DS 4300: Book Recommendation Engine

Sara Adra, Anika Das, Mirah Gordon, Genny Jawor

This notebook is meant to serve as the home of all data loading and cleaning from the raw book, author, and review json files.

The code here includes:

- loading in data to mongo and filtering data based on set parameters (average rating, number of ratings, and number of text reviews)
  - this was done exclusively in a local connection to mongo
- connecting to the mongo client
- cleaning the book data
- finding the top reviews for each book
- finding the author name for each book
- running sentiment analysis on reviews to find the most common words
- exporting the data needed to create neo4j graph

Loading Data

In [1]:

```
# import pymonogo to connect to the database and our collections
import pymongo
from pymongo import MongoClient
import pprint
import operator
# create a mongo client
client = MongoClient()
# connect the client to the local host
client = MongoClient('localhost', 27017)
```

List of Mongo Commands Used on Home Computer

These commands were run through the mongodb shell on a local computer instead of through pymongo to increase the efficiency of the command.

Loading Data Into Mongo Collections

- Reviews
  - /Users/mirahgordon/documents/MongoDB/bin/mongodb-tools/bin/mongoimport --db demo --collection reviews --file $HOME/data/reviews/goodreads_reviews.json
- Books
  - /Users/mirahgordon/documents/MongoDB/bin/mongodb-tools/bin/mongoimport --db demo --collection books --file $HOME/data/books/goodreads_books.json
- Authors

    - /Users/mirahgordon/documents/MongoDB/bin/mongodb-tools/bin/mongoimport --db demo --collection authors --file $HOME/data/authors/goodreads_book_authors.json

Deleting Unnecessary Fields From Collections

- Books
    - Fields to delete:
        - asin, country_code, edition_information, format, image_url, is_ebook, isbn, isbn13, kindle_asin, language_code, link, publication_day, publication_month, series, similar_books, url, work_id
    - Command:
        - db.books.updateMany({}, { $unset: {asin: "", country_code: "", edition_information: "", format: "", image_url: "", is_ebook: "", isbn: "", isbn13: "", kindle_asin: "", link: "", publication_day: "", publication_month: "", series: "", similar_books: "", url: "", work_id: "" }})
- Reviews
    - Fields to delete:
        - date_added, date_updated, read_at, started_at, user_id
    - Command:
        - db.reviews.updateMany({}, { $unset: {date_added: "", date_updated: "", read_at: "", started_at: "", user_id: "" }})

Filtering Books by Average Rating

- db.books.remove({ average_rating: { $lt: '4.00' } })

Filtering Books by Number of Text Reviews

- db.books.remove({ text_reviews_count: { $lt: '5000.0' } })

Filtering Books by Number of Ratings

- db.books.remove({ ratings_count: { $lt: '5000.0' } })

Create New Reviews Index using book_id

- db.reviews.createIndex({ 'book_id': -1 })

Create New Authors Index using author_id

- db.authors.createIndex({ 'author_id': -1 })

Setting up Mongo Client

In [2]:

```
# find and use the demo database
db = client.demo
# find and use the books collection
books = db.books
```

```
# find and use the authors collection
authors = db.authors
# find and use the reviews collection
reviews = db.reviews
```

In [3]:

```
# test to ensure the server is connected and will print out the first document i
for book in books.find().limit(2):
    pprint.pprint(book)
```

```
{'_id': ObjectId('6261cc491f034259ef58d8df'),
 'authors': [{'author_id': '4862', 'role': ''}],
 'average_rating': '4.26',
 'book_id': '89376',
 'description': 'What is Heaven really going to be like? What will we look '
                "like? What will we do? Won't Heaven get boring after a "
                'while?\n'
                'We all have questions about what Heaven will be like, and '
                'after 25 years of extensive research, Dr. Randy Alcorn has '
                'the answers.\n'
                'In the most comprehensive and definitive book on Heaven to '
                'date, Randy invites you to picture Heaven the way Scripture '
                'describes it-- a bright, vibrant, and physical New Earth, '
                'free from sin, suffering, and death, and brimming with '
                "Christ's presence, wondrous natural beauty, and the richness "
                'of human culture as God intended it.\n'
                'God has put eternity in our hearts. Now, Randy Alcorn brings '
                'eternity to light in a way that will surprise you, spark your '
                'imagination, and change how you live life today.\n'
                "If you've always thought of Heaven as a realm of disembodied "
                "spirits, clouds, and eternal harp strumming, you're in for a "
                'wonderful surprise.\n'
                'This is a book about real people with real bodies enjoying '
                'close relationships with God and each other, eating, '
                'drinking, working, playing, traveling, worshiping, and '
                'discovering on a New Earth. Earth as God created it. Earth as '
                'he intended it to be.\n'
                'And the next time you hear someone say, "We cant begin to '
                'imagine what Heaven will be like," you\'ll be able to tell '
                'them, "I can.""',
 'num_pages': '533',
 'popular_shelves': [{'count': '6393', 'name': 'to-read'},
                     {'count': '1206', 'name': 'currently-reading'},
                     {'count': '130', 'name': 'theology'},
                     {'count': '95', 'name': 'christian-life'},
                     {'count': '88', 'name': 'christian'},
                     {'count': '77', 'name': 'non-fiction'},
                     {'count': '59', 'name': 'christianity'},
                     {'count': '50', 'name': 'favorites'},
                     {'count': '44', 'name': 'religion'},
                     {'count': '40', 'name': 'christian-living'},
                     {'count': '36', 'name': 'faith'},
                     {'count': '34', 'name': 'nonfiction'},
                     {'count': '33', 'name': 'heaven'},
                     {'count': '31', 'name': 'christian-non-fiction'},
                     {'count': '29', 'name': 'owned'},
                     {'count': '26', 'name': 'spiritual'},
                     {'count': '25', 'name': 'eschatology'},
                     {'count': '19', 'name': 'christian-nonfiction'},
                     {'count': '18', 'name': 'kindle'},
                     {'count': '18', 'name': 'books-i-own'},
                     {'count': '16', 'name': 'bible-study'},
                     {'count': '14', 'name': 'spirituality'},
                     {'count': '13', 'name': 'inspirational'},
{'count': '12', 'name': 'default'},
{'count': '12', 'name': 'religious'},
{'count': '9', 'name': 'christian-theology'},
{'count': '9', 'name': 'christian-books'},
{'count': '9', 'name': 'my-library'},
{'count': '8', 'name': 'owned-books'},
{'count': '8', 'name': 'my-books'},
{'count': '7', 'name': 'library'},
{'count': '7', 'name': 'on-hold'},
{'count': '7', 'name': 'christian-reading'},
{'count': '6', 'name': 'spiritual-growth'},
{'count': '6', 'name': 'randy-alcorn'},
{'count': '6', 'name': 'did-not-finish'},
{'count': '6', 'name': 'never-finished'},
{'count': '6', 'name': 'audiobook'},
{'count': '5', 'name': 'wish-list'},
{'count': '5', 'name': 'ebooks'},
{'count': '4', 'name': 'to-read-faith'},
{'count': '4', 'name': 'bible'},
{'count': '4', 'name': 'audio-books'},
{'count': '4', 'name': 'to-read-christian'},
{'count': '4', 'name': 'own-it'},
{'count': '4', 'name': 'ministry'},
{'count': '4', 'name': 'church'},
{'count': '4', 'name': 'favorite-books'},
{'count': '4', 'name': 'kindle-books'},
{'count': '4', 'name': 'life'},
{'count': '4', 'name': 'books'},
{'count': '4', 'name': 'religon'},
{'count': '4', 'name': 'ebook'},
{'count': '4', 'name': 'christian-study'},
{'count': '4', 'name': 'audible'},
{'count': '4', 'name': 'christian-doctrine'},
{'count': '4', 'name': 'spiritual-formation'},
{'count': '4', 'name': 'personal-library'},
{'count': '4', 'name': 'partially-read'},
{'count': '3', 'name': 'afterlife'},
{'count': '3', 'name': 'shelfari-favorites'},
{'count': '3', 'name': 'shelfari-wishlist'},
{'count': '3', 'name': 'apologetics'},
{'count': '3', 'name': 'christian-thought'},
{'count': '3', 'name': 'christian-to-read'},
{'count': '3', 'name': 'hardback'},
{'count': '3', 'name': 'philosophy'},
{'count': '3', 'name': 'owned-to-read'},
{'count': '3', 'name': 'heaven-and-hell'},
{'count': '3', 'name': 'adult'},
{'count': '3', 'name': 'jesus'},
{'count': '3', 'name': 'recommended'},
{'count': '3', 'name': 'grief'},
{'count': '3', 'name': 'didn-t-finish'},
{'count': '3', 'name': 'living'},
{'count': '3', 'name': 'doctrine'},
{'count': '3', 'name': 'general'},
{'count': '3', 'name': 'pastoral'},
{'count': '3', 'name': 'christian-inspiration'},
{'count': '3', 'name': 'calibre'},
{'count': '3', 'name': 'on-my-shelf'},
{'count': '3', 'name': '2005'},
{'count': '2', 'name': 'tbr-nonfiction'},
{'count': '2', 'name': 'gave-up'},
{'count': '2', 'name': 'office'},
{'count': '2', 'name': 'c-life'},
{'count': '2', 'name': 'favorite'},
{'count': '2', 'name': 'ccc'},
                     {'count': '2', 'name': 'read-2016'},
                     {'count': '2', 'name': 'c-theology'},
                     {'count': '2', 'name': 'audio'},
                     {'count': '2', 'name': 'bookshelf'},
                     {'count': '2', 'name': 'systematic-theology'},
                     {'count': '2', 'name': 'eternity'},
                     {'count': '2', 'name': 'criticism'},
                     {'count': '2', 'name': 'american'},
                     {'count': '2', 'name': 'e-books'},
                     {'count': '2', 'name': '2016-to-read'},
                     {'count': '2', 'name': 'church-library'},
                     {'count': '2', 'name': 'read-2015'}],
 'publication_year': '',
 'publisher': '',
 'ratings_count': '7345',
 'text_reviews_count': '566',
 'title': 'Heaven',
 'title_without_series': 'Heaven'}
{'_id': ObjectId('6261cc491f034259ef58d8e9'),
 'authors': [{'author_id': '19158', 'role': ''}],
 'average_rating': '4.22',
 'book_id': '6066812',
 'description': "To Kara's astonishment, she discovers that a portal has "
                'opened in her bedroom closet and two goblins have fallen '
                'through! They refuse to return to the fairy realms and be '
                'drafted for an impending war. In an attempt to roust the '
                'pesky creatures, Kara falls through the portal, smack into '
                'the middle of a huge war. Kara meets Queen Selinda, who '
                'appoints Kara as a Fairy Princess and assigns her an '
                'impossible task: to put an end to the war using her '
                'diplomatic skills.\n'
                "All's Fairy In Love And War is the eighth book in Avalon: Web "
                'of Magic, a twelve-book fantasy series for middle grade '
                'readers. Through their magical journey, the teenage heroines '
                'discover who they really are . . . and run into plenty of '
                'good guys, bad guys, and cute guys. Out of print for two '
                'years, Seven Seas is pleased to return the Avalon series to '
                "print in editions targeted for today's readers, with new "
                'manga-style covers and interior illustrations.',
 'num_pages': '216',
 'popular_shelves': [{'count': '515', 'name': 'to-read'},
                     {'count': '25', 'name': 'fantasy'},
                     {'count': '11', 'name': 'owned'},
                     {'count': '11', 'name': 'books-i-own'},
                     {'count': '9', 'name': 'currently-reading'},
                     {'count': '9', 'name': 'favorites'},
                     {'count': '9', 'name': 'magic'},
                     {'count': '9', 'name': 'avalon'},
                     {'count': '8', 'name': 'young-adult'},
                     {'count': '6', 'name': 'series'},
                     {'count': '6', 'name': 'fiction'},
                     {'count': '5', 'name': 'books'},
                     {'count': '4', 'name': 'childrens-books'},
                     {'count': '4', 'name': 'owned-books'},
                     {'count': '4', 'name': 'adventure'},
                     {'count': '4', 'name': 'middle-grade'},
                     {'count': '4', 'name': 'children'},
                     {'count': '4', 'name': 'avalon-web-of-magic'},
                     {'count': '3', 'name': 'ya'},
                     {'count': '3', 'name': 'want'},
                     {'count': '3', 'name': 'teen'},
                     {'count': '3', 'name': 'faeries'},
                     {'count': '3', 'name': 'childrens'},
                     {'count': '2', 'name': 'children-s'},
                     {'count': '2', 'name': '4-stars'},
{'count': '2', 'name': 'library'},
{'count': '2', 'name': 'default'},
{'count': '2', 'name': 'dragons'},
{'count': '2', 'name': 'witches'},
{'count': '2', 'name': 'paranormal'},
{'count': '2', 'name': 'r'},
{'count': '2', 'name': 'childhood'},
{'count': '2', 'name': 'my-childhood'},
{'count': '2', 'name': 'children-s-books'},
{'count': '2', 'name': 'rachel-roberts'},
{'count': '2', 'name': 'fairies'},
{'count': '2', 'name': 'childhood-favorites'},
{'count': '2', 'name': 'kids-books'},
{'count': '2', 'name': 'kids'},
{'count': '2', 'name': 'my-books'},
{'count': '2', 'name': 'avalon--web-of-magic'},
{'count': '2', 'name': 'read-in-2009'},
{'count': '2', 'name': 'books-i-have'},
{'count': '2', 'name': 'to-buy'},
{'count': '1', 'name': 'own-own-unread'},
{'count': '1', 'name': 'own-own'},
{'count': '1', 'name': 'nostalgia'},
{'count': '1', 'name': 'kid-s-books'},
{'count': '1', 'name': 'america-and-canada'},
{'count': '1', 'name': 'low-fantasy'},
{'count': '1', 'name': '5th'},
{'count': '1', 'name': 'tbr-series'},
{'count': '1', 'name': 'part-of-a-series'},
{'count': '1', 'name': 'fantasy-ya'},
{'count': '1', 'name': 'novel'},
{'count': '1', 'name': 'where-i-started'},
{'count': '1', 'name': 'kj2016-17-2nd-3rd-grade'},
{'count': '1', 'name': 'kids-books-tracking'},
{'count': '1', 'name': 'book-library-now'},
{'count': '1', 'name': 'faerie-tales'},
{'count': '1', 'name': 'need-to-buy'},
{'count': '1', 'name': 'reread-in-2015'},
{'count': '1', 'name': 'personal-shelf'},
{'count': '1', 'name': 'childhood-favourites'},
{'count': '1', 'name': 'all-time-favourites'},
{'count': '1', 'name': 'novels'},
{'count': '1', 'name': 'to-buy-middlegrade'},
{'count': '1', 'name': 'literature'},
{'count': '1', 'name': 'bought'},
{'count': '1', 'name': 'bookshelf'},
{'count': '1', 'name': 'sci-fi-fantasy'},
{'count': '1', 'name': 'physical-copies'},
{'count': '1', 'name': 'childhood-books'},
{'count': '1', 'name': '0-books-i-own'},
{'count': '1', 'name': 'avalonwebofmagicseries'},
{'count': '1', 'name': 'shelfari-wishlist'},
{'count': '1', 'name': 'tbr-2014'},
{'count': '1', 'name': 'science-fiction-fantasy'},
{'count': '1', 'name': 'pc-100-199'},
{'count': '1', 'name': 'e-book'},
{'count': '1', 'name': 'c-purple'},
{'count': '1', 'name': 'author-r'},
{'count': '1', 'name': 'action-adventure'},
{'count': '1', 'name': 'a'},
{'count': '1', 'name': 'reg'},
{'count': '1', 'name': 'books-that-i-own'},
{'count': '1', 'name': 'middle-grade-lit'},
{'count': '1', 'name': 'child'},
{'count': '1', 'name': 'younger-reads'},
{'count': '1', 'name': 'fantasy-dystopians'},
                     {'count': '1', 'name': 'currently-have'},
                     {'count': '1', 'name': 'no-longer-have'},
                     {'count': '1', 'name': 'elements'},
                     {'count': '1', 'name': 'own-it'},
                     {'count': '1', 'name': 'currently-in-bookshelf'},
                     {'count': '1', 'name': 'my-favorites'},
                     {'count': '1', 'name': 'to-re-read'},
                     {'count': '1', 'name': 'not-available-on-kindle'},
                     {'count': '1', 'name': 'have-read'},
                     {'count': '1', 'name': 'childrens-and-middle-grade'}],
 'publication_year': '2009',
 'publisher': 'Seven Seas',
 'ratings_count': '98',
 'text_reviews_count': '6',
 'title': "All's Fairy in Love and War (Avalon: Web of Magic, #8)",
 'title_without_series': "All's Fairy in Love and War (Avalon: Web of Magic, "
                         '#8)'}
```

In [4]:

```python
# test to ensure the server is connected and will print out the first document i
for author in authors.find().limit(1):
    pprint.pprint(author)
```

```
{'_id': ObjectId('6261d1778687e60679f430fb'),
 'author_id': '3041852',
 'average_rating': '3.89',
 'name': 'Alfred J. Church',
 'ratings_count': '947',
 'text_reviews_count': '85'}
```

In [5]:

```python
# test to ensure the server is connected and will print out the first document i
for review in reviews.find().limit(1):
    pprint.pprint(review)
```

```
{'_id': ObjectId('6261ce995a5bd73d16c340b1'),
 'book_id': '1995421',
 'n_comments': 0,
 'n_votes': 4,
 'rating': 0,
 'review_id': '7350a30a2f5c785b190d9ebd1c0b4af9',
 'review_text': 'Kevin highly recommended on instagram'}
```

Cleaning Book Data

In [6]:

```python
# imports
import pandas as pd
import numpy as np
```

In [7]:

```python
# save the current list of book documents from mongo (using find command) to a d
book_df = pd.DataFrame(list(books.find()))
```

In [8]:

```python
# save book data frame to a csv
book_df.to_csv('books_clean.csv')

# open csv into data frame
book_df = pd.read_csv('books_clean.csv')
# drop the _id and duplicate index columns
```

```
book_df.drop(columns=['Unnamed: 0', '_id'], inplace=True)
book_df.head()
```

Out[8]:

	text_reviews_count	popular_shelves	average_rating	description	authors	publisher	nu
0	566	[{'count': '6393', 'name': 'to-read'}, {'count...	4.26	What is Heaven really going to be like? What w...	[{'author_id': '4862', 'role': ''}]	NaN	
1	6	[{'count': '515', 'name': 'to-read'}, {'count'...	4.22	To Kara's astonishment, she discovers that a p...	[{'author_id': '19158', 'role': ''}]	Seven Seas	
2	6	[{'count': '20', 'name': 'to-read'}, {'count':...	4.06	These are the stories that catapulted Superman...	[{'author_id': '81563', 'role': ''}, {'author_...	DC Comics	
3	6	[{'count': '735', 'name': 'to-read'}, {'count'...	4.02	O Alienista e uma celebre obra literaria do es...	[{'author_id': '22458', 'role': ''}, {'author_...	nshr lwH fkhr	
4	9	[{'count': '46', 'name': 'to-read'}, {'count':...	4.29	A killer by day and night, Zaire Pearson never...	[{'author_id': '4973079', 'role': ''}]	Jessica Watkins Presents	

In [9]:

```
# drop any rows that don't have a description or a number of pages
book_df['description'] = book_df['description'].replace('', np.nan)
book_df = book_df.dropna(axis=0, subset=['description', 'num_pages'])
```

In [10]:

```
# creates a df of 100 random books
books_100 = book_df.sample(n=100)

# creates a df of 10,000 books
books_10000 = book_df.sample(n=10000)
```

In [11]:

```
# list of author ids for 100 books -- only taking first author (in cases where t
books_100['authors'] = books_100['authors'].apply(lambda x: x.split("'")[3])
authorid_100 = books_100['authors'].tolist()

# list of author ids for 10,000 books -- only taking first author (in cases wher
books_10000['authors'] = books_10000['authors'].apply(lambda x: x.split("'")[3])
authorid_10000 = books_10000['authors'].tolist()

# list of book ids for 100 books
bookid_100 = books_100['book_id'].tolist()

# list of book ids for 10,000 books
bookid_10000 = books_10000['book_id'].tolist()
```

Retrieving Book Reviews

In [12]:

```python
# create dataframe to hold all the top reviews
top_reviews = pd.DataFrame()

# iterate through
for book in bookid_100:

    # mongo command to find the reviews with a specific book id
    book_reviews = reviews.find( { 'book_id' : str(book) }, { 'book_id':1, 'revi

    # collect all reviews as a list
    top5 = []

    for r in book_reviews:
        top5.append(r)

    # sort reviews by number of votes and keep the top 5
    sorted_by_votes = sorted(top5, key=lambda d: d['n_votes'])
    sorted_by_votes = sorted_by_votes[0:5]

    # append the top 5 reviews to the dataframe
    top_reviews = top_reviews.append(sorted_by_votes, ignore_index=True, sort=Fa
```

In [13]:

```python
# set the index as the book id
top_reviews.set_index('book_id', inplace=True)
# drop any row with an empty review
top_reviews['review_text'] = top_reviews['review_text'].replace('', np.nan)
top_reviews.dropna(axis=0, subset=['review_text'], inplace=True)
```

In [14]:

```python
top_reviews.head()
```

Out[14]:

	review_text	n_votes
book_id		
15759838	Life's Rhythms is full of beautiful poems. If ...	1
15759838	Life's Rhythms is a great selection of traditi...	1
18634307	lktb jyd f~ b$D \ln qT fymykhS <$ ml m` l Hyper...	0
18634307	Rollo is a manlet beta that doesn't lift but t...	0
18634307	Interesting book on gender communication and r...	0

In [15]:

```python
# save top 5 reviews for our 100 books to a csv
top_reviews.to_csv('top_reviews.csv')
```

Repeat the process for the full set of 10,000 books

In [16]:

```python
# create dataframe to hold all the top reviews
top_reviews_10000 = pd.DataFrame()

for book in bookid_10000:
    # mongo command to find the reviews with a specific book id
    book_reviews = reviews.find( { 'book_id' : str(book) }, { 'book_id':1, 'revi

    # collect all reviews as a list
    top5 = []

    for r in book_reviews:
        top5.append(r)

    # sort reviews by number of votes and keep the top 5
    sorted_by_votes = sorted(top5, key=lambda d: d['n_votes'])
    sorted_by_votes = sorted_by_votes[0:5]

    # append the top 5 reviews to the dataframe
    top_reviews_10000 = top_reviews_10000.append(sorted_by_votes, ignore_index=T
```

In [17]:

```
# set the index as the book id
top_reviews_10000.set_index('book_id', inplace=True)
# drop any row with an empty review
top_reviews_10000['review_text'] = top_reviews_10000['review_text'].replace('',
top_reviews_10000.dropna(axis=0, subset=['review_text'], inplace=True)
```

In [18]:

```
top_reviews_10000.head()
```

Out[18]:

	review_text	n_votes
book_id		
823653	First book in the Camel Club. Composed of elde...	0
823653	The Camel Club is a group of older men who are...	0
823653	I really liked this book just because there we...	0
823653	It could happen ... conspiracy theorists unite...	0
823653	SM	0

In [19]:

```
# save top 5 reviews for all 10,000 books to a csv
top_reviews_10000.to_csv('top_reviews_10000.csv')
```

Finding Book Author

In [20]:

```
# empty list to hold all author names
author_names = []

# find the name of the author for each book
for aid in authorid_100:
    # mongo command to find the author with a specific author id
    author = authors.find( { 'author_id': str(aid) }, { 'name':1, '_id':0 } )

    for a in author:
        name = a.get('name')
    author_names.append(name)

# create new author name column
books_100['author_name'] = author_names
```

In [21]:

```
# set the index to be the title of the book
books_100.set_index('title', inplace=True)
# drop unused columns of popular shelves and title without series
books_100.drop(columns=['popular_shelves', 'title_without_series'], inplace=True
```

In [22]:

```
"""
function to define the length of a given book
"""
def book_size(row):
    if row['num_pages'] <= 299:
        return "small"
    elif 300 <= row['num_pages'] <= 599:
        return "medium"
    elif row['num_pages'] > 600:
        return "large"
```

In [23]:

```
# apply book size function to book dataframe
books_100['book_size'] = books_100.apply(book_size, axis=1)
```

In [24]:

```
books_100.head()
```

Out[24]:

	text_reviews_count	average_rating	description	authors	publisher	num_pages
title						
Life's Rhythms	6	4.00	Ancient Japanese poetry with a modern twist. O...	4788773	Vickie Johnstone	116.0
The Rational Male	75	4.32	The Rational Maleis a rational and pragmatic a...	7328259	CreateSpace	300.0
The Tao Te Ching: 81 Verses by Lao Tzu with Introduction and Commentary	7	4.32	Tao Te Ching translates very roughly as "the w...	2622245	Watkins Publishing	224.0
Ex-Treme Measures	8	4.33	Men: You can't live with them, you can't kill ...	5828647	The Wild Rose Press	228.0

	text_reviews_count	average_rating	description	authors	publisher	num_pages
title						
Their Souls Met in Wishton	7	4.64	When the universe sends Liam Kincaid back into...	15386334	Solstice Publishing	202.(

In [25]:

```
# save the most cleaned dataframe to a csv to use with neo4j
books_100.to_csv('books_100.csv')
```

Repeat the process for the full set of 10,000 books

In [26]:

```
# empty list to hold all author names
author_names = []

# find the name of the author for each book
for aid in authorid_10000:
    # mongo command to find the author with a specific author id
    author = authors.find( { 'author_id': str(aid) }, { 'name':1, '_id':0 } )

    for a in author:
        name = a.get('name')

    author_names.append(name)

# create new author name column
books_10000['author_name'] = author_names
```

In [27]:

```
# set the index to be the title of the book
books_10000.set_index('title', inplace=True)
# drop unused columns of popular shelves and title without series
books_10000.drop(columns=['popular_shelves', 'title_without_series'], inplace=Tr
```

In [28]:

```
# apply book size function to book dataframe
books_10000['book_size'] = books_10000.apply(book_size, axis=1)
```

In [29]:

```
books_10000.head()
```

Out[29]:

	text_reviews_count	average_rating	description	authors	publisher	num_pa
title						
The Camel Club (Camel Club, #1)	66	4.02	It exists at the fringes of Washington, D.C., ...	9291	NaN	4

	text_reviews_count	average_rating	description	authors	publisher	num_pa
title						
Silver Dawn (Wishes, #4.5)	52	4.53	*Silver Dawn - book 4.5 of the Wishes Series*\...	6935697	G.J. Walker-Smith	2
Resurrections (Thin Ice #4)	8	4.69	Life sometimes takes strange twists, for seemi...	3297163	Kirabaco Publishing	2
La Rosa e il Deserto	6	4.11	Marylya, bella e vivace principessa del Regno ...	14370629	CreateSpace	4
Il principe (Shadowhunters - Le origini, #2)	63	4.46	In una Londra vittoriana fosca e inquietante, ...	150038	Mondadori	5

In [30]:

```
# save the most cleaned dataframe to a csv to use with neo4j
books_10000.to_csv('books_10000.csv')
```

Finding Common Words

In [31]:

```
# imports for stop words and counter
from collections import Counter
import nltk
from nltk.corpus import stopwords
nltk.download('stopwords')
set_stopwords = set(stopwords.words('english'))
set_stopwords.update(['book', 'books', 'author', 'story', 'read', "i've"])
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     /Users/mirahgordon/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

In [32]:

```
# compile all reviews for 100 books
reviews_list = top_reviews['review_text'].tolist()
```

In [33]:

```
"""
function to clean the given word (by making it all lower case
letters + removing any trailing punctuation if present)
"""
def clean_word(word):
    # make the word all lowercase letters
    cleaned_word = word.lower()
    # check if the last character in the word is a letter or not
    # remove last character if not a letter
    # (ex. ',' '.' '-' ' ')
    if (cleaned_word[-1].isalpha() == False):
         cleaned_word = cleaned_word[:-1]

    return cleaned_word
```

In [34]:

```
most_freq_word_column_labels = []
for idx in range(1, 11):
    most_freq_word_column_labels.append('most_freq_word_' + str(idx))

most_freq_word_column_labels
```

Out[34]:

```
['most_freq_word_1',
 'most_freq_word_2',
 'most_freq_word_3',
 'most_freq_word_4',
 'most_freq_word_5',
 'most_freq_word_6',
 'most_freq_word_7',
 'most_freq_word_8',
 'most_freq_word_9',
 'most_freq_word_10']
```

In [35]:

```
most_freq_words_all_books = dict()
most_freq_words_all_books_df = pd.DataFrame(columns=most_freq_word_column_labels

for book_id in top_reviews.index.unique():
    total_count_Counter = Counter()
    for index, review in top_reviews[top_reviews.index == book_id].iterrows():
        words_in_review_list = str(review['review_text']).split(' ')
        cleaned_words_in_review = filter(lambda word: (word not in set_stopwords
        cleaned_review_word_count_Counter = Counter(cleaned_words_in_review)
        total_count_Counter = Counter(total_count_Counter) + Counter(cleaned_rev

    most_freq_words = total_count_Counter.most_common(10)
    most_freq_words_all_books[book_id] = most_freq_words

    most_freq_words_list = [word_to_freq[0] for word_to_freq in most_freq_words]
    while len(most_freq_words_list) < 10: most_freq_words_list.append(np.nan)

    most_freq_words_all_books_df.loc[book_id] = most_freq_words_list

most_freq_words_all_books_df[:10]
```

Out[35]:

	most_freq_word_1	most_freq_word_2	most_freq_word_3	most_freq_word_4	most_fr
15759838	poems	life's	rhythms	review	
18634307	beta	lift	one	rollo	
31921810	commentary	ancient	text	modern	
26345942	novel	good	vanna	cheating	
17347049	end	able	spoilers	finally	

	most_freq_word_1	most_freq_word_2	most_freq_word_3	most_freq_word_4	most_fr
581607	writing	i'm	entries	actual	
25026385	und	der	das	die	
26237590	cady	mystery	really	like	
9699781	di	che	e	il	
10900312	198/8th/2	NaN	NaN	NaN	

In [36]:

```
set_freq_words = set()
for col in most_freq_word_column_labels:
    set_freq_words = set_freq_words.union(set(most_freq_words_all_books_df[col].


all_most_freq_words_df = pd.DataFrame(list(set_freq_words), columns=['most_freq_
all_most_freq_words_df.dropna(inplace=True)
all_most_freq_words_df
```

Out[36]:

	most_freq_words
1	dansk
2	austen's
3	four
4	tv
5	ich
...	...
673	reviews
674	lewis
675	sad
676	overstretched
677	folg

677 rows × 1 columns

In [37]:

```
# save the most frequent words as a csv
all_most_freq_words_df.to_csv('all_most_freq_words_df.csv')
```

Repeat process for the full set of 10,000 books

In [38]:

```
# compile all reviews for 10,000 books
reviews_list = top_reviews_10000['review_text'].tolist()
```

In [39]:

```
most_freq_word_column_labels = []
for idx in range(1, 11):
    most_freq_word_column_labels.append('most_freq_word_' + str(idx))
```

```
most_freq_word_column_labels
```

Out[39]:

```
['most_freq_word_1',
 'most_freq_word_2',
 'most_freq_word_3',
 'most_freq_word_4',
 'most_freq_word_5',
 'most_freq_word_6',
 'most_freq_word_7',
 'most_freq_word_8',
 'most_freq_word_9',
 'most_freq_word_10']
```

In [40]:

```
most_freq_words_all_books_10000 = dict()
most_freq_words_all_books_df_10000 = pd.DataFrame(columns=most_freq_word_column_

for book_id in top_reviews_10000.index.unique():
    total_count_Counter = Counter()
    for index, review in top_reviews_10000[top_reviews_10000.index == book_id].i
        words_in_review_list = str(review['review_text']).split(' ')
        cleaned_words_in_review = filter(lambda word: (word not in set_stopwords
        cleaned_review_word_count_Counter = Counter(cleaned_words_in_review)
        total_count_Counter = Counter(total_count_Counter) + Counter(cleaned_rev

    most_freq_words = total_count_Counter.most_common(10)
    most_freq_words_all_books_10000[book_id] = most_freq_words

    most_freq_words_list = [word_to_freq[0] for word_to_freq in most_freq_words]
    while len(most_freq_words_list) < 10: most_freq_words_list.append(np.nan)

    most_freq_words_all_books_df_10000.loc[book_id] = most_freq_words_list

most_freq_words_all_books_df_10000[:10]
```

Out[40]:

	most_freq_word_1	most_freq_word_2	most_freq_word_3	most_freq_word_4	most_fr
823653	club	camel	series	first	
22046993	novella	well	alex	one	
17315391	e	di	che	il	
14271	stories	writing	one	back	
714382	techniques	trims	section	nice	
1281730	little	journal	artists	well	
26815133	love	reading	erotic	one	
8575158	one	slight	improvement	nice	
22624779	mother	love	elisa	life	
1217489	certainly	always	loved	reading	

In [41]:

```
set_freq_words = set()
for col in most_freq_word_column_labels:
    set_freq_words = set_freq_words.union(set(most_freq_words_all_books_df_10000
all_most_freq_words_df_10000 = pd.DataFrame(list(set_freq_words), columns=['most
all_most_freq_words_df_10000.dropna(inplace=True)
all_most_freq_words_df_10000
```

Out[41]:

	most_freq_words
1	wright's
2	bowl
3	cantra
4	tudo
5	rinpoche
...	...
18064	rebuttal
18065	oldalnal
18066	polow
18067	deanna
18068	gue

18068 rows × 1 columns

In [42]:

```
# save the most frequent words as a csv
all_most_freq_words_df_10000.to_csv('all_most_freq_words_df_10000.csv')
```