1. New York Times Bestseller Books Chart Extraction

# libraries  
library(lubridate)

library(stringr)  
library(dplyr)

# functions  
scrape\_nytimes <- function(url, throttle = 0){  
  
   
 # Install / Load relevant packages  
 if(!"pacman" %in% installed.packages()[,"Package"]) install.packages("pacman")  
 pacman::p\_load(RCurl, XML, dplyr, stringr, rvest, purrr)  
   
 # Set throttle between URL calls  
 sec = 0  
 if(throttle < 0) warning("throttle was less than 0: set to 0")  
 if(throttle > 0) sec = max(0, throttle + runif(1, -1, 1))  
   
 # obtain HTML of URL  
 doc <- xml2::read\_html(url)  
   
 # Parse relevant elements from HTML  
 Title <- doc %>%  
 html\_nodes(".css-5pe77f") %>%  
 html\_text()  
   
 Author <- doc %>%  
 html\_nodes(".css-1j7a9fx") %>%  
 html\_text()  
   
 Publisher <- doc %>%  
 html\_nodes(".css-heg334") %>%  
 html\_text()  
   
 # Description <- doc %>%  
 # html\_nodes(".css-14lubdp") %>%  
 # html\_text()  
   
 Weeksonthelist <- doc %>%  
 html\_nodes(".css-1o26r9v") %>%  
 html\_text()  
   
 Weekdate <- doc %>%  
 html\_nodes(".css-1lm6q7y") %>%  
 html\_text()  
   
 Link <- doc %>%  
 html\_nodes(".css-wq7ea0") %>%  
 html\_attr("href")  
   
 Link <- Link[seq(from = 1, to = 45, by = 3)]  
   
   
 # Combine attributes into a single data frame  
 df <- data.frame(Title, Author, Publisher, Weeksonthelist, Weekdate, Link)  
   
 return(df)  
   
}  
  
get\_prod = function(x){  
 c = str\_split(x, "/")  
 c = unlist(c)  
 c = c[length(c)]  
 c = unlist(strsplit(c,''))  
 c = paste(c[-c(length(c))],collapse ='')  
 ;return (c)  
}

# getting list of sundays  
today <- Sys.Date()+7  
sundays = today   
nytimes\_start\_rank = as.Date("2011-02-13")  
repeat{  
 previous\_sunday <- floor\_date(today, "week")  
 sundays = c(sundays,previous\_sunday)  
 if(previous\_sunday == nytimes\_start\_rank){  
 sundays = sundays[-1]  
 break()}  
 today = previous\_sunday -1  
}  
head(sundays,5)

## [1] "2019-09-01" "2019-08-25" "2019-08-18" "2019-08-11" "2019-08-04"

# extracting fiction books rank from html  
ranks\_all= NULL  
  
for(page\_num in 1:length(sundays){  
   
 print(paste(as.character(round(page\_num/length(sundays)\*100)),"%"))  
 print(page\_num)  
   
 weekscrape = as.character(sundays[page\_num])  
 weekscrape = str\_replace\_all(weekscrape,"-","/")  
   
 url <- paste0("https://www.nytimes.com/books/best-sellers/combined-print-and-e-book-fiction/",weekscrape)  
 ranks <- scrape\_nytimes(url, throttle = 0)  
 ranks\_all <- rbind(ranks\_all, cbind(ranks))  
}

write.csv(ranks\_all,file = "nytimes chart fiction books.csv")  
  
head(ranks\_all,5)

## Title Author Publisher  
## 1 WHERE THE CRAWDADS SING by Delia Owens Putnam  
## 2 THE INN by James Patterson and Candice Fox Little, Brown  
## 3 OUTFOX by Sandra Brown Grand Central  
## 4 A DANGEROUS MAN by Robert Crais Putnam  
## 5 THE TURN OF THE KEY by Ruth Ware Scout  
## Weeksonthelist Weekdate  
## 1 48 weeks on the list August 25, 2019  
## 2 New this week August 25, 2019  
## 3 New this week August 25, 2019  
## 4 New this week August 25, 2019  
## 5 New this week August 25, 2019  
## Link  
## 1 https://www.amazon.com/Where-Crawdads-Sing-Delia-Owens/dp/0735219095?tag=NYTBS-20  
## 2 https://www.amazon.com/Inn-James-Patterson-ebook/dp/B07L2VQBG6?tag=NYTBS-20  
## 3 https://www.amazon.com/Outfox-Sandra-Brown/dp/1455572195?tag=NYTBS-20  
## 4 https://www.amazon.com/Dangerous-Elvis-Cole-Pike-Novel-ebook/dp/B07HW1BWHQ?tag=NYTBS-20  
## 5 https://www.amazon.com/Turn-Key-Ruth-Ware/dp/1501188771?tag=NYTBS-20

# extracting non fiction books rank from html  
ranks\_all= NULL  
  
for(page\_num in 1:length(sundays){  
   
 print(paste(as.character(round(page\_num/length(sundays)\*100)),"%"))  
 print(page\_num)  
   
 weekscrape = as.character(sundays[page\_num])  
 weekscrape = str\_replace\_all(weekscrape,"-","/")  
   
 url <- paste0("https://www.nytimes.com/books/best-sellers/combined-print-and-e-book-nonfiction/",weekscrape)  
 ranks <- scrape\_nytimes(url, throttle = 0)  
 ranks\_all <- rbind(ranks\_all, cbind(ranks))  
}

write.csv(ranks\_all,file = "nytimes chart nonfiction books.csv")  
  
head(ranks\_all,5)

## Title Author Publisher Weeksonthelist  
## 1 EDUCATED by Tara Westover Random House 77 weeks on the list  
## 2 TRICK MIRROR by Jia Tolentino Random House New this week  
## 3 BECOMING by Michelle Obama Crown 39 weeks on the list  
## 4 THREE WOMEN by Lisa Taddeo Avid Reader 5 weeks on the list  
## 5 THE PIONEERS by David McCullough Simon & Schuster 14 weeks on the list  
## Weekdate  
## 1 August 25, 2019  
## 2 August 25, 2019  
## 3 August 25, 2019  
## 4 August 25, 2019  
## 5 August 25, 2019  
## Link  
## 1 https://www.amazon.com/Educated-Memoir-Tara-Westover/dp/0399590501?tag=NYTBS-20  
## 2 https://www.amazon.com/Trick-Mirror-Self-Delusion-Jia-Tolentino/dp/0525510540?tag=NYTBS-20  
## 3 https://www.amazon.com/Becoming-Michelle-Obama/dp/1524763136?tag=NYTBS-20  
## 4 https://www.amazon.com/Three-Women-Lisa-Taddeo/dp/1451642296?tag=NYTBS-20  
## 5 https://www.amazon.com/Pioneers-Heroic-Settlers-Brought-American/dp/1501168681?tag=NYTBS-20

# latest week extraction of nytimes chart fiction books  
new\_sunday = as.character(sundays[1])  
new\_sunday = str\_replace\_all(new\_sunday,"-","/")  
  
url <- paste0("https://www.nytimes.com/books/best-sellers/combined-print-and-e-book-fiction/",new\_sunday)  
  
new\_rank <- scrape\_nytimes(url, throttle = 0)  
new\_ranks\_all <- read.csv("nytimes chart fiction books.csv")  
new\_ranks\_all = new\_ranks\_all[,-1]  
new\_ranks\_all <- rbind(new\_rank, new\_ranks\_all)  
  
write.csv(new\_ranks\_all,file = "nytimes chart fiction books.csv")  
  
head(new\_ranks\_all,5)

## Title Author  
## 1 WHERE THE CRAWDADS SING by Delia Owens  
## 2 THE BITTERROOTS by C.J. Box  
## 3 CONTRABAND by Stuart Woods  
## 4 THE INN by James Patterson and Candice Fox  
## 5 THE ART OF RACING IN THE RAIN by Garth Stein  
## Publisher Weeksonthelist Weekdate  
## 1 Putnam 49 weeks on the list September 1, 2019  
## 2 Minotaur New this week September 1, 2019  
## 3 Putnam New this week September 1, 2019  
## 4 Little, Brown 2 weeks on the list September 1, 2019  
## 5 HarperCollins 4 weeks on the list September 1, 2019  
## Link  
## 1 https://www.amazon.com/Where-Crawdads-Sing-Delia-Owens/dp/0735219095?tag=NYTBS-20  
## 2 https://www.amazon.com/Bitterroots-Novel-Cassie-Dewell/dp/1250051053?tag=NYTBS-20  
## 3 https://www.amazon.com/Contraband-Stone-Barrington-Novel-Book-ebook/dp/B07KNTLYYF?tag=NYTBS-20  
## 4 https://www.amazon.com/Inn-James-Patterson-ebook/dp/B07L2VQBG6?tag=NYTBS-20  
## 5 http://www.amazon.com/The-Racing-Rain-Garth-Stein-ebook/dp/B0017SWPXY?tag=NYTBS-20

# latest week extraction of nytimes chart nonfiction books  
new\_sunday = as.character(sundays[1])  
new\_sunday = str\_replace\_all(new\_sunday,"-","/")  
  
url <- paste0("https://www.nytimes.com/books/best-sellers/combined-print-and-e-book-nonfiction/",new\_sunday)  
  
new\_rank <- scrape\_nytimes(url, throttle = 0)  
new\_ranks\_all <- read.csv("nytimes chart nonfiction books.csv")  
new\_ranks\_all = new\_ranks\_all[,-1]  
new\_ranks\_all <- rbind(new\_rank, new\_ranks\_all)  
  
  
write.csv(new\_ranks\_all,file = "nytimes chart nonfiction books.csv")  
  
head(new\_ranks\_all,5)

## Title Author Publisher  
## 1 EDUCATED by Tara Westover Random House  
## 2 HOW TO BE AN ANTIRACIST by Ibram X. Kendi One World  
## 3 BECOMING by Michelle Obama Crown  
## 4 BORN A CRIME by Trevor Noah Spiegel & Grau  
## 5 THREE WOMEN by Lisa Taddeo Avid Reader  
## Weeksonthelist Weekdate  
## 1 78 weeks on the list September 1, 2019  
## 2 New this week September 1, 2019  
## 3 40 weeks on the list September 1, 2019  
## 4 57 weeks on the list September 1, 2019  
## 5 6 weeks on the list September 1, 2019  
## Link  
## 1 https://www.amazon.com/Educated-Memoir-Tara-Westover/dp/0399590501?tag=NYTBS-20  
## 2 https://www.amazon.com/How-Be-Antiracist-Ibram-Kendi/dp/0525509283?tag=NYTBS-20  
## 3 https://www.amazon.com/Becoming-Michelle-Obama/dp/1524763136?tag=NYTBS-20  
## 4 https://www.amazon.com/Born-Crime-Stories-African-Childhood-ebook/dp/B01DHWACVY?tag=NYTBS-20  
## 5 https://www.amazon.com/Three-Women-Lisa-Taddeo/dp/1451642296?tag=NYTBS-20

# tidy datasets  
dataf1 = read.csv("nytimes chart fiction books-knit.csv")  
dataf1$Type = "Fiction"  
dataf2 = read.csv("nytimes chart nonfiction books-knit.csv")  
dataf2$Type = "Non Fiction"  
dataf = rbind(dataf1,dataf2)  
  
dataf$Weekdate = format(as.Date(dataf$Weekdate, "%B %d,%Y"))  
dataf$Weekdate = as.Date(dataf$Weekdate,"%Y-%m-%d")  
  
dataf$Weeksonthelist = str\_remove\_all(dataf$Weeksonthelist," weeks on the list")  
dataf$Weeksonthelist = str\_replace\_all(dataf$Weeksonthelist,"New this week","1")  
dataf$Weeksonthelist = as.numeric(dataf$Weekso)  
  
dataf$Author = str\_remove\_all(dataf$Author,"by ")  
  
dataf$Title = str\_to\_title(dataf$Title)  
  
dataf$Link = str\_remove\_all(dataf$Link, "tag=NYTBS-20")  
  
for (i in 1:length(dataf$Link)){  
 dataf$Product[i] = get\_prod(dataf$Link[i])  
}  
  
write.csv(dataf, "nytimes chart books-knit.csv")  
  
head(dataf,5)

## X Title Author  
## 1 1 Where The Crawdads Sing Delia Owens  
## 2 2 The Bitterroots C.J. Box  
## 3 3 Contraband Stuart Woods  
## 4 4 The Inn James Patterson and Candice Fox  
## 5 5 The Art Of Racing In The Rain Garth Stein  
## Publisher Weeksonthelist Weekdate  
## 1 Putnam 49 2019-09-01  
## 2 Minotaur 1 2019-09-01  
## 3 Putnam 1 2019-09-01  
## 4 Little, Brown 2 2019-09-01  
## 5 HarperCollins 4 2019-09-01  
## Link  
## 1 https://www.amazon.com/Where-Crawdads-Sing-Delia-Owens/dp/0735219095?  
## 2 https://www.amazon.com/Bitterroots-Novel-Cassie-Dewell/dp/1250051053?  
## 3 https://www.amazon.com/Contraband-Stone-Barrington-Novel-Book-ebook/dp/B07KNTLYYF?  
## 4 https://www.amazon.com/Inn-James-Patterson-ebook/dp/B07L2VQBG6?  
## 5 http://www.amazon.com/The-Racing-Rain-Garth-Stein-ebook/dp/B0017SWPXY?  
## Type Product  
## 1 Fiction 0735219095  
## 2 Fiction 1250051053  
## 3 Fiction B07KNTLYYF  
## 4 Fiction B07L2VQBG6  
## 5 Fiction B0017SWPXY

2.Tidy New York Times Bestseller Books Chart

# libraries  
library(lubridate)

library(stringr)  
library(dplyr)

if(!"pacman" %in% installed.packages()[,"Package"]) install.packages("pacman")  
 pacman::p\_load(RCurl, XML, dplyr, stringr, rvest, purrr)

# getting the data  
nytimesdataf = read.csv("C:/Users/10/Documents/R/Julia Silge, David Robinson - Text Mining with R\_ A Tidy Approach/nytimes chart books.csv")  
  
nytimesdataf = nytimesdataf[,-c(1,2,8)]  
head(nytimesdataf,5)

## Title Author  
## 1 Where The Crawdads Sing Delia Owens  
## 2 The Bitterroots C.J. Box  
## 3 Contraband Stuart Woods  
## 4 The Inn James Patterson and Candice Fox  
## 5 The Art Of Racing In The Rain Garth Stein  
## Publisher Weeksonthelist Weekdate Type Product  
## 1 Putnam 49 2019-09-01 Fiction 0735219095  
## 2 Minotaur 1 2019-09-01 Fiction 1250051053  
## 3 Putnam 1 2019-09-01 Fiction B07KNTLYYF  
## 4 Little, Brown 2 2019-09-01 Fiction B07L2VQBG6  
## 5 HarperCollins 4 2019-09-01 Fiction B0017SWPXY

# sort out the titles by authors and publishers  
firstweek\_on\_the\_chart = nytimesdataf[nytimesdataf$Weeksonthelist ==1 ,]  
  
head(firstweek\_on\_the\_chart,5)

## Title Author Publisher  
## 2 The Bitterroots C.J. Box Minotaur  
## 3 Contraband Stuart Woods Putnam  
## 6 Blood Truth J.R. Ward Gallery  
## 12 The Wallflower Wager Tessa Dare Avon  
## 13 The Warning James Patterson and Robison Wells Grand Central  
## Weeksonthelist Weekdate Type Product  
## 2 1 2019-09-01 Fiction 1250051053  
## 3 1 2019-09-01 Fiction B07KNTLYYF  
## 6 1 2019-09-01 Fiction 1501195034  
## 12 1 2019-09-01 Fiction B07G14DRJJ  
## 13 1 2019-09-01 Fiction B07L2TXTS5

firstweek\_on\_the\_chart %>%  
count(Publisher, sort = TRUE) %>%  
ungroup()

## # A tibble: 431 x 2  
## Publisher n  
## <fct> <int>  
## 1 Simon & Schuster 179  
## 2 Putnam 148  
## 3 Penguin Group 143  
## 4 Grand Central 123  
## 5 Little, Brown 116  
## 6 St. Martin's 101  
## 7 Random House 100  
## 8 Berkley 91  
## 9 Ballantine 72  
## 10 Harper 66  
## # ... with 421 more rows

firstweek\_on\_the\_chart %>%  
count(Publisher, Author, sort = TRUE) %>%  
ungroup()

## # A tibble: 2,144 x 3  
## Publisher Author n  
## <fct> <fct> <int>  
## 1 Delacorte Danielle Steel 32  
## 2 Putnam Stuart Woods 24  
## 3 Grand Central David Baldacci 22  
## 4 Berkley Christine Feehan 20  
## 5 Ballantine Debbie Macomber 17  
## 6 Delacorte Lee Child 15  
## 7 St. Martin's Iris Johansen 15  
## 8 Harlequin Susan Mallery 14  
## 9 Scribner Stephen King 13  
## 10 Doubleday John Grisham 12  
## # ... with 2,134 more rows

firstweek\_on\_the\_chart %>%  
count(Publisher, Title, sort = TRUE) %>%  
ungroup()

## # A tibble: 3,314 x 3  
## Publisher Title n  
## <fct> <fct> <int>  
## 1 Algonquin An American Marriage 2  
## 2 Ballantine Before We Were Yours 2  
## 3 Bantam Tricky Twenty-Two 2  
## 4 Del Rey Thrawn: Alliances 2  
## 5 Delacorte Make Me 2  
## 6 Delacorte Never Go Back 2  
## 7 Doubleday Gray Mountain 2  
## 8 Doubleday Rogue Lawyer 2  
## 9 Doubleday Sycamore Row 2  
## 10 Farrar, Straus & Giroux Thinking, Fast And Slow 2  
## # ... with 3,304 more rows

firstweek\_on\_the\_chart %>%  
count(Author, sort = TRUE) %>%  
ungroup()

## # A tibble: 1,630 x 2  
## Author n  
## <fct> <int>  
## 1 Danielle Steel 41  
## 2 Christine Feehan 34  
## 3 Stuart Woods 31  
## 4 Nora Roberts 28  
## 5 Debbie Macomber 27  
## 6 Susan Mallery 27  
## 7 David Baldacci 26  
## 8 Robyn Carr 24  
## 9 Iris Johansen 20  
## 10 James Patterson 18  
## # ... with 1,620 more rows

firstweek\_on\_the\_chart %>%  
count(Author, Title, sort = TRUE) %>%  
ungroup()

## # A tibble: 3,308 x 3  
## Author Title n  
## <fct> <fct> <int>  
## 1 A.J. Finn The Woman In The Window 2  
## 2 Angela Duckworth Grit 2  
## 3 Anthony Doerr All The Light We Cannot See 2  
## 4 Arianna Huffington Thrive 2  
## 5 Brian Kilmeade and Don Yaeger Thomas Jefferson And The Trip~ 2  
## 6 Chris Kyle with Scott McEwen and J~ American Sniper 2  
## 7 Chris Smith The Daily Show (The Book) 2  
## 8 Dan Brown Inferno 2  
## 9 Daniel Kahneman Thinking, Fast And Slow 2  
## 10 David Baldacci The Escape 2  
## # ... with 3,298 more rows

nytimesdataf %>%  
count(Publisher, sort = TRUE) %>%  
ungroup()

## # A tibble: 456 x 2  
## Publisher n  
## <fct> <int>  
## 1 Little, Brown 793  
## 2 Simon & Schuster 746  
## 3 Random House 591  
## 4 Grand Central 456  
## 5 Penguin Group 436  
## 6 Doubleday 386  
## 7 Crown 363  
## 8 Knopf Doubleday Publishing 361  
## 9 Putnam 347  
## 10 Scribner 325  
## # ... with 446 more rows

nytimesdataf %>%  
count(Publisher, Author, sort = TRUE) %>%  
ungroup()

## # A tibble: 2,287 x 3  
## Publisher Author n  
## <fct> <fct> <int>  
## 1 Holt Bill O'Reilly and Martin Dugard 187  
## 2 Knopf Doubleday Publishing E. L. James 171  
## 3 Doubleday John Grisham 168  
## 4 Grand Central David Baldacci 144  
## 5 Random House Laura Hillenbrand 134  
## 6 Thomas Nelson Todd Burpo with Lynn Vincent 129  
## 7 Riverhead Paula Hawkins 115  
## 8 Scribner Stephen King 104  
## 9 Delacorte Lee Child 91  
## 10 Grand Central Nicholas Sparks 90  
## # ... with 2,277 more rows

nytimesdataf %>%  
count(Publisher, Title, sort = TRUE) %>%  
ungroup()

## # A tibble: 3,520 x 3  
## Publisher Title n  
## <fct> <fct> <int>  
## 1 Random House Unbroken 134  
## 2 Thomas Nelson Heaven Is For Real 129  
## 3 Riverhead The Girl On The Train 102  
## 4 HarperCollins Hillbilly Elegy 87  
## 5 Harper Sapiens 82  
## 6 Scribner All The Light We Cannot See 81  
## 7 Knopf Wild 79  
## 8 Random House Educated 78  
## 9 Crown Gone Girl 77  
## 10 Random House Publishing Unbroken 77  
## # ... with 3,510 more rows

nytimesdataf %>%  
count(Author, sort = TRUE) %>%  
ungroup()

## # A tibble: 1,686 x 2  
## Author n  
## <fct> <int>  
## 1 Bill O'Reilly and Martin Dugard 254  
## 2 E. L. James 235  
## 3 John Grisham 228  
## 4 Laura Hillenbrand 211  
## 5 David Baldacci 180  
## 6 Gillian Flynn 154  
## 7 Nicholas Sparks 134  
## 8 Todd Burpo with Lynn Vincent 129  
## 9 Daniel James Brown 122  
## 10 Nora Roberts 120  
## # ... with 1,676 more rows

nytimesdataf %>%  
count(Author, Title, sort = TRUE) %>%  
ungroup()

## # A tibble: 3,401 x 3  
## Author Title n  
## <fct> <fct> <int>  
## 1 Laura Hillenbrand Unbroken 211  
## 2 Todd Burpo with Lynn Vincent Heaven Is For Real 129  
## 3 Daniel James Brown The Boys In The Boat 122  
## 4 Gillian Flynn Gone Girl 122  
## 5 Cheryl Strayed Wild 119  
## 6 Paula Hawkins The Girl On The Train 102  
## 7 J.D. Vance Hillbilly Elegy 87  
## 8 Rebecca Skloot The Immortal Life Of Henrietta Lacks 87  
## 9 Yuval Noah Harari Sapiens 82  
## 10 Anthony Doerr All The Light We Cannot See 81  
## # ... with 3,391 more rows

# getting review of the best seller books by publishers  
best\_publishers = nytimesdataf %>%  
 count(Publisher, sort = TRUE) %>%  
 ungroup()  
head(best\_publishers,5)

## # A tibble: 5 x 2  
## Publisher n  
## <fct> <int>  
## 1 Little, Brown 793  
## 2 Simon & Schuster 746  
## 3 Random House 591  
## 4 Grand Central 456  
## 5 Penguin Group 436

titles\_by\_best\_publishers = firstweek\_on\_the\_chart[firstweek\_on\_the\_chart$Publisher == "Little, Brown",]  
head(titles\_by\_best\_publishers,5)

## Title Author  
## 17 The Inn James Patterson and Candice Fox  
## 115 Big Sky Kate Atkinson  
## 121 Summer Of '69 Elin Hilderbrand  
## 226 The 18th Abduction James Patterson and Maxine Paetro  
## 305 The Cornwalls Are Gone James Patterson and Brendan DuBois  
## Publisher Weeksonthelist Weekdate Type Product  
## 17 Little, Brown 1 2019-08-25 Fiction B07L2VQBG6  
## 115 Little, Brown 1 2019-07-14 Fiction 0316523097  
## 121 Little, Brown 1 2019-07-07 Fiction 0316420018  
## 226 Little, Brown 1 2019-05-19 Fiction B07CRJ2H4L  
## 305 Little, Brown 1 2019-04-14 Fiction 0316485551

productlist = as.character(titles\_by\_best\_publishers$Product)  
head(productlist,5)

## [1] "B07L2VQBG6" "0316523097" "0316420018" "B07CRJ2H4L" "0316485551"

3. Scraping Reviews From Amazon

library(lubridate)

library(stringr)  
library(dplyr)

if(!"pacman" %in% installed.packages()[,"Package"]) install.packages("pacman")  
 pacman::p\_load(RCurl, XML, dplyr, stringr, rvest, purrr)

#Function to scrape elements from Amazon reviews  
scrape\_amazon <- function(url, throttle = 0){  
  
 # Install / Load relevant packages  
 if(!"pacman" %in% installed.packages()[,"Package"]) install.packages("pacman")  
 pacman::p\_load(RCurl, XML, dplyr, stringr, rvest, purrr)  
  
 # Set throttle between URL calls  
 sec = 0  
 if(throttle < 0) warning("throttle was less than 0: set to 0")  
 if(throttle > 0) sec = max(0, throttle + runif(1, -1, 1))  
  
 # obtain HTML of URL  
 doc <- read\_html(url)  
  
 # Parse relevant elements from HTML  
 title <- doc %>%  
 html\_nodes("#cm\_cr-review\_list .a-color-base") %>%  
 html\_text()  
  
 author <- doc %>%  
 html\_nodes("#cm\_cr-review\_list .a-profile-name") %>%  
 html\_text()  
  
 date <- doc %>%  
 html\_nodes("#cm\_cr-review\_list .review-date") %>%  
 html\_text() %>%  
 gsub(".\*on ", "", .)  
  
 review\_format <- doc %>%  
 html\_nodes(".review-format-strip") %>%  
 html\_text()  
  
 stars <- doc %>%  
 html\_nodes("#cm\_cr-review\_list .review-rating") %>%  
 html\_text() %>%  
 str\_extract("\\d") %>%  
 as.numeric()  
  
 comments <- doc %>%  
 html\_nodes("#cm\_cr-review\_list .review-text") %>%  
 html\_text()  
  
 suppressWarnings(n\_helpful <- doc %>%  
 html\_nodes(".a-expander-inline-container") %>%  
 html\_text() %>%  
 gsub("\n\n \\s\*|found this helpful.\*", "", .) %>%  
 gsub("One", "1", .) %>%  
 map\_chr(~ str\_split(string = .x, pattern = " ")[[1]][1]) %>%  
 as.numeric())  
  
 # Combine attributes into a single data frame  
 df <- data.frame(title, author, date, review\_format, stars, comments, n\_helpful, stringsAsFactors = F)  
  
 return(df)  
}  
  
  
#loop over books  
  
books= # put the list of books you want to scrape  
  
prod1 = NULL  
eprod1 = NULL  
for(k in 1:length(books)){  
 #Product code  
 prod\_code <- books[k]  
  
 url <- paste0("https://www.amazon.com/dp/", prod\_code)  
 doc <- read\_html(url)  
  
 prod = NULL  
 # obtain the text in the node, remove "\n" from the text, and remove white space  
 prod <- html\_nodes(doc, "#productTitle") %>% html\_text() %>% gsub("\n", "", .) %>% trimws()  
 prod1 = c(prod1,prod)  
  
 eprod = NULL  
 eprod <- html\_nodes(doc, "#ebooksProductTitle") %>% html\_text() %>% gsub("\n", "", .) %>% trimws()  
 eprod1 = c(eprod1,eprod)  
  
}  
prod = c(prod1,eprod1)  
  
  
for(book\_num in 1:length(books)){  
#Product code  
prod\_code <- books[book\_num]  
  
url <- paste0("https://www.amazon.com/dp/", prod\_code)  
doc <- read\_html(url)  
  
product = NULL  
#obtain the text in the node, remove "\n" from the text, and remove white space  
product <- html\_nodes(doc, "#ebooksProductTitle") %>% html\_text() %>% gsub("\n", "", .) %>% trimws()  
if(length(product) == 0){  
 product <- html\_nodes(doc, "#productTitle") %>% html\_text() %>% gsub("\n", "", .) %>% trimws()  
 }  
  
# Set # of pages to scrape. Note: each page contains 10 reviews.  
reviewsnum<- html\_nodes(doc, "#acrCustomerReviewText") %>% html\_text() %>% gsub("\n", "", .) %>% trimws()  
all\_review\_pages = str\_remove(reviewsnum," customer reviews")  
all\_review\_pages = as.integer(str\_remove(all\_review\_pages,",") )  
all\_review\_pages <- floor(all\_review\_pages/10)+1  
  
#Getting number of Reviews Per each grading star  
xstar\_reviews = NULL  
xstar = c("one","two","three","four","five")  
for(x in xstar){  
 url <- paste0("https://www.amazon.com/product-reviews/",prod\_code,"/ref=cm\_cr\_arp\_d\_viewopt\_sr?filterByStar=",x,"\_star&pageNumber=1")  
 doc\_xstars = read\_html(url)  
 xstar\_review\_pages<- html\_nodes(doc\_xstars,"#filter-info-section > .a-size-base") %>% html\_text() %>% gsub("\n", "", .) %>% trimws()  
 xstar\_review\_pages = xstar\_review\_pages[1]  
 xstar\_review\_pages = gsub("[^0-9.]", "", xstar\_review\_pages)  
 xstar\_review\_pages = as.numeric(xstar\_review\_pages)  
 if(xstar\_review\_pages<11000){  
 xstar\_review\_pages = unlist(str\_extract\_all(as.character(xstar\_review\_pages),""))  
 xstar\_review\_pages = xstar\_review\_pages[length(xstar\_review\_pages)]  
  
 }else{  
 xstar\_review\_pages = unlist(str\_extract\_all(as.character(xstar\_review\_pages),""))  
 xstar\_review\_pages = xstar\_review\_pages[-c(1,2,3)]  
 for(i in 1:length(xstar\_review\_pages)-1){  
 a = xstar\_review\_pages[i]  
 a = paste(a,xstar\_review\_pages[i+1],collapse = "")  
  
 }  
  
 xstar\_review\_pages = str\_remove\_all(a," ")  
 }  
  
 xstar\_reviews = c(xstar\_reviews,xstar\_review\_pages)  
  
}  
  
xstar\_reviews = as.numeric(xstar\_reviews)  
xpages = floor(xstar\_reviews/10)+1  
for(i in c(1:5)){  
 if(xpages[i]>500){  
 xpages[i]=500  
 }  
}  
  
# create empty object to write data into  
reviews\_all <- NULL  
  
if(all\_review\_pages<=500){  
  
  
# loop over all pages  
 for(page\_num in 1:all\_review\_pages){  
 print(paste(as.character(round(page\_num/all\_review\_pages\*100)),"%"))  
 print(page\_num)  
 url <- paste0("http://www.amazon.com/product-reviews/",prod\_code,"/?pageNumber=", page\_num)  
 reviews <- scrape\_amazon(url, throttle = 0)  
 reviews\_all <- rbind(reviews\_all, cbind(prod, reviews))  
 }  
  
}else{  
 for(j in c(1:5)){  
 page\_num = NULL  
 for(page\_num in 1:xpages[j]){  
 print(paste(xstar[j],"star",as.character(round(page\_num/xpages[j]\*100)),"%"))  
 print(page\_num)  
 url <- paste0("https://www.amazon.com/product-reviews/",prod\_code,"/ref=cm\_cr\_arp\_d\_viewopt\_sr?filterByStar=",xstar[j],"\_star&pageNumber=",page\_num)  
 reviews <- scrape\_amazon(url, throttle = 0)  
 reviews\_all <- rbind(reviews\_all, cbind(prod, reviews))  
 }  
 }  
  
 }  
write.csv(reviews\_all,str\_remove(product,":"))  
}

4. Sentiment Analysis of Amazon Reviews

# libraries  
library(janeaustenr)

library(dplyr)

library(stringr)  
library(tidytext)

library(textdata)

library(tidyr)  
library(ggplot2)  
library(wordcloud)

library(reshape2)

# getting the data and tokenizing it by words  
data = read.csv("C:/Users/10/Desktop/GoodReads\_TextMining-master/books/The Time Traveler's Wife.csv")  
  
data\_tidy = data[]  
data\_tidy$date = format(as.Date(data\_tidy$date, "%d-%b-%y"))  
data\_tidy$comments = as.character(data\_tidy$comments)  
data\_tidy = as\_tibble(data\_tidy)  
data\_tidy = data\_tidy %>% arrange(desc(date))  
data\_tidy\_token\_word = as\_tibble(data\_tidy) %>%  
 unnest\_tokens(word, comments)

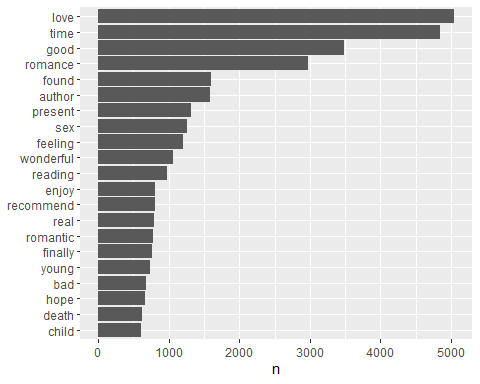
# sentiment analysis based on NRC lexicon  
data\_tidy\_sentiment\_nrc = data\_tidy\_token\_word %>%  
 inner\_join(lexicon\_nrc())

## Joining, by = "word"

data\_tidy\_sentiment\_nrc %>%  
 count(word, sort = TRUE)

## # A tibble: 2,657 x 2  
## word n  
## <chr> <int>  
## 1 love 5048  
## 2 time 4844  
## 3 good 3480  
## 4 romance 2968  
## 5 found 1602  
## 6 author 1584  
## 7 present 1315  
## 8 sex 1252  
## 9 feeling 1200  
## 10 wonderful 1064  
## # ... with 2,647 more rows

#count(word, sort = TRUE)  
  
data\_tidy\_sentiment\_nrc %>%  
count(word, sort = TRUE) %>%  
filter(n > 600) %>%  
mutate(word = reorder(word, n)) %>%  
ggplot(aes(word, n)) +  
geom\_col() +  
xlab(NULL) +  
coord\_flip()



nrcjoy <- lexicon\_nrc() %>%  
filter(sentiment == "joy")  
  
data\_tidy\_token\_word %>%  
inner\_join(nrcjoy) %>%  
count(word, sort = TRUE)

## Joining, by = "word"

## # A tibble: 387 x 2  
## word n  
## <chr> <int>  
## 1 love 2524  
## 2 good 696  
## 3 found 534  
## 4 romance 424  
## 5 sex 313  
## 6 beautiful 267  
## 7 wonderful 266  
## 8 present 263  
## 9 child 203  
## 10 enjoy 202  
## # ... with 377 more rows

nrcpositive <- lexicon\_nrc() %>%  
filter(sentiment == "positive")  
  
data\_tidy\_token\_word %>%  
inner\_join(nrcpositive) %>%  
count(word, sort = TRUE)

## Joining, by = "word"

## # A tibble: 1,121 x 2  
## word n  
## <chr> <int>  
## 1 love 2524  
## 2 reading 969  
## 3 author 792  
## 4 good 696  
## 5 found 534  
## 6 interesting 493  
## 7 traveling 437  
## 8 romance 424  
## 9 reader 403  
## 10 recommend 403  
## # ... with 1,111 more rows

nrctrust <- lexicon\_nrc() %>%  
filter(sentiment == "trust")  
  
data\_tidy\_token\_word %>%  
inner\_join(nrctrust) %>%  
count(word, sort = TRUE)

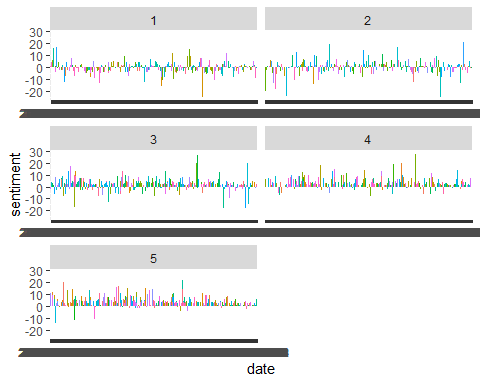
## Joining, by = "word"

## # A tibble: 553 x 2  
## word n  
## <chr> <int>  
## 1 author 792  
## 2 good 696  
## 3 found 534  
## 4 romance 424  
## 5 recommend 403  
## 6 real 395  
## 7 sex 313  
## 8 wonderful 266  
## 9 present 263  
## 10 fact 206  
## # ... with 543 more rows

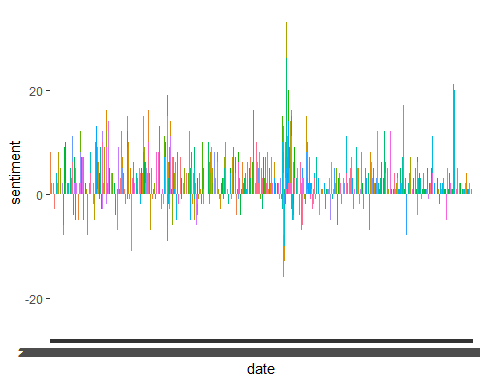
# sentiment analysis based on BING lexicon  
data\_tidy\_sentiment\_bing = data\_tidy\_token\_word %>%  
 inner\_join(get\_sentiments("bing")) %>%  
 count(X, date, title, stars, sentiment) %>%  
 spread(sentiment, n, fill = 0) %>%  
 mutate(sentiment = positive - negative)

## Joining, by = "word"

ggplot(data\_tidy\_sentiment\_bing, aes(date, sentiment, fill = title))+  
 geom\_col(show.legend = FALSE)+  
 facet\_wrap(~stars,ncol = 2, scales = "free\_x")



ggplot(data\_tidy\_sentiment\_bing %>%  
 arrange(date), aes(date, sentiment, fill = title))+  
 geom\_col(show.legend = FALSE)



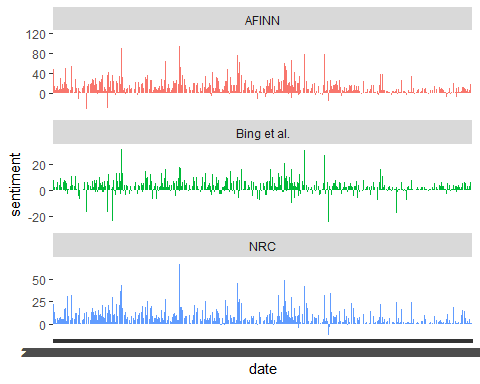
# sentiment analysis based on AFINN lexicon  
data\_tidy\_sentiment\_afinn <- data\_tidy\_token\_word %>%  
 inner\_join(lexicon\_afinn()) %>%  
 group\_by(title, date) %>%  
 summarise(sentiment = sum(value)) %>%  
 mutate(method = "AFINN")

## Joining, by = "word"

# comparing NRc,BING and AFINN  
data\_tidy\_sentiment\_bing\_and\_nrc <- bind\_rows(  
 data\_tidy\_token\_word %>%  
 inner\_join(lexicon\_bing()) %>%  
 mutate(method = "Bing et al."),  
 data\_tidy\_token\_word %>%  
 inner\_join(lexicon\_nrc() %>%  
 filter(sentiment %in% c("positive",  
 "negative"))) %>%  
 mutate(method = "NRC")) %>%  
 count(method, title, date, sentiment) %>%  
 spread(sentiment, n, fill = 0) %>%  
 mutate(sentiment = positive - negative)

## Joining, by = "word"  
## Joining, by = "word"

bind\_rows(data\_tidy\_sentiment\_afinn,  
data\_tidy\_sentiment\_bing\_and\_nrc) %>%  
ggplot(aes(date, sentiment, fill = method)) +  
geom\_col(show.legend = FALSE) +  
facet\_wrap(~method, ncol = 1, scales = "free\_y")

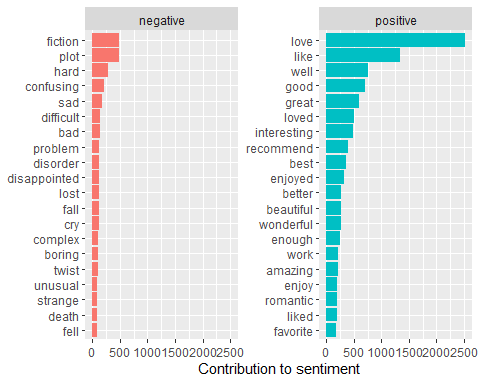


# Contribution to sentiment  
bing\_word\_counts <- data\_tidy\_token\_word %>%  
inner\_join(get\_sentiments("bing")) %>%  
count(word, sentiment, sort = TRUE) %>%  
ungroup()

## Joining, by = "word"

bing\_word\_counts %>%  
group\_by(sentiment) %>%  
top\_n(20) %>%  
ungroup() %>%  
mutate(word = reorder(word, n)) %>%  
ggplot(aes(word, n, fill = sentiment)) +  
geom\_col(show.legend = FALSE) +  
facet\_wrap(~sentiment, scales = "free\_y") +  
labs(y = "Contribution to sentiment",  
x = NULL) +  
coord\_flip()

## Selecting by n



custom\_stop\_words <- bind\_rows(data\_frame(word = c("fiction","time","travel","book","story"),  
lexicon = c("custom")),  
stop\_words)

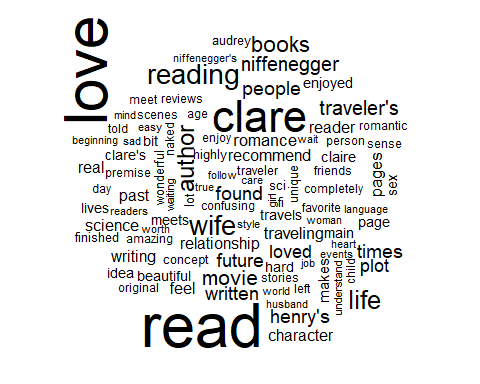
## Warning: `data\_frame()` is deprecated, use `tibble()`.  
## This warning is displayed once per session.

# wordclouds  
data\_tidy\_token\_word %>%  
anti\_join(custom\_stop\_words) %>%  
count(word) %>%  
with(wordcloud(word, n, max.words = 100))

## Joining, by = "word"

## Warning in wordcloud(word, n, max.words = 100): henry could not be fit on  
## page. It will not be plotted.

## Warning in wordcloud(word, n, max.words = 100): characters could not be fit  
## on page. It will not be plotted.



data\_tidy\_token\_word %>%  
inner\_join(get\_sentiments("bing")) %>%  
count(word, sentiment, sort = TRUE) %>%  
acast(word ~ sentiment, value.var = "n", fill = 0) %>%  
comparison.cloud(colors = c("red", "blue"),  
max.words = 100)

## Joining, by = "word"



# tokenizing comments based on sentences  
PandP\_sentences <- as\_tibble(data\_tidy) %>%  
unnest\_tokens(sentence, comments, token = "sentences")

book\_words <- as\_tibble(data\_tidy) %>%  
unnest\_tokens(word, comments) %>%  
count(title, word, stars,date, sort = TRUE) %>%  
ungroup()  
total\_words <- book\_words %>%  
group\_by(title) %>%  
summarize(total = sum(n))  
book\_words <- left\_join(book\_words, total\_words)

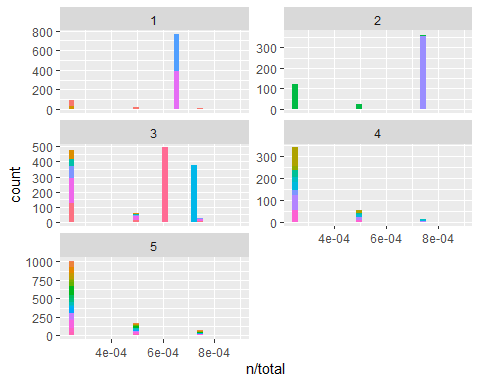
## Joining, by = "title"

ggplot(book\_words, aes(n/total, fill = date)) +  
geom\_histogram(show.legend = FALSE) +  
xlim(NA, 0.0009) +  
facet\_wrap(~stars, ncol = 2, scales = "free\_y")

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 239466 rows containing non-finite values (stat\_bin).

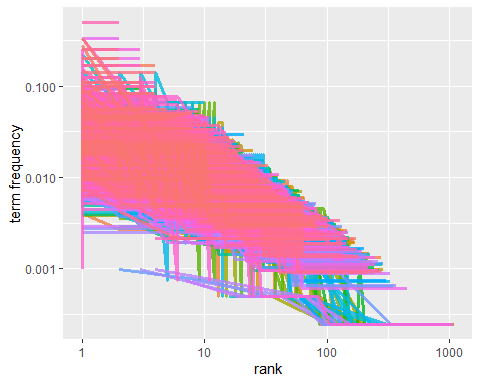
## Warning: Removed 45 rows containing missing values (geom\_bar).



# calculating word frequencies  
custom\_stop\_words <- bind\_rows(data\_frame(word = c("time","clare","henry","book","claire","story","page"),  
lexicon = c("custom")),  
stop\_words)  
  
freq\_by\_rank <- book\_words %>%  
anti\_join(custom\_stop\_words) %>%  
group\_by(title) %>%  
mutate(rank = row\_number(),  
`term frequency` = n/total)

## Joining, by = "word"

freq\_by\_rank %>%  
ggplot(aes(rank, `term frequency`, color = date)) +  
geom\_line(size = 1.1, alpha = 0.8, show.legend = FALSE) +  
scale\_x\_log10() +  
scale\_y\_log10()



# tf-idf  
book\_words <- book\_words %>%  
bind\_tf\_idf(word, title, n)

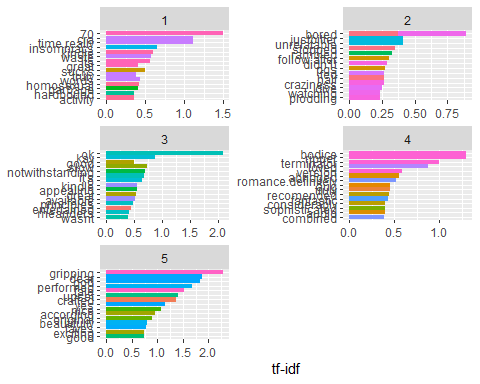
## Warning in bind\_tf\_idf.data.frame(., word, title, n): A value for tf\_idf is negative:  
## Input should have exactly one row per document-term combination.

book\_words %>%  
select(-total) %>%  
arrange(desc(tf\_idf))

## # A tibble: 244,307 x 8  
## title word stars date n tf idf tf\_idf  
## <fct> <chr> <int> <chr> <int> <dbl> <dbl> <dbl>  
## 1 "Well worth reading\n ~ grippi~ 5 2016-1~ 1 0.5 4.60 2.30  
## 2 "Longer than it should h~ ok 3 2018-0~ 1 0.5 4.20 2.10  
## 3 "Read it\n " dear 5 2014-0~ 1 0.333 5.62 1.87  
## 4 "Read it\n " god 5 2014-0~ 1 0.333 5.51 1.84  
## 5 "Once is not enough!\n ~ perfor~ 5 2015-0~ 1 0.25 6.72 1.68  
## 6 "Well worth reading\n ~ tale 5 2016-1~ 1 0.5 3.04 1.52  
## 7 "whats up with the price~ 70 1 2016-1~ 1 0.25 6.02 1.51  
## 8 "I wanted to keep readin~ upset 5 2016-0~ 1 0.25 5.62 1.40  
## 9 "A good story.\n ~ crafted 5 2017-0~ 1 0.333 4.10 1.37  
## 10 "Well-written but not or~ bodice 4 2003-1~ 1 0.167 7.81 1.30  
## # ... with 244,297 more rows

book\_words %>%  
arrange(desc(tf\_idf)) %>%  
mutate(word = factor(word, levels = rev(unique(word)))) %>%  
group\_by(stars) %>%  
top\_n(15) %>%  
ungroup %>%  
ggplot(aes(word, tf\_idf, fill = title)) +  
geom\_col(show.legend = FALSE) +  
labs(x = NULL, y = "tf-idf") +  
facet\_wrap(~stars, ncol = 2, scales = "free") +  
coord\_flip()

## Selecting by tf\_idf



# tokenizing comments based on two words together  
data\_tidy\_bigrams <- as\_tibble(data\_tidy) %>%  
unnest\_tokens(bigram, comments, token = "ngrams", n = 2)  
  
data\_tidy\_bigrams %>%  
count(bigram, sort = TRUE)

## # A tibble: 130,825 x 2  
## bigram n  
## <chr> <int>  
## 1 this book 2409  
## 2 of the 2076  
## 3 the book 1826  
## 4 in the 1375  
## 5 the time 1213  
## 6 time travel 1098  
## 7 the story 1085  
## 8 is a 1052  
## 9 it is 952  
## 10 i was 942  
## # ... with 130,815 more rows

bigrams\_separated <- data\_tidy\_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)

as\_tibble(data\_tidy) %>%  
unnest\_tokens(trigram, comments, 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: 9,830 x 4  
## word1 word2 word3 n  
## <chr> <chr> <chr> <int>  
## 1 time traveler's wife 539  
## 2 chrono displacement disorder 37  
## 3 beautiful love story 34  
## 4 <NA> <NA> <NA> 34  
## 5 henry's time traveling 32  
## 6 time travel stories 28  
## 7 time travel story 28  
## 8 henry time travels 27  
## 9 time travel aspect 27  
## 10 time traveller's wife 25  
## # ... with 9,820 more rows

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

## # A tibble: 665 x 3  
## title word1 n  
## <fct> <chr> <int>  
## 1 "Five Stars\n " favorite 4  
## 2 "There's a Reason It's So Popular ... One of the Best ~ fi 4  
## 3 "I miss my friends\n " audio 3  
## 4 "The Time Traveler's Wife\n " amazing 3  
## 5 "A sappy romance cleverly disguised as well-written sc~ fiction 2  
## 6 "Best book of the year\n " powerful 2  
## 7 "Five Stars\n " amazing 2  
## 8 "Great book!\n " wonderful 2  
## 9 "If you want to fall in love...\n " niffenegg~ 2  
## 10 "The Time Traveler's Wife\n " excellent 2  
## # ... with 655 more rows