Goodreads Project

Hananel Mandeleyl, Dmitry Zalyalyeyev

Phase 1 | **Data Import**

First, we shall import our dataset using the `read_delim()` function.
The reason we chose using the `read_delim()` function is When you run `read_delim()` it prints out a column specification that gives the name and type of each column:

```
data <- read_delim("books.csv", ",", quote = "\"")
```

```
## Warning: Missing column names filled in: 'X13' [13]
```

```
## Parsed with column specification:
## cols(
##   bookID = col_double(),
##   title = col_character(),
##   authors = col_character(),
##   average_rating = col_double(),
##   isbn = col_character(),
##   isbn13 = col_double(),
##   language_code = col_character(),
##   `  num_pages` = col_double(),
##   ratings_count = col_double(),
##   text_reviews_count = col_double(),
##   publication_date = col_character(),
##   publisher = col_character(),
##   X13 = col_logical()
## )
```

```
## Warning: 13 parsing failures.
##  row            col           expected
actual        file
## 3349 average_rating a double            Jr./Sam B. Warner
'books.csv'
## 3349   num_pages    a double           en-US
'books.csv'
## 3349 X13            1/0/T/F/TRUE/FALSE Harvard University Press
'books.csv'
## 4703 average_rating a double            one of the founding members of this Tolkien website)/Verlyn Flieger/Turgon (=Davi
d E. Smith) 'books.csv'
## 4703   num_pages    a double           eng
'books.csv'
## .... .............. ..................
.......................................................................................... ..........
## See problems(...) for more details.
```

We can see a few warnings, but we will ignore them for now and get back to them during the tidying phase.
For now, we'll just get to know our data better.

1.1. What are we dealing with?

Let's take a sneak peak first:

bookID	title	authors	average_rating	isbn	isbn13	language_code	num_pages	ratings_count	text_reviews_count	publication_date	publisher	X13
1	Harry Potter and the Half-Blood Prince (Harry Potter #6)	J.K. Rowling/Mary GrandPr<U+00E9>	4.57	439785960	9780440000000	eng	652	2095690	27591	9/16/2006	Scholastic Inc.	NA
2	Harry Potter and the Order of the Phoenix (Harry Potter #5)	J.K. Rowling/Mary GrandPr<U+00E9>	4.49	439358078	9780440000000	eng	870	2153167	29221	09/01/2004	Scholastic Inc.	NA
4	Harry Potter and the Chamber of Secrets (Harry Potter #2)	J.K. Rowling	4.42	439554896	9780440000000	eng	352	6333	244	11/01/2003	Scholastic	NA

Our dataset contains information about 11,127 books from the *Goodreads* online catalog.

> **Goodreads** is a social cataloging website that allows individuals to search freely its database of books, annotations, quotes, and reviews. Users can sign up and register books to generate library catalogs and reading lists.

1.2. Variables

Our dataset consists of 12 variables:

1. `bookID` : Book ID.
2. `title` : Title.
3. `authors` : Authors.
4. `average_rating` : Average rating.
5. `isbn` : ISBN number.
6. `isbn13` : ISBN13 number.
7. `language_code` : Language code.
8. `num_pages` : Number of pages.
9. `ratings_count` : Ratings count.
10. `text_reviews_count` : Text reviews count.
11. `publication_date` : Publication date.
12. `publisher` : Publisher.

We are dealing with more than one kind of variables; we have categorical variables, numerical variables, some are discrete and some are continuous, some are unique to each observation and some are not.
On our project we will research how different characteristics influence a title's average score, and how big an effect each has on it.

1.3. Some early statistics and insights

Even at such an early stage it is wise to get some raw insights about the data. A `summary()` will be great:

```
summary(data)
```

```
##      bookID         title              authors          average_rating
##  Min.   :    1   Length:11127       Length:11127       Min.   :0.000
##  1st Qu.:10287   Class :character   Class :character   1st Qu.:3.770
##  Median :20287   Mode  :character   Mode  :character   Median :3.960
##  Mean   :21311                                         Mean   :3.934
##  3rd Qu.:32105                                         3rd Qu.:4.140
##  Max.   :45641                                         Max.   :5.000
##                                                        NA's   :4
##      isbn               isbn13              language_code         num_pages
##  Length:11127       Min.   :    674842111   Length:11127       Min.   :   0.0
##  Class :character   1st Qu.:9780350000000   Class :character   1st Qu.: 192.0
##  Mode  :character   Median :9780580000000   Mode  :character   Median : 299.0
##                     Mean   :9757248897830                      Mean   : 336.4
##                     3rd Qu.:9780870000000                      3rd Qu.: 416.0
##                     Max.   :9790010000000                      Max.   :6576.0
##                     NA's   :1                                  NA's   :4
##  ratings_count     text_reviews_count publication_date    publisher
##  Min.   :      0   Min.   :    0.0    Length:11127       Length:11127
##  1st Qu.:    104   1st Qu.:    9.0    Class :character   Class :character
##  Median :    745   Median :   47.0    Mode  :character   Mode  :character
##  Mean   :  17936   Mean   :  541.9
##  3rd Qu.:   4994   3rd Qu.:  237.5
##  Max.   :4597666   Max.   :94265.0
##
##    X13
##  Mode:logical
##  NA's:11127
##
##
##
##
##
```

It looks like we have everything we need to explore and tidy-up the data.

Phase 2 | Data Tidying

In this phase we would like to clean the data up and address any issues which came up during the import phase. Those do not require data transformations per se, but rather cleaning and rearrangement (e.g. entry shifting and text leaks).

We can see that our data is pretty clean as it is; each row is an observation of its own, and each column is an understadable, intuitive variable; but all that doesn't mean it's flawless.

2.1. Row shifting and text leakage fix

Let's deal with those warnings we got while importing the data.
Fortunately, we have the `problems()` function, which makes it really easy to identify problematic entries:

```
data %>%
  problems %>%
  kable("html") %>%
  kable_styling(font_size = 9)
```

row	col	expected	actual	file
3349	average_rating	a double	Jr./Sam B. Warner	'books.csv'
3349	num_pages	a double	en-US	'books.csv'
3349	X13	1/0/T/F/TRUE/FALSE	Harvard University Press	'books.csv'
4703	average_rating	a double	one of the founding members of this Tolkien website)/Verlyn Flieger/Turgon (=David E. Smith)	'books.csv'
4703	num_pages	a double	eng	'books.csv'
4703	X13	1/0/T/F/TRUE/FALSE	Cold Spring Press	'books.csv'
5878	average_rating	a double	Rawles	'books.csv'
5878	isbn13	no trailing characters	X	'books.csv'
5878	num_pages	a double	eng	'books.csv'
5878	X13	1/0/T/F/TRUE/FALSE	Huntington House Publishers	'books.csv'
8980	average_rating	a double	Son & Ferguson	'books.csv'
8980	num_pages	a double	eng	'books.csv'
8980	X13	1/0/T/F/TRUE/FALSE	Brown Son & Ferguson Ltd.	'books.csv'

We've got a few problematic entries, but notice that they all come from 4 specific rows.
Another thing to notice is that while importing we got another warning — there are 13 columns identifed while only 12 were expected, which suggests we are dealing with a right shift of the said rows.

After a manual review we've narrowed the list of errors to:

Row	Error
3349	Text leakage, row shifting
4703	Text leakage, row shifting
5878	Text leakage, row shifting
8980	Text leakage, row shifting

After careful consideration we've decided that the best way to fix the errors would be applying manual row reconstruction, as R doesn't read problematic entries when it encounters parcing problems, but rather fills them with NA values, resulting in information loss.
Let's import the fixed dataset and make sure all the warnings are resolved:

```
data <- read_delim("books update 1 rows.csv", ",", quote = "\"")
```

```
## Parsed with column specification:
## cols(
##   bookID = col_double(),
##   title = col_character(),
##   authors = col_character(),
##   average_rating = col_double(),
##   isbn = col_character(),
##   isbn13 = col_double(),
##   language_code = col_character(),
##   `  num_pages` = col_double(),
##   ratings_count = col_double(),
##   text_reviews_count = col_double(),
##   publication_date = col_character(),
##   publisher = col_character()
## )
```

```
cat("The number of problems found:", data %>% problems %>% nrow)
```

```
## The number of problems found: 0
```

Success! No more row shifting and text leakage.

2.2. Nonuniformity of publisher names

We were suspicious that publisher names will be spelled nonuniformly across different titles, and so was the case. A manual fix has been applied for efficiency reasons.
Let's load a fixed version and continue:

```
data <- read_delim("books update 2 publishers.csv", ",", quote = "\"")
```

```
## Parsed with column specification:
## cols(
##   bookID = col_double(),
##   title = col_character(),
##   authors = col_character(),
##   average_rating = col_double(),
##   isbn = col_character(),
##   isbn13 = col_double(),
##   language_code = col_character(),
##   `  num_pages` = col_double(),
##   ratings_count = col_double(),
##   text_reviews_count = col_double(),
##   publication_date = col_character(),
##   publisher = col_character()
## )
```

2.3. Dialect-specific language codes grouping

Although distinguishing between different English dialects in the `language_code` variable can add a lot of valuable information to the study, we've chosen to combine all the English dialects except an old version called Middle English (1,100–1,500) to a single value — **"eng"**.
The reason is that in addition to dialect-specific English language codes (e.g. "en-GB"), the `language_code` variable also contains an ambiguous "eng" value, what might result in an inaccurate distribution of dialect influence, and in turn lead to an untruthful representation of it.
We've written simple code to convert dialect-specific language codes into generic "eng" entries:

```
before <- data$language_code[6]

data$language_code[grepl("en", data$language_code, fixed = TRUE) &
                     !grepl("eng", data$language_code, fixed = TRUE) &
                     !grepl("enm", data$language_code, fixed = TRUE)] <- "eng"

after <- data$language_code[6]
```

Let's test it by printing the values of `before` and `after`, which contain the value observation 6 before and after the fix:

```
## Language code of observation 6 before the fix: en-US
## Language code of observation 6 after the fix: eng
```

2.4. Publication date format fix

We noticed that the date format is nonuniform (e.g. some are *mm/dd/yyyy* while others are *m/dd/yyyy*).
The following code fills in the missing digit where required and changes the format to a local one; we'll test it

```
before <- data$publication_date[1]

str_to_date <- function(string) {
  if (nchar(string) < 10) {
    return(paste0("0", string) %>%
             as.Date(format = "%m/%d/%Y") %>%
             format("%d/%m/%Y"))
  } else {
    return(string %>%
             as.Date(format = "%m/%d/%Y") %>%
             format("%d/%m/%Y"))
  }
}

data$publication_date <- sapply(data$publication_date, str_to_date)

after <- data$publication_date[1]
```

Let's test it by printing the values of `before` and `after`, which now hold the value observation 1 before and after the fix:

```
## Publication date of observation 1 before the fix: 9/16/2006
## Publication date of observation 1 after the fix: 16/09/2006
```

Success!

2.5. Columns rearrangement

As our target variable is the `average_rating` variable, we would like to rearrange the dataset so that it would appear in the last column.
Also, we would like to fix some names (i.e remove whitespaces from column names), and group all the categorical variables on the left and all the numerical varibales on the right:

```
names(data) <- gsub(" ", "", names(data))

data <- data[, c("bookID", "title", "authors", "publisher", "language_code", "isbn", "isbn13", "publication_date", "num_pages", "ratings_count", "text_reviews_count", "average_rating")]
```

Looks like our dataset is clean and ready for the next phase:

```
data[1:3, ] %>%
  kable("html") %>%
  kable_styling(font_size = 9)
```

bookID	title	authors	publisher	language_code	isbn	isbn13	publication_date	num_pages	ratings_count	text_reviews_count	average_rating
1	Harry Potter and the Half-Blood Prince (Harry Potter #6)	J.K. Rowling/Mary GrandPr<U+00E9>	Scholastic	eng	439785960	9780000000000	16/09/2006	652	2095690	27591	4.57
2	Harry Potter and the Order of the Phoenix (Harry Potter #5)	J.K. Rowling/Mary GrandPr<U+00E9>	Scholastic	eng	439358078	9780000000000	01/09/2004	870	2153167	29221	4.49
4	Harry Potter and the Chamber of Secrets (Harry Potter #2)	J.K. Rowling	Scholastic	eng	439554896	9780000000000	01/11/2003	352	6333	244	4.42

Phase 3 | **Understanding the data**

Part A | **Transforming variables**

In this section, different transformations will applied to the data in order to extract the most useful information from the data, while removing the unnecessary parts to maintain the lowest dimension possible.

3.A.1. `NA` handling

Of all the `NA` entries we would first like to discuss the most important ones — where `average_rating == NA`. As we previously mentioned, the `average_rating` variable is our target, hence observations without it are ununseful to us as training samples:

```
data <- data[!is.na(data$average_rating), ]
```

Regarding the rest, we first need to check how bad is the situation anyway.
Here are the observations containing at least one `NA` entry:

```
data[rowSums(is.na(data)) > 0,] %>%
  head %>%
  kable("html") %>%
  kable_styling(font_size = 9)
```

bookID	title	authors	publisher	language_code	isbn	isbn13	publication_date	num_pages	ratings_count	text_reviews_count	average_rating
31373	In Pursuit of the Proper Sinner (Inspector Lynley #10)	Elizabeth George	Bantam	eng	553575104	9780000000000	NA	718	10608	295	4.10
45531	Montaillou village occitan de 1294 <U+00E0> 1324	Emmanuel Le Roy Ladurie/Emmanuel Le Roy-Ladurie	Folio	fre	2070323285	9780000000000	NA	640	15	2	3.96

```
which(is.na(data), arr.ind = TRUE)
```

```
##              row col
## 11/31/2000  8181   8
## 6/31/1982  11099   8
```

The dates that were entered don't exist. After quick search of the *goodreads* website we got the correct publication dates (October/June 30th instead of 31st), so thankfully no removals are needed. Let's fix the entries and check:

```
data[unique(which(is.na(data), arr.ind = TRUE)[, 1]),
     unique(which(is.na(data), arr.ind = TRUE)[, 2])] <- as.Date(c("10/31/2000", "06/30/1982"), format = "%m/%d/%Y") %>% for
mat("%d/%m/%Y")

cat("Amount of NA entries:", sum(is.na(data)))
```

```
## Amount of NA entries: 0
```

Great! Our data is free from `NA` entries.

3.A.2. Removal of observations with a small ratings sample size

The `average_rating` is averaged between `ratings_count` reviews, so we've decided that observations with `ratings_count << 30` will be regarded as non-representative and will be removed.
This is because their `average_rating` has been determinded using an insufficient sample size.

```
data <- data[data$ratings_count >= 30, ]
```

3.A.3. ISBN and ISBN13

> The **International Standard Book Number** (**ISBN**) is a numeric commercial book identifier which is intended to be unique.
> An ISBN is assigned to each separate edition and variation of a publication. For example, an e-book, a paperback and a hardcover edition of the same book will each have a different ISBN.

In other words, the `isbn` and `isbn13` (13-digit ISBN format) variables aren't really numerical — they are **categorical**, as we can clearly see in the `summary()` we've printed earlier.

Provided that each observation is a publication of its own, we can safely presume that every `isbn`, `isbn13` and `bookID` entry will be unique for each observation.
Hence, the `isbn` and `isbn13` doesn't offer any distinction between observations more than `bookID` does.

For the reason above we have decided to remove both the `isbn` and the `isbn13` variable, and convert the `book_id` variable to be strictly categorical:

```
# Dropping the isbn and isbn13 variables
drops <- c("isbn","isbn13")
data <- data[ , !(names(data) %in% drops)]

# Converting the bookID variable to categorical
data$bookID <- sapply(data$bookID, toString)
```

Let's have a look:

```
data %>%
  head %>%
  kable("html") %>%
  kable_styling(font_size = 9)
```

bookID	title	authors	publisher	language_code	publication_date	num_pages	ratings_count	text_reviews_count	average_rating
1	Harry Potter and the Half-Blood Prince (Harry Potter #6)	J.K. Rowling/Mary GrandPr<U+00E9>	Scholastic	eng	16/09/2006	652	2095690	27591	4.57
2	Harry Potter and the Order of the Phoenix (Harry Potter #5)	J.K. Rowling/Mary GrandPr<U+00E9>	Scholastic	eng	01/09/2004	870	2153167	29221	4.49
4	Harry Potter and the Chamber of Secrets (Harry Potter #2)	J.K. Rowling	Scholastic	eng	01/11/2003	352	6333	244	4.42
5	Harry Potter and the Prisoner of Azkaban (Harry Potter #3)	J.K. Rowling/Mary GrandPr<U+00E9>	Scholastic	eng	01/05/2004	435	2339585	36325	4.56
8	Harry Potter Boxed Set Books 1-5 (Harry Potter #1-5)	J.K. Rowling/Mary GrandPr<U+00E9>	Scholastic	eng	13/09/2004	2690	41428	164	4.78
10	Harry Potter Collection (Harry Potter #1-6)	J.K. Rowling	Scholastic	eng	12/09/2005	3342	28242	808	4.73

Beautiful, we can see the `bookID` column is left-aligned, suggesting it is now a categorical variable, as opposed to numerical columns which are right-aligned.

3.A.4. Title length transformation

The title might have a substantial contribution to a publication's score. We are influenced by marketing, social conventions and more.
Yet, due to performance considerations, we aren't going to decompose the titles, nor search for common phrases or buzzwords.
If so, you might wonder why the `title` variable wasn't removed along with `isbn` and `isbn13`.
The reasons are:

1. The `title` variable isn't unique per observation as `isbn` and `isbn13`.
2. There is undelying information hidden in the title.

An example for excellent information we can extract from the title is its length. We think that a title's length may affect the average score — different title lengths can be associated with different genres, levels, time periods and more. We even suspect there might be a title length "sweet spot". This is why we have decided to add a `title_len` variable.

Note: We are discussing the different influences on a publication's average rating specifically on *goodreads*, meaning that we are taking the length of the publications' title **as it appears on the website**.

```
# Adding the title length variable
data$title_len <- sapply(data$title, nchar)

# Column rearrangement
data <- data[, c("bookID", "title", "authors", "publisher", "language_code", "publication_date", "title_len", "num_pages",
"ratings_count", "text_reviews_count", "average_rating")]

data %>%
  head %>%
  kable("html") %>%
  kable_styling(font_size = 9)
```

bookID	title	authors	publisher	language_code	publication_date	title_len	num_pages	ratings_count	text_reviews_count	average_rating
1	Harry Potter and the Half-Blood Prince (Harry Potter #6)	J.K. Rowling/Mary GrandPr<U+00E9>	Scholastic	eng	16/09/2006	57	652	2095690	27591	4.57
2	Harry Potter and the Order of the Phoenix (Harry Potter #5)	J.K. Rowling/Mary GrandPr<U+00E9>	Scholastic	eng	01/09/2004	60	870	2153167	29221	4.49
4	Harry Potter and the Chamber of Secrets (Harry Potter #2)	J.K. Rowling	Scholastic	eng	01/11/2003	58	352	6333	244	4.42
5	Harry Potter and the Prisoner of Azkaban (Harry Potter #3)	J.K. Rowling/Mary GrandPr<U+00E9>	Scholastic	eng	01/05/2004	59	435	2339585	36325	4.56
8	Harry Potter Boxed Set Books 1-5 (Harry Potter #1-5)	J.K. Rowling/Mary GrandPr<U+00E9>	Scholastic	eng	13/09/2004	54	2690	41428	164	4.78
10	Harry Potter Collection (Harry Potter #1-6)	J.K. Rowling	Scholastic	eng	12/09/2005	44	3342	28242	808	4.73

3.A.5. From publication date to time since publication

The `publication_date` variable is categorical. However, we would like to transform it into a numeric variable which will be more useful in our further analysis — hence the "time since publication" variable. We will do so by calculating the number of weeks that have passed since each book's publication date until August 5th, 2020, i.e. the dataset's date of approval:

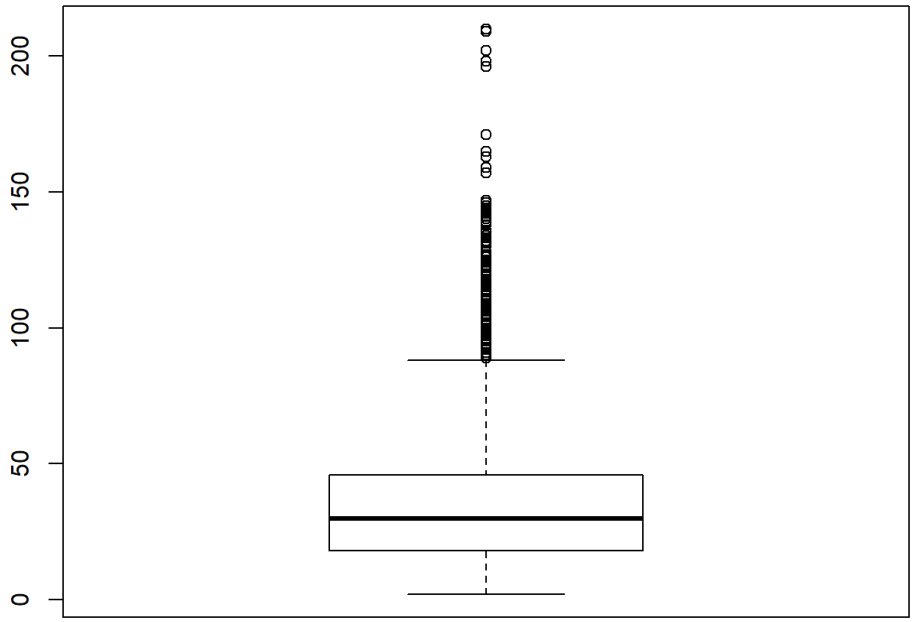
```
data$publication_date <- as.numeric(rep(as.Date("05/08/2020", format = "%d/%m/%Y"), times = nrow(data)) - as.Date(data$publi
cation_date, format = "%d/%m/%Y")) / 7

# Variable name change
names(data)[names(data) == "publication_date"] <- "weeks_since_publication"
```

3.A.6. Outliers removal

Outliers can negatively affect our research in multiple ways.
It it especially important to remove outliers now before performing any scaling on the data.
After careful consideration and review of numerical variables' boxplots, we've came to the conclusion that for our dataset, statistical outliers are'nt necessarily illogical.
For example, the following plot suggests that a 200-letters title is an obvious outlier:

```
boxplot(data$title_len)
```



But as we can see, **those are** titles:

```
data[data$title_len >= 200, ] %>%
  head %>%
  kable("html") %>%
  kable_styling(font_size = 9)
```

bookID	title	authors	publisher	language_code	weeks_since_publication	title_len	num_pages	ratings_count	text_reviews_count	average_rating
2810	Christian Mythmakers: C.S. Lewis Madeleine L'Engle J.R.R. Tolkien George MacDonald G.K. Chesterton Charles Williams Dante Alighieri John Bunyan Walter Wangerin Robert Siegel and Hannah Hurnard	Rolland Hein/Clyde S. Kilby	Cornerstone Press Chicago	eng	922.4286	202	303	448	18	3.93
10255	Dr. Mary's Monkey: How the Unsolved Murder of a Doctor a Secret Laboratory in New Orleans and Cancer-Causing Monkey Viruses are Linked to Lee Harvey Oswald the JFK Assassination and Emerging Global Epidemics	Edward T. Haslam/Jim Marrs	Trine Day	eng	696.4286	209	374	1023	164	3.92
39124	The covert war against rock: what you don't know about the deaths of Jim Morrison Tupac Shakur Michael Hutchence Brian Jones Jimi Hendrix Phil Ochs Bob Marley Peter Tosh John Lennon The Notorious B.I.G	Alex Constantine	Feral House	eng	1057.2857	210	280	85	8	3.58

A better indication of outliers in our dataset would be the minimum and maximum value of each numerical variable:

```
min_max_num_vars <- tibble(min_max = c("Minimum", "Maximum"),
                           weeks_since_publication = c(min(data$weeks_since_publication), max(data$weeks_since_publicatio
n)),
                           title_len = c(min(data$title_len), max(data$title_len)),
                           num_pages = c(min(data$num_pages), max(data$num_pages)),
                           ratings_count = c(min(data$ratings_count), max(data$ratings_count)),
                           text_reviews_count = c(min(data$text_reviews_count), max(data$text_reviews_count)),
                           average_rating = c(min(data$average_rating), max(data$average_rating))) %>% column_to_rownames(va
r = "min_max")

min_max_num_vars %>%
  kable("html") %>%
  kable_styling(font_size = 9)
```

	weeks_since_publication	title_len	num_pages	ratings_count	text_reviews_count	average_rating
Minimum	18.14286	2	0	30	0	2.40
Maximum	6292.28571	210	6576	4597666	94265	4.82

After careful review of the values and the dataset we can confirm that no additional outlier removal is necessary as all numbers make sense.

Note: We found out that among the authors presented in the dataset there is a “NOT A BOOK” value.
We have examined the relevant entries, and it appears that those refer to audio content. Therefore, we have decided to leave those entries in the dataset, as was the case with other audiobooks that were mentioned in the dataset.

3.A.7. Numerical variables scaling

And lastly before visualizations, ince a scaled version of the data might be valuable for our further analysis, we would thus like to create 2 versions of the dataset in which all the numeric variables appear scaled — with normal scaling in one, and min-max scaling in the other:

```
numer_vars <- c("weeks_since_publication", "title_len", "num_pages", "ratings_count", "text_reviews_count")

# Normal scaling
norm_scaled_data <- data
norm_scaled_data[, numer_vars] <- scale(norm_scaled_data[, numer_vars])

## Varible name changes
norm_scaled_data <- norm_scaled_data %>%
  rename(scaled_weeks_since_publication = weeks_since_publication,
         scaled_title_len = title_len,
         scaled_num_pages = num_pages,
         scaled_ratings_count = ratings_count,
         scaled_text_reviews_count = text_reviews_count)

# Min-max scaling
min_max_scaler <- function(x) {(x - min(x)) / (max(x) - min(x))}
min_max_scaled_data <- data
min_max_scaled_data[, numer_vars] <- min_max_scaler(min_max_scaled_data[, numer_vars])

## Varible name changes
min_max_scaled_data <- min_max_scaled_data %>%
  rename(scaled_weeks_since_publication = weeks_since_publication,
         scaled_title_len = title_len,
         scaled_num_pages = num_pages,
         scaled_ratings_count = ratings_count,
         scaled_text_reviews_count = text_reviews_count)
```

Part B | **Visualizing**

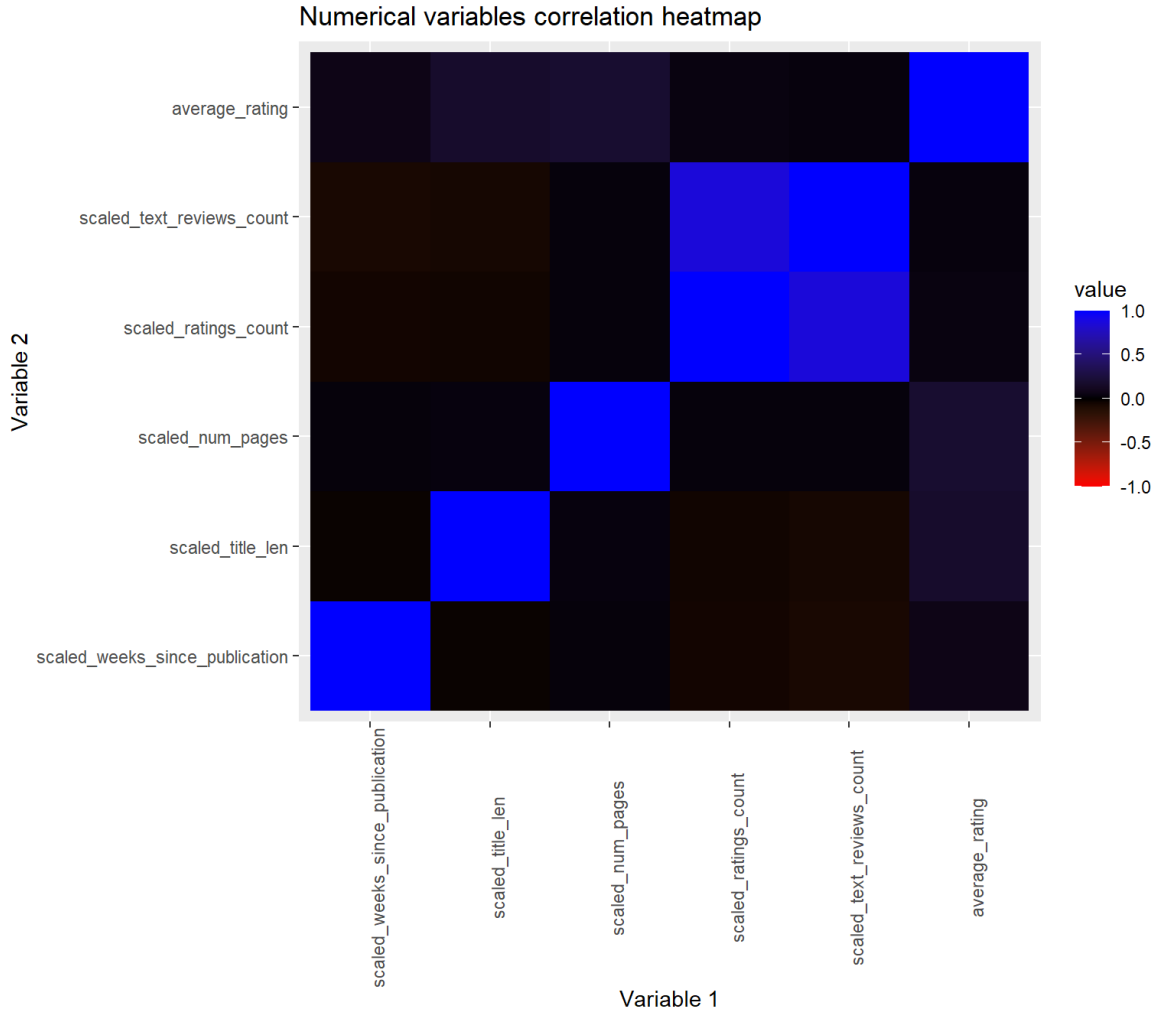
Visual information is incredibly valuable to us, and we will try to provide some good examples for it.

3.B.1. Correlation heatmap

The first visualization that we would like to introduce is the correlation between any two of the predictors. In order to do that, we will use our `norm_scaled_data` to produce we used a correlation heatmap:

```
scaled_numer_vars <- c("scaled_weeks_since_publication", "scaled_title_len", "scaled_num_pages", "scaled_ratings_count", "sc
aled_text_reviews_count", "average_rating")

ggplot(melt(cor(norm_scaled_data[, scaled_numer_vars])), aes(x = Var1, y = Var2, fill = value)) +
  geom_tile() +
  theme(axis.text.x = element_text(angle = 90)) +
  labs(x = "Variable 1", y = "Variable 2", title = "Numerical variables correlation heatmap") +
  scale_fill_gradient2(low = "red", high = "blue", mid = "black",
   midpoint = 0, limit = c(-1,1), space = "Lab")
```



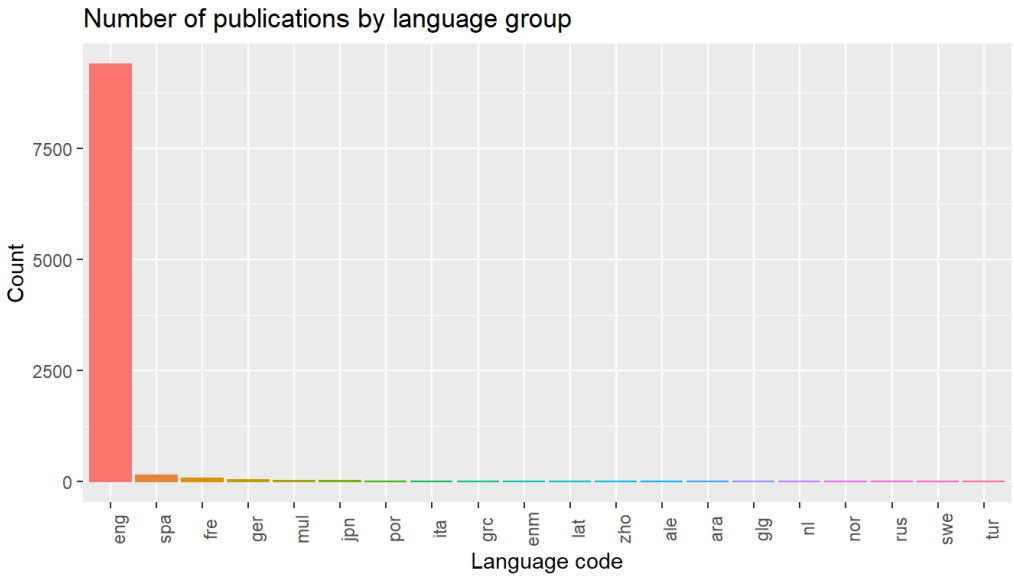
Unsurprisingly, the `scaled_ratings_count` and `scaled_text_reviews_count` variables are strongly correlated, while the correlations of the rest are only weak to moderate, but maybe still statistically significant.

3.B.2. Number of publications by language

In order to better understand which books our dataset stands for we have checked what is the amount of books published by their language.

We first looked at the number of books in each language

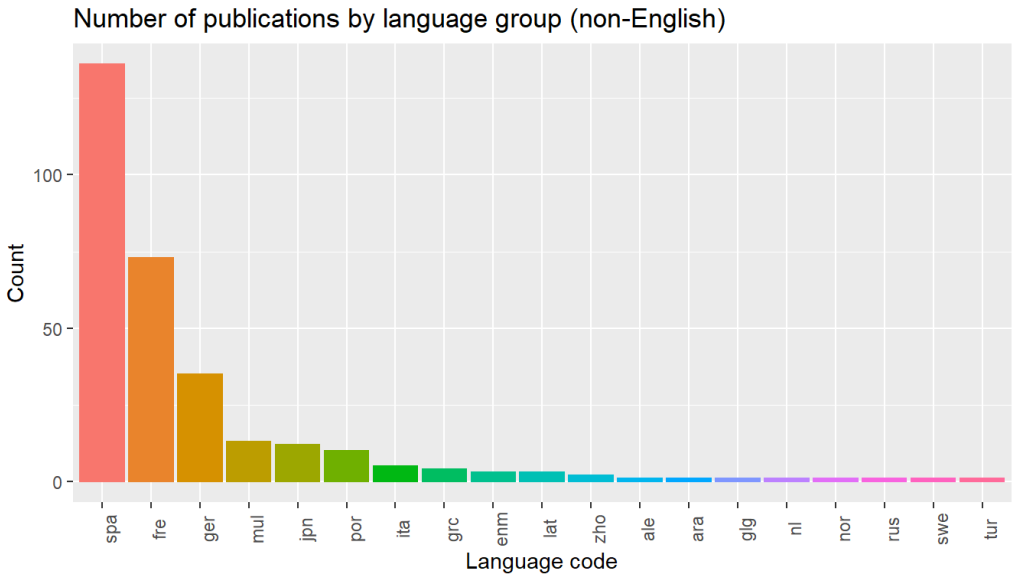
```
language <- forcats::fct_infreq(data$language_code)
ggplot(data, aes(x = language)) +
  geom_bar(mapping = aes(fill = language, color = language), show.legend = FALSE) +
  labs(y = "Count", x = "Language code", title = "Number of publications by language group") +
  theme(axis.text.x = element_text(angle = 90))
```



From the graph above we deduce that the number of books in the datatset that were published in English is significantly higher than those that were published in any other language. This is an important observation as it implies that the relevance of the results of our research might be predominantly limited to the English speaking community, and more so — does not take into account people's preferences in regards to the books that were published in other languages. We aknowledge the fact that such limitation inherently exists as we explore the data taken from a website the language of which is English.

In order to explore the amount of books published in each language besides English, we turn to a graph that refers to all books in the dataset, except those published in English:

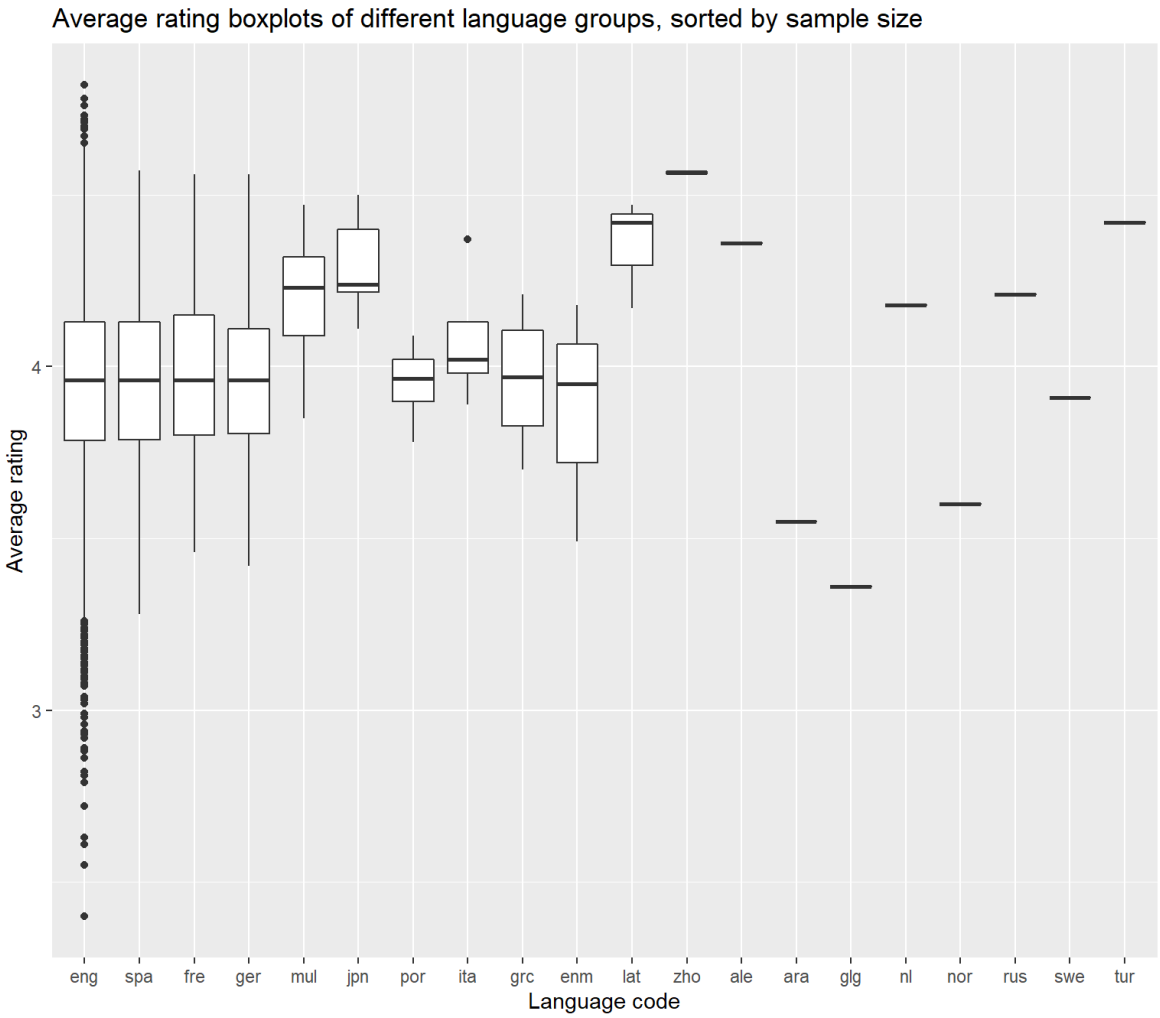
```
non_eng <- filter(data, language_code != "eng")
Language <- forcats::fct_infreq(non_eng$language_code)
ggplot(non_eng, aes(x = Language)) +
  geom_bar(mapping = aes(fill = Language, color = Language), show.legend = FALSE) +
  labs(y = "Count", x = "Language code", title = "Number of publications by language group (non-English)") +
  theme(axis.text.x = element_text(angle = 90))
```



3.B.3. Average rating by language group boxplots

We now turn to better understand the ratings of books written in a particular language, compared to others:

```
language <- forcats::fct_infreq(data$language_code)
avg_rating <- data$average_rating
ggplot(data) +
  geom_boxplot(mapping = aes(x = language, y = avg_rating)) +
  labs(y = "Average rating", x = "Language code", title = "Average rating boxplots of different language groups, sorted by s
ample size")
```

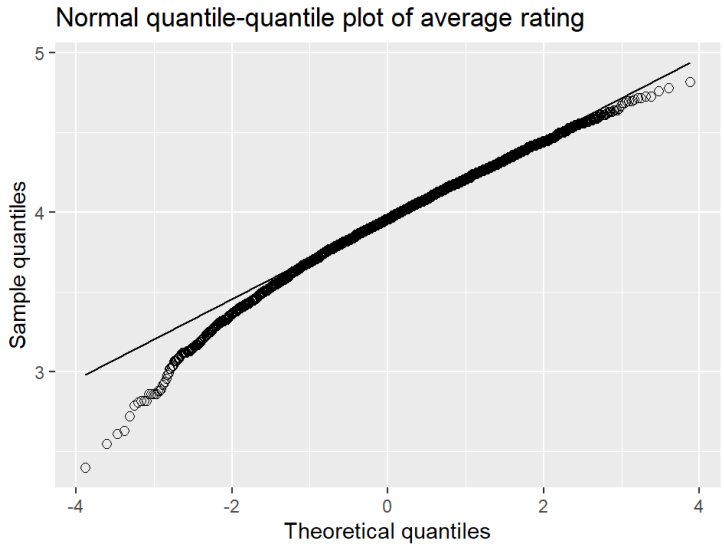


The boxplot graph provides us an interesting inference: the median average rating among each of the four most common languages is very similar — almost 4 points.

3.B.4. Average rating distribution

We would like to check whether the average rating distributes normally.
For this purpose we will use the normal Q-Q plot:

```
ggplot(data, aes(sample = average_rating)) +
  stat_qq(shape = 21, size = 2) +
  stat_qq_line() +
  labs(x = "Theoretical quantiles", y = "Sample quantiles", title = "Normal quantile-quantile plot of average rating")
```



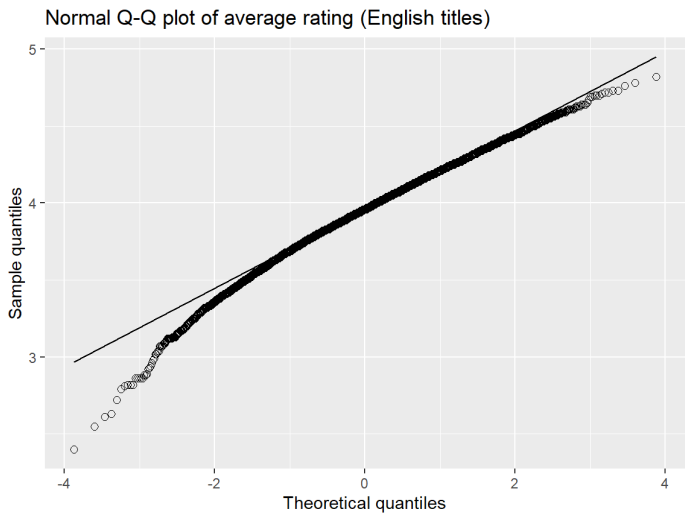
In addition, we would like to check whether the average rating distributes differently for English and non-English titles:

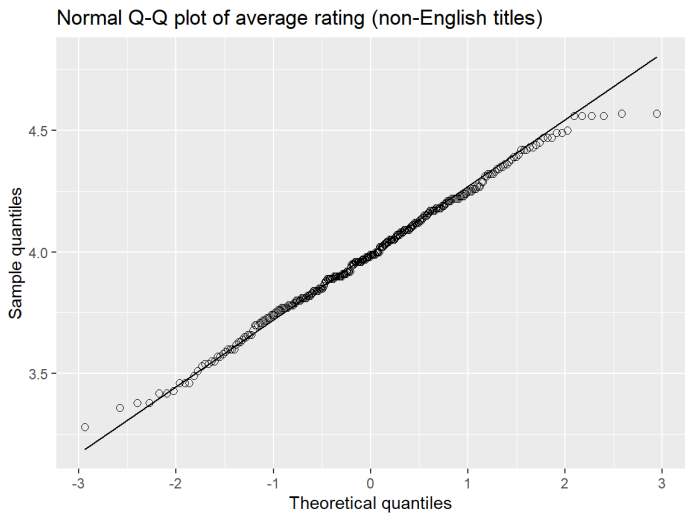
```
eng <- data %>% filter(language_code == "eng")
non_eng <- data %>% filter(language_code != "eng")

eng_qq <- ggplot(eng, aes(sample = average_rating)) +
  stat_qq(shape = 21, size = 2) +
  stat_qq_line() +
  labs(x = "Theoretical quantiles", y = "Sample quantiles", title = "Normal Q-Q plot of average rating (English titles)")

non_eng_qq <- ggplot(non_eng, aes(sample = average_rating)) +
  stat_qq(shape = 21, size = 2) +
  stat_qq_line() +
  labs(x = "Theoretical quantiles", y = "Sample quantiles", title = "Normal Q-Q plot of average rating (non-English titles)"
)

grid.arrange(eng_qq, non_eng_qq, ncol = 2)
```

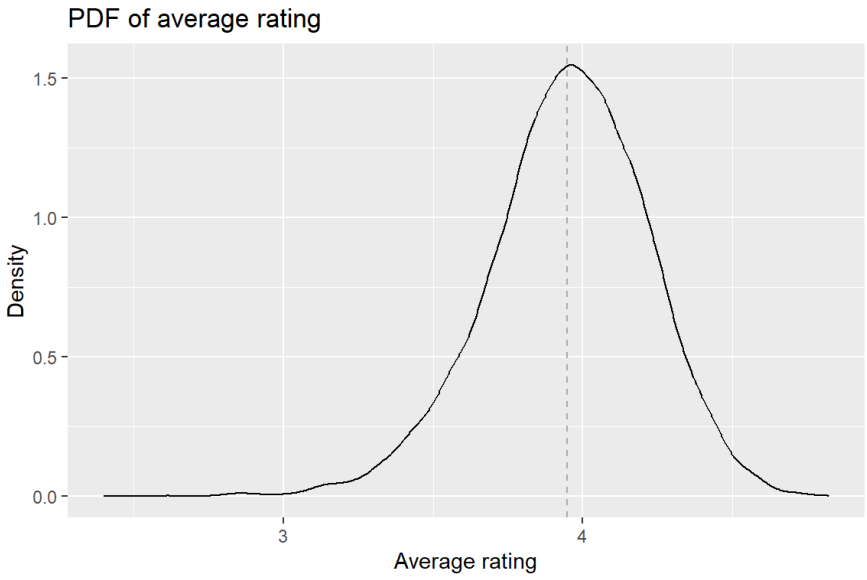




Curiously enough, non-English titles' average rating distribution appears to be approximately more normal than those of English titles, yet keep in mind that the former group's sample size is several orders of magnitude smaller than the latter's, thus only has small effect on the general curve.

Now, we would like to see exactly how the average rating distributes:

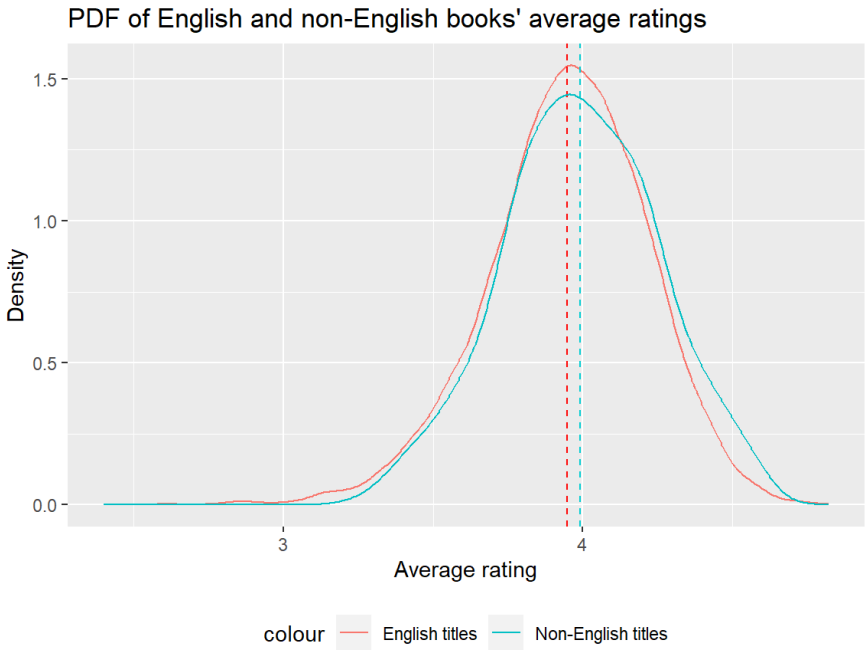
```
avg_rating <- data$average_rating
ggplot(data) +
  stat_density(aes(x = avg_rating), geom = "line", position = "identity") +
  labs(y = "Density", x = "Average rating", title = "PDF of average rating") +
  geom_vline(aes(xintercept = mean(average_rating)), linetype = "dashed", color = "dark grey")
```



Let's see how each language group's rating distributes seperately:

```
is_eng <- filter(data, language_code == "eng")
non_eng <- filter(data, language_code != "eng")
is_eng_avg_rating <- is_eng$average_rating
non_eng_avg_rating <- non_eng$average_rating

ggplot() +
  stat_density(aes(x = is_eng_avg_rating, color = "English titles"), geom = "line", position = "identity") +
  geom_vline(aes(xintercept = mean(is_eng_avg_rating)), linetype = "dashed", color = "red") +
  stat_density(aes(x = non_eng_avg_rating, color = "Non-English titles"), geom = "line", position = "identity") +
  geom_vline(aes(xintercept = mean(non_eng_avg_rating)), linetype = "dashed", color = "darkturquoise") +
  labs(y = "Density", x = "Average rating", title = "PDF of English and non-English books' average ratings") +
  theme(legend.position = "bottom")
```



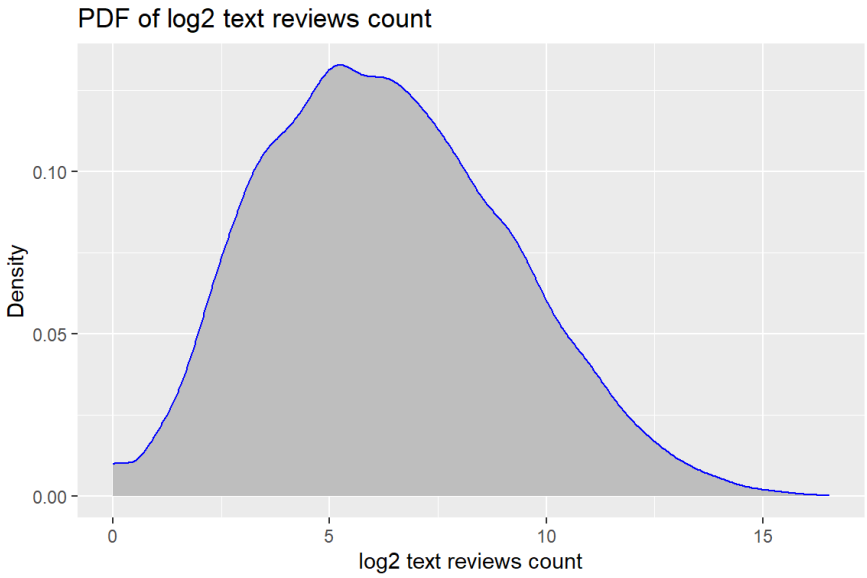
The graph is leading us to the conclusion that the average ratings of both language groups distribute similarly — approximately normal around the mean, yet denser around the left tail, as we would expected considering the mean rating is very close to 4, and the possible values being 0 to 5. This result reflects the Q-Q plots excellently, with the slight differences with the non-English titles explained by a smaller sample size.

3.B.5. Ratings count and text reviews

We will now look at the distribution of the text reviews count and the number of ratings predictors.
Since our data ranges across multiple orders of magnitude, we apply logarithm to the text reviews count and the ratings count. We use `log2()` as suggested on page 57 in the “R for Data Science” book.

Text reviews count:

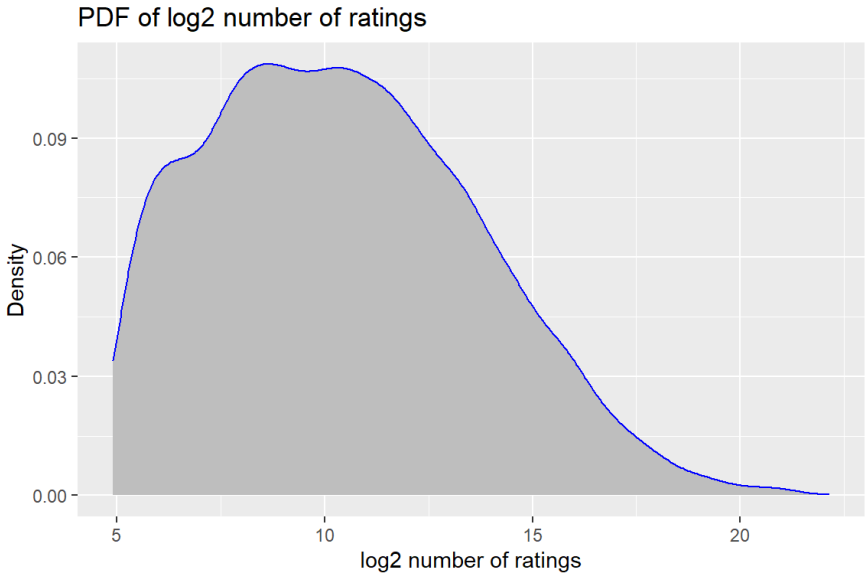
```
text_review_count <- log2(data$text_reviews_count)
ggplot(data, aes(x = text_review_count)) +
  geom_density(fill = "grey", color = "blue") +
  labs(y = "Density", x = "log2 text reviews count", title = "PDF of log2 text reviews count")
```



We can see that this distribution looks log normal with a slight skew to the left.

The number of ratings:

```
num_of_ratings <- log2(data$ratings_count)
ggplot(data, aes(x = num_of_ratings)) +
  geom_density(fill = "grey", color = "blue") +
  labs(y = "Density", x = "log2 number of ratings", title = "PDF of log2 number of ratings")
```



We can see that this distribution also looks log normal with a skew to the left.
Both distributions make sense, and it is no surprise that there is a high correlation between those two predictors, as we have shown in the correlation heatmap.

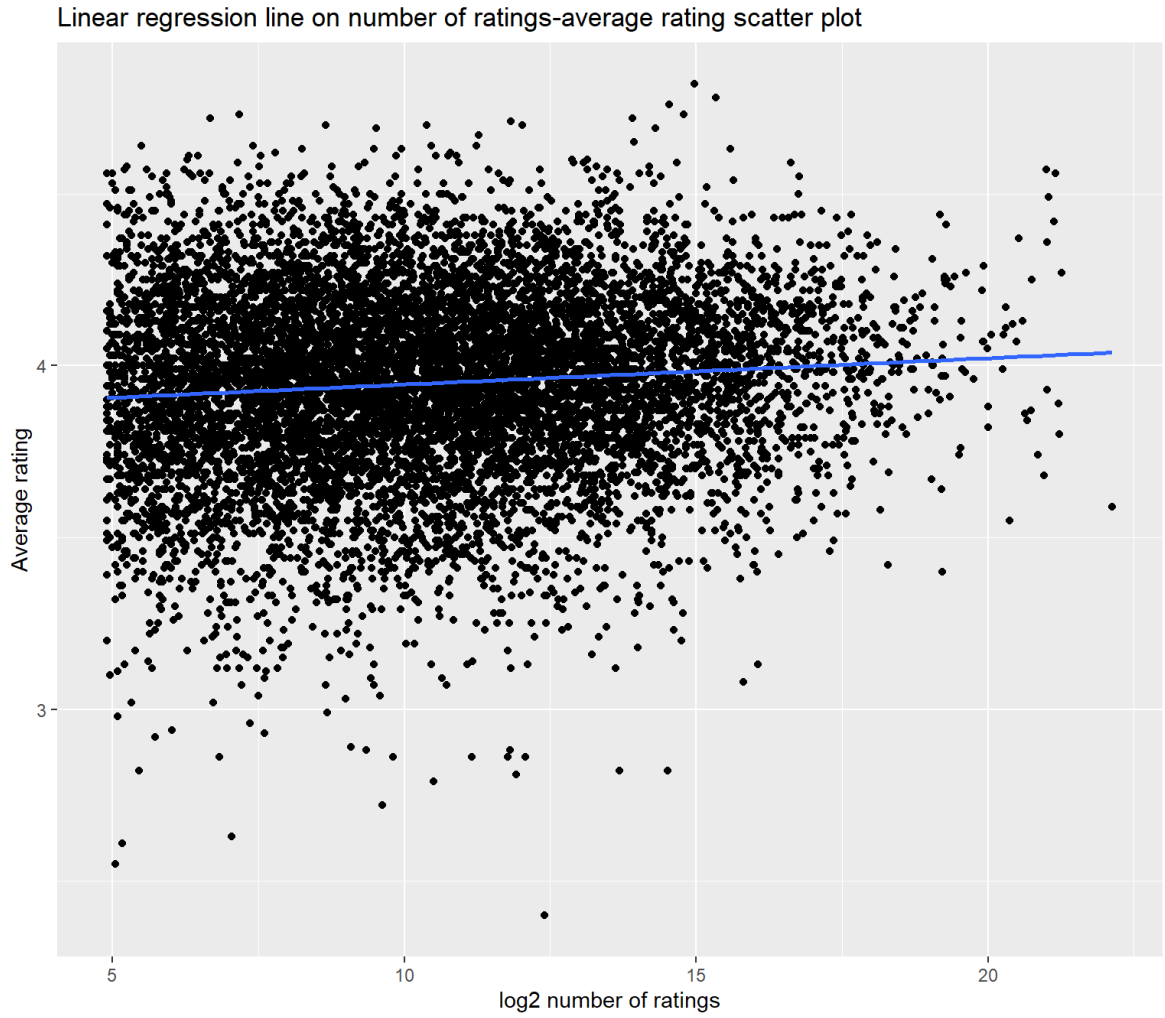
Note: Observations with less than 30 ratings have been removed, hence the slightly bigger skew to the left on the second graph.

3.B.6. Number of ratings and average rating association

In order to check the association between the number of ratings the books have and their ratings, I will use two methods. The first method is using a graph, and the second one is using a simple linear regression, which we will apply at the modelling part.

```
avg_rating = data$average_rating
num_of_ratings <- log2(data$ratings_count)
ggplot(data, mapping = aes(x = num_of_ratings, y =  avg_rating)) +
  geom_point() + stat_smooth(method = "lm", se = FALSE) +
  labs(y = "Average rating", x = "log2 number of ratings", title = "Linear regression line on number of ratings-average rati
ng scatter plot")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



We can't determine visually whether the relationship between the two variables is clearly linear or not, hence we will turn to a linear regression model for help in the next part.

Note: Observations with less than 30 ratings have been removed, hence the sharp end on the left.

Part C | **Modelling**

In this part we will implement various statistical models and tools, such as linear regression, hypothesis testing etc. in order to find interesting statistical information hidden within our data.

3.C.1. Simple linear regression

As we mentioned, we will apply a simple linear regression in order to check the relationship between the `ratings_count` the books have and their `average_rating`.
We will use our normally scaled dataset (`norm_scaled_data`):

```
numer_vars <- c("weeks_since_publication", "title_len", "num_pages", "ratings_count", "text_reviews_count", "average_rating"
)
scaled_numer_vars <- c("scaled_weeks_since_publication", "scaled_title_len", "scaled_num_pages", "scaled_ratings_count", "sc
aled_text_reviews_count", "average_rating")

numerical_data <- data[, numer_vars]
norm_scaled_numer_data <- norm_scaled_data[, scaled_numer_vars]

simple_lin_reg_model <- lm(average_rating ~ scaled_ratings_count, norm_scaled_numer_data)

summary(simple_lin_reg_model)
```

```
##
## Call:
## lm(formula = average_rating ~ scaled_ratings_count, data = norm_scaled_numer_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.5474 -0.1669  0.0130  0.1831  0.8700
##
## Coefficients:
##                       Estimate Std. Error  t value             Pr(>|t|)
## (Intercept)          3.948909   0.002735 1443.834 < 0.0000000000000002 ***
## scaled_ratings_count 0.011835   0.002735    4.327            0.0000153 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2694 on 9697 degrees of freedom
## Multiple R-squared:  0.001927,   Adjusted R-squared:  0.001824
## F-statistic: 18.72 on 1 and 9697 DF,  p-value: 0.00001526
```

The equation of linear regression is:

$$\hat{Y_i} = 3.949 + 0.012 \cdot x_i^{SRC}$$

Where the abbreviation stands for the variable name.

We can consider a linear model to be statistically significant when both p-values are less than the pre-determined statistical significance level α = 0.05, which is in fact the case as we can see according to the significance stars.

We have clearly established that there is a statistically significant possitive correlation between the amount of ratings and the average score.

3.C.2. Multivariate regression

We would also like to apply multivariate regression model on all of our numerical variables:

```
multi_lin_reg_model <- lm(average_rating ~
                            scaled_weeks_since_publication +
                            scaled_title_len +
                            scaled_num_pages +
                            scaled_ratings_count +
                            scaled_text_reviews_count,
                          norm_scaled_numer_data)

summary(multi_lin_reg_model)
```

```
##
## Call:
## lm(formula = average_rating ~ scaled_weeks_since_publication +
##     scaled_title_len + scaled_num_pages + scaled_ratings_count +
##     scaled_text_reviews_count, data = norm_scaled_numer_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.48063 -0.15779  0.01164  0.17739  0.88421
##
## Coefficients:
##                                Estimate Std. Error  t value
## (Intercept)                    3.948909   0.002634 1499.132
## scaled_weeks_since_publication 0.018780   0.002643    7.104
## scaled_title_len               0.046991   0.002641   17.791
## scaled_num_pages               0.050303   0.002639   19.061
## scaled_ratings_count           0.011676   0.005262    2.219
## scaled_text_reviews_count      0.001406   0.005273    0.267
##                                            Pr(>|t|)
## (Intercept)                    < 0.0000000000000002 ***
## scaled_weeks_since_publication     0.00000000000129 ***
## scaled_title_len               < 0.0000000000000002 ***
## scaled_num_pages               < 0.0000000000000002 ***
## scaled_ratings_count                         0.0265 *
## scaled_text_reviews_count                    0.7897
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2594 on 9693 degrees of freedom
## Multiple R-squared:  0.07458,    Adjusted R-squared:  0.0741
## F-statistic: 156.2 on 5 and 9693 DF,  p-value: < 0.00000000000000022
```

The equation of multivariate linear regression is:

$$\hat{Y}_i = 3.949 + 0.019 \cdot x_i^{SWSP} + 0.047 \cdot x_i^{STL} + 0.05 \cdot x_i^{SNP} + 0.012 \cdot x_i^{SRC} + 0.001 \cdot x_i^{STRC}$$

Where the abbreviations stand for variable names.

We have clearly established that there is a statistically significant possitive correlation between

- The number of pages
- The book's title length
- The weeks passed since publication
- The ratings count

and the average score. Unsurprisingly, the `text_reviews_count` variable got such a small weight as it is highly correlated with the `ratings_count` variable (as we've seen on our correlation heatmap), hence has almost the same effect on the dependent variable `average_rating` , which renders it redundant.
The result also leads us to the conclusion that our decision to add the `title_len` variable has paid off as indicated by it's relatively large weight and high significance level in our model.

3.C.3. Models evaluation and comparison

We can use the `anova()` function to get additional information about our models — an **ANOVA** (**Analysis of Variance**) table:

```
anova(multi_lin_reg_model)
```

```
## Analysis of Variance Table
##
## Response: average_rating
##                                Df Sum Sq Mean Sq  F value
## scaled_weeks_since_publication    1   3.39  3.3923  50.4068
## scaled_title_len                  1  22.65 22.6527 336.6035
## scaled_num_pages                  1  24.92 24.9182 370.2676
## scaled_ratings_count              1   1.60  1.6039  23.8333
## scaled_text_reviews_count         1   0.00  0.0048   0.0712
## Residuals                      9693 652.32  0.0673
##                                            Pr(>F)
## scaled_weeks_since_publication     0.000000000001337 ***
## scaled_title_len                < 0.00000000000000022 ***
## scaled_num_pages                < 0.00000000000000022 ***
## scaled_ratings_count               0.000001067246731 ***
## scaled_text_reviews_count                  0.7897
## Residuals
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We can also use it to compare our models:

```
anova(simple_lin_reg_model, multi_lin_reg_model)
```

```
## Analysis of Variance Table
##
## Model 1: average_rating ~ scaled_ratings_count
## Model 2: average_rating ~ scaled_weeks_since_publication + scaled_title_len +
##     scaled_num_pages + scaled_ratings_count + scaled_text_reviews_count
##   Res.Df    RSS Df Sum of Sq      F                Pr(>F)
## 1   9697 703.53
## 2   9693 652.32  4    51.213 190.25 < 0.00000000000000022 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

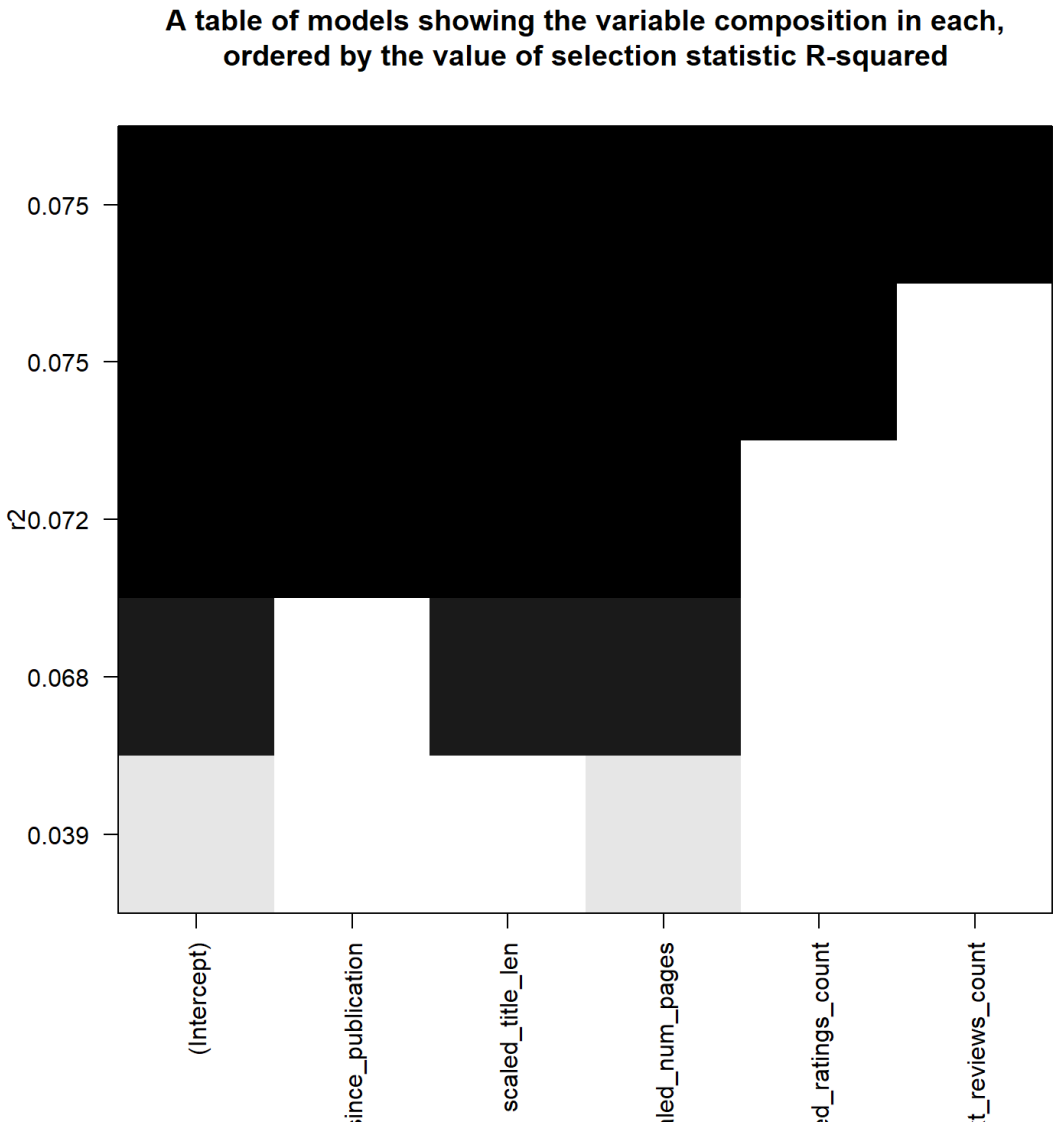
3.C.4. Variable selection using all-subsets regression

We've just decided which model to use and start to have a sense of the importance of each variable thanks to our ANOVA tables. Although this step usually belongs to the pre-processing phase, we will turn to it now as we wanted to first model using our "original" desired variables. We can perform variable selection by, for example, using all-subsets regression:

```
attach(norm_scaled_numer_data)

leaps <- regsubsets(average_rating ~
                      scaled_weeks_since_publication +
                      scaled_title_len +
                      scaled_num_pages +
                      scaled_ratings_count +
                      scaled_text_reviews_count,
                    data = norm_scaled_numer_data,
                    nbest = 1,
                    method = "exhaustive")

plot(leaps, scale = "r2")
title("A table of models showing the variable composition in each,\nordered by the value of selection statistic R-squared")
```



As we can see, our model/feature selection table precisely represents our recent conclusion: the `text_reviews_count` variable is highly correlated with the `ratings_count` variable, hence has almost the same effect on the dependent variable `average_rating`, which renders it redundant.

3.C.5. Hypothesis testing

Our hypothesis is that the mean `average_rating` of titles whose publication dates are **earlier** than the *goodreads* website launch date (January 2006) is different than of those whose publication dates are post-launch.

- $H_0 : \mu_{before} = \mu_{after}$
- $H_1 : \mu_{before} \neq \mu_{after}$
- $\alpha = 0.05$
- **Test statistic**: $t = \frac{\bar{X}_{before} - \bar{X}_{after}}{\sqrt{\frac{s^2_{before}}{n_{before}} + \frac{s^2_{after}}{n_{after}}}}$

  where:
  - $\bar{X}_{before}$ and $\bar{X}_{after}$ represent the mean values of the groups, respectively.
  - n_{before} and n_{after} represent the sizes of the group A and B, respectively.
  - S_{before} and S_{after} represent the standard deviations of the groups, respectively.
- The test's degrees of freedom are estimated as: $df = \frac{\frac{s^2_{before}}{n_{before}} + \frac{s^2_{after}}{n^2_{after}}}{\frac{s^4_{before}}{n^2_{before}(n_{after}-1)} + \frac{s^4_{after}}{n^2_{after}(n_{after}-1)}}$

Now, let's perform the test:

```
update_launch_diff <- as.numeric(as.Date("09/03/2020", format = "%d/%m/%Y") - as.Date("01/01/2006", format = "%d/%m/%Y")) /
7

weeks_since_publication <- data$weeks_since_publication

ratings_before_launch <- data[weeks_since_publication > update_launch_diff, "average_rating"]
ratings_after_launch <- data[weeks_since_publication <= update_launch_diff, "average_rating"]

t.test(x = ratings_before_launch,
       y = ratings_after_launch,
       alternative = "two.sided",
       var.equal = FALSE)
```

```
##
##  Welch Two Sample t-test
##
## data:  ratings_before_launch and ratings_after_launch
## t = 3.4616, df = 2416.4, p-value = 0.0005464
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.01122386 0.04055760
## sample estimates:
## mean of x mean of y
##  3.953514  3.927623
```

In other words, the alternative hypothesis $H_1 : \mu_{before} - \mu_{after} \neq 0$ is **true** with a p-value of 0.000009266 < α = 0.05, showing that the `average_rating` means of books published before and after the *goodreads* launch are in fact different, which might suggest that titles without a pre-existing status quo around them are rated more objectively, meaning that thanks to the website, readers are exposed to a more accurate and less biased rating of new publications.

Phase 4 | **Communicate**

In conclusion:

Our goal was to extract as much interesting statistical information as possible, while putting an emphasis on quality by:

- Maintaining a strict, structured and professional data science workflow.
- Focusing on clean data via careful management, maintanance and processing.

In order to reach this goal, we have thoroughly and methodically cleaned up and pre-processed our data, used various visualization techniques to obtain a better understanding and interesting insights about the data, and implemented various statistical models and tools, such as linear regression and hypothesis testing, in order to deduce conclusions about the relations between different variables and their effects on the average score.

Finally, our analysis has kindly invited, and left a room for, further research in various aspects related to the publications' ratings. We acknowledge the fact that our research has its limitations, but we also sincerely hope it could be of use for others.