BEST SELLER BOOK PREDICTION BASED ON SENTIMENT ANALYSIS

An approach via text mining and data mining techniques and with help of TidyText library

Abstract

Through massive amount of review data which is produced by amazon users , it is possible to get a sense of what people think about a book and with analyzing New York Times weekly chart of bestseller books we can reach to model which can predict the chance for a book to become a bestseller

Roozbeh Dadashzadeh – Dr. Mehran Yazdi

Roozbeh.dadashzadeh@gmail.com

Contents

New York Times Bestseller Books Chart Extraction

```r
# 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-FICT
ION/",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=NY
TBS-20
## 2           HTTPS://WWW.AMAZON.COM/INN-JAMES-PATTERSON-EBOOK/DP/B07L2VQBG6?TAG=NYT
BS-20
## 3                HTTPS://WWW.AMAZON.COM/OUTFOX-SANDRA-BROWN/DP/1455572195?TAG=NYT
BS-20
## 4 HTTPS://WWW.AMAZON.COM/DANGEROUS-ELVIS-COLE-PIKE-NOVEL-EBOOK/DP/B07HW1BWHQ?TAG=NY
TBS-20
## 5                 HTTPS://WWW.AMAZON.COM/TURN-KEY-RUTH-WARE/DP/1501188771?TAG=NYT
BS-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-NONF
ICTION/",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
##                                                                          L
INK
## 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-FICTIO
N/",NEW_SUNDAY)

NEW_RANK <- SCRAPE_NYTIMES(URL, THROTTLE = 0)
```

```r
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?T
AG=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?T
AG=NYTBS-20
## 5           HTTP://WWW.AMAZON.COM/THE-RACING-RAIN-GARTH-STEIN-EBOOK/DP/B0017SWPXY?
TAG=NYTBS-20
```

```r
# 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-NONFIC
TION/",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?TA
G=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?TA
G=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
```

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 MININ
G 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 == "LI
TTLE, 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"
```

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)
```

```r
  # 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, STRINGSA
SFACTORS = F)

  RETURN(DF)
}


#loop over books
```

```r
BOOKS= c("B07L2VQBG6") # 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", "", .) %>% TRI
MWS()
 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")
```

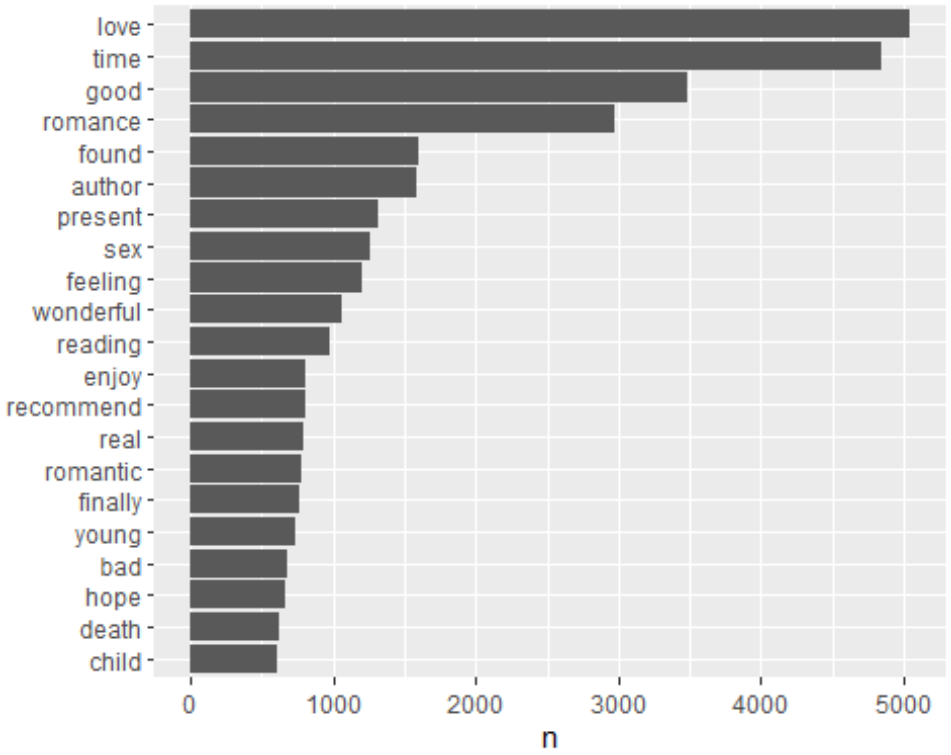
```r
ALL_REVIEW_PAGES = AS.INTEGER(STR_REMOVE(ALL_REVIEW_PAGES,",") )
ALL_REVIEW_PAGES <- FLOOR(ALL_REVIEW_PAGES/10)+1

#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_VIE
WOPT_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=", PAG
E_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,":"))
}
```

Sentiment Analysis of Amazon Reviews

```
# LIBRARIES
LIBRARY(JANEAUSTENR)

LIBRARY(DPLYR)

LIBRARY(STRINGR)
LIBRARY(TIDYTEXT)

LIBRARY(TEXTDATA)

LIBRARY(TIDYR)
LIBRARY(GGPLOT2)
LIBRARY(WORDCLOUD)

LIBRARY(RESHAPE2)
```

```r
# 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")
```

```r
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
```

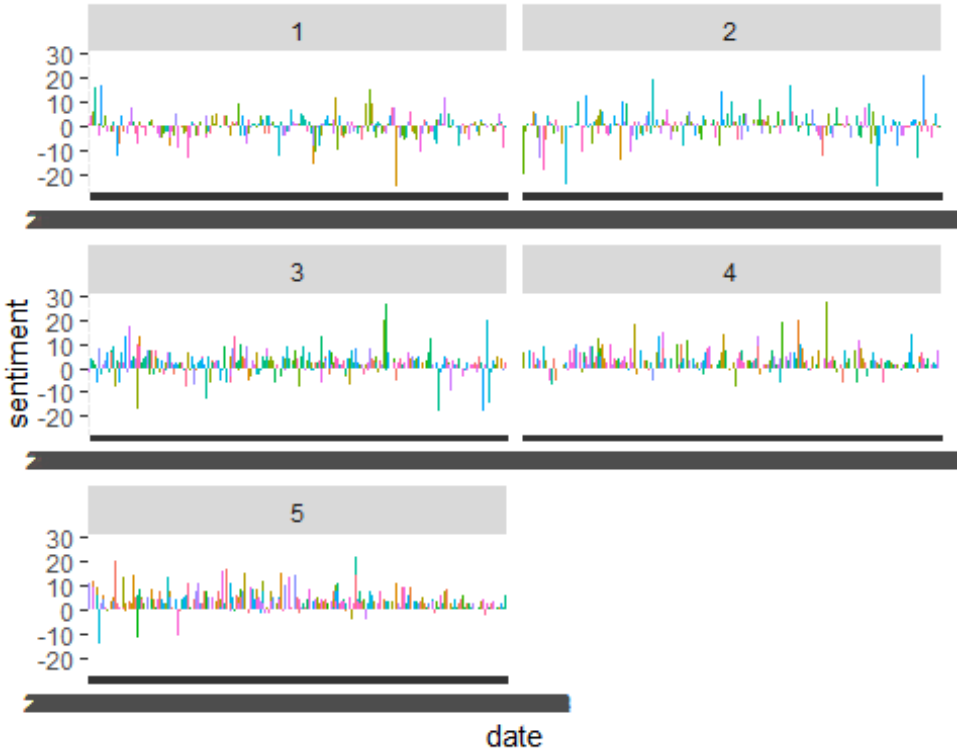
```r
NRCTRUST <- LEXICON_NRC() %>%
FILTER(SENTIMENT == "TRUST")

DATA_TIDY_TOKEN_WORD %>%
INNER_JOIN(NRCTRUST) %>%
COUNT(WORD, SORT = TRUE)
```

```
## JOINING, BY = "WORD"
```

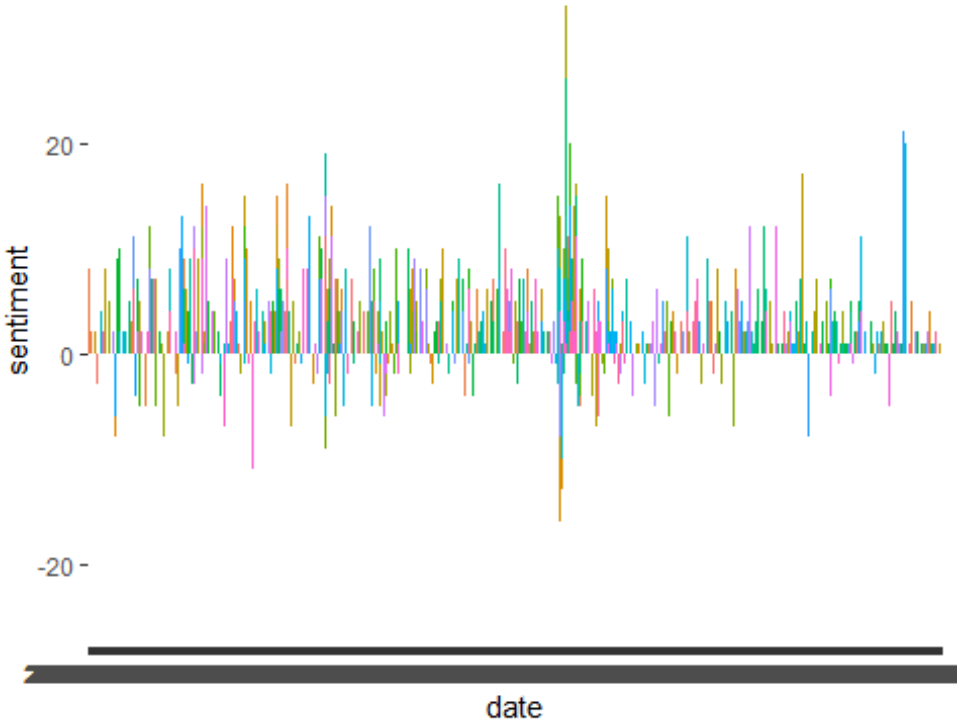
```
## # A TIBBLE: 553 X 2
##    WORD          N
##    <CHR>     <INT>
##  1 AUTHOR      792
##  2 GOOD        696
##  3 FOUND       534
##  4 ROMANCE     424
##  5 RECOMMEND   403
##  6 REAL        395
##  7 SEX         313
##  8 WONDERFUL   266
##  9 PRESENT     263
## 10 FACT        206
## # ... WITH 543 MORE ROWS
```

```r
# 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")
```

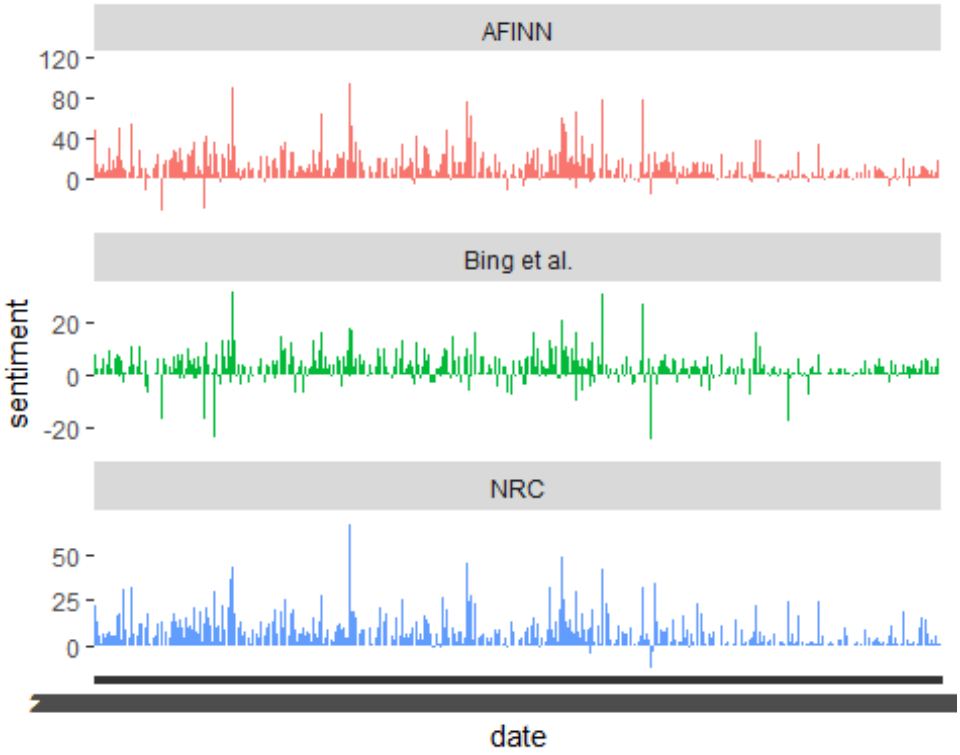



```
GGPLOT(DATA_TIDY_SENTIMENT_BING %>%
  ARRANGE(DATE), AES(DATE, SENTIMENT, FILL = TITLE))+
  GEOM_COL(SHOW.LEGEND = FALSE)
```



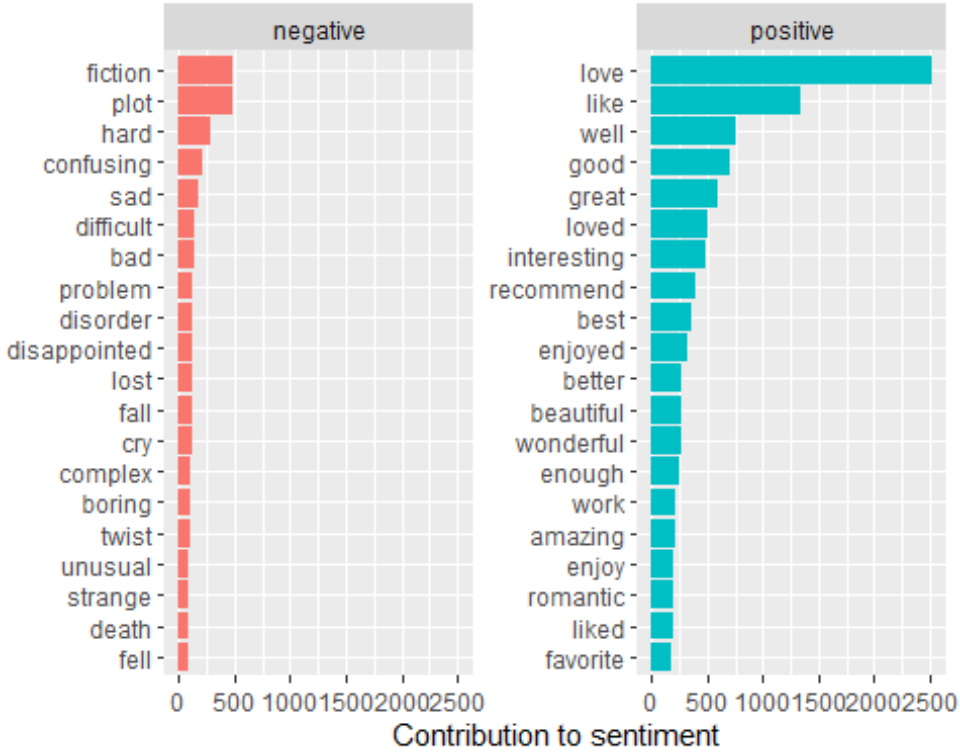

```
# SENTIMENT ANALYSIS BASED ON AFINN LEXICON
DATA_TIDY_SENTIMENT_AFINN <- DATA_TIDY_TOKEN_WORD %>%
  INNER_JOIN(LEXICON_AFINN()) %>%
  GROUP_BY(TITLE, DATE) %>%
  SUMMARISE(SENTIMENT = SUM(VALUE)) %>%
  MUTATE(METHOD = "AFINN")

## 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")
```

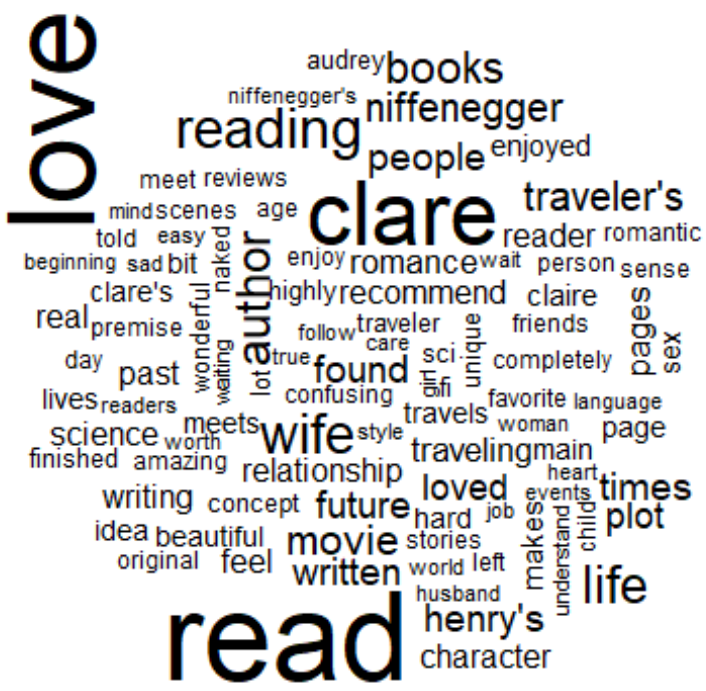



```
# CONTRIBUTION TO SENTIMENT
BING_WORD_COUNTS <- DATA_TIDY_TOKEN_WORD %>%
INNER_JOIN(GET_SENTIMENTS("BING")) %>%
COUNT(WORD, SENTIMENT, SORT = TRUE) %>%
UNGROUP()

## 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
```

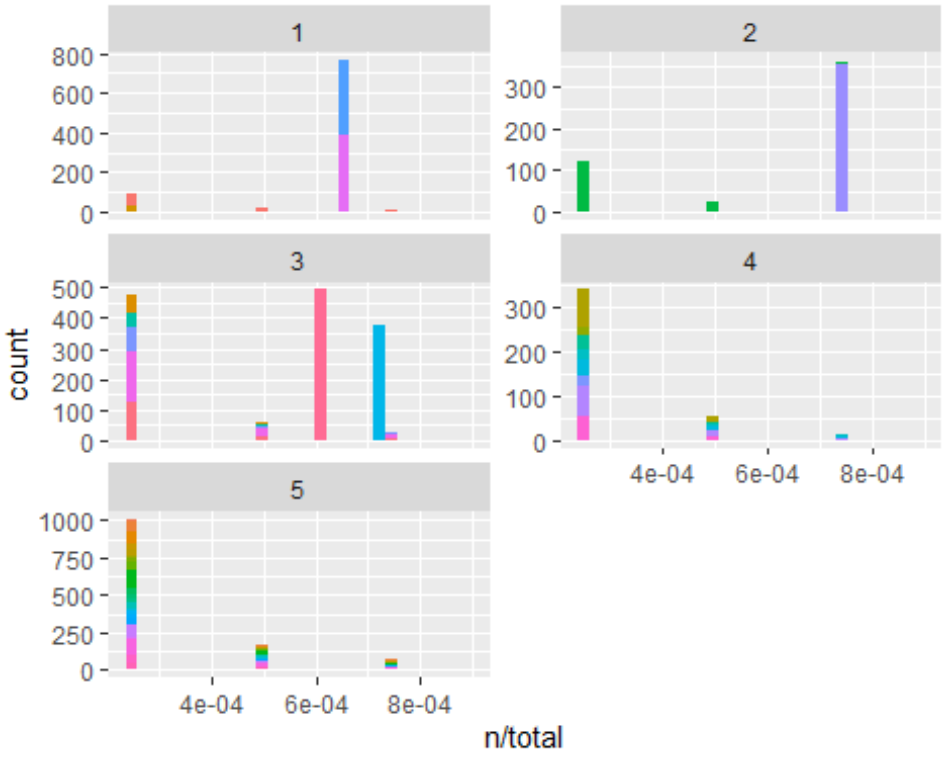



```
custom_stop_words <- bind_rows(data_frame(word = c("fiction","time","travel","book","stor
y"),
lexicon = c("custom")),
stop_words)

## 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.
```

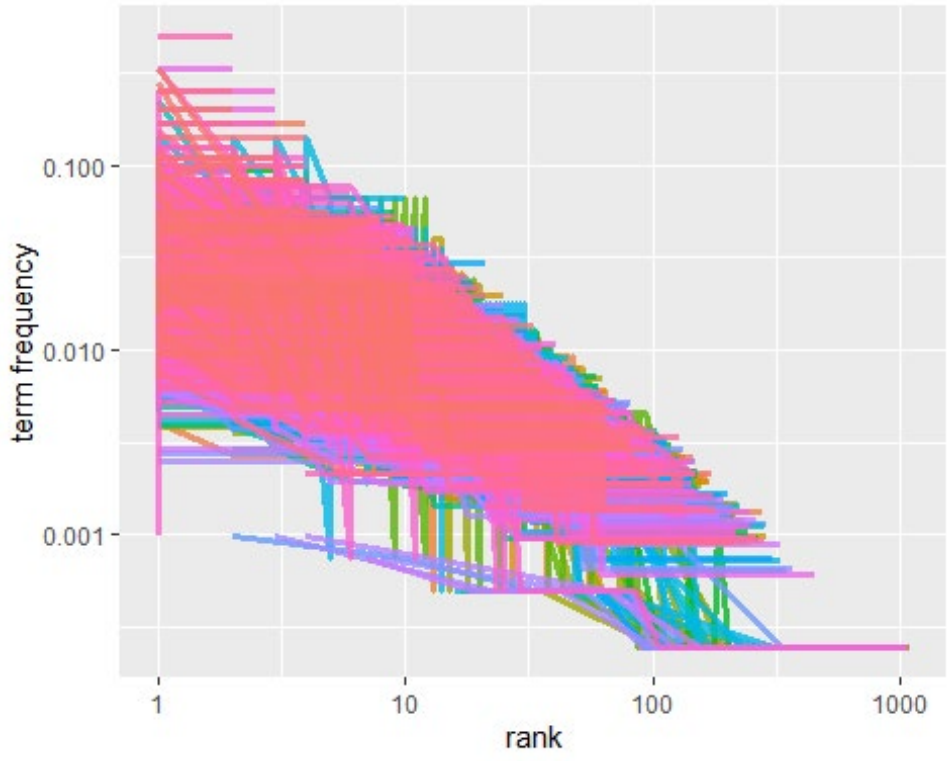



```
DATA_TIDY_TOKEN_WORD %>%
INNER_JOIN(GET_SENTIMENTS("BING")) %>%
COUNT(WORD, SENTIMENT, SORT = TRUE) %>%
ACAST(WORD ~ SENTIMENT, VALUE.VAR = "N", FILL = 0) %>%
COMPARISON.CLOUD(COLORS = C("RED", "BLUE"),
MAX.WORDS = 100)

## 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).
```



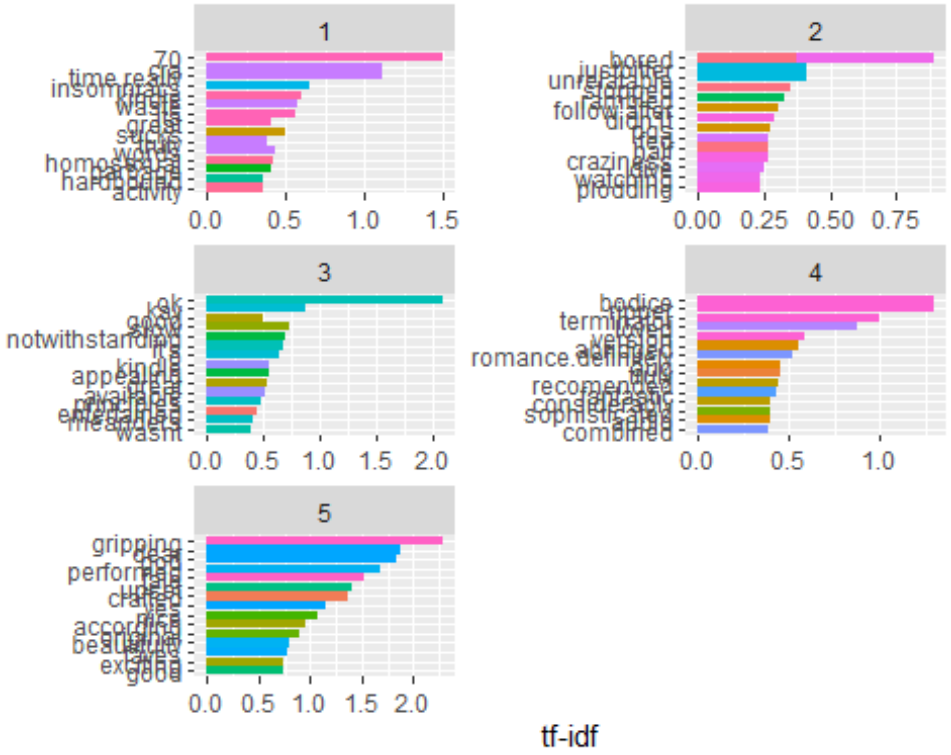

```
# CALCULATING WORD FREQUENCIES
CUSTOM_STOP_WORDS <- BIND_ROWS(DATA_FRAME(WORD = C("TIME","CLARE","HENRY","BOOK","CLAIRE"
,"STORY","PAGE"),
LEXICON = C("CUSTOM")),
STOP_WORDS)

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
```