**Text Analysis of Science books**

Jajwalya R. Karajgikar

##Tidy Data Tidy text format is a table with one-token-per row. Some of the packages used are dplyr (Wickham and Francois 2016), tidyr (Wickham 2016), ggplot2 (Wickham 2009). tm (Feinerer, Hornik, and Meyer 2008) and quanteda (Benoit and Nulty 2016) can also be used. #\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* # 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 #\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

The gutenbergr package (Robinson 2016) provides access to the public domain works from the [Project Gutenberg](https://www.gutenberg.org/) collection including metadata. The function gutenberg\_download() downloads one or more works from Project Gutenberg by ID. The texts Discourse on Floating Bodies by Galileo Galilei Treatise on Light by Christiaan Huygens Experiments with Alternate Currents of High Potential and High Frequency by Nikola Tesla Relativity: The Special and General Theory by Albert Einstein used are physics classics written across a 300-year timespan, and some of them were first written in other languages and then translated to English.

**library**(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

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

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

## Using mirror [http://aleph.gutenberg.org](http://aleph.gutenberg.org/)

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ## Pre-processing the corpus #\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ###How many times was each word used in each text? Unnest\_tokens breaks the text into individual tokens and performs tokenization. dplyr’s count() finds the most common words in all the books as a whole.

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

## # A tibble: 12,671 x 3  
## author word n  
## <chr> <chr> <int>  
## 1 Galilei, Galileo the 3760  
## 2 Tesla, Nikola the 3604  
## 3 Huygens, Christiaan the 3553  
## 4 Einstein, Albert the 2993  
## 5 Galilei, Galileo of 2049  
## 6 Einstein, Albert of 2028  
## 7 Tesla, Nikola of 1737  
## 8 Huygens, Christiaan of 1708  
## 9 Huygens, Christiaan to 1207  
## 10 Tesla, Nikola a 1176  
## # … with 12,661 more rows

These are the raw counts. The documents are all different lengths. The term-frequency(tf) quantifies the frequently occurring terms. The inverse term-frequency(idf) decreases the weight for commonly used words and increases the weight for words that are not used very much in a collection of documents. The tf-idf together is a statistic to measure the frequency of a term adjusted for how rarely it is used. It measures how important a word is in the text.

**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")))

**Chart 1**

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**()



#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ## Removing specific stopwords from the corpus #\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* The filter() function narrows down the selected text. A custom list of stop words can be included with an anti\_join() to remove them.

**library**(stringr)  
**data**(stop\_words)  
  
physics **%>%**  
 **filter**(**str\_detect**(text, "\_k\_")) **%>%**  
 **select**(text)

## # A tibble: 7 x 1  
## text   
## <chr>   
## 1 surface AB at the points AK\_k\_B. Then instead of the hemispherical   
## 2 would needs be that from all the other points K\_k\_B there should   
## 3 necessarily be equal to CD, because C\_k\_ is equal to CK, and C\_g\_ to   
## 4 the crystal at K\_k\_, all the points of the wave CO\_oc\_ will have   
## 5 O\_o\_ has reached K\_k\_. Which is easy to comprehend, since, of these   
## 6 CO\_oc\_ in the crystal, when O\_o\_ has arrived at K\_k\_, because it forms  
## 7 <U+03C1> is the average density of the matter and \_k\_ is a constant connected

physics **%>%**  
 **filter**(**str\_detect**(text, "RC")) **%>%**  
 **select**(text)

## # A tibble: 44 x 1  
## text   
## <chr>   
## 1 line RC, parallel and equal to AB, to be a portion of a wave of light,  
## 2 represents the partial wave coming from the point A, after the wave RC  
## 3 be the propagation of the wave RC which fell on AB, and would be the   
## 4 transparent body; seeing that the wave RC, having come to the aperture  
## 5 incident rays. Let there be such a ray RC falling upon the surface   
## 6 CK. Make CO perpendicular to RC, and across the angle KCO adjust OK,   
## 7 the required refraction of the ray RC. The demonstration of this is,   
## 8 explaining ordinary refraction. For the refraction of the ray RC is   
## 9 29. Now as we have found CI the refraction of the ray RC, similarly   
## 10 the ray \_r\_C is inclined equally with RC, the line C\_d\_ will   
## # … with 34 more rows

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") **%>%**  
 **anti\_join**(stop\_words)

## Joining, by = "word"

**head**(physics\_words)

## # A tibble: 6 x 3  
## author word n  
## <chr> <chr> <int>  
## 1 Galilei, Galileo water 828  
## 2 Galilei, Galileo gravity 240  
## 3 Huygens, Christiaan refraction 218  
## 4 Galilei, Galileo air 211  
## 5 Galilei, Galileo mass 208  
## 6 Huygens, Christiaan light 201

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ## Word cloud text frequency from the corpus #\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* The more number of times a specific word appears in a source of textual data, the bigger and bolder it appears in the word cloud.

**library**("SnowballC")  
**library**("wordcloud")

## Warning: package 'wordcloud' was built under R version 3.6.3

## Loading required package: RColorBrewer

**library**("RColorBrewer")  
  
physics\_words **%>%**  
 **count**(word)

## # A tibble: 7,490 x 2  
## word n  
## <chr> <int>  
## 1 \_1 1  
## 2 \_2 1  
## 3 \_3 1  
## 4 \_a 3  
## 5 \_a\_ 3  
## 6 \_a\_bsolute 1  
## 7 \_a\_ir 1  
## 8 \_a\_n 1  
## 9 \_a\_ncients 1  
## 10 \_ab\_ 2  
## # … with 7,480 more rows

**wordcloud**(physics\_words**$**word, physics\_words**$**n, min.freq = 50, max.words = 200, random.order=FALSE, rot.per=0.35, colors=**brewer.pal**(8, "Dark2"))



**Chart 2**

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**()

 #\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ## N-grams analysis from the corpus #\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

N-grams help to understand the relationships between words. Bigrams require unnesting tokens by groups of two words occurring together the most.

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

## # A tibble: 149,423 x 3  
## gutenberg\_id author bigram   
## <int> <chr> <chr>   
## 1 13476 Tesla, Nikola experiments with   
## 2 13476 Tesla, Nikola with alternate   
## 3 13476 Tesla, Nikola alternate currents  
## 4 13476 Tesla, Nikola currents of   
## 5 13476 Tesla, Nikola of high   
## 6 13476 Tesla, Nikola high potential   
## 7 13476 Tesla, Nikola potential and   
## 8 13476 Tesla, Nikola and high   
## 9 13476 Tesla, Nikola high frequency   
## 10 13476 Tesla, Nikola frequency a   
## # … with 149,413 more rows

The most common bigrams from dplyr’s count().

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

## # A tibble: 59,858 x 2  
## bigram n  
## <chr> <int>  
## 1 of the 2786  
## 2 to the 1175  
## 3 in the 980  
## 4 that the 553  
## 5 the water 505  
## 6 it is 502  
## 7 the same 471  
## 8 and the 389  
## 9 from the 349  
## 10 to be 335  
## # … with 59,848 more rows

Removing the stop words requires tidyr’s separate() splits a column into multiple based on a delimiter into two columns, “word1” and “word2”, removing cases where either is a stop-word.

**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

## # A tibble: 9,783 x 3  
## word1 word2 n  
## <chr> <chr> <int>  
## 1 straight line 115  
## 2 gravitational field 61  
## 3 reference body 56  
## 4 straight lines 49  
## 5 absolute gravity 43  
## 6 ordinate system 39  
## 7 special theory 37  
## 8 cone \_a 35  
## 9 illustration fig 33  
## 10 ethereal matter 31  
## # … with 9,773 more rows

tidyr’s unite() function recombines the columns into one.

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

## # A tibble: 14,312 x 3  
## gutenberg\_id author bigram   
## <int> <chr> <chr>   
## 1 13476 Tesla, Nikola alternate currents   
## 2 13476 Tesla, Nikola lecture delivered   
## 3 13476 Tesla, Nikola electrical engineers  
## 4 13476 Tesla, Nikola engineers london   
## 5 13476 Tesla, Nikola nikola tesla   
## 6 13476 Tesla, Nikola biographical sketch   
## 7 13476 Tesla, Nikola york 1892   
## 8 13476 Tesla, Nikola 1892 biographical   
## 9 13476 Tesla, Nikola biographical sketch   
## 10 13476 Tesla, Nikola nikola tesla   
## # … with 14,302 more rows

Trigrams are three consecutive terms occurring together frequently.

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)

## # A tibble: 3,213 x 4  
## word1 word2 word3 n  
## <chr> <chr> <chr> <int>  
## 1 degrees 40 minutes 17  
## 2 degrees 20 minutes 15  
## 3 grave \_in specie\_ 14  
## 4 \_x\_1 \_x\_2 \_x\_3 13  
## 5 45 degrees 20 11  
## 6 light \_in vacuo\_ 11  
## 7 space time continuum 11  
## 8 6 degrees 40 10  
## 9 disruptive discharge coil 10  
## 10 reference body \_k\_ 10  
## # … with 3,203 more rows

Performing a tf-idf analysis of the bigrams.

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

## # A tibble: 4 x 3  
## author word1 n  
## <chr> <chr> <int>  
## 1 Huygens, Christiaan solid 8  
## 2 Huygens, Christiaan obtuse 6  
## 3 Galilei, Galileo sharp 1  
## 4 Huygens, Christiaan equilateral 1

**Chart 3**

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

## # A tibble: 9,862 x 6  
## author bigram n tf idf tf\_idf  
## <chr> <chr> <int> <dbl> <dbl> <dbl>  
## 1 Einstein, Albert gravitational field 61 0.0163 1.39 0.0226  
## 2 Einstein, Albert reference body 56 0.0150 1.39 0.0207  
## 3 Galilei, Galileo absolute gravity 43 0.0135 1.39 0.0188  
## 4 Galilei, Galileo cone \_a 35 0.0110 1.39 0.0153  
## 5 Einstein, Albert ordinate system 39 0.0104 1.39 0.0144  
## 6 Einstein, Albert special theory 37 0.00988 1.39 0.0137  
## 7 Galilei, Galileo \_in specie\_ 30 0.00945 1.39 0.0131  
## 8 Galilei, Galileo specificall gravity 30 0.00945 1.39 0.0131  
## 9 Huygens, Christiaan ethereal matter 31 0.00882 1.39 0.0122  
## 10 Huygens, Christiaan 3 2 30 0.00853 1.39 0.0118  
## # … with 9,852 more rows

Visualizing the relationships is possible with network graphs. The source node is where the edge is originating from The target node is where the edge is terminating. The numeric value of the edge is the relationship strength value. The igraph package has many powerful functions for manipulating and analyzing networks. One way to create an igraph object from tidy data is the graph\_from\_data\_frame() function, which takes a data frame of edges with columns for “from”, “to”, and edge attributes (in this case n).

**library**(igraph)

##   
## Attaching package: 'igraph'

## The following object is masked from 'package:tidyr':  
##   
## crossing

## The following objects are masked from 'package:dplyr':  
##   
## as\_data\_frame, groups, union

## The following objects are masked from 'package:stats':  
##   
## decompose, spectrum

## The following object is masked from 'package:base':  
##   
## union

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

## IGRAPH a24c2c4 DN-- 43 23 --   
## + attr: name (v/c), n (e/n)  
## + edges from a24c2c4 (vertex names):  
## [1] straight ->line gravitational->field   
## [3] reference ->body straight ->lines   
## [5] absolute ->gravity ordinate ->system   
## [7] special ->theory cone ->\_a   
## [9] illustration ->fig ethereal ->matter   
## [11] lorentz ->transformation \_in ->specie\_   
## [13] 3 ->2 specificall ->gravity   
## [15] irregular ->refraction semi ->diameter   
## + ... omitted several edges

|  |
| --- |
| The ggraph package (Pedersen 2017) converts an igraph object into a ggraph with the ggraph function. ### Chart 4 |
| ```r 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) ``` |
|  |

We add the edge\_alpha aesthetic to the link layer to make links transparent based on how common or rare the bigram is We add directionality with an arrow, constructed using grid::arrow(), including an end\_cap option that tells the arrow to end before touching the node We tinker with the options to the node layer to make the nodes more attractive (larger, blue points) We add a theme that’s useful for plotting networks, theme\_void()

**Chart 5**

**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 HYPERLINK "Amazon"& HYPERLINK "Amazon"tag=juliasilge-20 HYPERLINK "Amazon"& HYPERLINK "Amazon"camp=1789 HYPERLINK "Amazon"& HYPERLINK "Amazon"creative=9325 HYPERLINK "Amazon"& HYPERLINK "Amazon"linkCode=as2 HYPERLINK "Amazon"& HYPERLINK "Amazon"creativeASIN=1491981652 HYPERLINK "Amazon"& HYPERLINK "Amazon"linkId=0e92d44b0aa39ab34608ffa582dbd490](Amazon).

**citation**("dplyr")

##   
## To cite package 'dplyr' in publications use:  
##   
## Hadley Wickham, Romain François, Lionel Henry and Kirill Müller  
## (2019). dplyr: A Grammar of Data Manipulation. R package version  
## 0.8.3. [https://CRAN.R-project.org/package=dplyr](https://cran.r-project.org/package=dplyr)  
##   
## A BibTeX entry for LaTeX users is  
##   
## @Manual{,  
## title = {dplyr: A Grammar of Data Manipulation},  
## author = {Hadley Wickham and Romain François and Lionel Henry and Kirill Müller},  
## year = {2019},  
## note = {R package version 0.8.3},  
## url = {[https://CRAN.R-project.org/package=dplyr](https://cran.r-project.org/package=dplyr)},  
## }

**citation**("tidytext")

##   
## Silge J, Robinson D (2016). "tidytext: Text Mining and Analysis Using  
## Tidy Data Principles in R." \_JOSS\_, \*1\*(3). doi: 10.21105/joss.00037  
## (URL: <https://doi.org/10.21105/joss.00037>), <URL:  
## <http://dx.doi.org/10.21105/joss.00037>>.  
##   
## A BibTeX entry for LaTeX users is  
##   
## @Article{,  
## title = {tidytext: Text Mining and Analysis Using Tidy Data Principles in R},  
## author = {Julia Silge and David Robinson},  
## doi = {10.21105/joss.00037},  
## url = {<http://dx.doi.org/10.21105/joss.00037>},  
## year = {2016},  
## publisher = {The Open Journal},  
## volume = {1},  
## number = {3},  
## journal = {JOSS},  
## }

**citation**("ggplot2")

##   
## To cite ggplot2 in publications, please use:  
##   
## H. Wickham. ggplot2: Elegant Graphics for Data Analysis.  
## Springer-Verlag New York, 2016.  
##   
## A BibTeX entry for LaTeX users is  
##   
## @Book{,  
## author = {Hadley Wickham},  
## title = {ggplot2: Elegant Graphics for Data Analysis},  
## publisher = {Springer-Verlag New York},  
## year = {2016},  
## isbn = {978-3-319-24277-4},  
## url = {[https://ggplot2.tidyverse.org](https://ggplot2.tidyverse.org/)},  
## }

**citation**("gutenbergr")

##   
## To cite package 'gutenbergr' in publications use:  
##   
## David Robinson (2019). gutenbergr: Download and Process Public Domain  
## Works from Project Gutenberg. R package version 0.1.5.  
## [https://CRAN.R-project.org/package=gutenbergr](https://cran.r-project.org/package=gutenbergr)  
##   
## A BibTeX entry for LaTeX users is  
##   
## @Manual{,  
## title = {gutenbergr: Download and Process Public Domain Works from Project Gutenberg},  
## author = {David Robinson},  
## year = {2019},  
## note = {R package version 0.1.5},  
## url = {[https://CRAN.R-project.org/package=gutenbergr](https://cran.r-project.org/package=gutenbergr)},  
## }

**citation**("forcats")

##   
## To cite package 'forcats' in publications use:  
##   
## Hadley Wickham (2019). forcats: Tools for Working with Categorical  
## Variables (Factors). R package version 0.4.0.  
## [https://CRAN.R-project.org/package=forcats](https://cran.r-project.org/package=forcats)  
##   
## A BibTeX entry for LaTeX users is  
##   
## @Manual{,  
## title = {forcats: Tools for Working with Categorical Variables (Factors)},  
## author = {Hadley Wickham},  
## year = {2019},  
## note = {R package version 0.4.0},  
## url = {[https://CRAN.R-project.org/package=forcats](https://cran.r-project.org/package=forcats)},  
## }

**citation**("stringr")

##   
## To cite package 'stringr' in publications use:  
##   
## Hadley Wickham (2019). stringr: Simple, Consistent Wrappers for  
## Common String Operations. R package version 1.4.0.  
## [https://CRAN.R-project.org/package=stringr](https://cran.r-project.org/package=stringr)  
##   
## A BibTeX entry for LaTeX users is  
##   
## @Manual{,  
## title = {stringr: Simple, Consistent Wrappers for Common String Operations},  
## author = {Hadley Wickham},  
## year = {2019},  
## note = {R package version 1.4.0},  
## url = {[https://CRAN.R-project.org/package=stringr](https://cran.r-project.org/package=stringr)},  
## }

**citation**("wordcloud")

##   
## To cite package 'wordcloud' in publications use:  
##   
## Ian Fellows (2018). wordcloud: Word Clouds. R package version 2.6.  
## [https://CRAN.R-project.org/package=wordcloud](https://cran.r-project.org/package=wordcloud)  
##   
## A BibTeX entry for LaTeX users is  
##   
## @Manual{,  
## title = {wordcloud: Word Clouds},  
## author = {Ian Fellows},  
## year = {2018},  
## note = {R package version 2.6},  
## url = {[https://CRAN.R-project.org/package=wordcloud](https://cran.r-project.org/package=wordcloud)},  
## }  
##   
## ATTENTION: This citation information has been auto-generated from the  
## package DESCRIPTION file and may need manual editing, see  
## 'help("citation")'.

**citation**("tidyr")

##   
## To cite package 'tidyr' in publications use:  
##   
## Hadley Wickham and Lionel Henry (2020). tidyr: Tidy Messy Data. R  
## package version 1.0.2. [https://CRAN.R-project.org/package=tidyr](https://cran.r-project.org/package=tidyr)  
##   
## A BibTeX entry for LaTeX users is  
##   
## @Manual{,  
## title = {tidyr: Tidy Messy Data},  
## author = {Hadley Wickham and Lionel Henry},  
## year = {2020},  
## note = {R package version 1.0.2},  
## url = {[https://CRAN.R-project.org/package=tidyr](https://cran.r-project.org/package=tidyr)},  
## }

**citation**("igraph")

##   
## To cite 'igraph' in publications use:  
##   
## Csardi G, Nepusz T: The igraph software package for complex network  
## research, InterJournal, Complex Systems 1695. 2006. [http://igraph.org](http://igraph.org/)  
##   
## A BibTeX entry for LaTeX users is  
##   
## @Article{,  
## title = {The igraph software package for complex network research},  
## author = {Gabor Csardi and Tamas Nepusz},  
## journal = {InterJournal},  
## volume = {Complex Systems},  
## pages = {1695},  
## year = {2006},  
## url = {[http://igraph.org](http://igraph.org/)},  
## }

**citation**("ggraph")

##   
## To cite package 'ggraph' in publications use:  
##   
## Thomas Lin Pedersen (2020). ggraph: An Implementation of Grammar of  
## Graphics for Graphs and Networks. R package version 2.0.1.  
## [https://CRAN.R-project.org/package=ggraph](https://cran.r-project.org/package=ggraph)  
##   
## A BibTeX entry for LaTeX users is  
##   
## @Manual{,  
## title = {ggraph: An Implementation of Grammar of Graphics for Graphs and Networks},  
## author = {Thomas Lin Pedersen},  
## year = {2020},  
## note = {R package version 2.0.1},  
## url = {[https://CRAN.R-project.org/package=ggraph](https://cran.r-project.org/package=ggraph)},  
## }

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