EDS Theory Activity No. 1

Name: Sarthak Bhosale

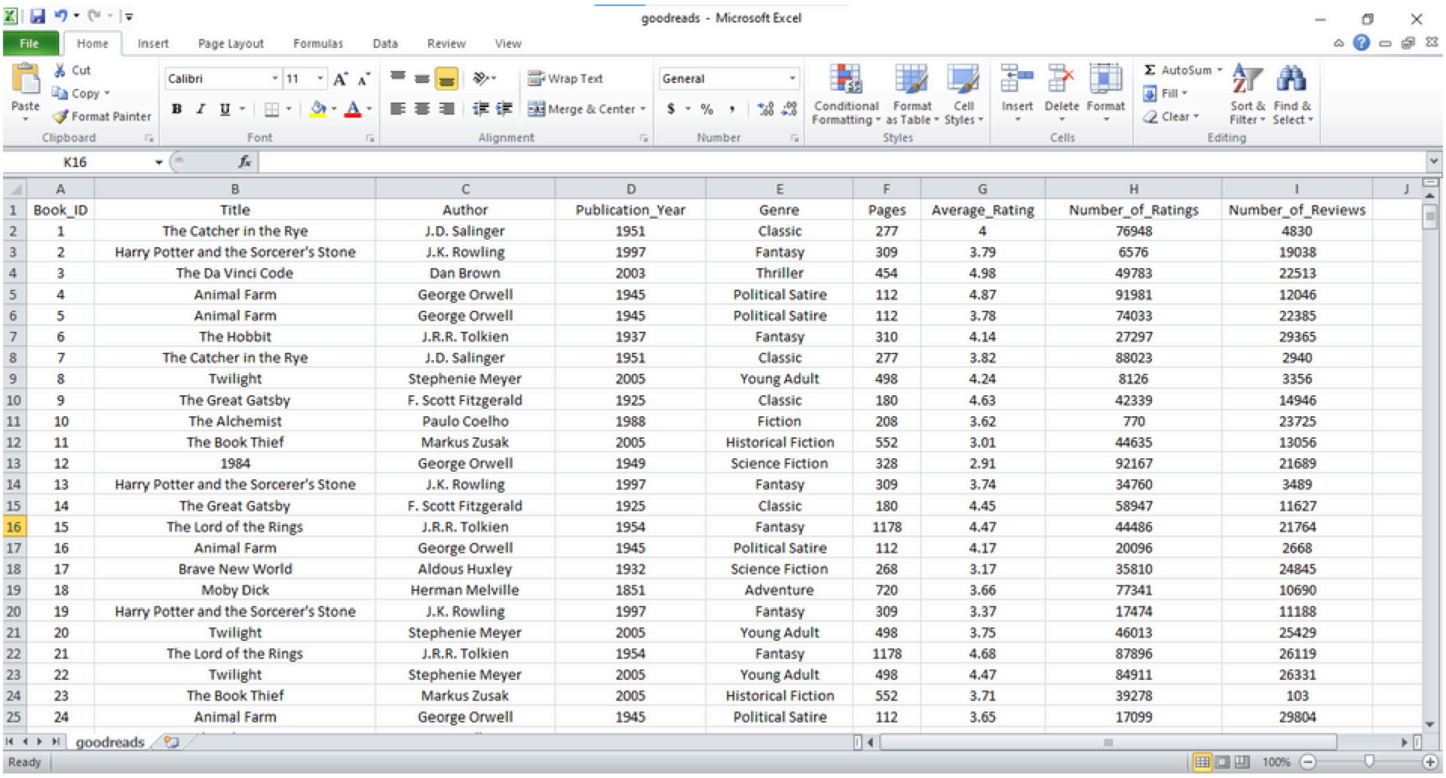
Batch:- CS5

Roll no.:- 30

PRN:- 202401100051

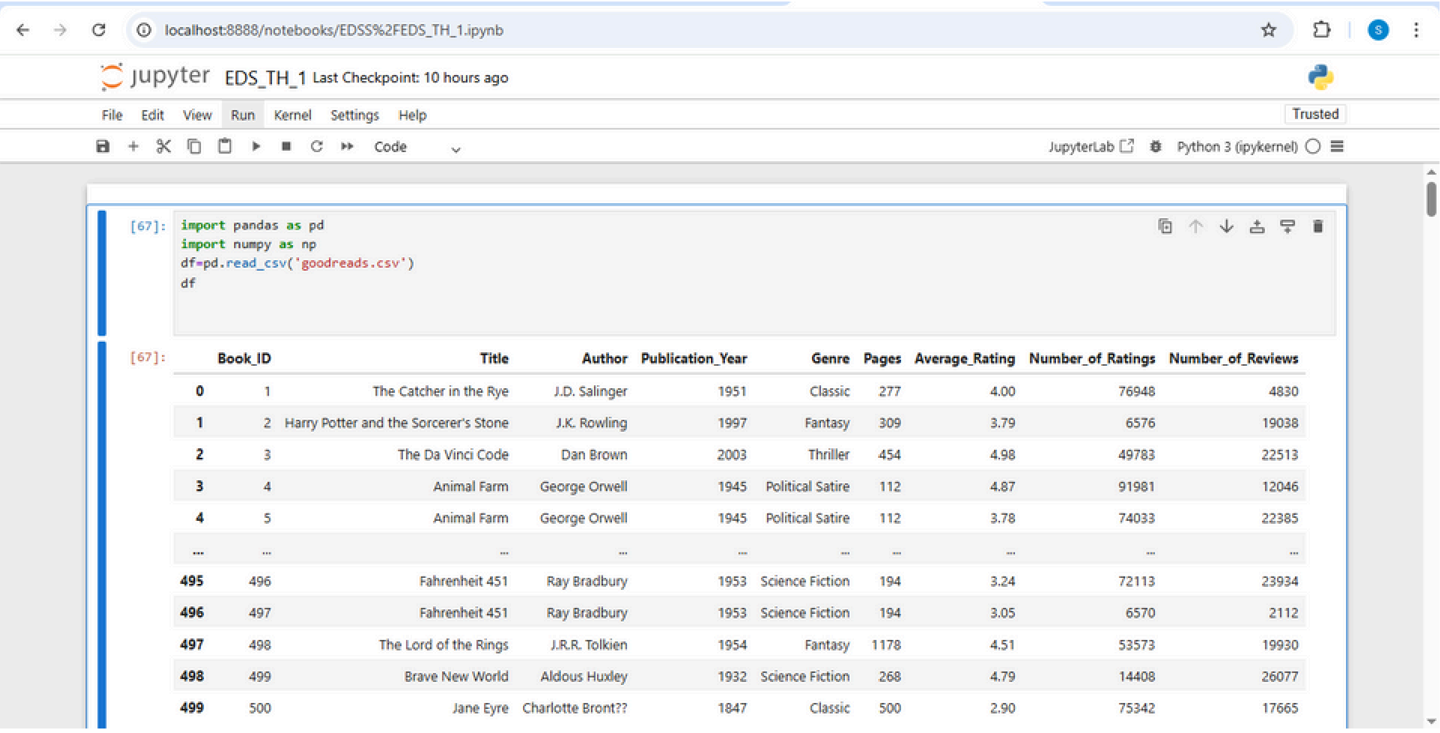
TOPIC:- Goodreads Book Reviews

Screen shot of the data set I have used which was downloaded from Kaggle.



	A	B	C	D	E	F	G	H	I
1	Book_ID	Title	Author	Publication_Year	Genre	Pages	Average_Rating	Number_of_Ratings	Number_of_Reviews
2	1	The Catcher in the Rye	J.D. Salinger	1951	Classic	277	4	76948	4830
3	2	Harry Potter and the Sorcerer's Stone	J.K. Rowling	1997	Fantasy	309	3.79	6576	19038
4	3	The Da Vinci Code	Dan Brown	2003	Thriller	454	4.98	49783	22513
5	4	Animal Farm	George Orwell	1945	Political Satire	112	4.87	91981	12046
6	5	Animal Farm	George Orwell	1945	Political Satire	112	3.78	74033	22385
7	6	The Hobbit	J.R.R. Tolkien	1937	Fantasy	310	4.14	27297	29365
8	7	The Catcher in the Rye	J.D. Salinger	1951	Classic	277	3.82	88023	2940
9	8	Twilight	Stephenie Meyer	2005	Young Adult	498	4.24	8126	3356
10	9	The Great Gatsby	F. Scott Fitzgerald	1925	Classic	180	4.63	42339	14946
11	10	The Alchemist	Paulo Coelho	1988	Fiction	208	3.62	770	23725
12	11	The Book Thief	Markus Zusak	2005	Historical Fiction	552	3.01	44635	13056
13	12	1984	George Orwell	1949	Science Fiction	328	2.91	92167	21689
14	13	Harry Potter and the Sorcerer's Stone	J.K. Rowling	1997	Fantasy	309	3.74	34760	3489
15	14	The Great Gatsby	F. Scott Fitzgerald	1925	Classic	180	4.45	58947	11627
16	15	The Lord of the Rings	J.R.R. Tolkien	1954	Fantasy	1178	4.47	44486	21764
17	16	Animal Farm	George Orwell	1945	Political Satire	112	4.17	20096	2668
18	17	Brave New World	Aldous Huxley	1932	Science Fiction	268	3.17	35810	24845
19	18	Moby Dick	Herman Melville	1851	Adventure	720	3.66	77341	10690
20	19	Harry Potter and the Sorcerer's Stone	J.K. Rowling	1997	Fantasy	309	3.37	17474	11188
21	20	Twilight	Stephenie Meyer	2005	Young Adult	498	3.75	46013	25429
22	21	The Lord of the Rings	J.R.R. Tolkien	1954	Fantasy	1178	4.68	87896	26119
23	22	Twilight	Stephenie Meyer	2005	Young Adult	498	4.47	84911	26331
24	23	The Book Thief	Markus Zusak	2005	Historical Fiction	552	3.71	39278	103
25	24	Animal Farm	George Orwell	1945	Political Satire	112	3.65	17099	29804

Screenshot of initial Preparations.



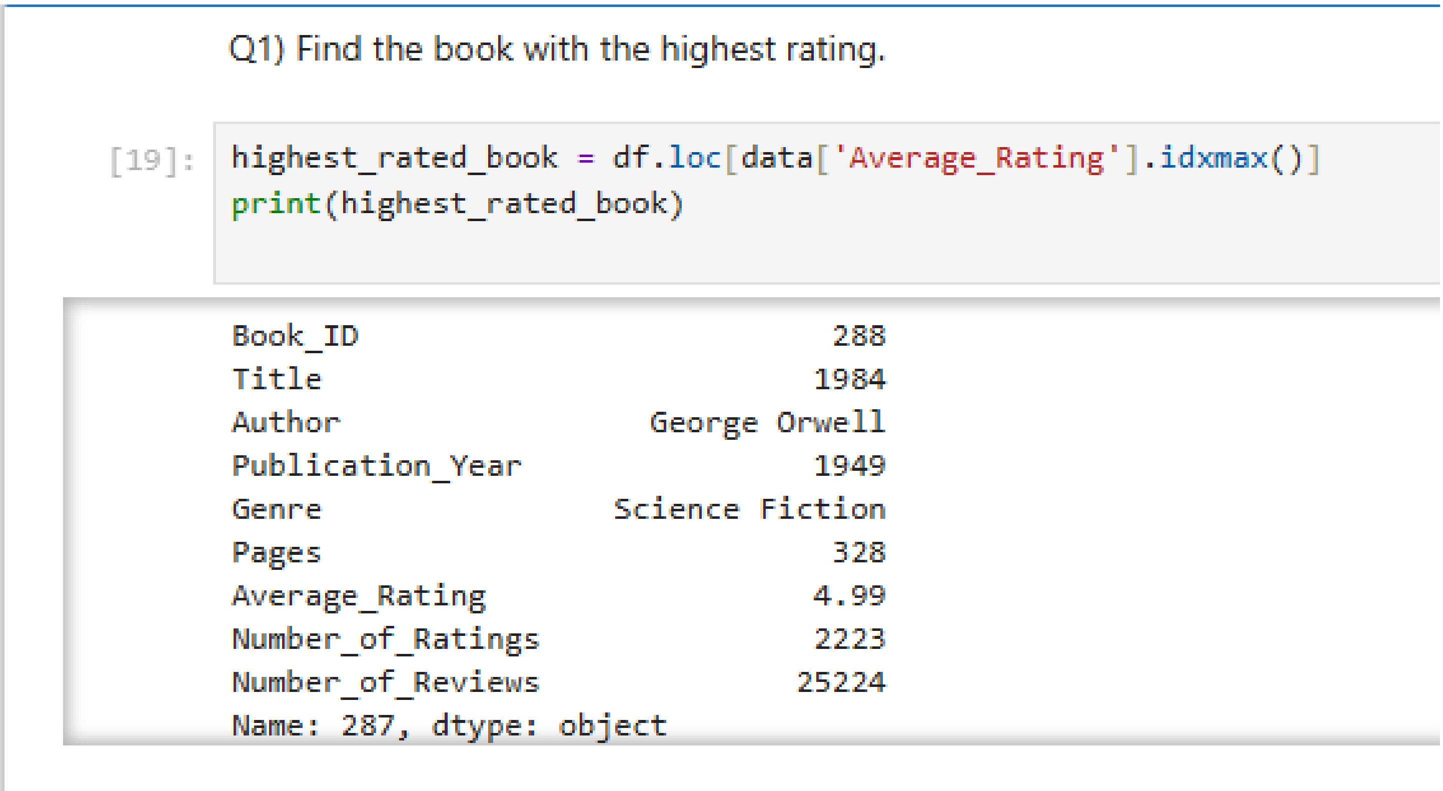
```
[67]: import pandas as pd
      import numpy as np
      df=pd.read_csv('goodreads.csv')
      df
```

[67]:

	Book_ID	Title	Author	Publication_Year	Genre	Pages	Average_Rating	Number_of_Ratings	Number_of_Reviews
0	1	The Catcher in the Rye	J.D. Salinger	1951	Classic	277	4.00	76948	4830
1	2	Harry Potter and the Sorcerer's Stone	J.K. Rowling	1997	Fantasy	309	3.79	6576	19038
2	3	The Da Vinci Code	Dan Brown	2003	Thriller	454	4.98	49783	22513
3	4	Animal Farm	George Orwell	1945	Political Satire	112	4.87	91981	12046
4	5	Animal Farm	George Orwell	1945	Political Satire	112	3.78	74033	22385
...	...	...	...	...	...	...	...	...	...
495	496	Fahrenheit 451	Ray Bradbury	1953	Science Fiction	194	3.24	72113	23934
496	497	Fahrenheit 451	Ray Bradbury	1953	Science Fiction	194	3.05	6570	2112
497	498	The Lord of the Rings	J.R.R. Tolkien	1954	Fantasy	1178	4.51	53573	19930
498	499	Brave New World	Aldous Huxley	1932	Science Fiction	268	4.79	14408	26077
499	500	Jane Eyre	Charlotte Bront??	1847	Classic	500	2.90	75342	17665

Problem statements along with their solutions.

1.

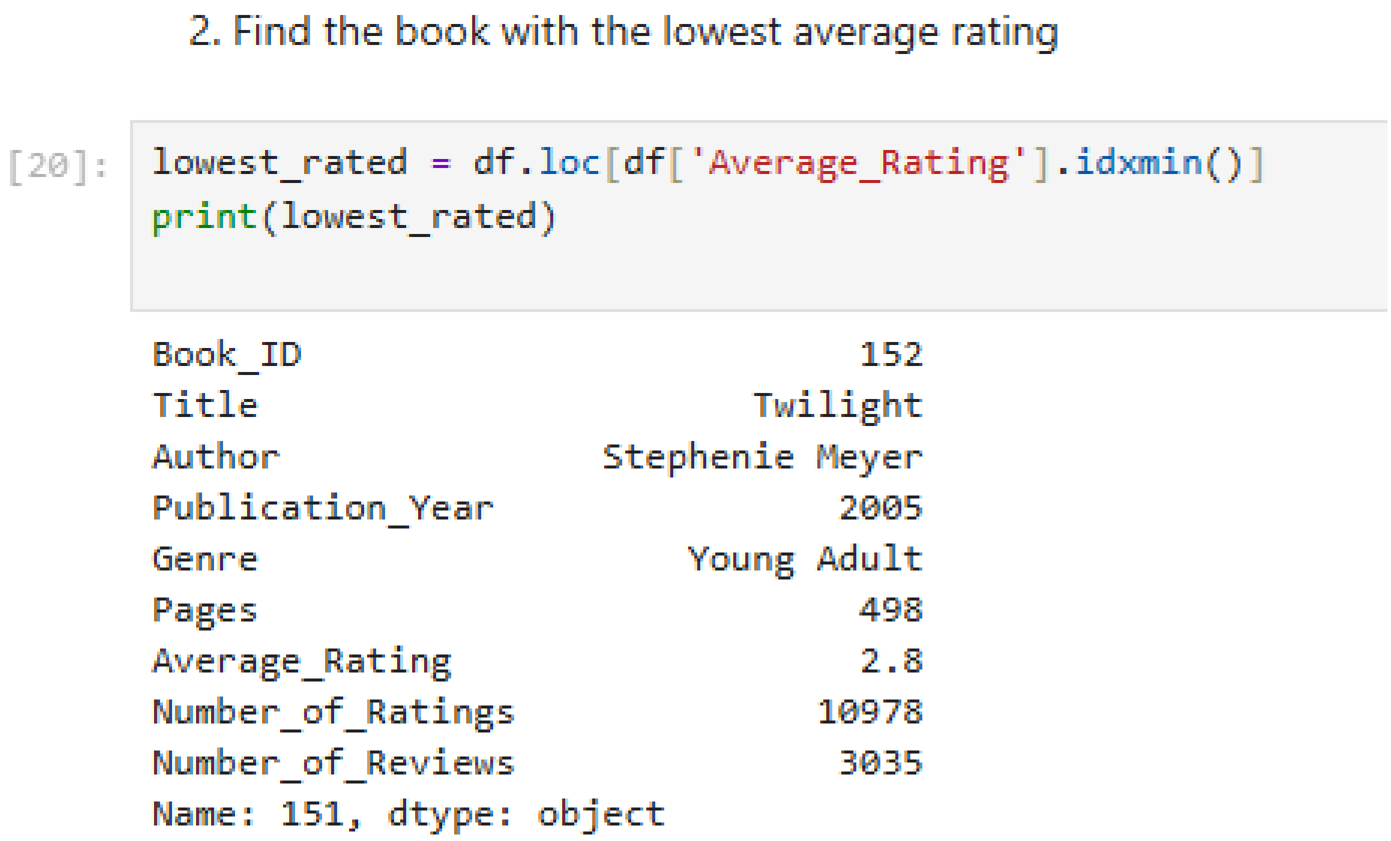


Q1) Find the book with the highest rating.

```
[19]: highest_rated_book = df.loc[data['Average_Rating'].idxmax()]
      print(highest_rated_book)
```

```
Book_ID                               288
Title                                1984
Author                      George Orwell
Publication_Year                     1949
Genre                     Science Fiction
Pages                                 328
Average_Rating                       4.99
Number_of_Ratings                    2223
Number_of_Reviews                   25224
Name: 287, dtype: object
```

2.

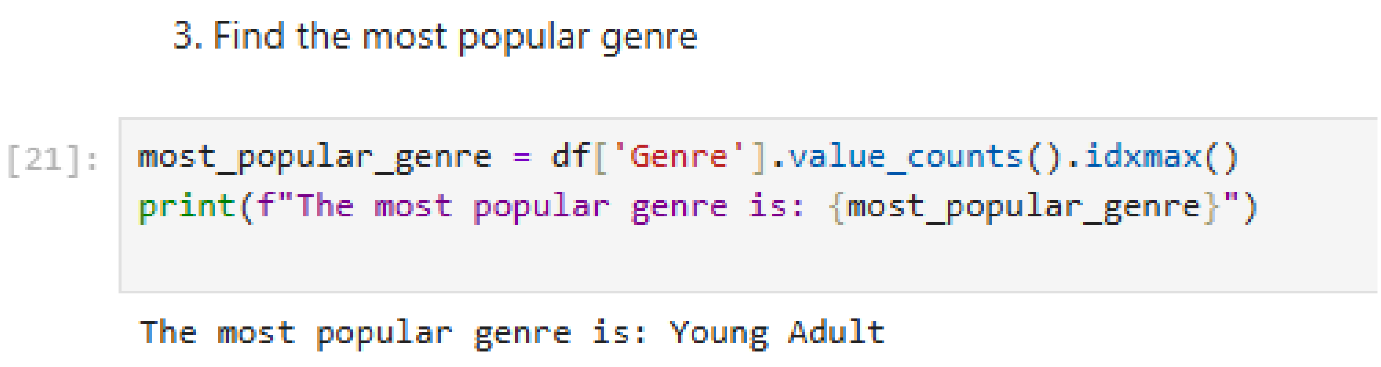


2. Find the book with the lowest average rating

```
[20]: lowest_rated = df.loc[df['Average_Rating'].idxmin()]
      print(lowest_rated)
```

```
Book_ID                              152
Title                           Twilight
Author                   Stephenie Meyer
Publication_Year                    2005
Genre                        Young Adult
Pages                                498
Average_Rating                       2.8
Number_of_Ratings                  10978
Number_of_Reviews                   3035
Name: 151, dtype: object
```

3.



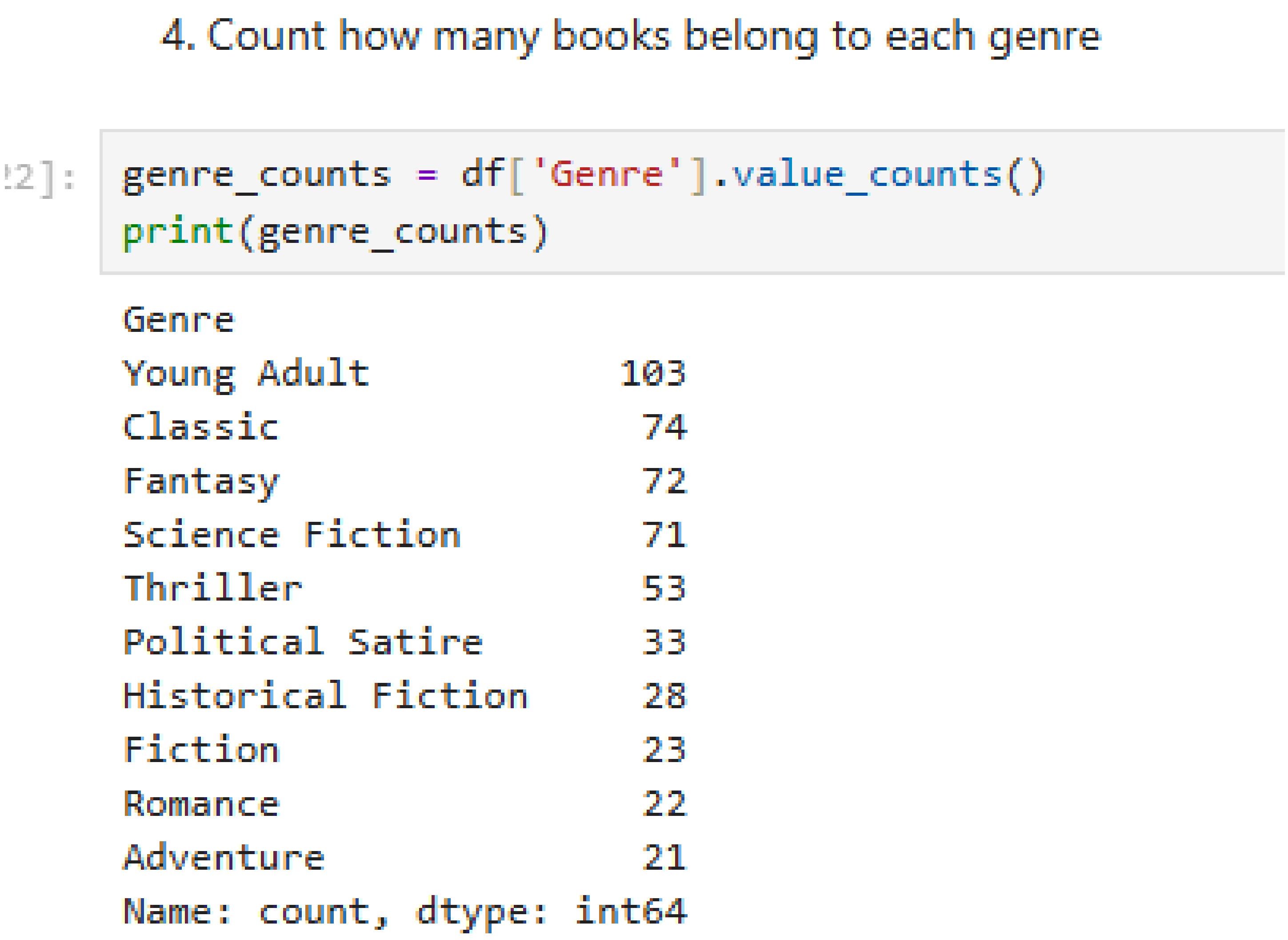
3. Find the most popular genre

[21]:

```
most_popular_genre = df['Genre'].value_counts().idxmax()
print(f"The most popular genre is: {most_popular_genre}")
```

```
The most popular genre is: Young Adult
```

4.



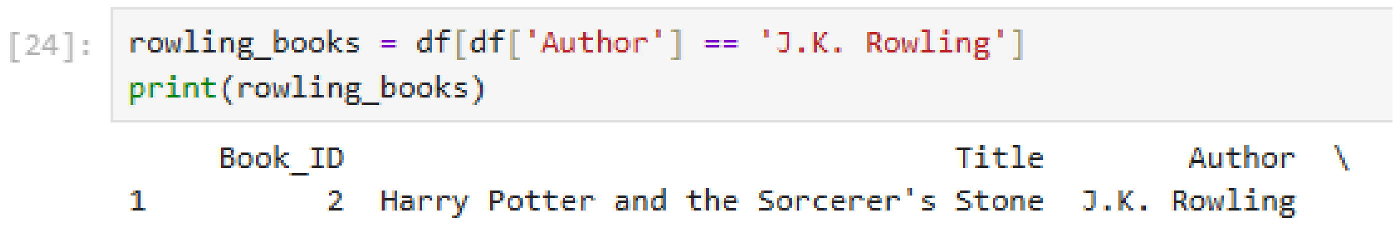
4. Count how many books belong to each genre

2]:

```
genre_counts = df['Genre'].value_counts()
print(genre_counts)
```

```
Genre
Young Adult          103
Classic               74
Fantasy               72
Science Fiction       71
Thriller              53
Political Satire      33
Historical Fiction    28
Fiction               23
Romance               22
Adventure             21
Name: count, dtype: int64
```

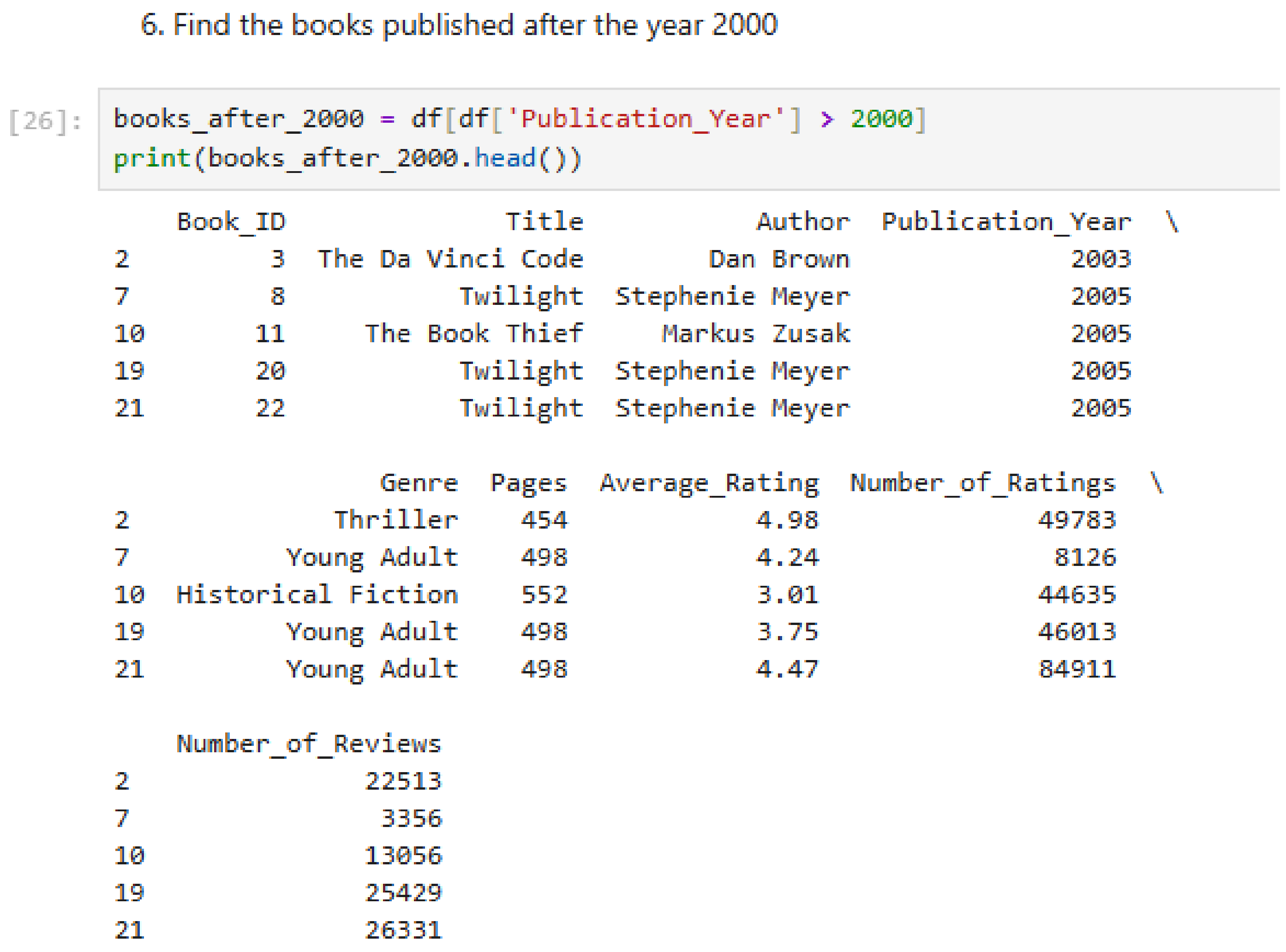
5.



```
[24]: rowling_books = df[df['Author'] == 'J.K. Rowling']
      print(rowling_books)
```

```
   Book_ID                                  Title        Author  \
1        2  Harry Potter and the Sorcerer's Stone  J.K. Rowling
```

6.



6. Find the books published after the year 2000

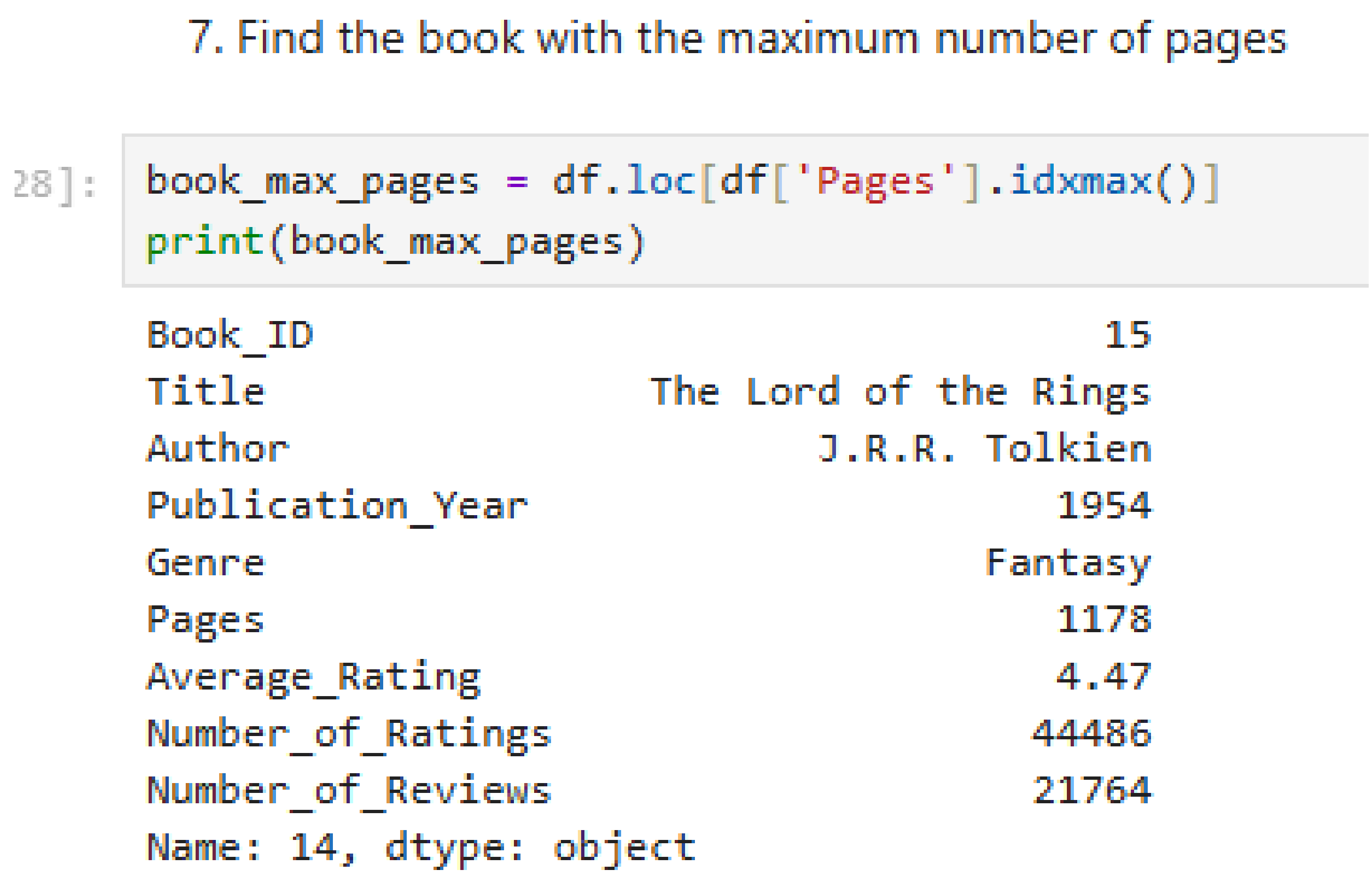
```
[26]: books_after_2000 = df[df['Publication_Year'] > 2000]
      print(books_after_2000.head())
```

```
    Book_ID              Title           Author  Publication_Year  \
2         3  The Da Vinci Code        Dan Brown              2003
7         8           Twilight  Stephenie Meyer              2005
10       11     The Book Thief     Markus Zusak              2005
19       20           Twilight  Stephenie Meyer              2005
21       22           Twilight  Stephenie Meyer              2005

                 Genre  Pages  Average_Rating  Number_of_Ratings  \
2             Thriller    454            4.98              49783
7          Young Adult    498            4.24               8126
10  Historical Fiction    552            3.01              44635
19         Young Adult    498            3.75              46013
21         Young Adult    498            4.47              84911

    Number_of_Reviews
2               22513
7                3356
10              13056
19              25429
21              26331
```

7.



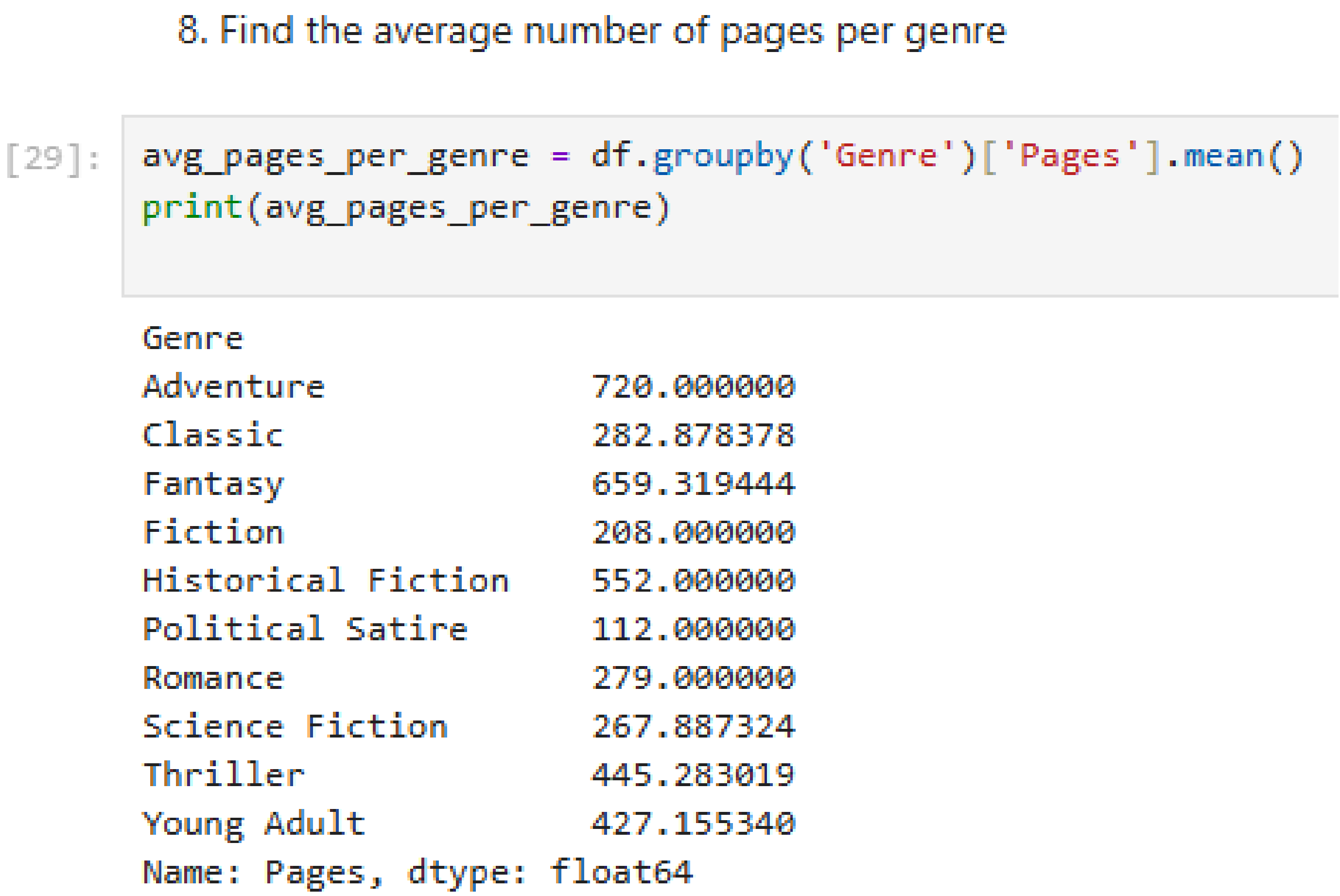
7. Find the book with the maximum number of pages

28]:

```
book_max_pages = df.loc[df['Pages'].idxmax()]
print(book_max_pages)
```

```
Book_ID                                 15
Title                The Lord of the Rings
Author                     J.R.R. Tolkien
Publication_Year                      1954
Genre                              Fantasy
Pages                                 1178
Average_Rating                        4.47
Number_of_Ratings                    44486
Number_of_Reviews                    21764
Name: 14, dtype: object
```

8.



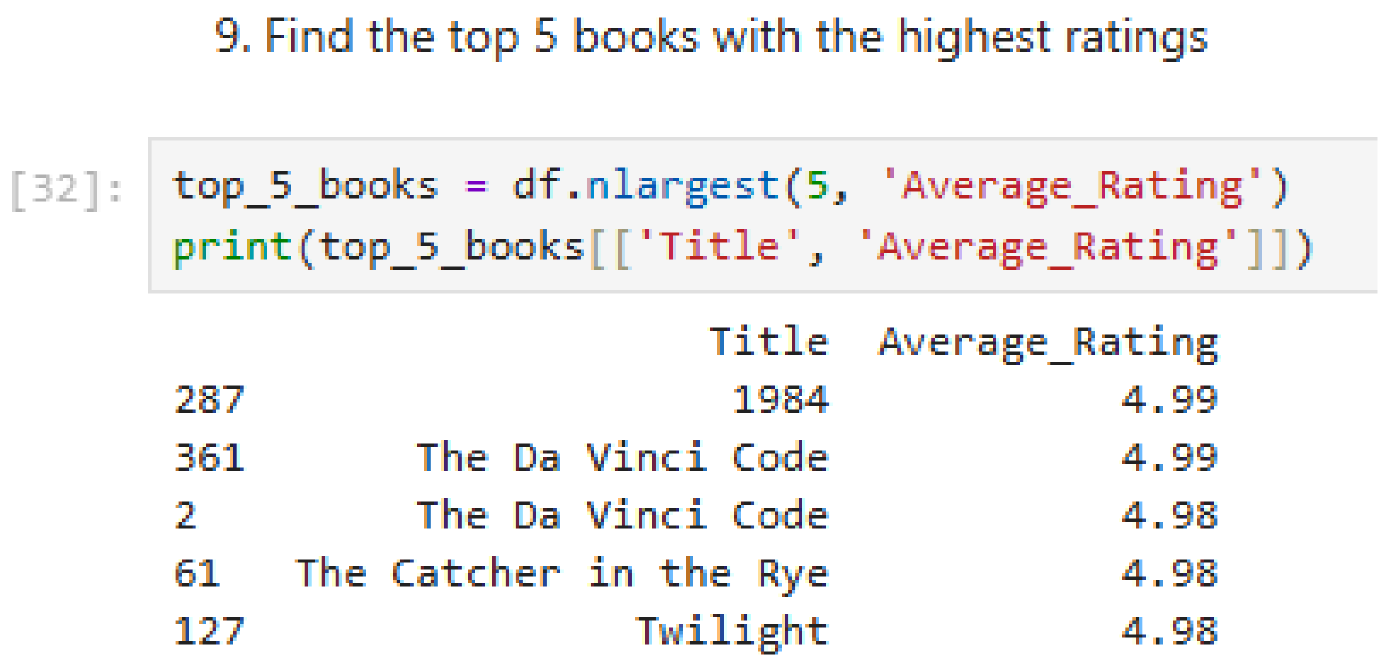
8. Find the average number of pages per genre

[29]:

```
avg_pages_per_genre = df.groupby('Genre')['Pages'].mean()
print(avg_pages_per_genre)
```

```
Genre
Adventure             720.000000
Classic               282.878378
Fantasy               659.319444
Fiction               208.000000
Historical Fiction    552.000000
Political Satire      112.000000
Romance               279.000000
Science Fiction       267.887324
Thriller              445.283019
Young Adult           427.155340
Name: Pages, dtype: float64
```

9.

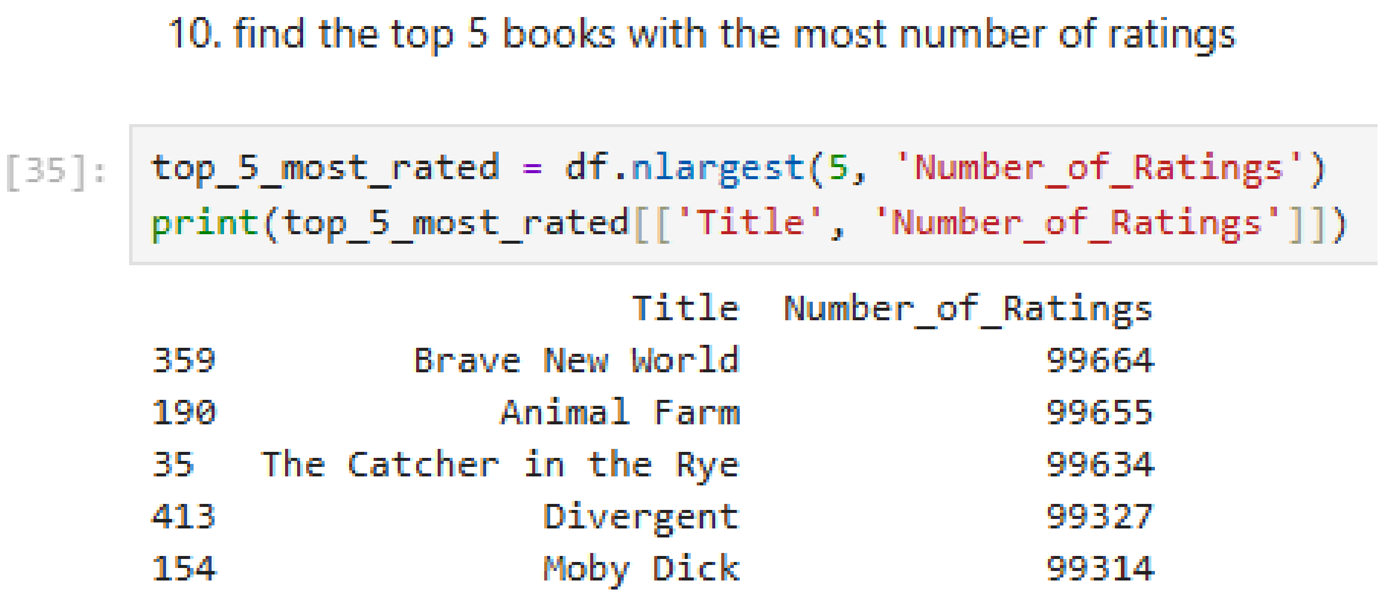


9. Find the top 5 books with the highest ratings

```
[32]: top_5_books = df.nlargest(5, 'Average_Rating')
      print(top_5_books[['Title', 'Average_Rating']])
```

```
                      Title  Average_Rating
287                    1984            4.99
361       The Da Vinci Code            4.99
2         The Da Vinci Code            4.98
61   The Catcher in the Rye            4.98
127                Twilight            4.98
```

10.

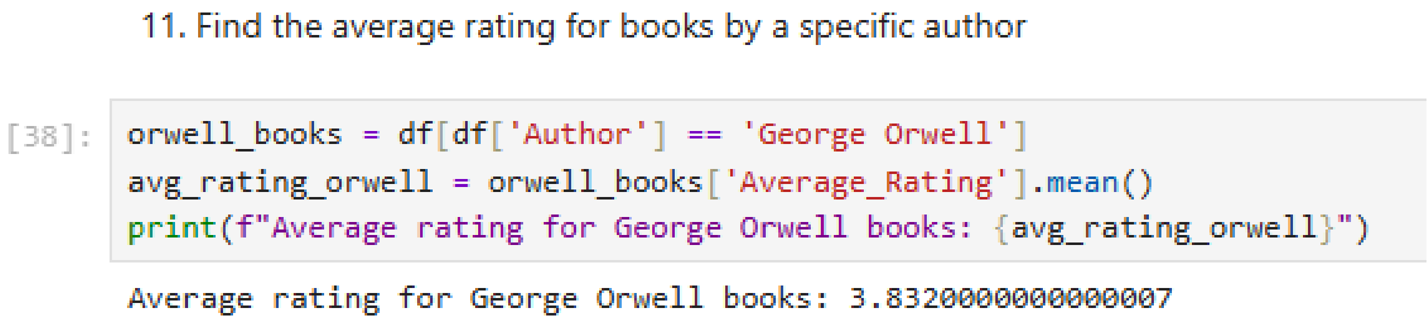


10. find the top 5 books with the most number of ratings

```
[35]: top_5_most_rated = df.nlargest(5, 'Number_of_Ratings')
      print(top_5_most_rated[['Title', 'Number_of_Ratings']])
```

```
                      Title  Number_of_Ratings
359         Brave New World              99664
190             Animal Farm              99655
35   The Catcher in the Rye              99634
413               Divergent              99327
154               Moby Dick              99314
```

11.

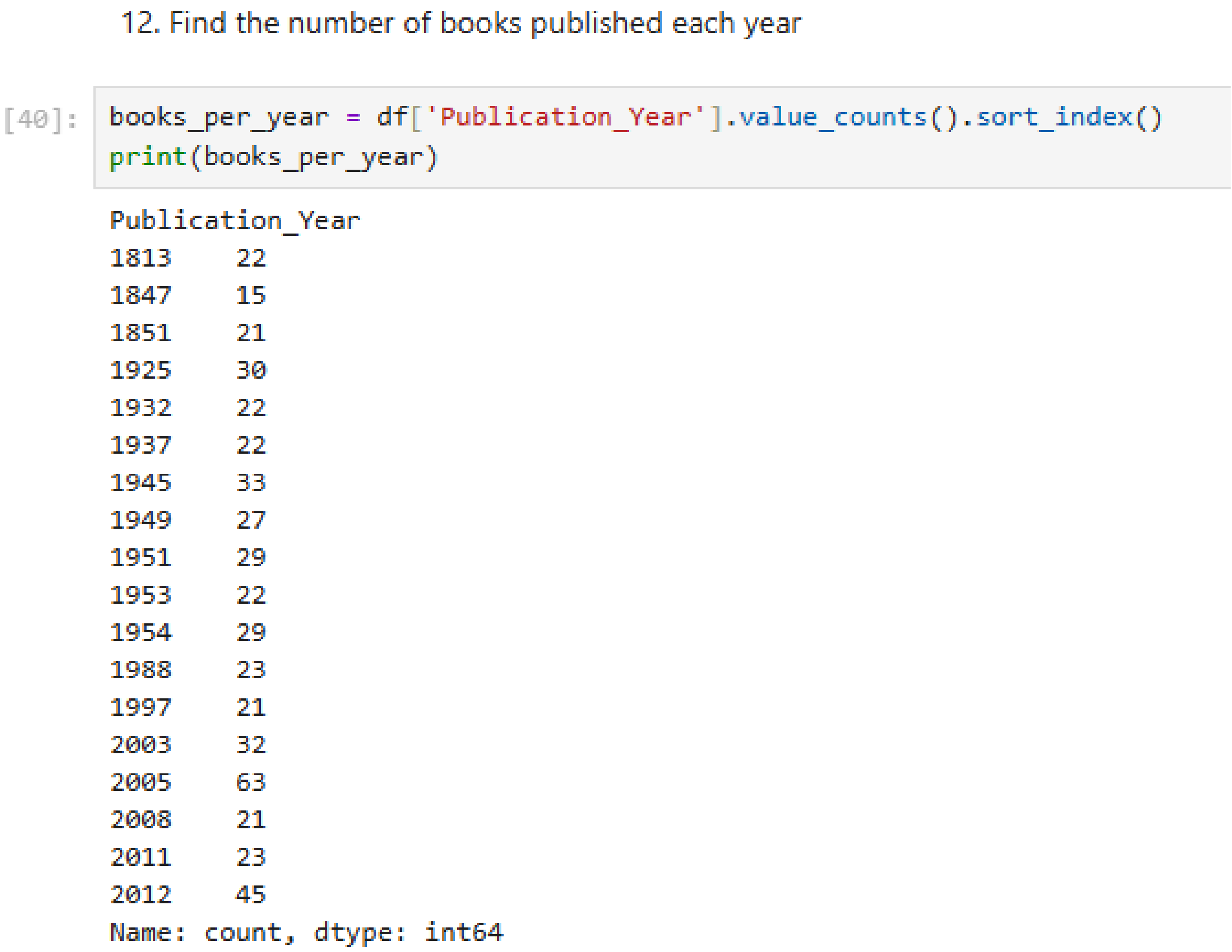


11. Find the average rating for books by a specific author

```
[38]: orwell_books = df[df['Author'] == 'George Orwell']
      avg_rating_orwell = orwell_books['Average_Rating'].mean()
      print(f"Average rating for George Orwell books: {avg_rating_orwell}")
```

```
Average rating for George Orwell books: 3.8320000000000007
```

12.

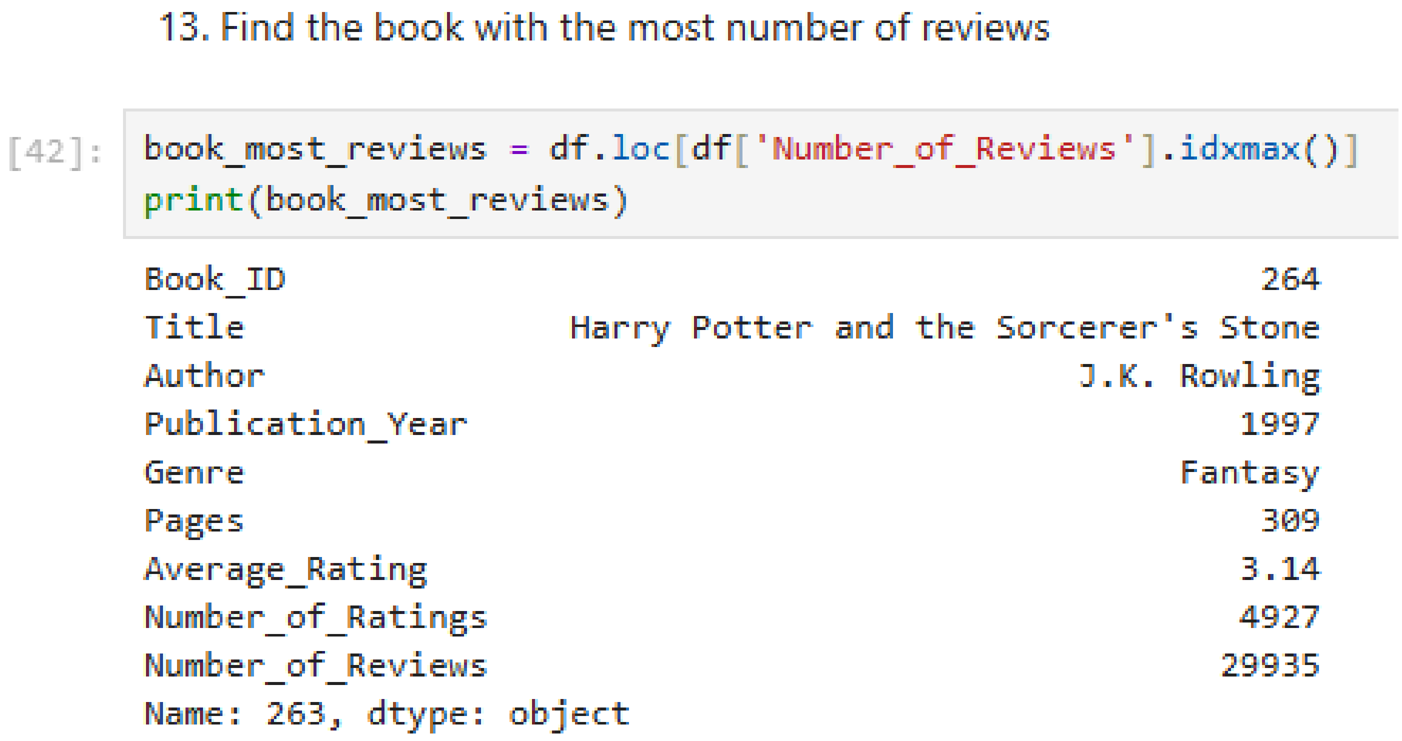


12. Find the number of books published each year

```
[40]: books_per_year = df['Publication_Year'].value_counts().sort_index()
      print(books_per_year)
```

```
Publication_Year
1813    22
1847    15
1851    21
1925    30
1932    22
1937    22
1945    33
1949    27
1951    29
1953    22
1954    29
1988    23
1997    21
2003    32
2005    63
2008    21
2011    23
2012    45
Name: count, dtype: int64
```

13.



13. Find the book with the most number of reviews

```
[42]: book_most_reviews = df.loc[df['Number_of_Reviews'].idxmax()]
      print(book_most_reviews)
```

```
Book_ID                                                264
Title                Harry Potter and the Sorcerer's Stone
Author                                        J.K. Rowling
Publication_Year                                      1997
Genre                                              Fantasy
Pages                                                  309
Average_Rating                                        3.14
Number_of_Ratings                                     4927
Number_of_Reviews                                    29935
Name: 263, dtype: object
```

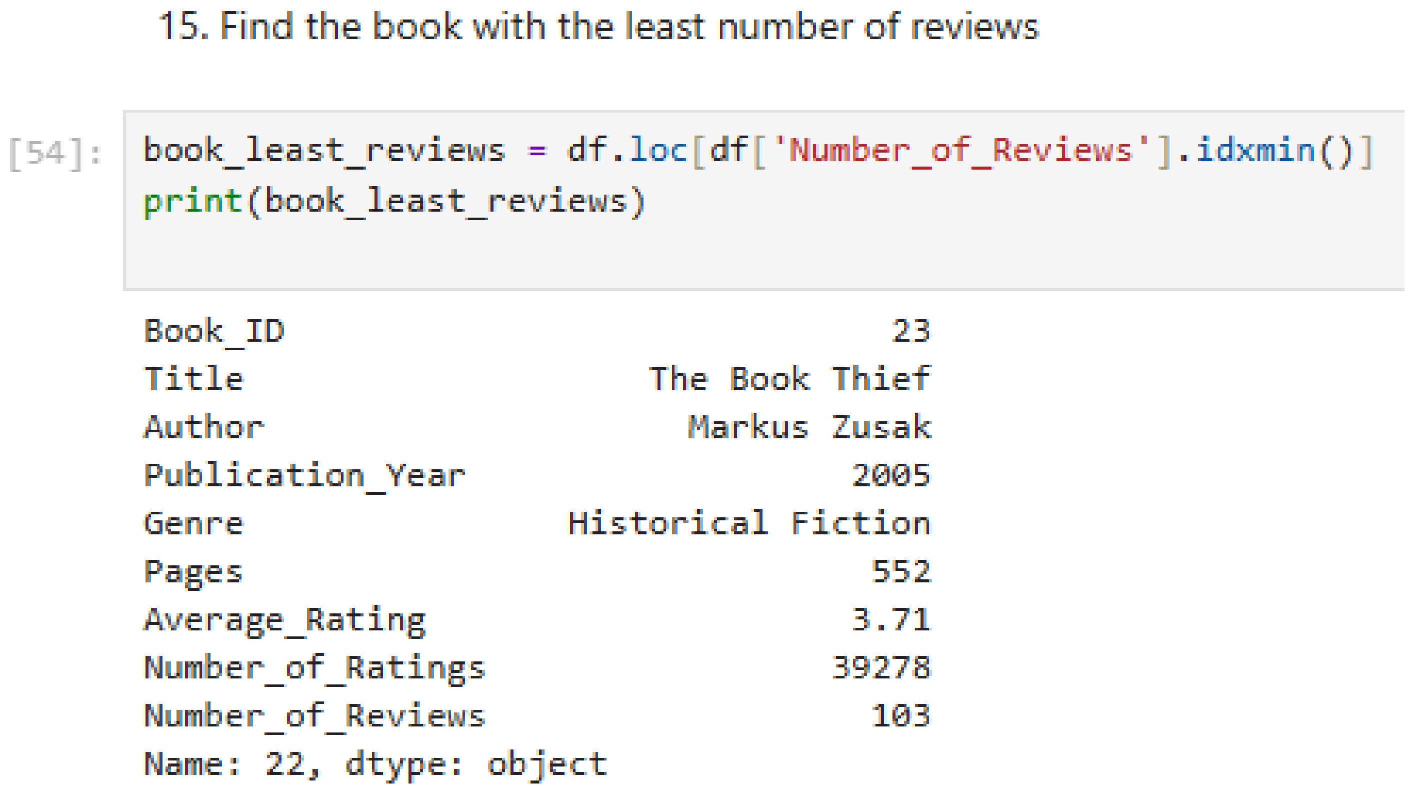
14.

14. Find the books that have been reviewed more than 29000 times

```
[53]: popular_books = df[df['Number_of_Reviews'] > 29000]
      print(popular_books[['Title', 'Number_of_Reviews']])
```

```
                                       Title  Number_of_Reviews
5                                 The Hobbit              29365
23                               Animal Farm              29804
38                    The Catcher in the Rye              29113
89                          The Great Gatsby              29349
135                           The Book Thief              29738
197                            The Alchemist              29088
205                   The Fault in Our Stars              29181
232                   The Fault in Our Stars              29327
261                      Pride and Prejudice              29184
263     Harry Potter and the Sorcerer's Stone              29935
295                                Divergent              29327
315                               The Hobbit              29351
358                        The Da Vinci Code              29142
365                        The Da Vinci Code              29042
389                         The Hunger Games              29521
437                        The Da Vinci Code              29343
443                                Moby Dick              29051
478                           The Book Thief              29104
487                         The Hunger Games              29499
492                                Moby Dick              29318
```

15.

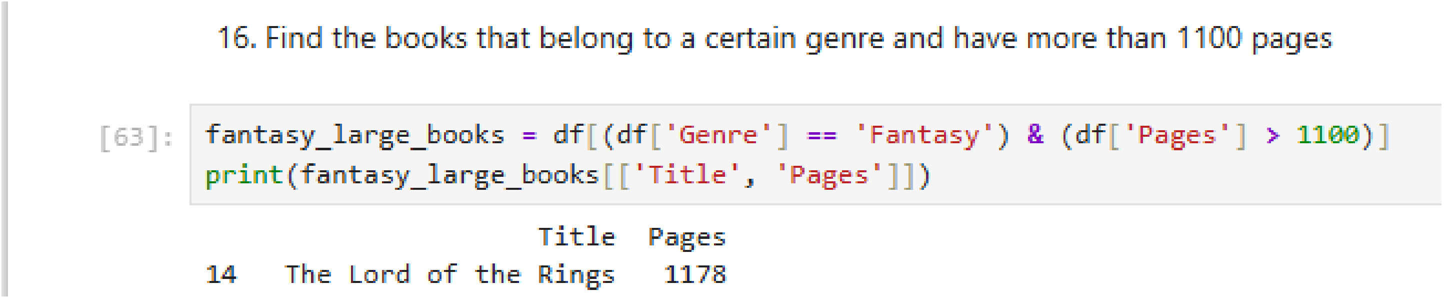


15. Find the book with the least number of reviews

```
[54]: book_least_reviews = df.loc[df['Number_of_Reviews'].idxmin()]
      print(book_least_reviews)
```

```
Book_ID                              23
Title                    The Book Thief
Author                     Markus Zusak
Publication_Year                   2005
Genre                Historical Fiction
Pages                               552
Average_Rating                     3.71
Number_of_Ratings                 39278
Number_of_Reviews                   103
Name: 22, dtype: object
```

16.

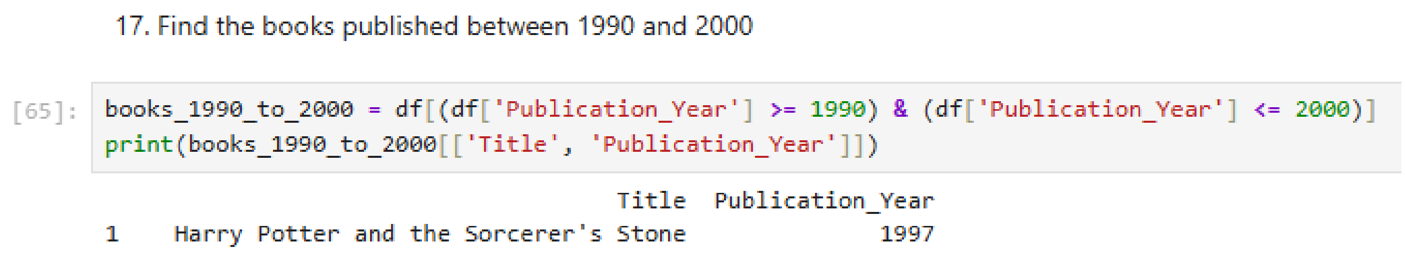


16. Find the books that belong to a certain genre and have more than 1100 pages

```
[63]: fantasy_large_books = df[(df['Genre'] == 'Fantasy') & (df['Pages'] > 1100)]
      print(fantasy_large_books[['Title', 'Pages']])
```

```
                    Title  Pages
14  The Lord of the Rings   1178
```

17.



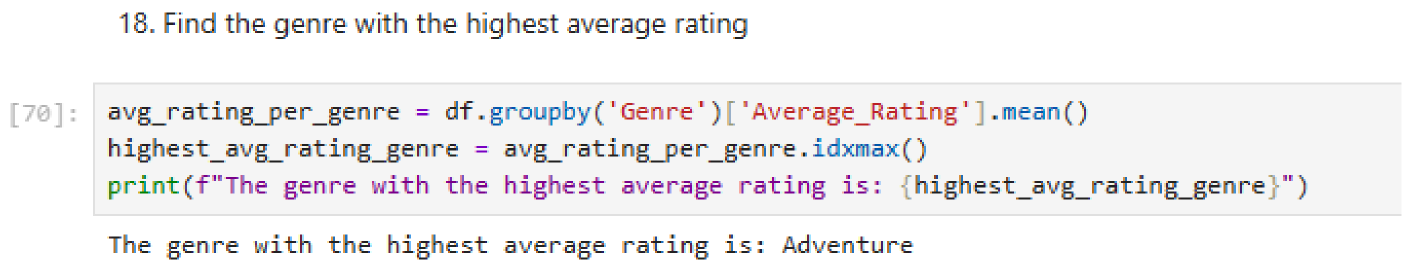
17. Find the books published between 1990 and 2000

[65]:

```
books_1990_to_2000 = df[(df['Publication_Year'] >= 1990) & (df['Publication_Year'] <= 2000)]
print(books_1990_to_2000[['Title', 'Publication_Year']])
```

```
                                Title  Publication_Year
1    Harry Potter and the Sorcerer's Stone              1997
```

18.



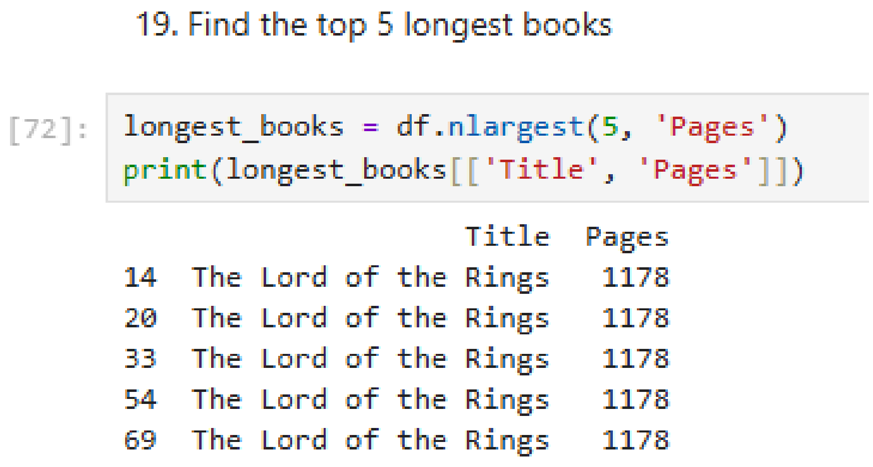
18. Find the genre with the highest average rating

[70]:

```
avg_rating_per_genre = df.groupby('Genre')['Average_Rating'].mean()
highest_avg_rating_genre = avg_rating_per_genre.idxmax()
print(f"The genre with the highest average rating is: {highest_avg_rating_genre}")
```

```
The genre with the highest average rating is: Adventure
```

19.

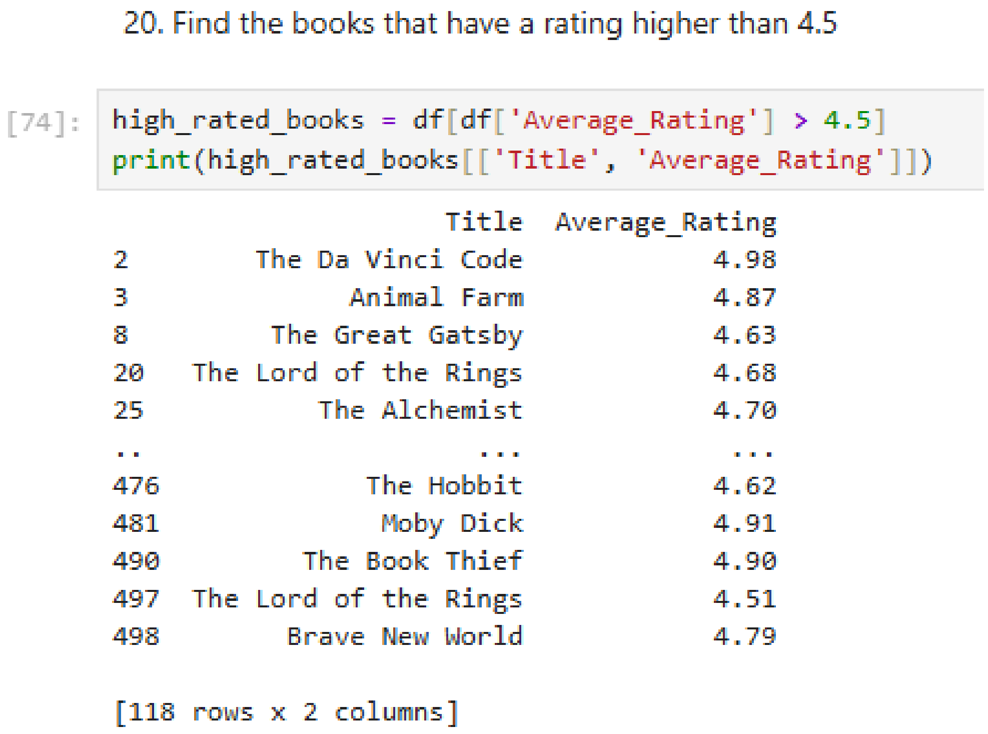


19. Find the top 5 longest books

```
[72]: longest_books = df.nlargest(5, 'Pages')
      print(longest_books[['Title', 'Pages']])
```

```
                    Title  Pages
14  The Lord of the Rings   1178
20  The Lord of the Rings   1178
33  The Lord of the Rings   1178
54  The Lord of the Rings   1178
69  The Lord of the Rings   1178
```

20.



20. Find the books that have a rating higher than 4.5

```
[74]: high_rated_books = df[df['Average_Rating'] > 4.5]
      print(high_rated_books[['Title', 'Average_Rating']])
```

```
                     Title  Average_Rating
2        The Da Vinci Code            4.98
3              Animal Farm            4.87
8         The Great Gatsby            4.63
20   The Lord of the Rings            4.68
25           The Alchemist            4.70
..                     ...             ...
476             The Hobbit            4.62
481              Moby Dick            4.91
490         The Book Thief            4.90
497  The Lord of the Rings            4.51
498        Brave New World            4.79

[118 rows x 2 columns]
```

~Thank You~