Shao Ru Zhang
CIS 3120- Programming for Analytics
Project #2
25th May 2021

In this project, my goal was to scrape the data from a wikipedia table and utilize the NY Times API to recommend books to readers who may be interested in an author's other best-seller books. I scraped a table from "The New York Times Fiction Best Sellers of 2020" wikipedia page. The table consists of a list of adult fiction books that are in the combined print & e-book fiction category. Using beautifulsoup, I was able to find the table needed and get data from the three columns available: dates, titles, and authors. After scraping this data, I converted into a dataframe called DF1. I also created a new author's list since some of the authors repeat, and we only want a unique author's name for later on.

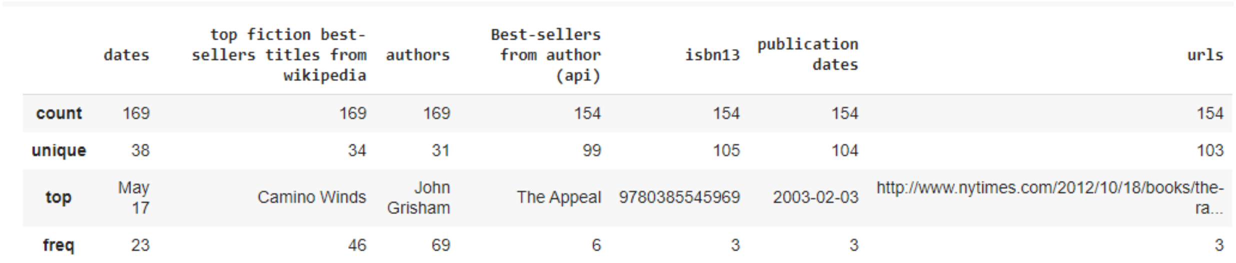
Then, using The New York Times api, I made a request for each of the author's names. The results return a list of the author's best-seller books. The data retrieved from the API included book title, author, isbn13, publication date, and url. The data was then used to create another dataframe called DF2. Finally, both dataframe were merged together using left merge on author's names. The end-result of the dataframe shows all the rows from the wikipedia, including some repetition in order to keep the names of the author consistent for the result from the API. Furthermore, some of the wikipedia authors had no other NY Times best-sellers books so the resulting row shows a "NaN". I kept "NaN" in this case in order to keep the reader informed.

Finally, I wanted to create a bar graph of each author and their number of NY Times best-seller books. I started off by creating two lists. One list included all authors whose name appeared in DF2, which is the NY Times API, and found how many times their name appeared and put them into a list. Then, the author whose name does not appear go onto another list. In order to stay consistent with the numbers returned and the author's names, I had to sort both the value and the author's name. Then, I added the string lists together and transformed the values and the author's into a dataframe. I had to fill the remaining authors who had no value with 0's, and finally create a bar chart.

The statistical summary below only shows the categorical perspective since I have no numerical columns. It shows the count for each column, the API count will be less than the wikipedia count since "NaN" values are not factored in. The "unique" shows how many unique rows there are. The wikipedia columns will have a lot of repetitive rows in order to match the author's name with other books from that author using the API. The "top" summary statistics shows the row with the highest repetition of author, John Grisham, which is consistent with the bar graph below. The "frequency" shows the frequency of the most common value, which is John Grisham.

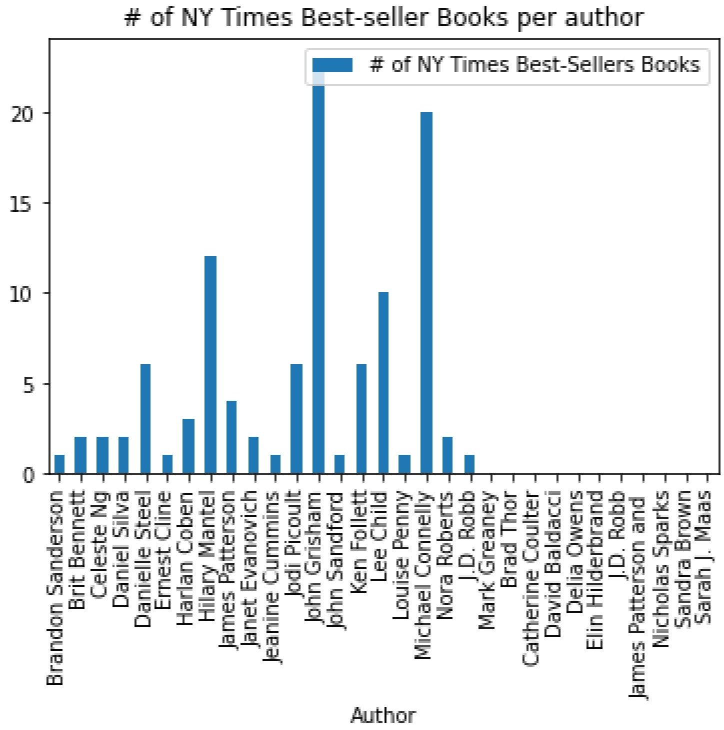
Using the pandas dataframe, statistical summary, and graph, people can use this information to determine best-seller fiction books they may want to read during their free time. They can utilize this information to make a decision based on the top ranking best-seller fiction book during various times throughout the year of 2020, or look into other best-selling books by that same author. Readers may also utilize the bar chart to decide which author's books they may want to read based on their number of NY Times best-sellers.

Statistical Summary:



	dates	top fiction best-sellers titles from wikipedia	authors	Best-sellers from author (api)	isbn13	publication dates	urls
count	169	169	169	154	154	154	154
unique	38	34	31	99	105	104	103
top	May 17	Camino Winds	John Grisham	The Appeal	9780385545969	2003-02-03	http://www.nytimes.com/2012/10/18/books/the-ra...
freq	23	46	69	6	3	3	3

Bar Graph:


# of NY Times Best-seller Books per author
# of NY Times Best-Sellers Books
Brandon Sanderson
Brit Bennett
Celeste Ng
Daniel Silva
Danielle Steel
Ernest Cline
Harlan Coben
Hilary Mantel
James Patterson
Janet Evanovich
Jeanine Cummins
Jodi Picoult
John Grisham
John Sandford
Ken Follett
Lee Child
Louise Penny
Michael Connelly
Nora Roberts
J.D. Robb
Mark Greaney
Brad Thor
Catherine Coulter
David Baldacci
Delia Owens
Elin Hilderbrand
J.D. Robb
James Patterson and
Nicholas Sparks
Sandra Brown
Sarah J. Maas
Author