Data Narrative

I. Overview of the Dataset

The Goodbooks-10k dataset consists of information of 10,000 books, with ratings, book metadata and book tags. It is available on GitHub and created by Zygmunt Zajac.

The dataset is presented in Comma-Separated Values (CSV) format, which facilitates a wide range of data analysis tools. It is organized to provide contents like book marked to read by the users, author, publication year, genre and six million ratings by users.

This dataset will be useful to explore various aspects related to publishing industry, book classification, etc. As this dataset is about 10,000 books, which may not provide information about all the books.

Overall, Goodbooks-10k is a good source for the book readers.

II. Scientific Questions / Hypotheses

1. What is the distribution of books across different languages, and which language has the highest number of books published?
2. What are the top five most frequently used book tags?
3. Which user has read maximum number of books?
4. What is the distribution of books published by year, and in which year were the most books published?
5. Hypothesis: Books with a higher number of ratings will have a lower average rating than those with a lower number of ratings.

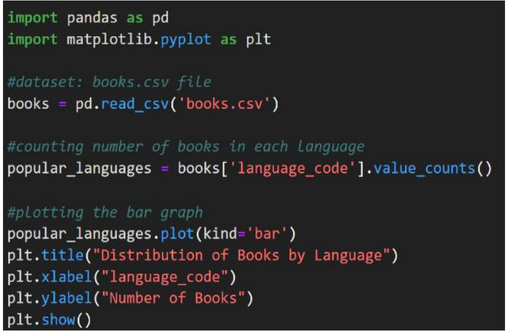
III. Details of Libraries and Functions

- pandas [Library]: Pandas is a Python library used for working with data sets. It has functions for analysing, cleaning, exploring, and manipulating data.[1]
- read_csv: Read a comma-separated values (csv) file into DataFrame.[2]
- value_counts: Return a Series containing counts of unique values.[2]
- len: Finds length.
- head: Return the first *n* rows.[2]
- append: Will add new element in the list.
- iloc: Purely integer-location based indexing for selection by position.[2]
- merge: Merge DataFrame or named Series objects with a database-style join.[2]
- groupby: Group DataFrame using a mapper or by a Series of columns.[2]
- size: Return an int representing the number of elements in this object.[2]
- reset_index: Reset the index, or a level of it.[2]
- sort_values: Sort by the values along either axis.[2]
- max: Return the maximum of the values over the requested axis.[2]
- matplotlib [Library]: Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python.[3]
- pyplot: mainly intended for interactive plots and simple cases of programmatic plot generation.[3]
- plot: Plot y versus x as lines and/or markers.[3]
- hist: Compute and plot a histogram.[3]
- bar: Make a bar plot.[3]
- xlabel: Labels the x-axis.
- ylabel: Labels the y-axis.
- show: Display all open figures.[3]
- numpy [Library]: NumPy is a Python library used for working with arrays.[1]
- where: Return elements chosen from *x* or *y* depending on *condition*.[4]

IV. Answers to the Questions

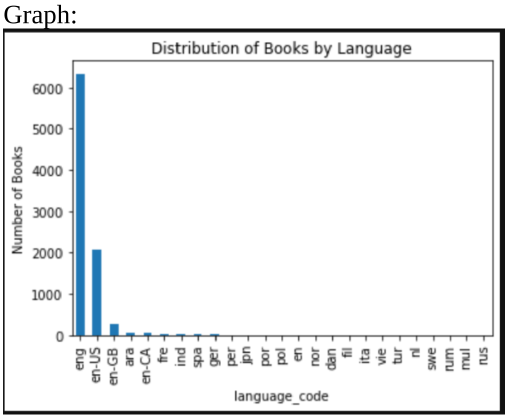
1. Let’s start by analyzing books.csv file. We can count the number of books available in each language using ‘language_code’ column, and plot the results on a bar chart.

   Here is the code to visualize the graph:



```
import pandas as pd
import matplotlib.pyplot as plt

#dataset: books.csv file
books = pd.read_csv('books.csv')

#counting number of books in each language
popular_languages = books['language_code'].value_counts()

#plotting the bar graph
popular_languages.plot(kind='bar')
plt.title("Distribution of Books by Language")
plt.xlabel("language_code")
plt.ylabel("Number of Books")
plt.show()
```

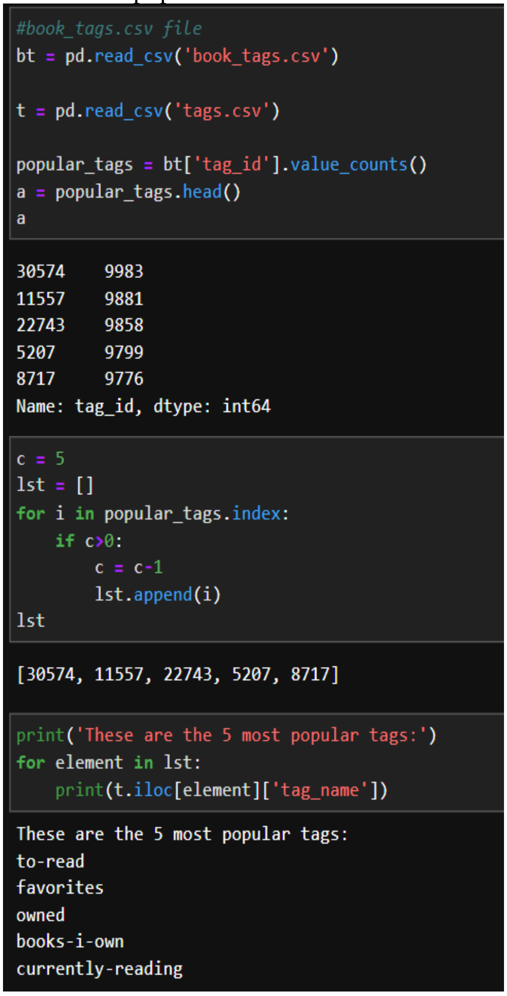
   Graph:



   The resulting plot shows the distribution, and we can see that highest number of books are in ‘eng’ language_code.

There are 25 different languages in the dataset and English has the highest number of books published with a count of 6,341.

2. To answer this question, we will use booktags.csv and tags.csv file. We can count the number of times each tag appears using the ‘tag_id’ column from booktags.csv file, sort them in descending order, and select the top five tags. Then we are finding corresponding ‘tag_name’ for that ‘tag_id’. Top five tags will be the most popular one.



```
#book_tags.csv file
bt = pd.read_csv('book_tags.csv')

t = pd.read_csv('tags.csv')

popular_tags = bt['tag_id'].value_counts()
a = popular_tags.head()
a
```

```
30574    9983
11557    9881
22743    9858
5207     9799
8717     9776
Name: tag_id, dtype: int64
```

```
c = 5
lst = []
for i in popular_tags.index:
    if c>0:
        c = c-1
        lst.append(i)
lst
```

```
[30574, 11557, 22743, 5207, 8717]
```

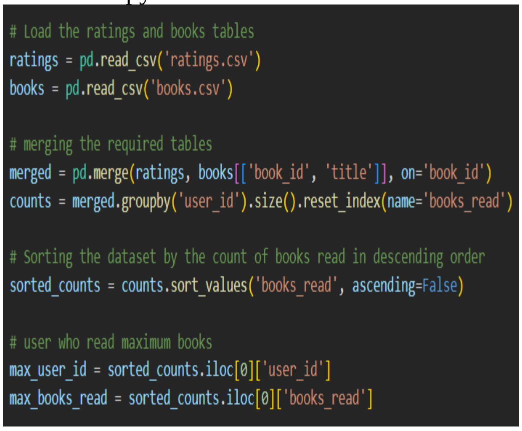
```
print('These are the 5 most popular tags:')
for element in lst:
    print(t.iloc[element]['tag_name'])
```

```
These are the 5 most popular tags:
to-read
favorites
owned
books-i-own
currently-reading
```

According to analysis, these are the five most frequently used book tags:
a. to-read
b. favorites
c. owned
d. books-i-own
e. currently-reading

3. To answer this question, we need ratings.csv and books.csv files. From them we will take book titles and authors for each rating. Group the merged dataset by user ID to get the count of books read by each user. Sort the resulting dataset by the count of books read in descending order to identify the user who has read the most books.

Here is the python code:

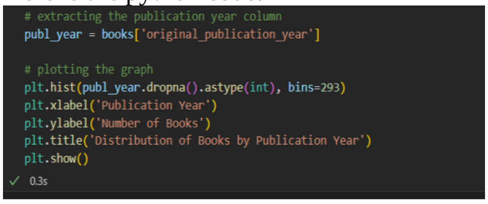


```
# Load the ratings and books tables
ratings = pd.read_csv('ratings.csv')
books = pd.read_csv('books.csv')

# merging the required tables
merged = pd.merge(ratings, books[['book_id', 'title']], on='book_id')
counts = merged.groupby('user_id').size().reset_index(name='books_read')

# Sorting the dataset by the count of books read in descending order
sorted_counts = counts.sort_values('books_read', ascending=False)

# user who read maximum books
max_user_id = sorted_counts.iloc[0]['user_id']
max_books_read = sorted_counts.iloc[0]['books_read']
```

From the output we can say, the user with ID 30944 has read the most books, with a total of 200 books.

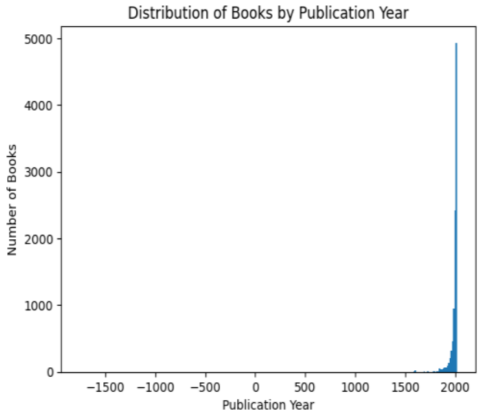
4. Let’s start by looking at the distribution of books published by year in books.csv file. We can create histogram to visualize it.

Here is the python code:



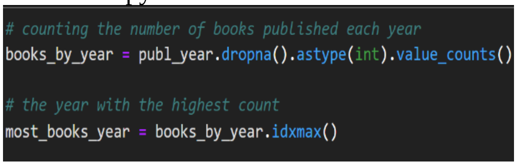
```
# extracting the publication year column
publ_year = books['original_publication_year']

# plotting the graph
plt.hist(publ_year.dropna().astype(int), bins=293)
plt.xlabel('Publication Year')
plt.ylabel('Number of Books')
plt.title('Distribution of Books by Publication Year')
plt.show()
```

✓ 0.3s



From the histogram, we can see that the distribution of books published is in large number towards right i.e., many books are published in late 2000s. To determine the year in which the most books published, we will count the number of books published in each year and then find the year with highest count.

Here is the python code for it:

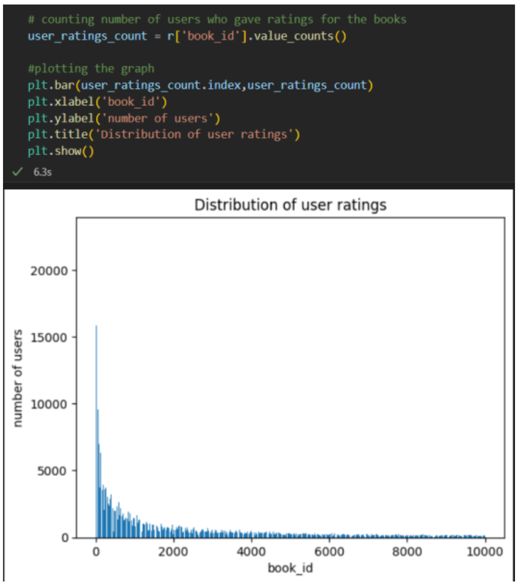


```
# counting the number of books published each year
books_by_year = publ_year.dropna().astype(int).value_counts()

# the year with the highest count
most_books_year = books_by_year.idxmax()
```

So, according to the output, the year 2012 had the highest number of books published, with, 1,760 books published in that year.

5. To answer this question, we need ‘ratings.csv’ and ‘books.csv’ files. We can count number of users who gave ratings for a specific book_id. Let’s create a bar graph to visualize this.

Here is the python code for it:

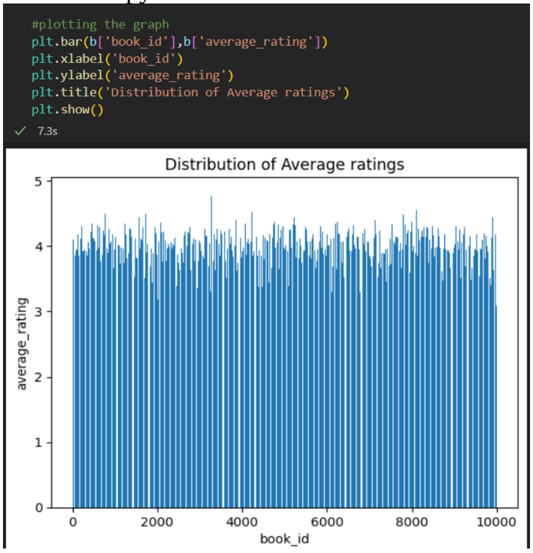


```
# counting number of users who gave ratings for the books
user_ratings_count = r['book_id'].value_counts()

#plotting the graph
plt.bar(user_ratings_count.index,user_ratings_count)
plt.xlabel('book_id')
plt.ylabel('number of users')
plt.title('Distribution of user ratings')
plt.show()
```

✓ 6.3s

Now let’s plot a bar graph for average_ratings corresponding to its book_id.

Here is the python code for it:



```
#plotting the graph
plt.bar(b['book_id'],b['average_rating'])
plt.xlabel('book_id')
plt.ylabel('average_rating')
plt.title('Distribution of Average ratings')
plt.show()
```

✓ 7.3s

According to analysis, books with higher number of ratings tend to have a slightly lower average rating, than some of the books with lower number of average ratings. This could be because popular books are more likely to have a wider range of ratings, including more low ratings, while less popular books may have fewer ratings but a higher proportion of high ratings.

There is a lot of variation in the data, so it's difficult to draw strong conclusions from this analysis alone.

V. SUMMARY OF THE OBSERVATIONS

Most books in the Goodbooks-10k dataset are written in English, and some other languages are also present. It contains data about a lot of books which are published in late 2000s. Book tags include genres which are in high demand among readers. Books that receive more ratings tend to have a higher average rating, suggesting that well-liked books are typically popular. However, it is also observed that books with higher ratings tend to have slightly lower average ratings than books with lower ratings, possibly due to a wider range of ratings for popular books. The Goodbooks-10k dataset provides valuable information on book-related trends and patterns, and talk about the reading habits of a diverse group of book readers.

VI. REFERNCES

[1] “W3Schools,” [Online]. Available:

https://www.w3schools.com/

[2] “pandas documentation,” [Online]. Available:

https://pandas.pydata.org/docs/

[3] “Matplotlib: Visualization with Python,” [Online].

Available: https://matplotlib.org/stable/index.html

[4] “NumPy Documentation,” [Online]. Available:

https://numpy.org/doc/

VII. ACKNOWLEDGEMENTS

This analysis of the Goodbooks-10k dataset was only possible due to the hard work of its creators who collected and organized the data. I would like to thank them and many researchers who have contributed to this field, making it possible to get the information from large datasets like this one.