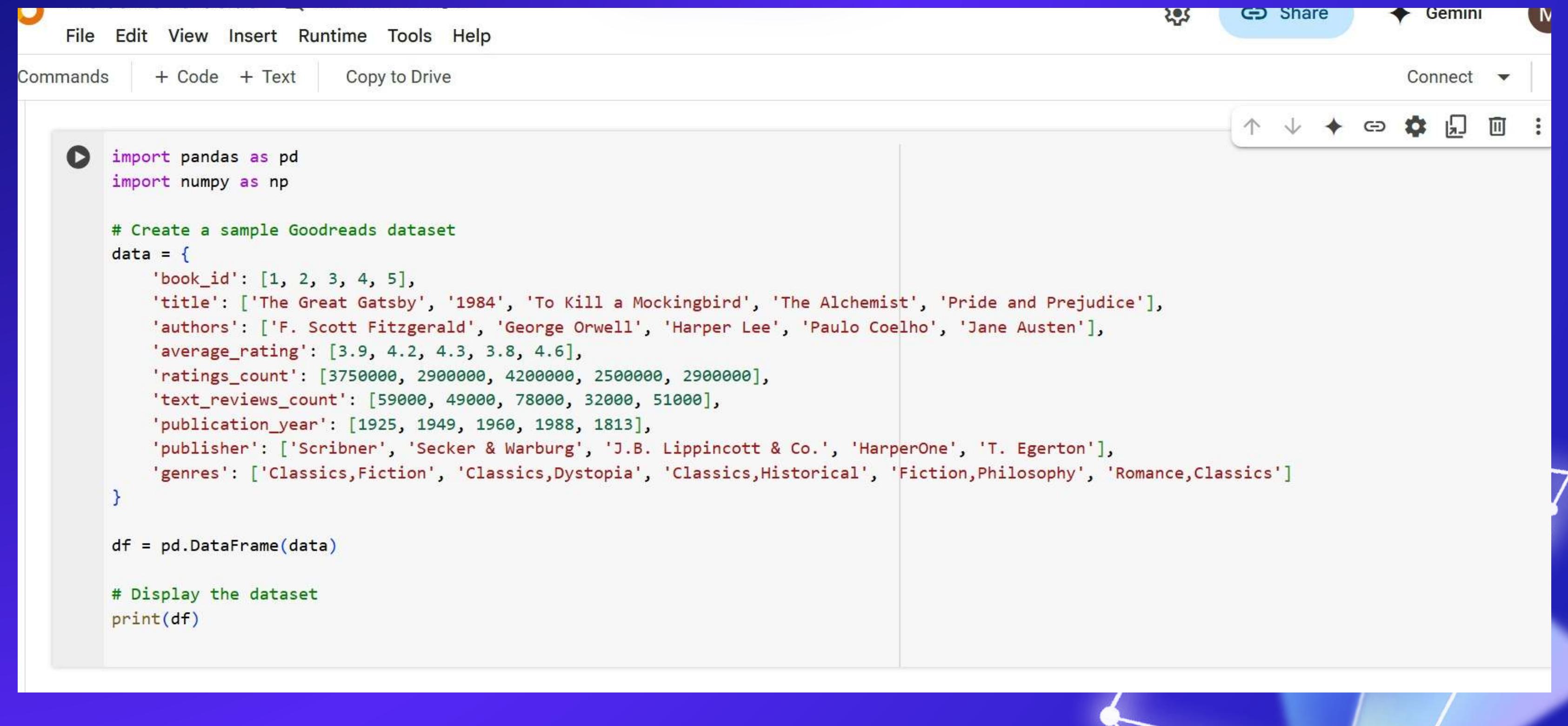


GOODREADS BOOK REVIEWS

ESSENTIAL OF DATA SCIENCE



HERE'S A SMALL DATAFRAME LIKE WE LOADED A REAL DATASET:



The screenshot shows a Jupyter Notebook interface with the following details:

- Header:** File, Edit, View, Insert, Runtime, Tools, Help.
- Toolbar:** Share, Gemini, Connect.
- Cells:** One code cell containing Python code to create a sample dataset from a dictionary.
- Code Content:**

```
import pandas as pd
import numpy as np

# Create a sample Goodreads dataset
data = {
    'book_id': [1, 2, 3, 4, 5],
    'title': ['The Great Gatsby', '1984', 'To Kill a Mockingbird', 'The Alchemist', 'Pride and Prejudice'],
    'authors': ['F. Scott Fitzgerald', 'George Orwell', 'Harper Lee', 'Paulo Coelho', 'Jane Austen'],
    'average_rating': [3.9, 4.2, 4.3, 3.8, 4.6],
    'ratings_count': [3750000, 2900000, 4200000, 2500000, 2900000],
    'text_reviews_count': [59000, 49000, 78000, 32000, 51000],
    'publication_year': [1925, 1949, 1960, 1988, 1813],
    'publisher': ['Scribner', 'Secker & Warburg', 'J.B. Lippincott & Co.', 'HarperOne', 'T. Egerton'],
    'genres': ['Classics,Fiction', 'Classics,Dystopia', 'Classics,Historical', 'Fiction,Philosophy', 'Romance,Classics']
}

df = pd.DataFrame(data)

# Display the dataset
print(df)
```

HERE ARE 20 PROBLEM STATEMENTS WITH ASSUMED DATASET

The screenshot shows a Jupyter Notebook interface with the following details:

- Toolbar:** Includes "Commands", "+ Code", "+ Text", and "Cancel pending changes".
- Status Bar:** Shows "1s completed at 8:27PM".
- Code Cell:** Contains Python code to find the total number of books in a dataset.
- Output Cell:** Displays the dataset as a pandas DataFrame with three sections:
 - Books by Author (5 rows)
 - Books by Publisher (5 rows)
 - Books by Genres (5 rows)
- Console Output:** Shows the total number of books as 5.

```
#1. Find total number of books
total_books = df['book_id'].nunique()
print("Total number of books:", total_books)

{x}
{X} book_id          title      authors  average_rating \
0      1  The Great Gatsby  F. Scott Fitzgerald      3.9
1      2            1984  George Orwell        4.2
2      3  To Kill a Mockingbird  Harper Lee        4.3
3      4       The Alchemist  Paulo Coelho      3.8
4      5  Pride and Prejudice  Jane Austen        4.6

  ratings_count  text_reviews_count  publication_year      publisher \
0      3750000             59000           1925      Scribner
1      2900000             49000           1949  Secker & Warburg
2      4200000             78000           1960  J.B. Lippincott & Co.
3      2500000             32000           1988     HarperOne
4      2900000             51000           1813      T. Egerton

      genres
0  Classics,Fiction
1  Classics,Dystopia
2  Classics,Historical
3  Fiction,Philosophy
4  Romance,Classics
Total number of books: 5
```

2. FIND THE BOOK WITH THE HIGHEST AVERAGE RATING

The screenshot shows a Jupyter Notebook interface with the following content:

```
#1. Find total number of books
total_books = df['book_id'].nunique()
print("Total number of books:", total_books)
```

Execution results:

```
{x} 1s 1s  Cannot save changes RAM Disk
```

book_id	title	authors	average_rating
0	The Great Gatsby	F. Scott Fitzgerald	3.9
1	1984	George Orwell	4.2
2	To Kill a Mockingbird	Harper Lee	4.3
3	The Alchemist	Paulo Coelho	3.8
4	Pride and Prejudice	Jane Austen	4.6

ratings_count	text_reviews_count	publication_year	publisher
3750000	59000	1925	Scribner
2900000	49000	1949	Secker & Warburg
4200000	78000	1960	J.B. Lippincott & Co.
2500000	32000	1988	HarperOne
2900000	51000	1813	T. Egerton

genres
0 Classics, Fiction
1 Classics, Dystopia
2 Classics, Historical
3 Fiction, Philosophy
4 Romance, Classics

Total number of books: 5

RAM Disk

1s completed at 8:27 PM

3. CALCULATE THE MEAN AND

MEDIAN OF RATINGS COUNT

PYTHON COPYED IT

The screenshot shows a Jupyter Notebook interface with the following content:

```
#3. Calculate the mean and median of ratings count
mean_ratings = df['ratings_count'].mean()
median_ratings = df['ratings_count'].median()

print(f"Mean ratings count: {mean_ratings}")
print(f"Median ratings count: {median_ratings}")
```

The notebook displays a table of book data:

book_id	title	authors	average_rating
0	The Great Gatsby	F. Scott Fitzgerald	3.9
1	1984	George Orwell	4.2
2	To Kill a Mockingbird	Harper Lee	4.3
3	The Alchemist	Paulo Coelho	3.8
4	Pride and Prejudice	Jane Austen	4.6

	ratings_count	text_reviews_count	publication_year	publisher
0	3750000	59000	1925	Scribner
1	2900000	49000	1949	Secker & Warburg
2	4200000	78000	1960	J.B. Lippincott & Co.
3	2500000	32000	1988	HarperOne
4	2900000	51000	1813	T. Egerton

	genres
0	Classics, Fiction
1	Classics, Dystopia
2	Classics, Historical
3	Fiction, Philosophy
4	Romance, Classics

Mean ratings count: 3250000.0
Median ratings count: 2900000.0

4.TOP 3 BOOKS WITH MOST RATINGS

```
Commands + Code + Text ⚙ Cannot save changes

#3. Calculate the mean and median of ratings count
mean_ratings = df['ratings_count'].mean()
median_ratings = df['ratings_count'].median()

print(f"Mean ratings count: {mean_ratings}")
print(f"Median ratings count: {median_ratings}")

book_id          title           authors  average_rating \
0      1  The Great Gatsby  F. Scott Fitzgerald      3.9
1      2                  1984        George Orwell      4.2
2      3  To Kill a Mockingbird       Harper Lee      4.3
3      4      The Alchemist     Paulo Coelho      3.8
4      5  Pride and Prejudice      Jane Austen      4.6

   ratings_count  text_reviews_count  publication_year  publisher \
0      3750000            59000          1925      Scribner
1      2900000            49000          1949  Secker & Warburg
2      4200000            78000          1960  J.B. Lippincott & Co.
3      2500000            32000          1988    HarperOne
4      2900000            51000          1813      T. Egerton

      genres
0  Classics,Fiction
1  Classics,Dystopia
2  Classics,Historical
3  Fiction,Philosophy
4  Romance,Classics
Mean ratings count: 3250000.0
Median ratings count: 2900000.0
```

5.FIND THE OLDEST AND NEWEST BOOKS

```
#5.Find the oldest and newest books
oldest_book = df.loc[df['publication_year'].idxmin()]
newest_book = df.loc[df['publication_year'].idxmax()]

print("Oldest Book:", oldest_book['title'], "Year:", oldest_book['publication_year'])
print("Newest Book:", newest_book['title'], "Year:", newest_book['publication_year'])

book_id          title           authors  average_rating \
0      1  The Great Gatsby  F. Scott Fitzgerald      3.9
1      2                  1984        George Orwell      4.2
2      3  To Kill a Mockingbird       Harper Lee      4.3
3      4      The Alchemist     Paulo Coelho      3.8
4      5  Pride and Prejudice      Jane Austen      4.6

   ratings_count  text_reviews_count  publication_year  publisher \
0      3750000            59000          1925      Scribner
1      2900000            49000          1949  Secker & Warburg
2      4200000            78000          1960  J.B. Lippincott & Co.
3      2500000            32000          1988    HarperOne
4      2900000            51000          1813      T. Egerton

      genres
0  Classics,Fiction
1  Classics,Dystopia
2  Classics,Historical
3  Fiction,Philosophy
4  Romance,Classics
Oldest Book: Pride and Prejudice Year: 1813
Newest Book: The Alchemist Year: 1988
```

6. GROUP BOOKS BY AUTHORS AND FIND AVERAGE RATING PER AUTHOR

```
author_rating = df.groupby('authors')['average_rating'].mean()
print("Average rating per author:")
print(author_rating)

book_id      title          authors  average_rating \
0           1  The Great Gatsby  F. Scott Fitzgerald      3.9
1           2                1984  George Orwell        4.2
2           3  To Kill a Mockingbird  Harper Lee        4.3
3           4            The Alchemist  Paulo Coelho      3.8
4           5  Pride and Prejudice  Jane Austen        4.6

r   ratings_count  text_reviews_count  publication_year  publisher \
0       3750000                  59000            1925      Scribner
1       2900000                  49000            1949  Secker & Warburg
2       4200000                  78000            1960  J.B. Lippincott & Co.
3       2500000                  32000            1988    HarperOne
4       2900000                  51000            1813      T. Egerton

genres
0  Classics,Fiction
1  Classics,Dystopia
2  Classics,Historical
3  Fiction,Philosophy
4  Romance,Classics
Average rating per author:
authors
F. Scott Fitzgerald      3.9
George Orwell        4.2
Harper Lee        4.3
Jane Austen        4.6
Paulo Coelho      3.8
Name: average_rating, dtype: float64
```

7. FIND CORRELATION

BETWEEN RATINGS

COUNT AND TEXT

REVIEWS COUNT

PYTHON COPYEDIT

```
# Display the dataset
print(df)

#7. Find correlation between ratings count and text reviews count
correlation = df['ratings_count'].corr(df['text_reviews_count'])
print(f"Correlation between ratings and text reviews: {correlation:.2f}")

book_id          title           authors  average_rating \
0      1  The Great Gatsby  F. Scott Fitzgerald      3.9
1      2            1984  George Orwell      4.2
2      3  To Kill a Mockingbird  Harper Lee      4.3
3      4        The Alchemist  Paulo Coelho      3.8
4      5  Pride and Prejudice  Jane Austen      4.6

   ratings_count  text_reviews_count  publication_year  publisher \
0      3750000             59000          1925      Scribner
1      2900000             49000          1949  Secker & Warburg
2      4200000             78000          1960  J.B. Lippincott & Co.
3      2500000             32000          1988     HarperOne
4      2900000             51000          1813       T. Egerton

           genres
0  Classics,Fiction
1  Classics,Dystopia
2  Classics,Historical
3  Fiction,Philosophy
4  Romance,Classics

Correlation between ratings and text reviews: 0.95
```

8.CLASSIFY BOOKS AS

HIGH-RATED (≥ 4.0)

OR LOW-RATED (< 4.0)

```
#8.Classify books as High-rated (>=4.0) or Low-rated (<4.0)
df['rating_category'] = np.where(df['average_rating'] >= 4.0, 'High Rated', 'Low Rated')
print(df[['title', 'average_rating', 'rating_category']])
```

book_id	title	authors	average_rating	rating_category
0	The Great Gatsby	F. Scott Fitzgerald	3.9	Low Rated
1	1984	George Orwell	4.2	High Rated
2	To Kill a Mockingbird	Harper Lee	4.3	High Rated
3	The Alchemist	Paulo Coelho	3.8	Low Rated
4	Pride and Prejudice	Jane Austen	4.6	High Rated

book_id	ratings_count	text_reviews_count	publication_year	publisher
0	3750000	59000	1925	Scribner
1	2900000	49000	1949	Secker & Warburg
2	4200000	78000	1960	J.B. Lippincott & Co.
3	2500000	32000	1988	HarperOne
4	2900000	51000	1813	T. Egerton

book_id	genres
0	Classics, Fiction
1	Classics, Dystopia
2	Classics, Historical
3	Fiction, Philosophy
4	Romance, Classics

book_id	title	average_rating	rating_category
0	The Great Gatsby	3.9	Low Rated
1	1984	4.2	High Rated
2	To Kill a Mockingbird	4.3	High Rated
3	The Alchemist	3.8	Low Rated
4	Pride and Prejudice	4.6	High Rated

9.FIND BOOKS CONTAINING THE WORD "GREAT" IN TITLE

```
#9.Find books containing the word "Great" in title
great_books = df[df['title'].str.contains('Great', case=False, na=False)]
print("Books containing 'Great' in title:")
print(great_books[['title', 'authors']])


book_id          title      authors  average_rating \
0            1  The Great Gatsby  F. Scott Fitzgerald      3.9
1            2           1984    George Orwell      4.2
2            3  To Kill a Mockingbird  Harper Lee      4.3
3            4       The Alchemist  Paulo Coelho      3.8
4            5  Pride and Prejudice  Jane Austen      4.6

   ratings_count  text_reviews_count  publication_year  publisher \
0        3750000             59000            1925      Scribner
1        2900000             49000            1949  Secker & Warburg
2        4200000             78000            1960  J.B. Lippincott & Co.
3        2500000             32000            1988     HarperOne
4        2900000             51000            1813      T. Egerton

      genres
0  Classics,Fiction
1  Classics,Dystopia
2  Classics,Historical
3  Fiction,Philosophy
4  Romance,Classics
Books containing 'Great' in title:
      title      authors
0  The Great Gatsby  F. Scott Fitzgerald
```

10.COUNT HOW MANY BOOKS BELONG TO "CLASSICS" GENRE

```
#10.Count how many books belong to "Classics" genre
classic_books = df[df['genres'].str.contains('Classics', case=False, na=False)]
print(f"Number of books under 'Classics': {classic_books.shape[0]}")

book_id          title      authors  average_rating \
0            1  The Great Gatsby  F. Scott Fitzgerald      3.9
1            2           1984    George Orwell      4.2
2            3  To Kill a Mockingbird  Harper Lee      4.3
3            4       The Alchemist  Paulo Coelho      3.8
4            5  Pride and Prejudice  Jane Austen      4.6

   ratings_count  text_reviews_count  publication_year  publisher \
0        3750000             59000            1925      Scribner
1        2900000             49000            1949  Secker & Warburg
2        4200000             78000            1960  J.B. Lippincott & Co.
3        2500000             32000            1988     HarperOne
4        2900000             51000            1813      T. Egerton

      genres
0  Classics,Fiction
1  Classics,Dystopia
2  Classics,Historical
3  Fiction,Philosophy
4  Romance,Classics
Number of books under 'Classics': 4
```

11. REPLACE MISSING PUBLISHER

NAMES WITH 'UNKNOWN'

```
Commands + Code + Text Cannot save changes
1s
#11. Replace missing publisher names with 'Unknown'
df['publisher'] = df['publisher'].fillna('Unknown')
print(df[['title', 'publisher']])

book_id          title      authors  average_rating \
0           1   The Great Gatsby  F. Scott Fitzgerald       3.9
1           2                1984        George Orwell       4.2
2           3  To Kill a Mockingbird    Harper Lee       4.3
3           4        The Alchemist  Paulo Coelho       3.8
4           5  Pride and Prejudice   Jane Austen       4.6

   ratings_count  text_reviews_count  publication_year  \
0      3750000             59000            1925
1      2900000             49000            1949
2      4200000             78000            1960
3      2500000             32000            1988
4      2900000             51000            1813

                                publisher \
0                         Scribner
1        Secker & Warburg
2  J.B. Lippincott & Co.
3        HarperOne
4         T. Egerton

      genres
0  Classics,Fiction
1  Classics,Dystopia
2 Classics,Historical
3  Fiction,Philosophy
4  Romance,Classics

      title      publisher
0   The Great Gatsby      Scribner
1                1984  Secker & Warburg
2  To Kill a Mockingbird  J.B. Lippincott & Co.
3        The Alchemist      HarperOne
4  Pride and Prejudice      T. Egerton
```

12. FIND THE AVERAGE

PUBLICATION YEAR OF ALL BOOKS

```
Commands + Code + Text Cannot save changes
1s
print(df)

#12. Find the average publication year of all books
avg_year = df['publication_year'].mean()
print(f"Average publication year: {avg_year:.0f}")

book_id          title      authors  average_rating \
0           1   The Great Gatsby  F. Scott Fitzgerald       3.9
1           2                1984        George Orwell       4.2
2           3  To Kill a Mockingbird    Harper Lee       4.3
3           4        The Alchemist  Paulo Coelho       3.8
4           5  Pride and Prejudice   Jane Austen       4.6

   ratings_count  text_reviews_count  publication_year  \
0      3750000             59000            1925
1      2900000             49000            1949
2      4200000             78000            1960
3      2500000             32000            1988
4      2900000             51000            1813

                                publisher \
0                         Scribner
1        Secker & Warburg
2  J.B. Lippincott & Co.
3        HarperOne
4         T. Egerton

      genres
0  Classics,Fiction
1  Classics,Dystopia
2 Classics,Historical
3  Fiction,Philosophy
4  Romance,Classics
Average publication year: 1927
```

13. CREATE A NEW COLUMN

THAT SHOWS THE DECADE OF
PUBLICATION
PYTHON COPYEDIT

```
Commands + Code + Text Cannot save changes
0s #13. Create a new column that shows the decade of publication
df['decade'] = (df['publication_year'] // 10) * 10
print(df[['title', 'publication_year', 'decade']])

book_id          title      authors  average_rating \
0           1  The Great Gatsby  F. Scott Fitzgerald      3.9
1           2                1984  George Orwell      4.2
2           3  To Kill a Mockingbird  Harper Lee      4.3
3           4        The Alchemist  Paulo Coelho      3.8
4           5  Pride and Prejudice  Jane Austen      4.6

ratings_count  text_reviews_count  publication_year  publisher \
0       3750000                 59000            1925      Scribner
1       2900000                 49000            1949  Secker & Warburg
2       4200000                 78000            1960  J.B. Lippincott & Co.
3       2500000                 32000            1988    HarperOne
4       2900000                 51000            1813     T. Egerton

genres
0  Classics,Fiction
1  Classics,Dystopia
2 Classics,Historical
3  Fiction,Philosophy
4  Romance,Classics

      title  publication_year  decade
0  The Great Gatsby            1925   1920
1                1984            1949   1940
2  To Kill a Mockingbird      1960   1960
3        The Alchemist            1988   1980
4  Pride and Prejudice            1813  1810
```

14. LIST ALL PUBLISHERS AND THE NUMBER OF BOOKS THEY HAVE PUBLISHED

PYTHON COPYEDIT

```
Commands + Code + Text ⚙ Cannot save changes
▶ publisher_count = df['publisher'].value_counts()
print("Number of books by publisher:")
print(publisher_count)

→   book_id          title      authors  average_rating \
0       1  The Great Gatsby  F. Scott Fitzgerald      3.9
1       2                1984  George Orwell      4.2
2       3  To Kill a Mockingbird  Harper Lee      4.3
3       4            The Alchemist  Paulo Coelho      3.8
4       5  Pride and Prejudice  Jane Austen      4.6

           ratings_count  text_reviews_count  publication_year  \
0             3750000                  59000            1925
1             2900000                  49000            1949
2             4200000                  78000            1960
3             2500000                  32000            1988
4             2900000                  51000            1813

                                         genres
0  Classics, Fiction
1  Classics, Dystopia
2  Classics, Historical
3  Fiction, Philosophy
4  Romance, Classics

Number of books by publisher:
publisher
Scribner              1
Secker & Warburg      1
J.B. Lippincott & Co.    1
HarperOne               1
T. Egerton               1

Name: count, dtype: int64
```

15. FIND BOOKS WITH TEXT REVIEW COUNT MORE THAN 50,000

```
Commands + Code + Text Cannot save changes

#15. Find books with text review count more than 50,000
high_text_reviews = df[df['text_reviews_count'] > 50000]
print("Books with more than 50,000 text reviews:")
print(high_text_reviews[['title', 'text_reviews_count']])

book_id          title      authors  average_rating \
0      1    The Great Gatsby  F. Scott Fitzgerald      3.9
1      2                1984        George Orwell      4.2
2      3  To Kill a Mockingbird       Harper Lee      4.3
3      4        The Alchemist      Paulo Coelho      3.8
4      5   Pride and Prejudice      Jane Austen      4.6

ratings_count  text_reviews_count  publication_year      publisher \
0      3750000             59000            1925      Scribner
1      2900000             49000            1949  Secker & Warburg
2      4200000             78000            1960  J.B. Lippincott & Co.
3      2500000             32000            1988     HarperOne
4      2900000             51000            1813      T. Egerton

genres
0  Classics,Fiction
1  Classics,Dystopia
2 Classics,Historical
3  Fiction,Philosophy
4  Romance,Classics
Books with more than 50,000 text reviews:
      title  text_reviews_count
0    The Great Gatsby           59000
2  To Kill a Mockingbird         78000
4  Pride and Prejudice           51000
```

16. FIND THE MINIMUM RATING COUNT IN THE DATASET

```
Commands + Code + Text Cannot save changes

#16. Find the minimum rating count in the dataset
min_ratings = df['ratings_count'].min()
print(f"Minimum rating count: {min_ratings}")

book_id          title      authors  average_rating \
0      1    The Great Gatsby  F. Scott Fitzgerald      3.9
1      2                1984        George Orwell      4.2
2      3  To Kill a Mockingbird       Harper Lee      4.3
3      4        The Alchemist      Paulo Coelho      3.8
4      5   Pride and Prejudice      Jane Austen      4.6

ratings_count  text_reviews_count  publication_year      publisher \
0      3750000             59000            1925      Scribner
1      2900000             49000            1949  Secker & Warburg
2      4200000             78000            1960  J.B. Lippincott & Co.
3      2500000             32000            1988     HarperOne
4      2900000             51000            1813      T. Egerton

genres
0  Classics,Fiction
1  Classics,Dystopia
2 Classics,Historical
3  Fiction,Philosophy
4  Romance,Classics
Minimum rating count: 2500000
```

18. FIND NUMBER OF UNIQUE GENRES APPEARING ACROSS ALL BOOKS

```
Commands + Code + Text ⚙ Cannot save changes
#18. Find number of unique genres appearing across all books
# Split genres and find unique ones
all_genres = df['genres'].str.split(',')
flat_genres = [genre for sublist in all_genres for genre in sublist]
unique_genres = set(flat_genres)

print(f"Number of unique genres: {len(unique_genres)}")
print(f"Genres: {unique_genres}")

book_id          title           authors  average_rating \
0      1   The Great Gatsby  F. Scott Fitzgerald      3.9
1      2             1984        George Orwell      4.2
2      3  To Kill a Mockingbird       Harper Lee      4.3
3      4        The Alchemist       Paulo Coelho      3.8
4      5  Pride and Prejudice       Jane Austen      4.6

   ratings_count  text_reviews_count  publication_year           publisher \
0      3750000                  59000            1925           Scribner
1      2900000                  49000            1949  Secker & Warburg
2      4200000                  78000            1960  J.B. Lippincott & Co.
3      2500000                  32000            1988      HarperOne
4      2900000                  51000            1813        T. Egerton

           genres
0  Classics,Fiction
1  Classics,Dystopia
2 Classics,Historical
3  Fiction,Philosophy
4  Romance,Classics
Number of unique genres: 6
Genres: {'Philosophy', 'Historical', 'Fiction', 'Romance', 'Dystopia', 'Classics'}
```

19. CALCULATE STANDARD DEVIATION OF AVERAGE RATINGS

```
print(df)

#19. Calculate standard deviation of average ratings
rating_std = df['average_rating'].std()
print(f"Standard deviation of ratings: {rating_std:.2f}")

    book_id          title      authors  average_rating \
0       1   The Great Gatsby  F. Scott Fitzgerald      3.9
1       2                  1984        George Orwell      4.2
2       3  To Kill a Mockingbird      Harper Lee      4.3
3       4           The Alchemist     Paulo Coelho      3.8
4       5  Pride and Prejudice      Jane Austen      4.6

    ratings_count  text_reviews_count  publication_year      publisher \
0      3750000             59000            1925      Scribner
1      2900000             49000            1949  Secker & Warburg
2      4200000             78000            1960  J.B. Lippincott & Co.
3      2500000             32000            1988      HarperOne
4      2900000             51000            1813      T. Egerton

      genres
0  Classics,Fiction
1  Classics,Dystopia
2  Classics,Historical
3  Fiction,Philosophy
4  Romance,Classics
Standard deviation of ratings: 0.32
```

20. FIND AUTHORS WHO HAVE BOOKS PUBLISHED BEFORE 1950

```
#20. Find authors who have books published before 1950
authors_before_1950 = df[df['publication_year'] < 1950]['authors'].unique()
print("Authors with books published before 1950:")
print(authors_before_1950)

    book_id          title      authors  average_rating \
0       1   The Great Gatsby  F. Scott Fitzgerald      3.9
1       2                  1984        George Orwell      4.2
2       3  To Kill a Mockingbird      Harper Lee      4.3
3       4           The Alchemist     Paulo Coelho      3.8
4       5  Pride and Prejudice      Jane Austen      4.6

    ratings_count  text_reviews_count  publication_year      publisher \
0      3750000             59000            1925      Scribner
1      2900000             49000            1949  Secker & Warburg
2      4200000             78000            1960  J.B. Lippincott & Co.
3      2500000             32000            1988      HarperOne
4      2900000             51000            1813      T. Egerton

      genres
0  Classics,Fiction
1  Classics,Dystopia
2  Classics,Historical
3  Fiction,Philosophy
4  Romance,Classics
Authors with books published before 1950:
['F. Scott Fitzgerald' 'George Orwell' 'Jane Austen']
```

THANK YOU!

~MAITHILI GOUD

