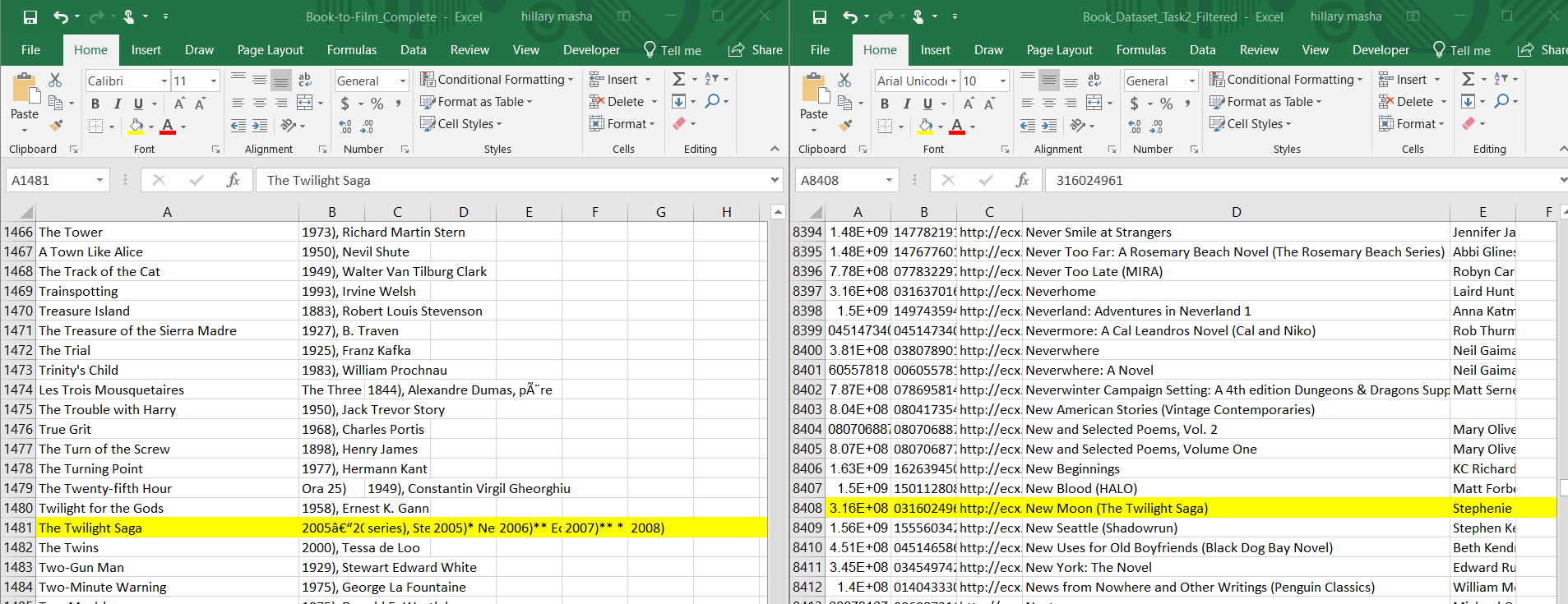
**Data retrieval and Manipulation**

Initial Process

Using wiki\_scraper.py lists of all books that were made into feature films(included list of movies). The information from the wikipedia page was mined with the help of a wikipedia library that specializes in finding specific wikipedia pages by their titles, and parsing the HTML from the page. The library BeautifulSoup was then used in order to read from the HTML and gather each entry from the tables in which the information was organized. Then the parsed data was written into a csv file and further formatted into a table.This list was filtered out to only include books that are in English and that were fiction, this was a comprehensive list of 1600 books.

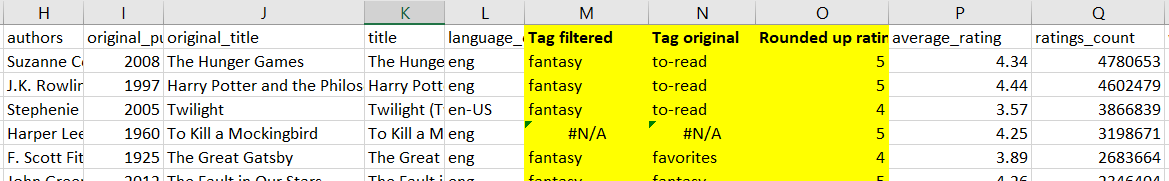
We found a data set of amazon books(17,670) that contained a list of all books[insert url], their genre, ASIN(amazon’s identification code) and authors.

We then attempted to classify the book titles in the list by determining if the book title belonged in the Wikipedia books-to-movies list. After running several R algorithms, found in Explanatory\_class.R ,and also attempting VLOOKUP’s to find exact matches, this failed due to the disparity in the naming convention of between the two datasets, as seen below, when searching for ‘Twilight’ by Stephanie Myers.



The next step was to find datasets for the same data source. We were able to find a file(from a github repository) with dataset of 10,000 goodreads books. [insert url] that contained a list of books there goodreads id, author, ratings, count of users who rated the books and many more details. This dataset also contained a file called books\_tag.csv that mapped every book id to tag id’s that the users had assigned and the number of users that tagged it with the specific tag, it was ordered from highest to lowest count of tag id’s. There also contained a dataset of all tag id’s and their corresponding tag names.

After mapping the tag names to the tag id’s and the tag id’s to the books, it was found that the tag with the highest number of users assignment did not always correspond to the genre of the book, i.e the tag ‘to-read’. The next step was to create a list of only tags that we care about, i.e, fiction, fantasy, thriller etc, and tags that looked similar to it, i.e fiction-thriller, fictional-book, etc. After these tags were applied to the books list , there were some books that did not have a tag, and it was also discovered that some of the tags did not match the books as expected, as seen below i.e, the great gatsby is not fantasy as the filtered tag indicated.



We decided to leave out the tags/genres for further investigation.

**Merging Data**

The next step involved getting books that were made into movies from good reads, this was done by scraping lists from their page that references books being movies, i.e ‘The book was better than the movie’, ‘I only watched the movie and not the book’, ‘Books turning into movies in 2018’, etc, this was scraped using a chrome extension called ‘data miner’ that retrieved the book titles. These lists were then converged into one huge list of unique titles. The tag ‘movie’ was also searched for in the tags.csv file and mapped to the associated books, this was added to the list. Also, it was identified that Wikipedia's book information could be parsed into the simplest title form, from "The Secret Life of Bees (2002), Sue Monk Kidd" to "The Secret Life of Bees" and using the VBA "instr" function, could be matched with the data from Goodreads, which organizes the information as: "Sue Monk Kidd 2001 The Secret Life of Bees". Through this process, ~200 books from the Wikipedia list was matched to Goodreads and was added onto the list of "Books to Movies" that was used. This became our list of books turned into movies. Using Explanatory\_class.R we assigned class variable ‘yes’ to books found in this list and ‘no’ to all other books.

**Explanatory Data Analysis**

We analyzed how the authors and number of books on the list varied. It was observed as seen below that most of the authors had less than ten books on the list and there were very few authors who had more than ten books as compared to the entire list of authors. This better helps us understand the authors who dominate the list to better interpret our results.

**Analyzing Average Ratings to number of books**

**Scatter Plot**

To better understand how users, rank books, we created a histogram of the ratings mapped to the number of books assigned that rating. It was discovered that this list does not contain ratings that are less than ‘2.47’. There were 185 unique review values i.e, ‘4.25, 3.87’, etc. It was also observed that there were clusters of users around certain numbers, this led us to believe that using all the ratings as is would result in overfitting, further analysis was done as seen below.

**Pareto Histogram**

Upon selecting the data to be made into a histogram, excel suggested the chart below which plots the distribution of data in descending order of frequency and with a cumulative line on the secondary axis as a percentage of the total.

