CSDA 1040: Assignment 1 - Group 4

Jose German, Anjana Pradeep Kumar, Anupama Radhakrishnan Kowsalya, and Xenel Nazar

6/20/2020

1.0 Abstract

Book readers encompass a large industry with ever changing tastes. Finding the next great book that users may like, would not only help the industry but also help satisfy avid readers. Data from the BookCrossing community helped provide the basis of a book recommendation system using the CRISP-DM methodology framework. A Shiny web application was developed providing a usable interface for readers to generate a list of recommended books based on their ratings of previous books they have read.

2.0 Introduction

Books remain an important medium in today's society. In 2018, over 675 million print books were sold in the U.S. and book stores bring in almost 10 million dollars per year (Watson, 2019). While, in Canada, 54 million books; accounting for 1.1 billion dollars in retail value, were sold in 2018 (Ontario Creates, 2020).

Consumption of books remains relevant, as 74% of U.S. consumers have noted that they have consumed at least one book in the past year, and 98% of readers say pleasure is their main reason for reading books (Watson, 2019).

3.0 Background

While the industry remains a fixture in every person's life, the underlying trends the industry is facing is a major concern. From 2014 to 2019, the average industry growth in Canada was -3.6% (IBISWorld, 2019). This may point to various reasons, including but not limited to, a change in consumer tastes or even slow adoption of technology and new mediums by the industry.

A solution the industry might look into, is to gain insight into readers' tastes and see what books they have liked and did not like. Recommending books that boost the "pleasure" consumers gain from reading a book and that aligns with their tastes may help keep consumers reading and help the industry overall.

4.0 Objectives

The objective is to find an effective recommender system to recommend the next book a user should read, based on ratings of past books they have read.

5.0 Data Understanding

5.1 About the Data

The data is from a crawl by Cai-Nicolas Ziegler of the Department of Computer Science from the University of Freiburg on the BookCrossing community for a four week period in August to September 2004 (Ziegler, 2005).

The BookCrossing community releases "books into the wild for a stranger to find, or via controlled release to another BookCrossing member" (BookCrossing, 2020). The data is a compilation of anonymized users with their locations, information about books they have read, as well as their explicit and implicit ratings of those books.

The data is compiled into three distinct csv files on User data ("BX-Users.csv"), Books ("BX-Books.csv"), and Ratings ("BX-Book-Ratings.csv").

Each csv file contains various attributes, such as:

Users:

- User-Id: Represents each anonymized user

- Location: The user's location
- Age: The user's age

Books:

- ISBN: International Standard Book Number or unique identifier for each book (International ISBN Agency, 2014)
- Book-Title: Title of the book
- Book-Author: The book's author
- Year-of-Publication: Year when the book was published
- Publisher: The book's publisher
- Image-URL-S: Small cover image, linked to the Amazon website
- Image-URL-M: Medium cover image, linked to the Amazon website
- Image-URL-L: Large cover image, linked to the Amazon website

Ratings:

- User-Id: Represents each anonymized user
- ISBN: International Standard Book Number or unique identifier for each book (International ISBN Agency, 2014)
- Book-Rating: Either explicit, expressed on a scale from 1-10 (higher values denoting higher appreciation), or implicit, expressed by 0.

5.2 Import Packages

```
knitr::opts_chunk$set(echo=TRUE)
```

```
#Path setup
#path<- "C:/RProjects/bookcrossing"
#setwd(path)
getwd()
```

```
## [1] "/Users/xen/Documents/CSDA 1040/ASSIGNMENT1/CSDA1040Assignment1"
```

```
library(data.table)
library(plotly)
```

```
## Loading required package: ggplot2
```

```
##
## Attaching package: 'plotly'
```

```
## The following object is masked from 'package:ggplot2':
##
##     last_plot
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
## The following object is masked from 'package:graphics':
##
##     layout
```

```
library(ggplot2)
library(RColorBrewer)
library(stringr)
library(plyr)
```

```
##
## Attaching package: 'plyr'
```

```
## The following objects are masked from 'package:plotly':
##
##     arrange, mutate, rename, summarise
```

```
library(tidyr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

```
## The following objects are masked from 'package:data.table':
##
##     between, first, last
```

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

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

```
library(rcompanion)
library(e1071)
library(outliers)
library(recommenderlab)
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

```
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```
## Loading required package: arules
```

```
##
## Attaching package: 'arules'
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
## The following objects are masked from 'package:base':
##
##     abbreviate, write
```

```
## Loading required package: proxy
```

```
##
## Attaching package: 'proxy'
```

```
## The following object is masked from 'package:Matrix':
##
##     as.matrix
```

```
## The following objects are masked from 'package:stats':
##
##     as.dist, dist
```

```
## The following object is masked from 'package:base':
##
##     as.matrix
```

```
## Loading required package: registry
```

```
## Registered S3 methods overwritten by 'registry':
##   method               from
##   print.registry_field proxy
##   print.registry_entry proxy
```

5.3 Import Data

```
# Import CSV Files

# Import Books CSV
books=read.csv("BX-Books.csv", header = TRUE, sep = ";", quote="", stringsAsFactors = FALSE, col.names
= c("ISBN", "Book_Title", "Book_Author", "Year_of_Publication", "Publisher", "Image_S", "Image_M", "Im
age_L"))

# Import Users CSV
users=read.csv("BX-Users.csv", header = TRUE, sep = ";", quote="", stringsAsFactors = FALSE, col.names
= c("UserId","Location","Age"))

# Import Ratings CSV
ratings=read.csv("BX-Book-Ratings.csv", header = TRUE, sep=";", quote="", stringsAsFactors = FALSE, co
l.names = c("UserId","ISBN","Rating"))
```

6.0 Data Exploration and Preparation

6.1 Initial Data cleanup

We first need to remove quotes from the data.

```
# Function to remove quotes from the DataFrames
clean <- colwise(function(ttt) str_replace_all(ttt, '\"', ""))
```

```
# Apply clean function to remove quotes from each DataFrame
books[] <- clean(books)
ratings[] <- clean(ratings)
users[] <- clean(users)
```

We also need to ensure certain columns are numeric.

```
# Convert Certain Columns to Numeric

# Convert UserID and Age in Users DataFrame
users$UserId<-as.numeric(users$UserId)
```

```
## Warning: NAs introduced by coercion
```

```
users$Age<-as.numeric(users$Age)
```

```
## Warning: NAs introduced by coercion
```

```
# Convert Year of Publication in Books DataFrame
books$Year_of_Publication<-as.numeric(books$Year_of_Publication)
```

```
## Warning: NAs introduced by coercion
```

```
# Convert UserId and Rating in Ratings DataFrame
ratings$UserId<-as.numeric(ratings$UserId)
ratings$Rating<-as.numeric(ratings$Rating)
```

Note: We are not converting the ISBN columns as numeric, as they may contain the Roman Numeral “X” instead of the number 10 (R.R. Bowker LLC., 2014).

We are also removing image links from the books DataFrame, as it will not be pertinent for later analysis. This may be included later during deployment or future iterations of the recommender system.

```
# Remove Columns "Image_S", "Image_M", "Image_L" from Books DataFrame
books<-books[-c(6,7,8)]
```

6.2 Verify Data Types

```
# Data Types under Books DataFrame
str (books)
```

```
## 'data.frame':    292746 obs. of  5 variables:
##  $ ISBN               : chr  "0195153448" "0002005018" "0060973129" "0374157065" ...
##  $ Book_Title         : chr  "Classical Mythology" "Clara Callan" "Decision in Normandy" "Flu: The
Story of the Great Influenza Pandemic of 1918 and the Search for the Virus That Caused It" ...
##  $ Book_Author        : chr  "Mark P. O. Morford" "Richard Bruce Wright" "Carlo D'Este" "Gina Bari
Kolata" ...
##  $ Year_of_Publication: num  2002 2001 1991 1999 1999 ...
##  $ Publisher          : chr  "Oxford University Press" "HarperFlamingo Canada" "HarperPerennial" "F
arrar Straus Giroux" ...
```

```
# Data Types for Users DataFrame
str (users)
```

```
## 'data.frame':    279071 obs. of  3 variables:
##  $ UserId  : num  1 2 3 4 5 6 7 8 9 10 ...
##  $ Location: chr  "nyc, new york, usa" "stockton, california, usa" "moscow, yukon territory, russi
a" "porto, v.n.gaia, portugal" ...
##  $ Age     : num  NA 18 NA 17 NA 61 NA NA NA 26 ...
```

```
# Data Types for Ratings DataFrame
str(ratings)
```

```
## 'data.frame':    1149780 obs. of  3 variables:
##  $ UserId: num  276725 276726 276727 276729 276729 ...
##  $ ISBN  : chr  "034545104X" "0155061224" "0446520802" "052165615X" ...
##  $ Rating: num  0 5 0 3 6 0 8 6 7 10 ...
```

6.3 Summary of the Variables

```
# Details of Books DataFrame
summary (books)
```

```
##      ISBN            Book_Title        Book_Author        Year_of_Publication
##  Length:292746      Length:292746      Length:292746      Min.   :   0
##  Class :character   Class :character   Class :character   1st Qu.:1989
##  Mode  :character   Mode  :character   Mode  :character   Median :1995
##                                                           Mean   :1959
##                                                           3rd Qu.:2000
##                                                           Max.   :2050
##                                                           NA's   :27423
##   Publisher
##  Length:292746
##  Class :character
##  Mode  :character
##
##
##
##
```

```
# Details of Users DataFrame
summary (users)
```

```
##      UserId         Location              Age
##  Min.   :     1   Length:279071      Min.   :  0.00
##  1st Qu.: 69673   Class :character   1st Qu.: 24.00
##  Median :139402   Mode  :character   Median : 32.00
##  Mean   :139402                      Mean   : 34.76
##  3rd Qu.:209130                      3rd Qu.: 44.00
##  Max.   :278858                      Max.   :244.00
##  NA's   :157                         NA's   :111075
```

```
# Details of Ratings DataFrame
summary (ratings)
```

```
##      UserId           ISBN               Rating
##  Min.   :     2   Length:1149780     Min.   : 0.000
##  1st Qu.: 70345   Class :character   1st Qu.: 0.000
##  Median :141010   Mode  :character   Median : 0.000
##  Mean   :140386                      Mean   : 2.867
##  3rd Qu.:211028                      3rd Qu.: 7.000
##  Max.   :278854                      Max.   :10.000
```

6.4 Replace NULL Values with NA

```
# Replace NULL values with NA in all DataFrames
users_df <-users %>% replace(.=="NULL", NA)
books_df <-books %>% replace(.=="NULL", NA)
ratings_df<-ratings %>% replace(.=="NULL", NA)
```

6.5 Remove rows with NA

```
# Remove rows with NA in all DataFrames
users_df<-users_df %>% drop_na(UserId)
users_df<-users_df %>% drop_na(Location)
books_df<-na.omit(books_df)
```

6.6 Filter out rows with web addresses under ISBN

There was likely an error during the initial compilation of the web crawl, resulting in some links showing under ISBN instances. These web addresses need to be removed from our data to improve our analysis.

```
# Filter out rows containing web addresses under ISBN (e.g. "http")
books_df<-books_df[!grepl("http", books_df$ISBN),]
```

6.7 Further cleanup

We will focus on readers between the ages of 11 and 99, and we will need need to replace NA values with the mean age of 34.

```
# Replace ages less than 10 and greater than 100
users_df$Age[users_df$Age > 100] <- NA
users_df$Age[users_df$Age < 10] <- NA
```

```
# Replace missing age with mean
ageMean = trunc(mean(users_df$Age, na.rm = TRUE))
users_df$Age[is.na(users_df$Age)] <- ageMean
```

We will also need to replace Years of Publication values that show as NAs or zeroes with the mean year of publication, which is 1959.

```
# Replace Years of Publication showing as 0 or NA, with the mean value
yearMean = trunc(mean(books_df$Year_of_Publication, na.rm = TRUE))
books_df$Year_of_Publication[books_df$Year_of_Publication ==0] <- NA
books_df$Year_of_Publication[is.na(books_df$Year_of_Publication)] <- yearMean
```

```
# Save into a RDS file
 saveRDS(books_df, file="booksDf.RDS")
```

6.8 Merge DataFrames

We will now have to combine the DataFrames into a single DataFrame for proper analysis.

```
# Merge Ratings and Books DataFrames to "ratings_books" DataFrame
ratings_books<-left_join(ratings_df,books_df,by="ISBN")
```

```
# Merge ratings_books and Users DataFrames to "ratings_books_users" DataFrame
ratings_books_users<-left_join(ratings_books,users_df,by="UserId")
```

6.9 Remove NAs from DataFrame

```
# Remove NAs from DataFrame
ratings_books_users<-na.omit(ratings_books_users)
```

6.10 Generate Book List

```
#Generate Book List for Word Cloud for Shiny application
cloud<-ratings_books_users%>%
    group_by(Book_Title)%>%
    dplyr::summarize(count_Book_Title=n())%>%
    filter(count_Book_Title>500)%>%
    select(Book_Title,count_Book_Title)%>%
    head()
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
# Save books list into a RDS file
saveRDS(cloud, file = "wordCloud.Rds")
```

6.11 Subset DataFrame to Explicit and Implicit Ratings

The ratings are actually categorized into two groups, explicit and implicit ratings. Explicit ratings are listed from 1-10, with higher values reflecting the consumers' appreciation of the book, while implicit ratings are listed as 0s (Ziegler, 2005). We will need to subset the data to only look at the explicit ratings.

```
# Subset DataFrame to Explicit and Implicit Ratings
ratings_explicit<-subset(ratings_books_users,ratings_books_users$Rating>0)
ratings_implicit<-subset(ratings_books_users,ratings_books_users$Rating==0)
```

6.12 Simple Analysis

We can now gain some insight in the data and determine some of the most popular books and top rated books overall. This can be used to get a sense of what the overall reader base of "BookCrossing" tend to prefer.

We can identify the most popular books.

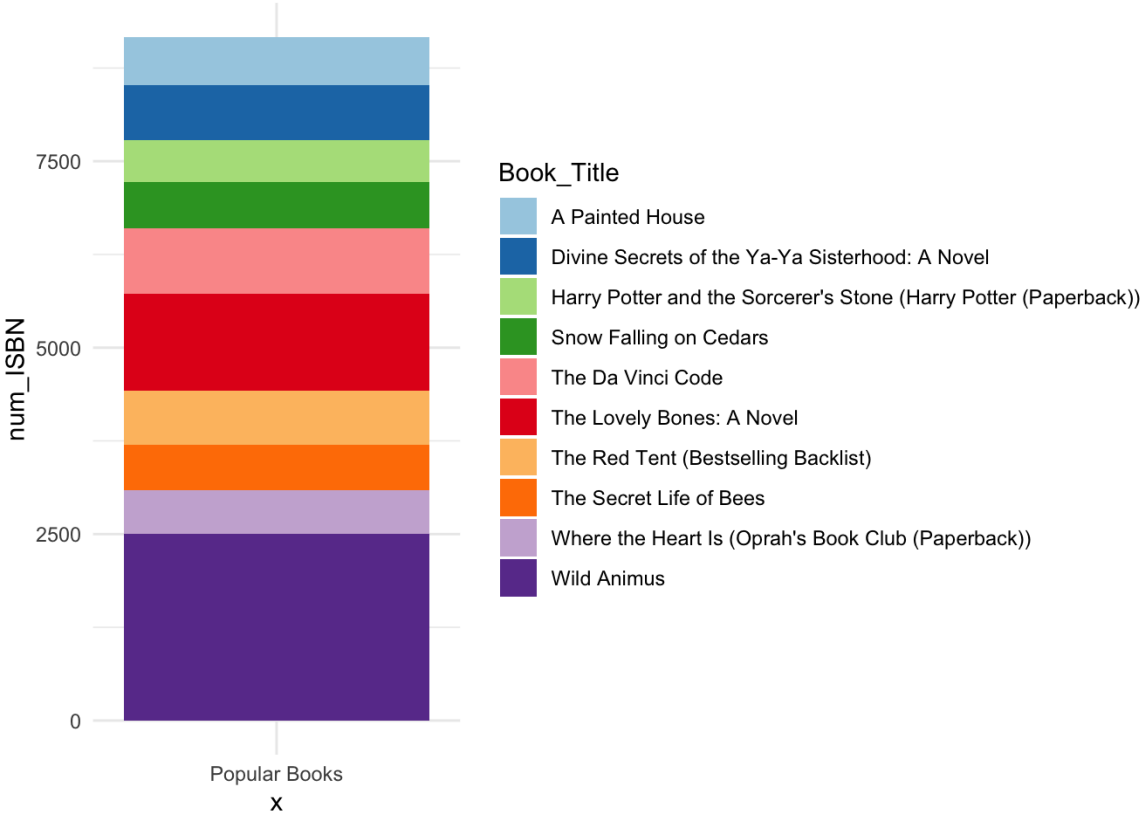
```
# Popular books
most_popular_books<-ratings_books_users%>%
  group_by(ISBN,Book_Title,Book_Author,Publisher)%>%
  summarize(num_ISBN=n())%>%
  filter(num_ISBN>200)%>%
  arrange(-num_ISBN)%>%
  select(ISBN,Book_Title,Book_Author,Publisher,num_ISBN)%>%
  head(10)
```

```
## `summarise()` regrouping output by 'ISBN', 'Book_Title', 'Book_Author' (override with `.groups` arg
ument)
```

```
print(most_popular_books)
```

```
## # A tibble: 10 x 5
## # Groups:   ISBN, Book_Title, Book_Author [10]
##    ISBN    Book_Title                       Book_Author  Publisher      num_ISBN
##    <chr>   <chr>                            <chr>        <chr>             <int>
##  1 097188… Wild Animus                      Rich Shapero Too Far            2502
##  2 031666… The Lovely Bones: A Novel        Alice Sebold Little, Brown      1295
##  3 038550… The Da Vinci Code                Dan Brown    Doubleday           883
##  4 006092… Divine Secrets of the Ya-Ya Sis… Rebecca Wel… Perennial           732
##  5 031219… The Red Tent (Bestselling Backl… Anita Diama… Picador USA         723
##  6 044023… A Painted House                  John Grisham Dell Publishi…      647
##  7 014200… The Secret Life of Bees          Sue Monk Ki… Penguin Books       615
##  8 067976… Snow Falling on Cedars           David Guter… Vintage Books…      614
##  9 044667… Where the Heart Is (Oprah's Boo… Billie Letts Warner Books        585
## 10 059035… Harry Potter and the Sorcerer's… J. K. Rowli… Arthur A. Lev…      571
```

```
# Plot Popular Books
bp<- ggplot(most_popular_books, aes(x="Popular Books", y=num_ISBN, fill=Book_Title))+
  geom_bar(width = 1, stat = "identity")

bp + scale_fill_brewer(palette="Paired")+theme_minimal()
```



We can identify the top rated books as well.

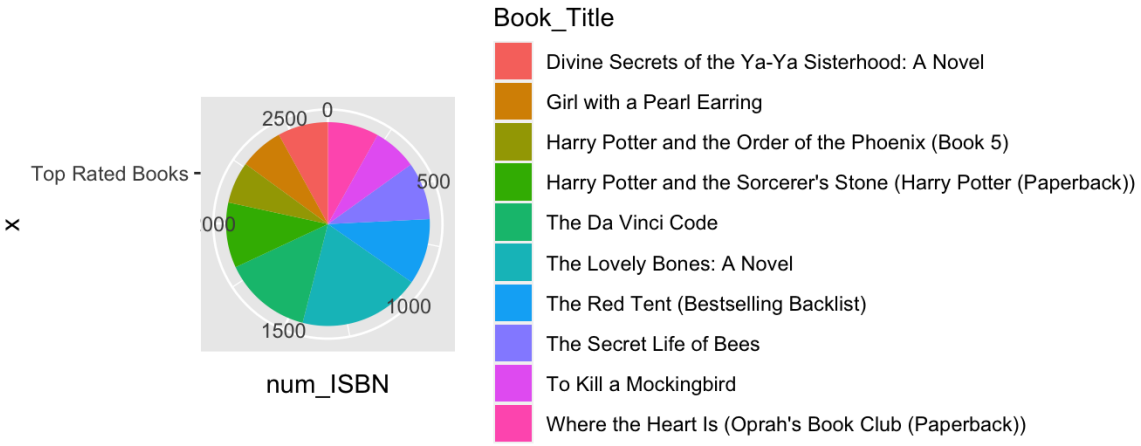
```
# Top Rated books
top_rated_books<-ratings_explicit%>%
  group_by(ISBN,Book_Title,Book_Author,Publisher)%>%
  filter(Rating>7)%>%
  summarize(num_ISBN=n())%>%
  filter(num_ISBN>50)%>%
  arrange(-num_ISBN)%>%
  select(ISBN,Book_Title,Book_Author,Publisher,num_ISBN)%>%
  head(10)
```

```
## `summarise()` regrouping output by 'ISBN', 'Book_Title', 'Book_Author' (override with `.groups` arg
ument)
```

```
print(top_rated_books)
```

```
## # A tibble: 10 x 5
## # Groups:   ISBN, Book_Title, Book_Author [10]
##    ISBN     Book_Title                          Book_Author  Publisher      num_ISBN
##    <chr>    <chr>                               <chr>        <chr>             <int>
##  1 031666… The Lovely Bones: A Novel           Alice Sebold Little, Brown       515
##  2 038550… The Da Vinci Code                   Dan Brown    Doubleday           374
##  3 031219… The Red Tent (Bestselling Backl…    Anita Diama… Picador USA         278
##  4 059035… Harry Potter and the Sorcerer's…    J. K. Rowli… Arthur A. Lev…      276
##  5 014200… The Secret Life of Bees             Sue Monk Ki… Penguin Books       243
##  6 044667… Where the Heart Is (Oprah's Boo…    Billie Letts Warner Books        216
##  7 006092… Divine Secrets of the Ya-Ya Sis…    Rebecca Wel… Perennial           213
##  8 044631… To Kill a Mockingbird               Harper Lee   Little Brown …      186
##  9 045228… Girl with a Pearl Earring           Tracy Cheva… Plume Books         186
## 10 043935… Harry Potter and the Order of t…    J. K. Rowli… Scholastic          176
```

```
# Plot Top Rated books
pie<- ggplot(top_rated_books, aes(x="Top Rated Books", y=num_ISBN, fill=Book_Title))+
  geom_bar(width = 1, stat = "identity")

pie + coord_polar("y", start=0)
```



6.13 Generate Lists

```
# Save most read books in a list
poplist<-most_popular_books%>%
  select(ISBN,Book_Title,Book_Author,Publisher)
```

```
# Save top rated books in a list
topratelist<-top_rated_books%>%
  select(ISBN,Book_Title,Book_Author,Publisher)
```

6.14 Save lists as RDS files for Shiny Deployment

```
# Save list of most popular books into a RDS file
saveRDS(poplist, file = "poplist.Rds")

# Save list of top rated books into a RDS file
saveRDS(topratelist, file="topratedlist.Rds")
```

6.15 Evaluate Rating Distribution

```
# Ratings Details
summary(ratings_explicit$Rating)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   7.000   8.000   7.625   9.000  10.000
```

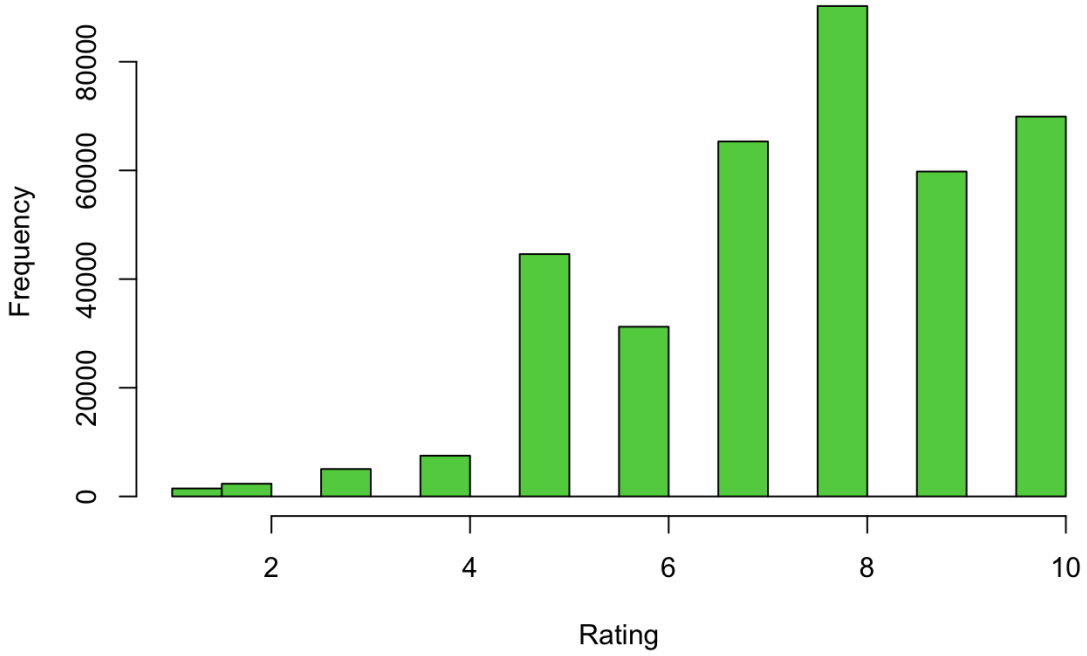
```
range(ratings_explicit$Rating)
```

```
## [1]  1 10
```

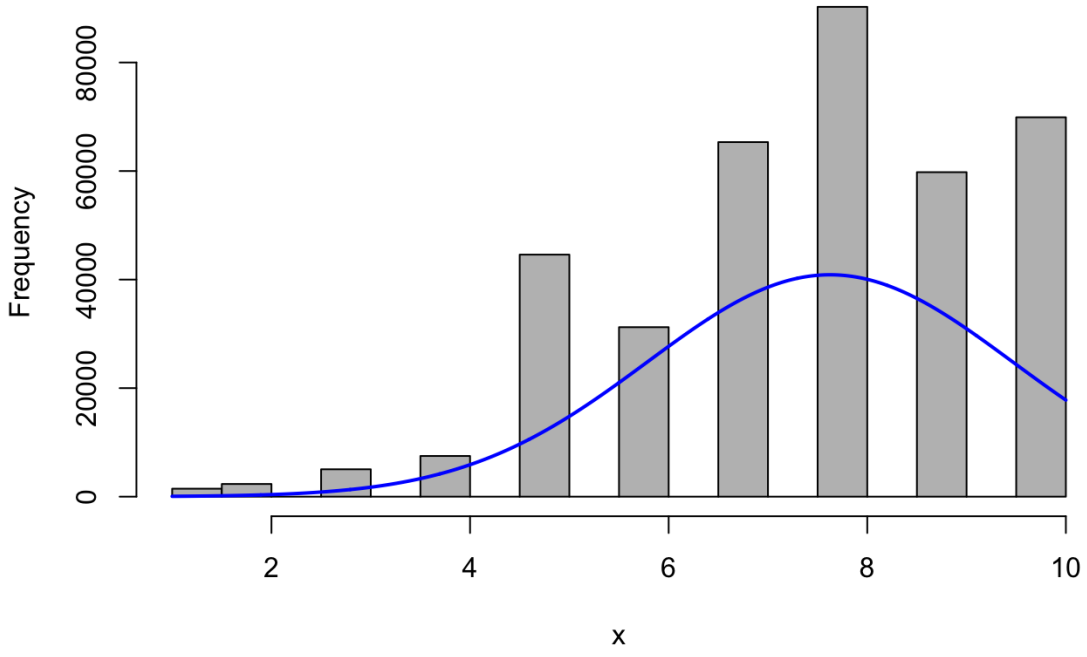
```
print(skewness(ratings_explicit$Rating))# Left skewed by -0.6613279
```

```
## [1] -0.6613279
```

```
# Rating Distribution
hist(ratings_explicit$Rating,
     breaks = 'Sturges',
     xlab = 'Rating',
     main = '',
     col = 3)
```




```
# Rating Distribution
rating_dist = ratings_explicit$Rating
plotNormalHistogram(rating_dist)
```



The ratings are distributed from the range 1-10, and the ratings are left skewed by -0.6613279.

6.16 Subset Ratings

Based on the ratings distribution, we can lessen how skewed the data is. We can subset the data by focusing on ratings greater than 3.

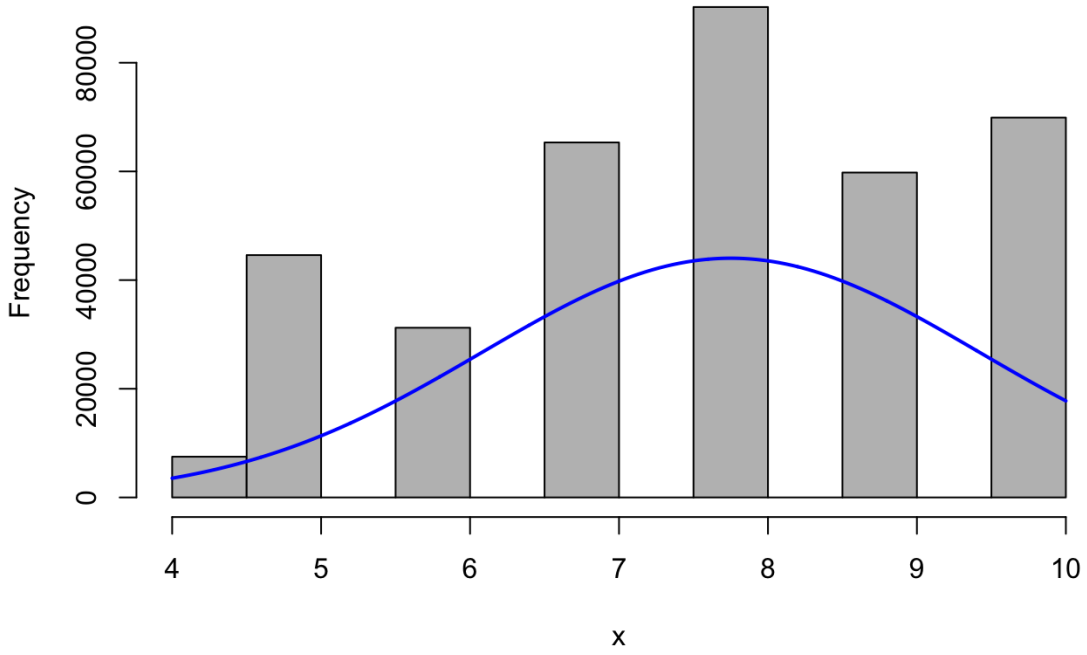
```
outlier(ratings_explicit$Rating)
```

```
## [1] 1
```

```
# Subset data to only ratings greater than 3
ratings_norm<-subset(ratings_explicit,ratings_explicit$Rating>3)
print(skewness(ratings_norm$Rating)) # skewness after -0.3522745
```

```
## [1] -0.3522745
```

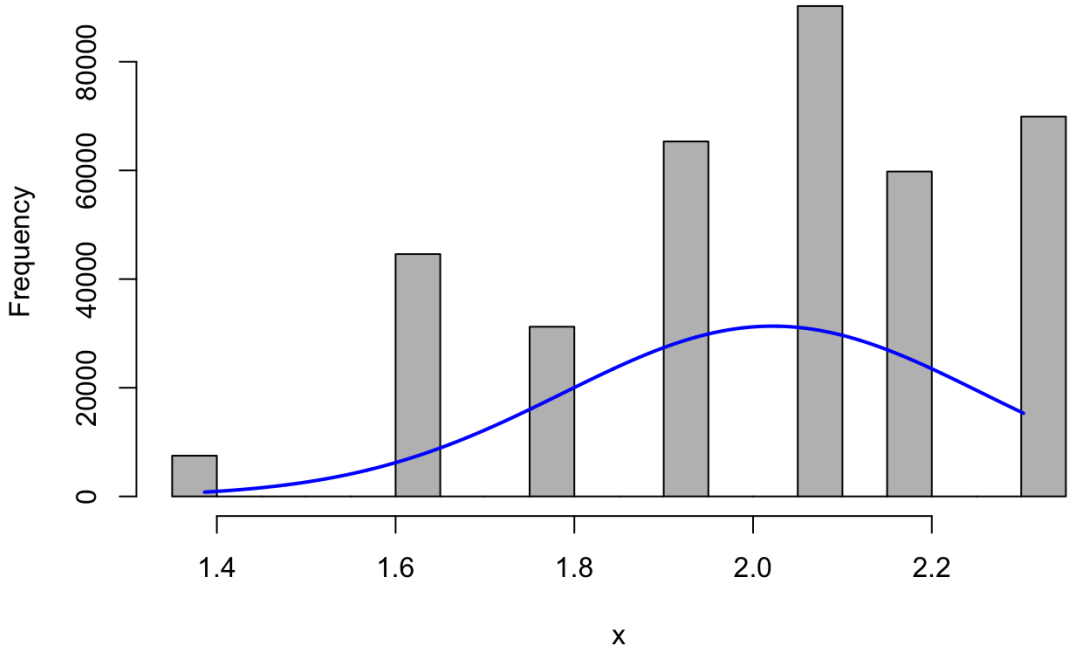
```
# Visualize Rating Distribution
rating_subset_dist = ratings_norm$Rating
plotNormalHistogram(rating_subset_dist)
```



Subsetting the data has lessened the skewness of the data and is now left skewed to -0.3522745 compared to -0.6613279 before.

We can also visualize the effect if the ratings data is normalized.

```
# Normalize ratings data and plot
rating_log_norm = log(ratings_norm$Rating)
plotNormalHistogram(rating_log_norm)
```



7.0 Modeling

7.1 Subset DataFrame

For purposes of modeling, the data needs to be non-normalized.

We need narrow down the DataFrame, we will first subset the DataFrame to users who have rated 50 or more books.

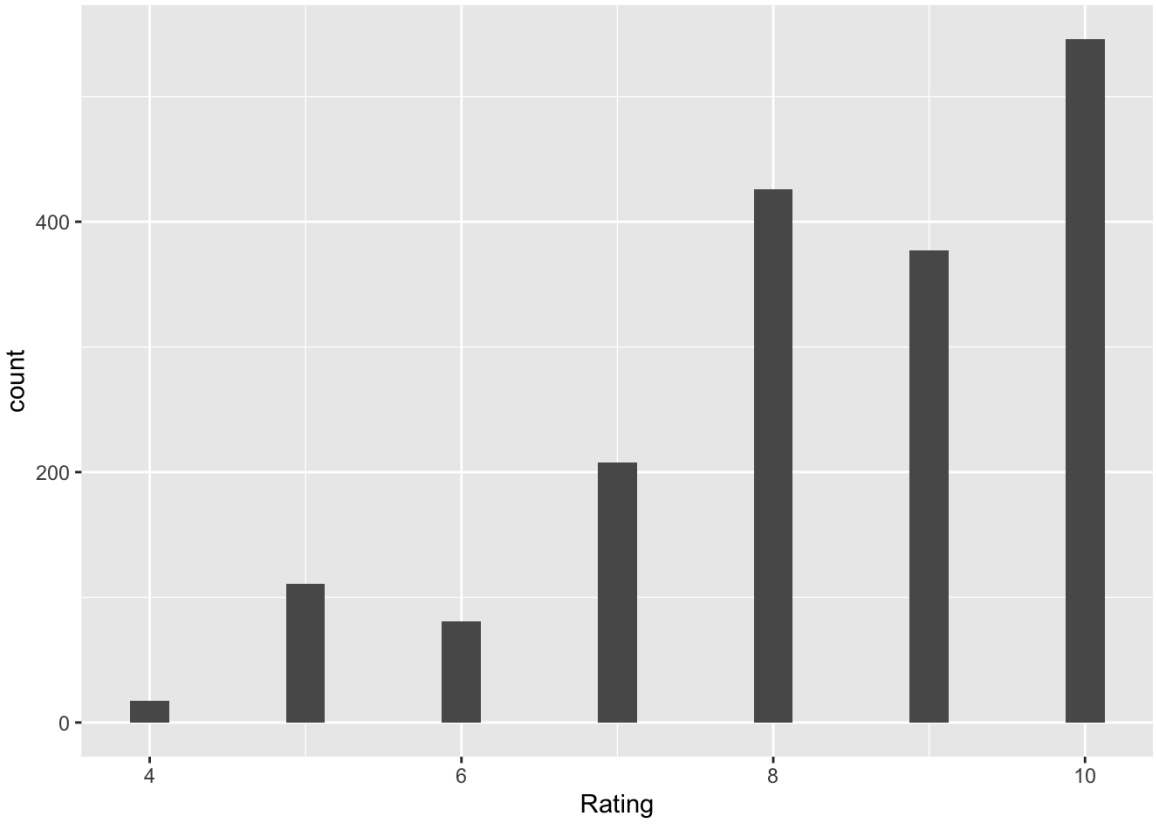
```
# Subset data to users who have rated 50 or more books
rating_matrix_long<-ratings_norm%>%
  add_count(UserId)%>%
  dplyr::filter(n>50)%>%
  select(ISBN,UserId,Rating)%>%
  arrange(UserId)
```

We then further narrow the data down to books that have been rated 50 or more times.

```
# Narrow down to only books that were rated 50 or more times
rating_matrix_long<-rating_matrix_long%>%
  add_count(ISBN)%>%
  dplyr::filter(n>50)%>%
  select(ISBN,UserId,Rating)%>%
  arrange(ISBN)
```

Plot Rating Distribution and User count for the subset

```
#Rating Distribution and user count for the subset
ggplot(rating_matrix_long, aes(Rating)) + geom_histogram(binwidth = 0.25)
```



7.2 Extract Book List

We also need to extract the book list to use for the Shiny web application.

```
rating_matrix_long2<-ratings_norm%>%
  add_count(UserId)%>%
  dplyr::filter(n>50)%>%
  select(ISBN,UserId,Rating,Book_Title)%>%
  arrange(UserId)

rating_matrix_long2<-rating_matrix_long2%>%
  add_count(ISBN)%>%
  dplyr::filter(n>50)%>%
  select(ISBN,UserId,Rating,Book_Title)%>%
  arrange(UserId)

# Extract book list
 booklist<-rating_matrix_long2%>%
   select(ISBN,Book_Title)%>%
   arrange(Book_Title)%>%
   unique()
```

```
# Save list of books into a RDS file
 saveRDS(booklist, file="bookList.RDS")
```

This list will be what the user sees in making their selection

7.3 Pivot Matrix

```
# User-Book Rating matrix
rating_matrix<-rating_matrix_long%>%pivot_wider(names_from = ISBN,values_from=Rating)
```

7.4 Remove Columns

```
# Removing UserId and Book_Title from Rating Matrix
rating_matrix_final = as.matrix(rating_matrix[,-1])
```

```
# Save into a RDS file
 saveRDS(rating_matrix_final, file="ratingMatrix.RDS")
```

7.5 Convert into a Real Matrix

```
# Real matrix
object.size(rating_matrix)
```

```
## 157760 bytes
```

```
# Convert Rating Matrix into a recommenderlab Sparse Matrix
rating_matrix = as(rating_matrix_final, "realRatingMatrix")
object.size(rating_matrix)
```

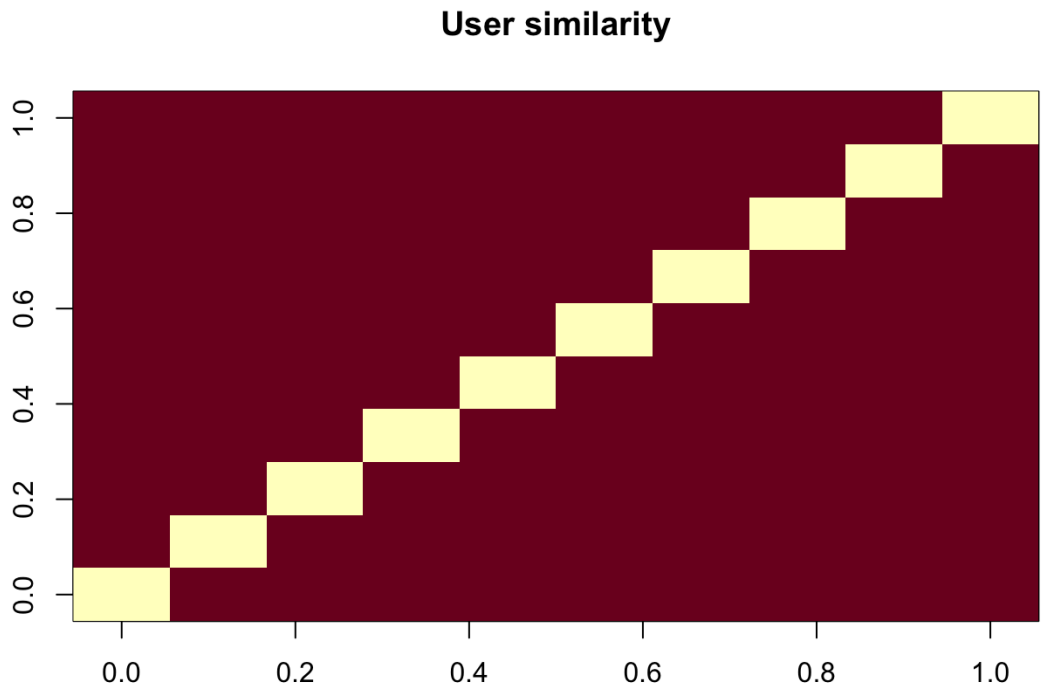
```
## 25624 bytes
```

7.6 Check Similarities

```
# Similarity Between Users
similarity_users <- similarity(rating_matrix[1:10, ],
                               method = "jaccard",
                               which = "UserId")
as.matrix(similarity_users)
```

```
##    1 2 3 4 5 6 7 8 9 10
## 1  0 1 1 1 1 1 1 1 1  1
## 2  1 0 1 1 1 1 1 1 1  1
## 3  1 1 0 1 1 1 1 1 1  1
## 4  1 1 1 0 1 1 1 1 1  1
## 5  1 1 1 1 0 1 1 1 1  1
## 6  1 1 1 1 1 0 1 1 1  1
## 7  1 1 1 1 1 1 0 1 1  1
## 8  1 1 1 1 1 1 1 0 1  1
## 9  1 1 1 1 1 1 1 1 0  1
## 10 1 1 1 1 1 1 1 1 1  0
```

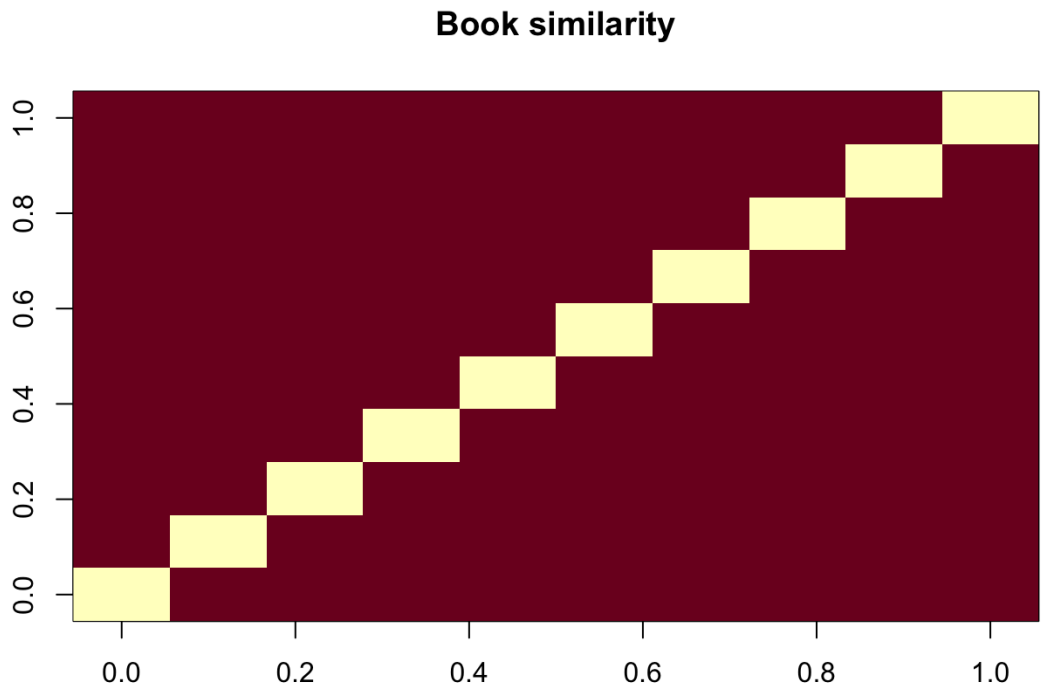
```
image(as.matrix(similarity_users), main = "User similarity")
```



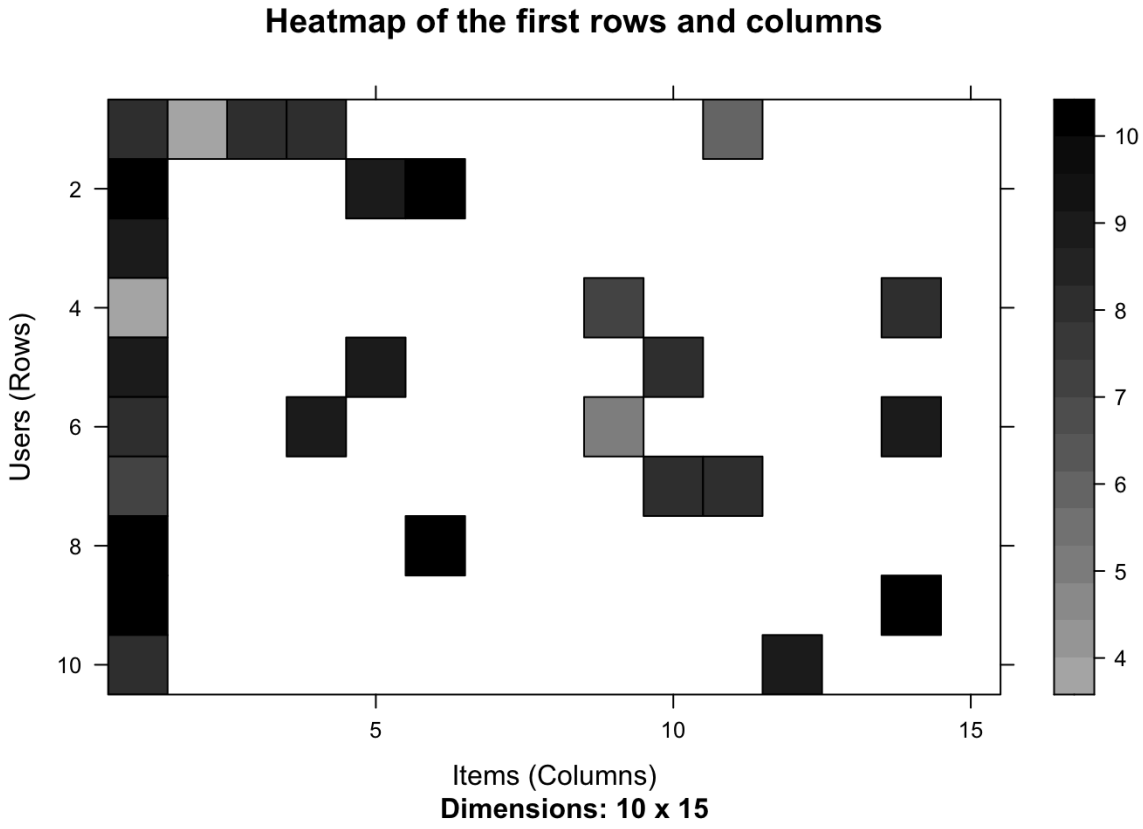

```
# Similarity between Books

similarity_books <- similarity(rating_matrix[, 1:10], method =
                                "cosine", which = "items")
as.matrix(similarity_books)
```

```
##            0060928336 014028009X 0142001740 0312195516 0316601950 0316666343
## 0060928336  0.0000000  0.9242021  0.9838561  0.9760232  0.9925833  0.9857949
## 014028009X  0.9242021  0.0000000  0.9612893  0.9387433  0.9838492  0.9625365
## 0142001740  0.9838561  0.9612893  0.0000000  0.9641161  0.9838371  0.9868710
## 0312195516  0.9760232  0.9387433  0.9641161  0.0000000  0.9654146  0.9665212
## 0316601950  0.9925833  0.9838492  0.9838371  0.9654146  0.0000000  0.9901919
## 0316666343  0.9857949  0.9625365  0.9868710  0.9665212  0.9901919  0.0000000
## 0345313860  0.9824294  0.9902913  0.9984557  0.9883222  0.9838520  0.9838067
## 0345337662  0.9505772  0.9924087  0.9929577  0.9541136  1.0000000  0.9883568
## 0345370775  0.9688835  0.9858296  0.9585288  0.9737634  0.9952211  0.9792926
## 0375727345  0.9740929  0.9504698  0.9660979  0.9839074  0.9706755  0.9640018
##            0345313860 0345337662 0345370775 0375727345
## 0060928336  0.9824294  0.9505772  0.9688835  0.9740929
## 014028009X  0.9902913  0.9924087  0.9858296  0.9504698
## 0142001740  0.9984557  0.9929577  0.9585288  0.9660979
## 0312195516  0.9883222  0.9541136  0.9737634  0.9839074
## 0316601950  0.9838520  1.0000000  0.9952211  0.9706755
## 0316666343  0.9838067  0.9883568  0.9792926  0.9640018
## 0345313860  0.0000000  0.9885441  0.9523786  0.9990779
## 0345337662  0.9885441  0.0000000  0.9450775  0.9970593
## 0345370775  0.9523786  0.9450775  0.0000000  0.9666512
## 0375727345  0.9990779  0.9970593  0.9666512  0.0000000
```

```
image(as.matrix(similarity_books), main = "Book similarity")
```




```
#Heat Map of first 10 rows and 15 columns
image(rating_matrix[1:10, 1:15], main = "Heatmap of the first rows and columns")
```



7.7 Generate Recommender Model

```
# Recommender Model
ratingmodel = Recommender(rating_matrix, method = "UBCF", param=list(method="Cosine",nn=10))
```

```
# Save into a RDS file
 saveRDS(ratingmodel, file="ratingModel.RDS")
```

7.8 Evaluation

```
# Evaluation Scheme
evaluation_scheme <- evaluationScheme(rating_matrix,
                                      method="cross-validation",
                                      k=5, given=1,
                                      goodRating=3) #k=5 meaning a 5-fold cross validation. given=3
 meaning a Given-3 protocol

# Evaluation Results
  evaluation_results <- evaluate(evaluation_scheme,
                                 method="UBCF",
                                 n=c(1,3,5,10,15,20))
```

```
## UBCF run fold/sample [model time/prediction time]
##   1  [0.023sec/0.081sec]
##   2  [0.001sec/0.217sec]
##   3  [0.001sec/0.059sec]
##   4  [0.001sec/0.074sec]
##   5  [0.002sec/0.091sec]
```

```
  eval_results <- getConfusionMatrix(evaluation_results)[[1]]
  print(eval_results)
```

```
##            TP         FP        FN       TN  precision     recall        TPR
## 1  0.06569343  0.6058394 1.6496350 23.67883 0.09782609 0.02891156 0.02891156
## 3  0.14598540  1.8686131 1.5693431 22.41606 0.07246377 0.06164966 0.06164966
## 5  0.25547445  3.1021898 1.4598540 21.18248 0.07608696 0.12164588 0.12164588
## 10 0.51094891  6.2043796 1.2043796 18.08029 0.07608696 0.25345805 0.25345805
## 15 0.81751825  9.2554745 0.8978102 15.02920 0.08115942 0.39158163 0.39158163
## 20 1.08759124 12.3430657 0.6277372 11.94161 0.08097826 0.51638322 0.51638322
##           FPR
## 1  0.02532126
## 3  0.07812891
## 5  0.12977218
## 10 0.25976699
## 15 0.38692179
## 20 0.51571078
```

7.9 Check Predictions

```
# Predictions
# Obtain top 5 recommendations for 2nd user in dataset
Top_5_pred = predict(ratingmodel, rating_matrix[2], n=5)
Top_5_List = as(Top_5_pred, "list")
```

```
# Recommended predictions for a user based on previous ratings
recomdf=data.frame(Top_5_List)
colnames(recomdf)="ISBN"
bookrecomdf<-left_join(recomdf,books_df,by="ISBN")%>%
  select(Book_Title)%>%
  print()
```

```
##                                         Book_Title
## 1                The Red Tent (Bestselling Backlist)
## 2                                    The Da Vinci Code
## 3   Harry Potter and the Chamber of Secrets (Book 2)
## 4 Where the Heart Is (Oprah's Book Club (Paperback))
## 5                                      Jurassic Park
```

8.0 Conclusions and Recommendations

The current model is constrained with the limited size of the initial data crawl. Expanding the time period to a year or more would capture even more ratings of books from more users.

The increased time period could help us segment the data by season, and provide some insight if certain books are more popular in a season; helping improve book recommendations based on the time period the user inputs their ratings. This would also help book publishers prepare for certain seasons by increasing or decreasing marketing expenses when a new season starts (e.g. Holiday Season).

Information from other mediums, such as Audiobooks and eBooks, would also provide more information on users' appreciation of books. As well as provide a sense if purchases of a certain medium is increasing / declining, helping book publishers allocate funds accordingly.

Collection of other data such as genre, language, book size, or number of pages would benefit the model and help users narrow down what readers may be looking for.

Partnering with certain book stores, publishers, and online retailers would provide additional data, including links to purchase, pictures of the books, and other metadata.

Overall data collection would also need to be reevaluated after a set number of years, as consumer tastes change.

9.0 Deployment

The underlying code of this markdown report, can be found on Github (https://github.com/xnazar/CSDA1040Assignment1) and on the Shiny web application About page as listed on shinyapps.io (https://jose-g.shinyapps.io/BookRecommender/)

The Shiny web application provides a working user interface for readers to provide their ratings on three different books, and will then generate a list of reccomendations based on their ratings. It also lists the current most popular books, and the top rated books.

10.0 Bibliography

BookCrossing. (2020). About BookCrossing. Retrieved June 10, 2020, from https://www.bookcrossing.com/about (https://www.bookcrossing.com/about)

IBISWorld. (2019). Industry Market Research, Reports, and Statistics. Retrieved June 15, 2020, from https://www.ibisworld.com/canada/market-research-reports/book-publishing-industry/ (https://www.ibisworld.com/canada/market-research-reports/book-publishing-industry/)

International ISBN Agency. (2014). International ISBN Agency. Retrieved June 15, 2020, from https://www.isbn-international.org/content/what-isbn (https://www.isbn-international.org/content/what-isbn)

Ontario Creates. (2020, May). MAY 2020 PROFILE. Retrieved June 15, 2020, from http://www.ontariocreates.ca/collaboration/research_and_industry_information/industry_profiles/Book_Industry_Profile.htm#footnote-2 (http://www.ontariocreates.ca/collaboration/research_and_industry_information/industry_profiles/Book_Industry_Profile.htm#footnote-2)

R.R. Bowker LLC. (2014). FAQs: General Questions. Retrieved June 15, 2020, from https://www.isbn.org/faqs_general_questions#isbn_faq5 (https://www.isbn.org/faqs_general_questions#isbn_faq5)

Watson, A. (2019, January 16). Topic: Book Industry. Retrieved June 15, 2020, from https://www.statista.com/topics/1177/book-market/ (https://www.statista.com/topics/1177/book-market/)

Ziegler, C. (2005, May 14). Book-Crossing Dataset … mined by Cai-Nicolas Ziegler, DBIS Freiburg. Retrieved June 10, 2020, from http://www2.informatik.uni-freiburg.de/~cziegler/BX/ (http://www2.informatik.uni-freiburg.de/~cziegler/BX/)