigram, c("word1", "word2"), sep = " ")  
  
bigrams\_filtered <- bigrams\_separated %>%  
 filter(!word1 %in% stop\_words$word) %>%  
 filter(!word2 %in% stop\_words$word)  
  
# new bigram counts:  
bigram\_counts <- bigrams\_filtered %>%   
 count(word1, word2, sort = TRUE)  
  
bigram\_counts

bigrams\_united <- bigrams\_filtered %>%  
 unite(bigram, word1, word2, sep = " ")  
  
bigrams\_united

physics %>%  
 unnest\_tokens(trigram, text, token = "ngrams", n = 3) %>%  
 separate(trigram, c("word1", "word2", "word3"), sep = " ") %>%  
 filter(!word1 %in% stop\_words$word,  
 !word2 %in% stop\_words$word,  
 !word3 %in% stop\_words$word) %>%  
 count(word1, word2, word3, sort = TRUE)

bigrams\_filtered %>%  
 filter(word2 == "angle") %>%  
 count(author, word1, sort = TRUE)

bigram\_tf\_idf <- bigrams\_united %>%  
 count(author, bigram) %>%  
 bind\_tf\_idf(bigram, author, n) %>%  
 arrange(desc(tf\_idf))  
  
bigram\_tf\_idf

library(igraph)  
  
bigram\_graph <- bigram\_counts %>%  
 filter(n > 20) %>%  
 graph\_from\_data\_frame()  
  
bigram\_graph

library(ggraph)  
set.seed(2000)  
  
ggraph(bigram\_graph, layout = "fr") +  
 geom\_edge\_link() +  
 geom\_node\_point() +  
 geom\_node\_text(aes(label = name), vjust = 1, hjust = 1)

set.seed(2000)  
  
a <- grid::arrow(type = "closed", length = unit(.15, "inches"))  
  
ggraph(bigram\_graph, layout = "fr") +  
 geom\_edge\_link(aes(edge\_alpha = n), show.legend = FALSE,  
 arrow = a, end\_cap = circle(.07, 'inches')) +  
 geom\_node\_point(color = "lightblue", size = 5) +  
 geom\_node\_text(aes(label = name), vjust = 1, hjust = 1) +  
 theme\_void()

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* # Citations #\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

This project is heavily adapted from the [website](https://www.tidytextmining.com/) for Text Mining with R! Visit the [GitHub repository](https://github.com/dgrtwo/tidy-text-mining) for this site, find the [book at O’Reilly](http://shop.oreilly.com/product/0636920067153.do?cmp=af-strata-books-video-product_cj_0636920067153_4428796), or buy it on [https://www.amazon.com/gp/product/1491981652/ref=as\_li\_tl?ie=UTF8&tag=juliasilge-20&camp=1789&creative=9325&linkCode=as2&creativeASIN=1491981652&linkId=0e92d44b0aa39ab34608ffa582dbd490](Amazon).

citation(“dplyr”) citation(“tidytext”) citation(“ggplot2”) citation(“gutenbergr”) citation(“forcats”) citation(“stringr”) citation(“wordcloud”) citation(“tidyr”) citation(“igraph”) citation(“ggraph”)

Silge J. and Robinson D. (2017) Text Mining with R: A Tidy Approach (1st. ed.). O’Reilly Media, Inc.