

```
1 from google.colab import drive
2 drive.mount('/content/drive')
```

Mounted at /content/drive

```
1 !cd /content/drive/MyDrive/Kaggle/
2 !ls /content/drive/MyDrive/Kaggle/
```

kaggle.json newwww.drawio 'Untitled Diagram.drawio'

```
1 !pip install skimpy
```

```
Requirement already satisfied: ipywidgets<extension=3.0.0 ||| /usr/local/lib/python3.10/dist-packages (from ipywidgets->jupyter<2.0.0,>=1.0.0->skimpy) (3.0.0)
Requirement already satisfied: jupyterlab-widgets>=1.0.0 in /usr/local/lib/python3.10/dist-packages (from ipywidgets->jupyter<2.0.0,>=1.0.0->skimpy) (3.0.8)
Requirement already satisfied: Ixml in /usr/local/lib/python3.10/dist-packages (from nbconvert->jupyter<2.0.0,>=1.0.0->skimpy) (4.9.3)
Requirement already satisfied: beautifulsoup4 in /usr/local/lib/python3.10/dist-packages (from nbconvert->jupyter<2.0.0,>=1.0.0->skimpy) (4.11.2)
Requirement already satisfied: bleach in /usr/local/lib/python3.10/dist-packages (from nbconvert->jupyter<2.0.0,>=1.0.0->skimpy) (6.0.0)
Requirement already satisfied: defusedxml in /usr/local/lib/python3.10/dist-packages (from nbconvert->jupyter<2.0.0,>=1.0.0->skimpy) (0.7.1)
Requirement already satisfied: entrypoints>=0.2.2 in /usr/local/lib/python3.10/dist-packages (from nbconvert->jupyter<2.0.0,>=1.0.0->skimpy) (0.4)
Requirement already satisfied: jinja2>=3.0 in /usr/local/lib/python3.10/dist-packages (from nbconvert->jupyter<2.0.0,>=1.0.0->skimpy) (3.1.2)
Requirement already satisfied: jupyterlab-pygments in /usr/local/lib/python3.10/dist-packages (from nbconvert->jupyter<2.0.0,>=1.0.0->skimpy) (0.2.2)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from nbconvert->jupyter<2.0.0,>=1.0.0->skimpy) (2.1.3)
Requirement already satisfied: mistune<2,>=0.8.1 in /usr/local/lib/python3.10/dist-packages (from nbconvert->jupyter<2.0.0,>=1.0.0->skimpy) (0.8.4)
Requirement already satisfied: nbclient>=0.5.0 in /usr/local/lib/python3.10/dist-packages (from nbconvert->jupyter<2.0.0,>=1.0.0->skimpy) (0.8.0)
Requirement already satisfied: nbformat>=5.1 in /usr/local/lib/python3.10/dist-packages (from nbconvert->jupyter<2.0.0,>=1.0.0->skimpy) (5.9.2)
Requirement already satisfied: pandocfilters>=1.4.1 in /usr/local/lib/python3.10/dist-packages (from nbconvert->jupyter<2.0.0,>=1.0.0->skimpy) (1.5.0)
Requirement already satisfied: tinycc2 in /usr/local/lib/python3.10/dist-packages (from nbconvert->jupyter<2.0.0,>=1.0.0->skimpy) (1.2.1)
Requirement already satisfied: argon2-cffi in /usr/local/lib/python3.10/dist-packages (from notebook->jupyter<2.0.0,>=1.0.0->skimpy) (23.1.0)
Requirement already satisfied: Send2Trash>=1.8.0 in /usr/local/lib/python3.10/dist-packages (from notebook->jupyter<2.0.0,>=1.0.0->skimpy) (18.2)
Requirement already satisfied: terminado>=0.8.3 in /usr/local/lib/python3.10/dist-packages (from notebook->jupyter<2.0.0,>=1.0.0->skimpy) (0.17.1)
Requirement already satisfied: prometheus-client in /usr/local/lib/python3.10/dist-packages (from notebook->jupyter<2.0.0,>=1.0.0->skimpy) (0.17.1)
Requirement already satisfied: nbclassic>=0.4.7 in /usr/local/lib/python3.10/dist-packages (from notebook->jupyter<2.0.0,>=1.0.0->skimpy) (1.0.0)
Collecting qtpy>=2.0.1 (from qtconsole->jupyter<2.0.0,>=1.0.0->skimpy)
  Downloading QtPy-2.3.1-py3-none-any.whl (84 kB)
```

84.9/84.9 kB 11.7 MB/s eta 0:00:00

```
Requirement already satisfied: parso<0.9.0,>=0.8.3 in /usr/local/lib/python3.10/dist-packages (from jedi>=0.16->ipython>=7.23.1->ipykernel<7.0.0,>=6.7.0->skimpy)
Requirement already satisfied: jupyter-server>=1.8 in /usr/local/lib/python3.10/dist-packages (from nbclassic>=0.4.7->notebook->jupyter<2.0.0,>=1.0.0->skimpy) (1.2)
Requirement already satisfied: notebook-shim>=0.2.3 in /usr/local/lib/python3.10/dist-packages (from nbclassic>=0.4.7->notebook->jupyter<2.0.0,>=1.0.0->skimpy) (0.2.3)
Requirement already satisfied: fastjsonschema in /usr/local/lib/python3.10/dist-packages (from nbformat>=5.1->nbconvert->jupyter<2.0.0,>=1.0.0->skimpy) (2.18.0)
Requirement already satisfied: jsonschema>=2.6 in /usr/local/lib/python3.10/dist-packages (from nbformat>=5.1->nbconvert->jupyter<2.0.0,>=1.0.0->skimpy) (4.19.0)
Requirement already satisfied: ptyprocess>=0.5 in /usr/local/lib/python3.10/dist-packages (from pexpect>4.3->ipython>=7.23.1->ipykernel<7.0.0,>=6.7.0->skimpy) (0.5)
Requirement already satisfied: wcwidth in /usr/local/lib/python3.10/dist-packages (from prompt-toolkit!=3.0.0!=3.0.1,>=2.0.0->ipython>=7.23.1->ipykernel<7.0.0,>=6.7.0->skimpy) (0.1.1)
Requirement already satisfied: argon2-cffi-bindings in /usr/local/lib/python3.10/dist-packages (from argon2-cffi->notebook->jupyter<2.0.0,>=1.0.0->skimpy) (21.2.0)
Requirement already satisfied: soupsieve>1.2 in /usr/local/lib/python3.10/dist-packages (from beautifulsoup4->nbconvert->jupyter<2.0.0,>=1.0.0->skimpy) (2.4.1)
Requirement already satisfied: webencodings in /usr/local/lib/python3.10/dist-packages (from bleach->nbconvert->jupyter<2.0.0,>=1.0.0->skimpy) (0.5.1)
Requirement already satisfied: attrs>=22.2.0 in /usr/local/lib/python3.10/dist-packages (from jsonschema>=2.6->nbformat>=5.1->nbconvert->jupyter<2.0.0,>=1.0.0->skimpy) (22.2.0)
Requirement already satisfied: jsonschema-specifications>=2023.03.6 in /usr/local/lib/python3.10/dist-packages (from jsonschema>=2.6->nbformat>=5.1->nbconvert->jupyter<2.0.0,>=1.0.0->skimpy) (2023.03.6)
Requirement already satisfied: referencing>=0.28.4 in /usr/local/lib/python3.10/dist-packages (from jsonschema>=2.6->nbformat>=5.1->nbconvert->jupyter<2.0.0,>=1.0.0->skimpy) (0.28.4)
Requirement already satisfied: rpdps-py>=0.7.1 in /usr/local/lib/python3.10/dist-packages (from jsonschema>=2.6->nbformat>=5.1->nbconvert->jupyter<2.0.0,>=1.0.0->skimpy) (0.7.1)
Requirement already satisfied: anyio<4,>=3.1.0 in /usr/local/lib/python3.10/dist-packages (from jupyter-server>=1.8->nbclassic>=0.4.7->notebook->jupyter<2.0.0,>=1.0.0->skimpy) (3.1.0)
Requirement already satisfied: websocket-client in /usr/local/lib/python3.10/dist-packages (from jupyter-server>=1.8->nbclassic>=0.4.7->notebook->jupyter<2.0.0,>=1.0.0->skimpy) (1.0.1)
Requirement already satisfied: cffi>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from argon2-cffi-bindings->argon2-cffi->notebook->jupyter<2.0.0,>=1.0.0->skimpy) (1.0.1)
Requirement already satisfied: idna>=2.8 in /usr/local/lib/python3.10/dist-packages (from anyio<4,>=3.1.0->jupyter-server>=1.8->nbclassic>=0.4.7->notebook->jupyter<2.0.0,>=1.0.0->skimpy) (2.8)
Requirement already satisfied: sniffio>=1.1 in /usr/local/lib/python3.10/dist-packages (from anyio<4,>=3.1.0->jupyter-server>=1.8->nbclassic>=0.4.7->notebook->jupyter<2.0.0,>=1.0.0->skimpy) (1.1)
Requirement already satisfied: exceptiongroup in /usr/local/lib/python3.10/dist-packages (from anyio<4,>=3.1.0->jupyter-server>=1.8->nbclassic>=0.4.7->notebook->jupyter<2.0.0,>=1.0.0->skimpy) (0.1.0)
Requirement already satisfied: pyparser in /usr/local/lib/python3.10/dist-packages (from cffi>=1.0.1->argon2-cffi-bindings->argon2-cffi->notebook->jupyter<2.0.0,>=1.0.0->skimpy) (0.1.0)
Installing collected packages: tzdata, typeguard, qtpy, jedi, comm, pandas, ipykernel, qtconsole, jupyter, skimpy
Attempting uninstall: pandas
  Found existing installation: pandas 1.5.3
  Uninstalling pandas-1.5.3:
    Successfully uninstalled pandas-1.5.3
Attempting uninstall: ipykernel
  Found existing installation: ipykernel 5.5.6
  Uninstalling ipykernel-5.5.6:
    Successfully uninstalled ipykernel-5.5.6

```

**ERROR:** pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the source of the following dependency conflict:  
google-colab 1.0.0 requires ipykernel==5.5.6, but you have ipykernel 6.25.1 which is incompatible.  
google-colab 1.0.0 requires pandas==1.5.3, but you have pandas 2.0.3 which is incompatible.

Successfully installed comm-0.1.4 ipykernel-6.25.1 jedi-0.19.0 jupyter-1.0.0 pandas-2.0.3 qtconsole-5.4.3 qtpy-2.3.1 skimpy-0.0.10 typeguard-2.13.3 tzdata-2023.3

```
1 !mkdir ~/.kaggle
2 !touch ~/.kaggle/kaggle.json
3
4 api_token = {"username": "mahimathakur", "key": "7503a502a434f3f056a256ba01ffa25d"}
5
6 import json
7
8 with open('/root/.kaggle/kaggle.json', 'w') as file:
9     json.dump(api_token, file)
```

```

10
11 !chmod 600 ~/.kaggle/kaggle.json

```

```
1 !kaggle datasets download -d mdhamani/goodreads-books-100k
```

Downloading goodreads-books-100k.zip to /content  
 75% 33.0M/44.2M [00:00<00:00, 136MB/s]  
 100% 44.2M/44.2M [00:00<00:00, 148MB/s]

```

1 from zipfile import ZipFile
2
3 zip = ZipFile('goodreads-books-100k.zip')
4 zip.extractall()

```

```
1 !pip install colorama
```

Collecting colorama  
 Downloading colorama-0.4.6-py3.py3-none-any.whl (25 kB)  
 Installing collected packages: colorama  
 Successfully installed colorama-0.4.6

```

1 import pandas as pd
2 import seaborn as sns
3 import matplotlib.pyplot as plt
4 import numpy as np
5 from PIL import Image as im
6 from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator
7 !pip install colorama
8 %matplotlib inline
9
10 from colorama import Fore, Back, Style
11 red = Fore.RED
12 grn = Fore.GREEN
13 blu = Fore.BLUE
14 ylw = Fore.YELLOW
15 wht = Fore.WHITE
16 mag = Fore.MAGENTA
17
18 palette = sns.color_palette("bright", 15)
19 sns.palplot(palette)
20 sns.set_palette("Paired")

```

Requirement already satisfied: colorama in /usr/local/lib/python3.10/dist-packages (0.4.6)



Double-click (or enter) to edit

```

1 import re
2 import numpy as np
3 import pandas as pd
4 import plotly.graph_objects as go
5 import plotly.express as px
6 import plotly.colors as colors
7 from wordcloud import WordCloud
8 from sklearn.neighbors import NearestNeighbors
9 import warnings
10 warnings.filterwarnings("ignore")

```

```
1
```

```
1 !ls
```

```
drive GoodReads_100k_books.csv goodreads-books-100k.zip sample_data
```

```

1 data = pd.read_csv('GoodReads_100k_books.csv')
2 data[:50]

```

	author	bookformat	desc	genre	img	is
0	Laurence M. Hauptman	Hardcover	Reveals that several hundred thousand Indians ...	History,Military History,Civil War,American Hi...	<a href="https://i.gr-assets.com/images/S/compressed.ph...">https://i.gr-assets.com/images/S/compressed.ph...</a>	00291418
1	Charlotte Fiell,Emmanuelle Dirix	Paperback	Fashion Sourcebook - 1920s is the first book i...	Couture,Fashion,Historical,Art,Nonfiction	<a href="https://i.gr-assets.com/images/S/compressed.ph...">https://i.gr-assets.com/images/S/compressed.ph...</a>	19068634
2	Andy Anderson	Paperback	The seminal history and analysis of the Hungar...	Politics,History	<a href="https://i.gr-assets.com/images/S/compressed.ph...">https://i.gr-assets.com/images/S/compressed.ph...</a>	9489841
3	Carlotta R. Anderson	Hardcover	"All-American Anarchist" chronicles the life a...	Labor,History	<a href="https://i.gr-assets.com/images/S/compressed.ph...">https://i.gr-assets.com/images/S/compressed.ph...</a>	8143270
4	Jean Leveille	NaN	Aujourdâ€™hui, lâ€™oiseau nous invite à sa ta...	NaN	<a href="https://i.gr-assets.com/images/S/compressed.ph...">https://i.gr-assets.com/images/S/compressed.ph...</a>	27619208
5	Jeffrey Pfeffer	Hardcover	Why is common sense so uncommon when it comes ...	Business,Leadership,Romance,Historical Romance...	<a href="https://i.gr-assets.com/images/S/compressed.ph...">https://i.gr-assets.com/images/S/compressed.ph...</a>	8758484
6	Jeffrey Pfeffer	Paperback	"Competitive Advantage Through People" explore...	Business,Leadership,Business,Management,Romanc...	<a href="https://i.gr-assets.com/images/S/compressed.ph...">https://i.gr-assets.com/images/S/compressed.ph...</a>	08758471
7	Edward Joesting	Paperback	"Even if you know Hawaiian history you will fi...	History,Nonfiction	<a href="https://i.gr-assets.com/images/S/compressed.ph...">https://i.gr-assets.com/images/S/compressed.ph...</a>	3930090
8	Nick Le Neve Walmsley	Paperback	At the time of her construction in the late 19...	NaN	<a href="https://i.gr-assets.com/images/S/compressed.ph...">https://i.gr-assets.com/images/S/compressed.ph...</a>	7524568
9	B. Alan Wallace,Dalai Lama XIV	Hardcover	Discover your personal path to bliss,"This boo...	Religion,Buddhism,Philosophy,Spirituality,Psyc...	<a href="https://i.gr-assets.com/images/S/compressed.ph...">https://i.gr-assets.com/images/S/compressed.ph...</a>	04714698
10	Mark Verman	Paperback	The earliest medieval Jewish mystical writings...	NaN	<a href="https://i.gr-assets.com/images/S/compressed.ph...">https://i.gr-assets.com/images/S/compressed.ph...</a>	7914072
11	Brian Morris	Paperback	In this important, scholarly and wide-ranging ...	Anthropology,Religion,Nonfiction,Academic,Soci...	<a href="https://i.gr-assets.com/images/S/compressed.ph...">https://i.gr-assets.com/images/S/compressed.ph...</a>	05213399
12	Graham Purchase	Paperback	This book outlines the history of our slow ali...	Biology,Ecology,Nonfiction,Politics	<a href="https://i.gr-assets.com/images/S/compressed.ph...">https://i.gr-assets.com/images/S/compressed.ph...</a>	15516402
13	Richard Layard	Paperback	There is a paradox at the heart of our lives. ...	Psychology,Nonfiction,Economics,Science,Philos...	<a href="https://i.gr-assets.com/images/S/compressed.ph...">https://i.gr-assets.com/images/S/compressed.ph...</a>	1430370
14	Graham Purchase	Paperback	In this wide-ranging book, Graham Purchase, on...	Biology,Ecology	<a href="https://i.gr-assets.com/images/S/compressed.ph...">https://i.gr-assets.com/images/S/compressed.ph...</a>	9613289
15	Philip Mansel	Hardcover	Throughout history rulers have used clothes as...	History,Couture,Fashion,Nonfiction,Historical...	<a href="https://i.gr-assets.com/images/S/compressed.ph...">https://i.gr-assets.com/images/S/compressed.ph...</a>	3001069
16	Umberto Eco	Hardcover	In the course of the long debate on the nature...	History	<a href="https://i.gr-assets.com/images/S/compressed.ph...">https://i.gr-assets.com/images/S/compressed.ph...</a>	90272329

17	Helen Fremont	Paperback	"To this day, I don't even know what my mother..."	Autobiography,Memoir,Nonfiction,World War II,H...	<a href="https://i.gr-assets.com/images/S/compressed.ph...">https://i.gr-assets.com/images/S/compressed.ph...</a>	3853337
18	William James	Paperback	This is the first inexpensive edition of the c...	Psychology,Philosophy,Nonfiction,Science,Class...	<a href="https://i.gr-assets.com/images/S/compressed.ph...">https://i.gr-assets.com/images/S/compressed.ph...</a>	4862038
19	Dan Sleigh,AndrÃ© Brink	Hardcover	This novel of epic proportions from South Afr...	Historical,Historical Fiction,Cultural,Africa,...	<a href="https://i.gr-assets.com/images/S/compressed.ph...">https://i.gr-assets.com/images/S/compressed.ph...</a>	01510111
20	Christopher Peterson,Martin E.P. Seligman	Hardcover	Character has become a front-and-center topic ...	Psychology,Philosophy,Nonfiction,Reference,Sel...	<a href="https://i.gr-assets.com/images/S/compressed.ph...">https://i.gr-assets.com/images/S/compressed.ph...</a>	1951670
21	Christopher Morley	Paperback	Roger Mifflin is part pixie, part sage, part n...	Fiction,Writing,Books About Books,Classics,Hum...	<a href="https://i.gr-assets.com/images/S/compressed.ph...">https://i.gr-assets.com/images/S/compressed.ph...</a>	14142706
22	Fernando Henrique Cardoso,Brian Winter	Paperback	Fernando Henrique Cardoso received a phone cal...	History,Cultural,Brazil,Nonfiction,Biography,P...	<a href="https://i.gr-assets.com/images/S/compressed.ph...">https://i.gr-assets.com/images/S/compressed.ph...</a>	15864832
23	Gustavo Gorriti	Paperback	First published in Peru in 1990, ,The Shining ...	History,Nonfiction,Politics	<a href="https://i.gr-assets.com/images/S/compressed.ph...">https://i.gr-assets.com/images/S/compressed.ph...</a>	8078467
24	Robert J. Shiller	Paperback	As Robert Shiller's new 2009 preface to his ...	Economics,Economics,Finance,Business,Nonfictio...	<a href="https://i.gr-assets.com/images/S/compressed.ph...">https://i.gr-assets.com/images/S/compressed.ph...</a>	7679236
25	Marilyn Butler	Paperback	It is often said that Jane Austen in the count...	Criticism,Literary Criticism,Nonfiction,Critic...	<a href="https://i.gr-assets.com/images/S/compressed.ph...">https://i.gr-assets.com/images/S/compressed.ph...</a>	1981296
26	Mark Lee	Paperback	Photojournalist Nicky Bettencourt thinks he's ...	Fiction,Cultural,Australia,Eastern Africa,Ugan...	<a href="https://i.gr-assets.com/images/S/compressed.ph...">https://i.gr-assets.com/images/S/compressed.ph...</a>	1560295
27	Leo Bagrow,Raleigh Ashlin Skelton	Hardcover	This illustrated work is intended to acquaint ...		NaN	9137503
28	B. Mark Smith	Paperback	The stock market is big news today. Over 50 pe...	Economics,Finance,Economics,History,Finance,Pe...	<a href="https://i.gr-assets.com/images/S/compressed.ph...">https://i.gr-assets.com/images/S/compressed.ph...</a>	3745285
29	Rocky McElveen	Paperback	In ,Wild Men, Wild Alaska, professional huntin...	Nonfiction,Nature,Outdoors,Christian,Environme...	<a href="https://i.gr-assets.com/images/S/compressed.ph...">https://i.gr-assets.com/images/S/compressed.ph...</a>	7852890
30	Luc Milne	Paperback	NaN	Erotica,Gay Erotica	<a href="https://i.gr-assets.com/images/S/compressed.ph...">https://i.gr-assets.com/images/S/compressed.ph...</a>	9435956
31	Carly Mine,Carly Mine	Paperback	When adult entertainment first appeared on the...	Nonfiction,Sociology,Sexuality,Sex Work,Femini...	<a href="https://i.gr-assets.com/images/S/compressed.ph...">https://i.gr-assets.com/images/S/compressed.ph...</a>	7867155
32	Martin Evans	Hardcover	Invaded in 1830, populated by one million sett...	History,Cultural,Africa,Northern Africa,Algeri...	<a href="https://i.gr-assets.com/images/S/compressed.ph...">https://i.gr-assets.com/images/S/compressed.ph...</a>	1928035
33	Jerome Murphy-O'Connor	Paperback	For someone who has exercised such a profound ...	Religion,Religion/Theology,Christianity,New Te...	<a href="https://i.gr-assets.com/images/S/compressed.ph...">https://i.gr-assets.com/images/S/compressed.ph...</a>	1992838
34	John H. Dunning	Paperback	Between 1790 and 1840.		<a href="https://i.gr-assets.com/images/S/compressed.ph...">https://i.gr-assets.com/images/S/compressed.ph...</a>	6701056

34	Sumpter Priddy	Hardcover	millions of middle-clas...	NaN	assets.com/images/S/compressed.ph...	https://i.gr-assets.com/images/S/compressed.ph...	9159925
35	Marion Weinstein	Paperback	Marion Weinstein was one of the first witches ...	Religion,Wicca,Witchcraft,Spirituality,Nonfict...	assets.com/images/S/compressed.ph...	https://i.gr-assets.com/images/S/compressed.ph...	15641463
36	Wilford C. Wood	Hardcover	NaN	NaN	NaN	NaN	N
37	Ovid,David Malouf,James Michie	Paperback	NaN	Poetry,Classics,Nonfiction,Philosophy,Romance,...	assets.com/images/S/compressed.ph...	https://i.gr-assets.com/images/S/compressed.ph...	3757611
38	Ronald Jackson II	ebook	NaN	NaN	NaN	https://i.gr-assets.com/images/S/compressed.ph...	7914823
39	Richard Allen	Paperback	NaN	NaN	NaN	https://i.gr-assets.com/images/S/compressed.ph...	9159925
40	Janice G. Raymond	Paperback	Fifteen years ago, when it was first published...	Feminism,Nonfiction,Gender,GLBT,Queer,LGBT,LGB...	assets.com/images/S/compressed.ph...	https://i.gr-assets.com/images/S/compressed.ph...	8077627
41	Joseph Staten	Paperback	This is how it began....It is the year 2524. H...	Science Fiction,Fiction,Games,Video Games,Spor...	assets.com/images/S/compressed.ph...	https://i.gr-assets.com/images/S/compressed.ph...	7653156
42	Alice A. Carter	Paperback	A beautiful art book and a richly illustrated ...	Art,Biography,Nonfiction,Art,Art History,Histo...	assets.com/images/S/compressed.ph...	https://i.gr-assets.com/images/S/compressed.ph...	8109906
43	Hong Ying,Nicky Harman,Henry Zhao	Paperback	Set in 1930s China, this is a true but tragic ...	Fiction,Historical,Historical Fiction,Asian Li...	assets.com/images/S/compressed.ph...	https://i.gr-assets.com/images/S/compressed.ph...	7145307
44	Renate Klein,Janice G. Raymond,Lynette Dumble	NaN	À classic text for health activists and femin...	Feminism	assets.com/images/S/compressed.ph...	https://i.gr-assets.com/images/S/compressed.ph...	9630083
45	Janice G. Raymond	Paperback	'A Passion for Friends is not a sentimental hy...	Feminism,Nonfiction,Philosophy,Feminism,Womens...	assets.com/images/S/compressed.ph...	https://i.gr-assets.com/images/S/compressed.ph...	8070673
46	Brian G. Hedges,Donald S. Whitney	Paperback	Christ Formed in You is a book about spiritual...	Religion/Theology,Christian,Christian Living,C...	assets.com/images/S/compressed.ph...	https://i.gr-assets.com/images/S/compressed.ph...	09824387
47	Richard L. Kagan	Hardcover	This engrossing book examines the particular i...	History	assets.com/images/S/compressed.ph...	https://i.gr-assets.com/images/S/compressed.ph...	3000831
48	Richard L. Kagan	Hardcover	Setting aside the pastiche of bullfighters and...	NaN	assets.com/images/S/compressed.ph...	https://i.gr-assets.com/images/S/compressed.ph...	2520272
49	Louis L'Amour	Audiobook	Louis L'Amour said the West was no place for th...	Westerns,Fiction,Historical,Historical Fiction...	assets.com/images/S/compressed.ph...	https://i.gr-assets.com/images/S/compressed.ph...	14332020

1 df = data

```
1 from skimpy import skim
2
3 skim(df)
```

Data Summary			Data Types						
dataframe	Values	Column Type	Count						
Number of rows	100000	string	9						
Number of columns	13	int64	3						
		float64	1						
number									
column_name	NA	NA %	mean	sd	p0	p25	p75	p100	hist
pages	0	0	260	370	0	140	340	70000	
rating	0	0	3.8	0.62	0	3.7	4.1	5	
reviews	0	0	180	1400	0	3	67	160000	
totalratings	0	0	3000	36000	0	31	740	3800000	
string									
column_name	NA	NA %	words per row			total words			
author	0	0				2.9			
bookformat	3228	3.23				290091			
desc	6772	6.77				290091			
genre	10467	10.47				290091			
img	3045	3.04				290091			
isbn	14482	14.48				290091			
isbn13	11435	11.44				290091			
link	0	0				290091			
title	1	0				290091			

```
1 data.isnull().sum()
```

```
author      0
bookformat  3228
desc       6772
genre     10467
img        3045
isbn      14482
isbn13    11435
link        0
pages      0
rating      0
reviews      0
title       1
totalratings 0
dtype: int64
```

```
1 def clean(feature):
2     cleaned_feature = list()
3     for feat in data[feature]:
4         if len(feat.split(',')) > 1:
5             temp = feat.split(',')
6             for a in temp:
7                 cleaned_feature.append(a)
8         else:
9             cleaned_feature.append(feat)
10    return cleaned_feature
```

```
1 def plot_distribution(data,feat,color,alpha=1):
2     sns.set(font_scale=1);
3     sns.distplot(data[feat],color=color,kde=True,aspect=15/10,alpha = alpha,);
4     print(f'{red}Max value of {feat} is {data[feat].max()}\n{blue}Min value of {feat} is {data[feat].min()}\n{green}Std value of {feat} is {data[feat].std()}\n{yellow}Mean value of {feat} is {data[feat].mean()}'
```

```
1 all_authors = clean('author')
2 authors = pd.DataFrame(all_authors, columns = ['authors'])
3 author_counts = authors.value_counts()
4 authors['books_count'] = authors['authors'].apply(lambda x: author_counts[x])
5 sorted_authors = authors.sort_values(by='books_count',ascending=False)
6 sorted_authors = sorted_authors.drop_duplicates(['authors'])
7 sorted_authors = sorted_authors.reset_index(drop = True)
```

```
1 sorted_authors
```

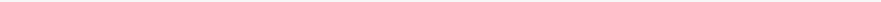
	authors	books_count	grid
0	Mi-Ri Hwang	137	grid
1	Anonymous	126	grid
2	Brian Michael Bendis	97	
3	H.P. Lovecraft	92	
4	Joss Whedon	91	
...	...	...	
87305	The Historical Society of Frankford	1	
87306	Brian H. Harris	1	
87307	Grant W. Reid	1	
87308	Frank Stephenson	1	

```

1 # Python program to generate WordCloud
2
3 # importing all necessary modules
4 from wordcloud import WordCloud, STOPWORDS
5 import matplotlib.pyplot as plt
6 import pandas as pd
7
8 comment_words = ""
9 stopwords = set(STOPWORDS)
10
11 # iterate through the csv file
12 for author in sorted_authors.authors:
13
14     # typecaste each val to string
15     author = str(author)
16
17     # split the value
18     tokens = author.split()
19
20     # Converts each token into lowercase
21     for i in range(len(tokens)):
22         tokens[i] = tokens[i].lower()
23
24     comment_words += " ".join(tokens)+" "
25
26 wordcloud = WordCloud(width = 800, height = 800,
27                       background_color ='white',
28                       stopwords = stopwords,
29                       min_font_size = 10).generate(comment_words)
30
31 # plot the WordCloud image
32 plt.figure(figsize = (8, 8), facecolor = None)
33 plt.imshow(wordcloud)
34 plt.axis("off")
35 plt.tight_layout(pad = 0)
36
37 plt.show()
38

```



1  100% 20% 40% 60% 80% 100%

1 . harri ' e - kevin ' elur a a c in a

```
1 from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator
```

```
1 data['genre'] = data['genre'].fillna('None')
2 all_genres = clean('genre')
3 genres = pd.DataFrame(all_genres, columns = ['genres'])
4 genre_counts = genres.value_counts()
5 genres['genre_count'] = genres['genres'].apply(lambda x: genre_counts[x])
6 sorted_genres = genres.sort_values(by='genre_count', ascending=False)
7 sorted_genres = sorted_genres.drop_duplicates(['genres'])
8 sorted_genres = sorted_genres.reset_index(drop=True)
```

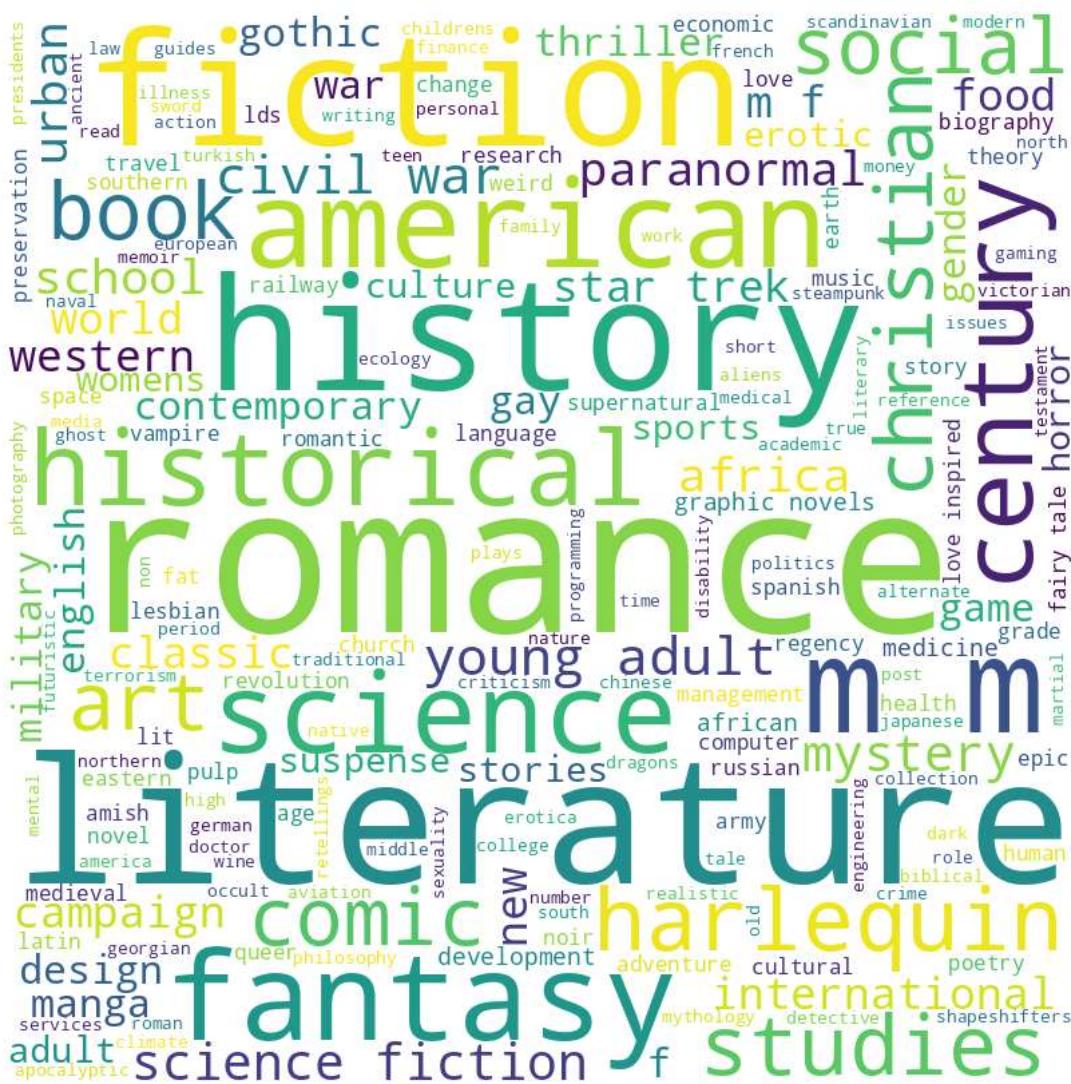
1 sorted genres

	genres	genre_count	
0	Romance	34324	
1	Fantasy	30798	
2	Fiction	29743	
3	Nonfiction	29446	
4	Historical	18183	
...	...	...	
1178	Scientific Method	1	
1179	Foster Parents	1	
1180	Islamic Terrorism	1	
1181	Nairobi	1	
1182	...	1	

1183 rows × 2 columns

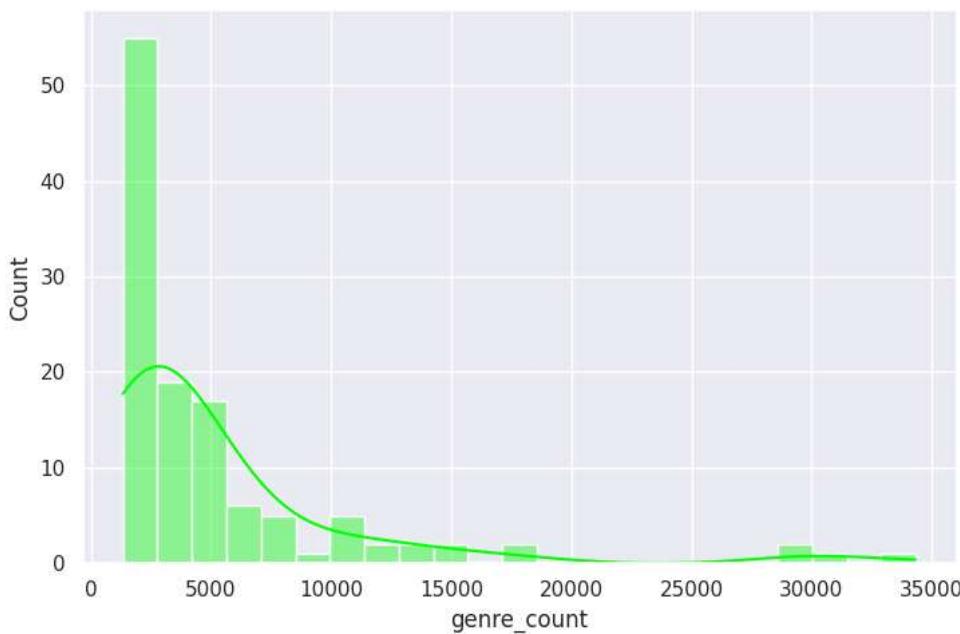
```
1 # Python program to generate WordCloud  
2  
3 # importing all necessary modules  
4 # from wordcloud import WordCloud, STOPWORDS  
5 # import matplotlib.pyplot as plt  
6 # import pandas as pd  
7  
8 comment_words = "
```

```
9 stopwords = set(STOPWORDS)
10
11 # iterate through the csv file
12 for genre in sorted_genres.genres:
13
14     # typecaste each val to string
15     genre = str(genre)
16
17     # split the value
18     tokens = genre.split()
19
20     # Converts each token into lowercase
21     for i in range(len(tokens)):
22         tokens[i] = tokens[i].lower()
23
24     comment_words += " ".join(tokens)+" "
25
26 wordcloud = WordCloud(width = 800, height = 800,
27                         background_color ='white',
28                         stopwords = stopwords,
29                         min_font_size = 10).generate(comment_words)
30
31 # plot the WordCloud image
32 plt.figure(figsize = (8, 8), facecolor = None)
33 plt.imshow(wordcloud)
34 plt.axis("off")
35 plt.tight_layout(pad = 0)
36
37 plt.show()
38
```



```
1 plot_distribution(sorted_genres[:120],'genre_count','lime',0.4);
```

```
Max value of genre_count is 34324
Min value of genre_count is 1345
Std value of genre_count is 5995.688840041371
Mean value of genre_count is 5257.575
Median value of genre_count is 3043.5
```



```
1 data.rating.mean()
```

```
3.8330552
```

```
1 # What are the most frequently repeated books on the list? (Top 20 books)
2 top_20_books = data['title'].value_counts()[:20]
3
4 # Define a color palette with enough unique colors
5 color_palette = colors.qualitative.Plotly
6
7 # Create a list of distinct colors for each bar
8 bar_colors = [color_palette[i % len(color_palette)] for i in range(len(top_20_books))]
9
10 # Create a Plotly bar plot
11 fig = go.Figure(data=[go.Bar(
12     x=top_20_books,
13     y=data.index,
14     orientation='h',
15     marker=dict(color=bar_colors) # Set the bar colors using the list of colors
16 )])
17
18 # Update the layout with dark background and white font
19 fig.update_layout(
20     plot_bgcolor='rgb(17, 17, 17)', # Dark background color
21     paper_bgcolor='rgb(17, 17, 17)', # Dark background color
22     font=dict(color='white'), # White font color
23     title="Most Occurring Books",
24     xaxis_title="Number of occurrences",
25     yaxis_title="Books"
26 )
27
28 # Show the plot
29 fig.show()
```



```

1 top_20_most_rated_books = data.sort_values('totalratings', ascending=False).head(20)
2
3 # Define the number of bars and obtain a color palette with enough unique colors
4 num_bars = len(top_20_most_rated_books)
5 color_palette = colors.qualitative.Set3
6
7 # Create a list of distinct colors for each bar
8 bar_colors = color_palette[:num_bars]
9
10 # Create a Plotly bar plot
11 fig = go.Figure(data=[go.Bar(
12     x=top_20_most_rated_books['totalratings'],
13     y=data['title'],
14     orientation='h',
15     marker=dict(color=bar_colors) # Set the bar colors using the list of colors
16 )])
17
18 # Update the layout with dark background and white font
19 fig.update_layout(
20     plot_bgcolor='rgb(17, 17, 17)', # Dark background color
21     paper_bgcolor='rgb(17, 17, 17)', # Dark background color
22     font=dict(color='white'), # White font color
23     title="Top 20 Most Rated Books",
24     xaxis_title="Number of Ratings",
25     yaxis_title="Book Title"
26 )
27
28 # Show the plot
29 fig.show()

```



```
1 top_rated_books = data[data['rating'] > 3.7]
2 top_rated_books = top_rated_books.sort_values(by='rating', ascending=False).head(20)
```

```
1 top_rated_books
```

	author	bookformat	desc	genre	img	isbn	isbn13
92610	Jack Griggs	Paperback	Owls, Hawks, Swifts & Swallows, Songbirds, Hum...	None	https://i.gr-assets.com/images/S/compressed.ph...	60533366	9.78006E+12
43539	Richard McKinney	Paperback	When man first walked the Earth, he learned th...	None	https://i.gr-assets.com/images/S/compressed.ph...	NaN	9.78E+12
72529	Wendy Ramshaw,David Watkins	Hardcover	Paper jewelry - - bold, simple, innovative -- i...	None	https://i.gr-assets.com/images/S/compressed.ph...	500510199	9.7805E+12

```

1 top_books = data[data['totalratings'] > 2990]
2 top_books = top_books.sort_values(by='totalratings', ascending=False).head(20)
    amazing f...

```

```
1 top_books.rating.mean()
```

4.108

Capitole: Lev

```

1 top_reviewed_rated = top_books[top_books['rating'] > 4.0]
2 top_reviewed_rated = top_books.sort_values(by='rating', ascending=False).head(20)

```

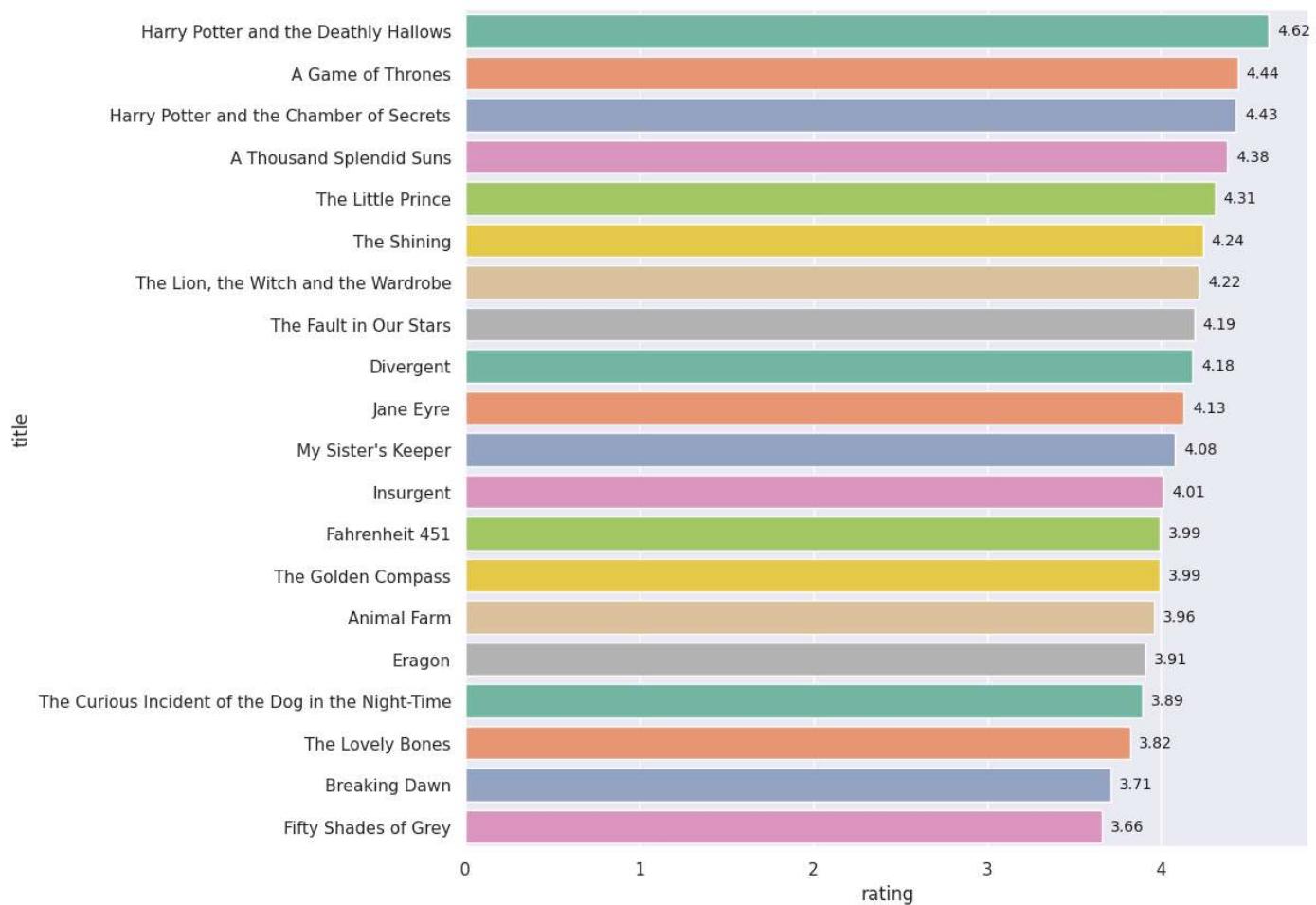
63294	Principe I. Invd De	Paperback	the most	Esoterica,Alchemv	https://i.gr-	941901327	9.78094E+12
1	top_reviewed_rated.head()						

	author	bookformat	desc	genre	img	isbn
57742	J.K. Rowling	Hardcover	Harry Potter is leaving Privet Drive for the l...	Fantasy,Young Adult,Fiction,Fantasy,Magic,Chil...	https://i.gr-assets.com/images/S/compressed.ph...	NaN
52796	George R.R. Martin	Mass Market Paperback	Here is the first volume in George R. R. Marti...	Fantasy,Fiction,Fantasy,Epic Fantasy,Adult,Sci...	https://i.gr-assets.com/images/S/compressed.ph...	553588486
81236	J.K. Rowling,Mary GrandPrÃ©	Hardcover	Ever since Harry Potter had come home for the ...	Fantasy,Young Adult,Fiction,Fantasy,Magic,Chil...	https://i.gr-assets.com/images/S/compressed.ph...	439064864
40192	Khaled Hosseini	Hardcover	A Thousand Splendid Suns, is a breathtaking st...	Fiction,Historical,Historical Fiction,Contempo...	https://i.gr-assets.com/images/S/compressed.ph...	1594489505
78182	Antoine de Saint-ExupÃ©ry,Richard Howard,Ivan ...	Paperback	A PBS Great American Read Top 100 Pick,Few sto...	Classics,Fiction,Fantasy,Childrens,Cultural,Fr...	https://i.gr-assets.com/images/S/compressed.ph...	NaN

```

1 sns.set(style="darkgrid")
2 plt.figure(figsize=(10, 10))
3
4 color = sns.color_palette("Set2")
5 ax = sns.barplot(x="rating", y="title", data=top_reviewed_rated, palette=color)
6
7 for i in ax.patches:
8     ax.text(i.get_width() + .05, i.get_y() + 0.5, str(i.get_width()), fontsize = 10, color = 'k')
9 plt.show()

```



Double-click (or enter) to edit

1  
1

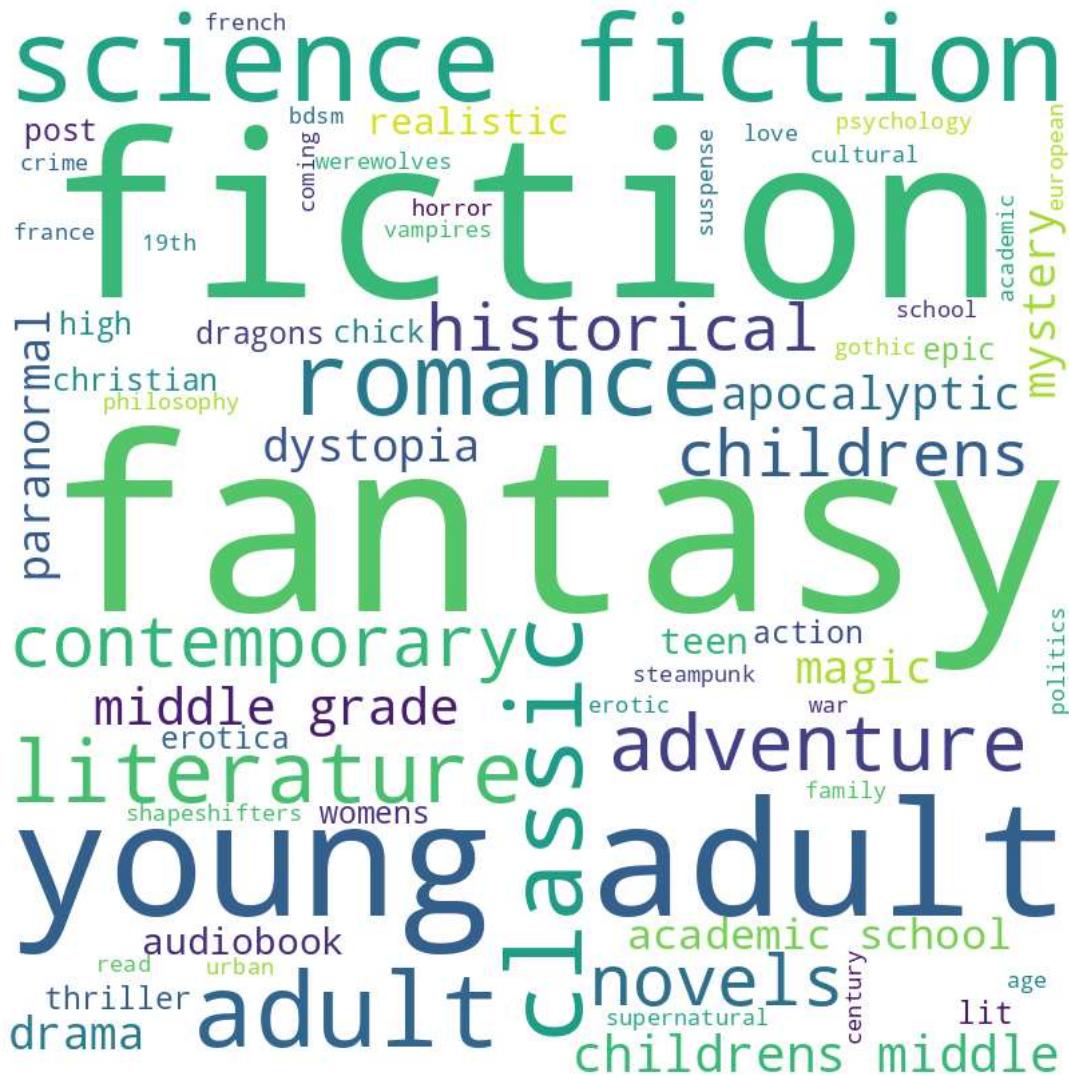
So Top Rated and Reviewed Books fall into which category??

```

1 # Python program to generate WordCloud
2
3 # importing all necessary modules
4 # from wordcloud import WordCloud, STOPWORDS
5 # import matplotlib.pyplot as plt
6 # import pandas as pd
7
8 comment_words = ""
9 stopwords = set(STOPWORDS)
10
11 # iterate through the csv file
12 for genre in top_reviewed_rated.genre:
13
14     # typecaste each val to string
15     genre = str(genre)
16
17     # split the value
18     tokens = genre.split()
19
20     # Converts each token into lowercase
21     for i in range(len(tokens)):
22         tokens[i] = tokens[i].lower()
23

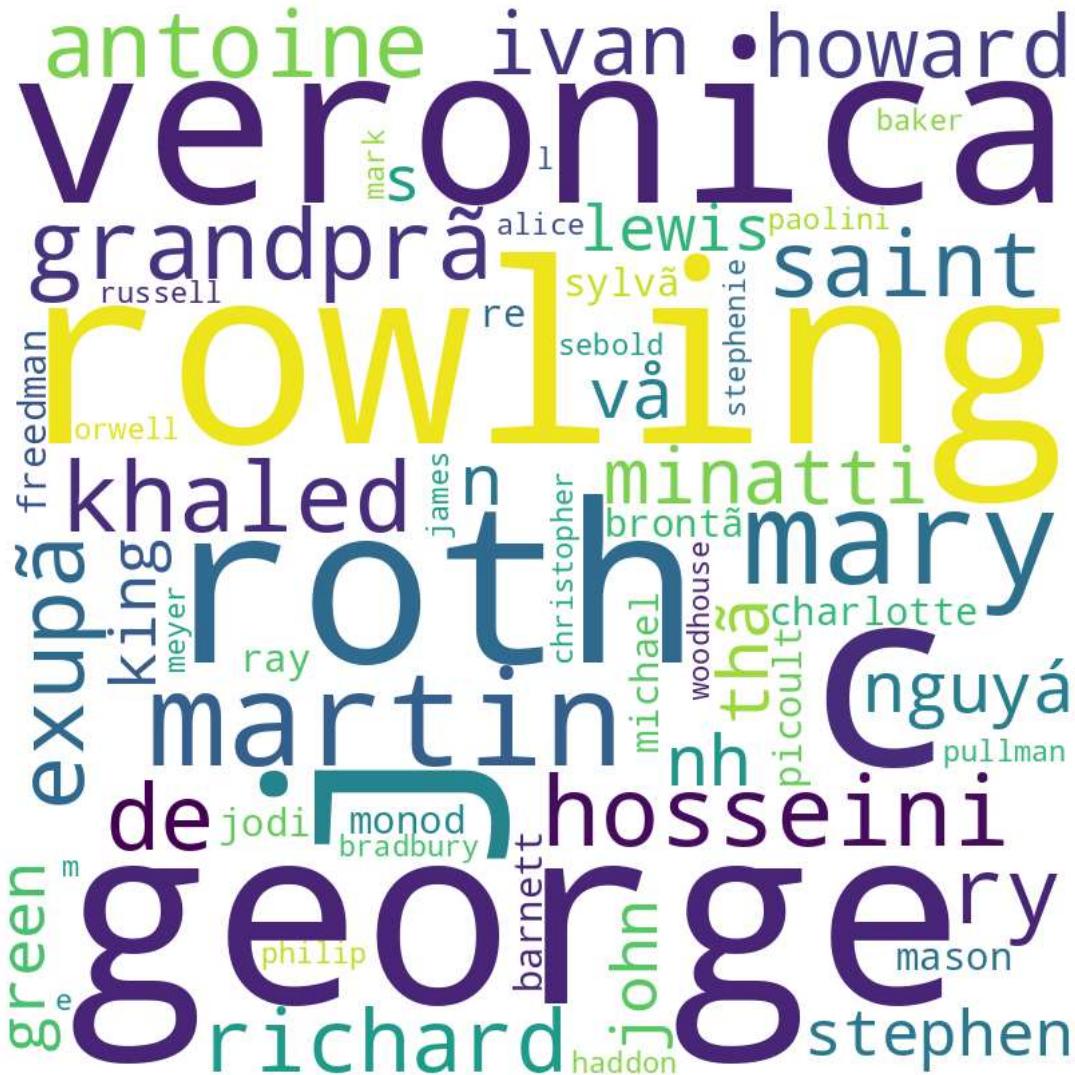
```

```
24     comment_words += " ".join(tokens)+" "
25
26 wordcloud = WordCloud(width = 800, height = 800,
27                       background_color ='white',
28                       stopwords = stopwords,
29                       min_font_size = 10).generate(comment_words)
30
31 # plot the WordCloud image
32 plt.figure(figsize = (8, 8), facecolor = None)
33 plt.imshow(wordcloud)
34 plt.axis("off")
35 plt.tight_layout(pad = 0)
36
37 plt.show()
38
```



```
1 # Python program to generate WordCloud
2
3 # importing all necessary modules
4 # from wordcloud import WordCloud, STOPWORDS
5 # import matplotlib.pyplot as plt
6 # import pandas as pd
7
8 comment_words = ""
9 stopwords = set(STOPWORDS)
10
11 # iterate through the csv file
12 for author in top_reviewed_rated.author:
13
14     # typecaste each val to string
15     auyhor = str(author)
16
```

```
17 # split the value
18 tokens = author.split()
19
20 # Converts each token into lowercase
21 for i in range(len(tokens)):
22     tokens[i] = tokens[i].lower()
23
24 comment_words += " ".join(tokens)+" "
25
26 wordcloud = WordCloud(width = 800, height = 800,
27                       background_color ='white',
28                       stopwords = stopwords,
29                       min_font_size = 10).generate(comment_words)
30
31 # plot the WordCloud image
32 plt.figure(figsize = (8, 8), facecolor = None)
33 plt.imshow(wordcloud)
34 plt.axis("off")
35 plt.tight_layout(pad = 0)
36
37 plt.show()
38
```



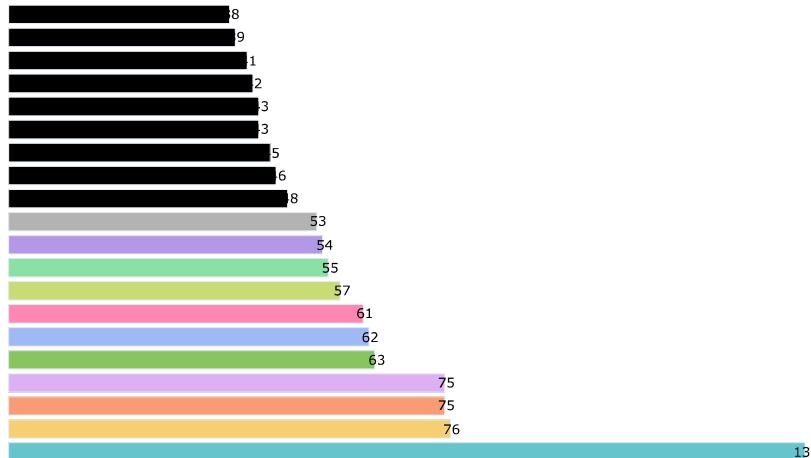
```
1 most_books = data.groupby('author')['title'].count().reset_index().sort_values('title', ascending=False).head(20)
2 most_books.head()
```

author	title	
44920	Mi-Ri Hwang	137

```

1 # Group the dataframe by authors and count the number of titles for each author
2 most_books = data.groupby('author')['title'].count().reset_index().sort_values('title', ascending=False).head(20)
3
4 # Create a Plotly bar plot
5 fig = go.Figure(data=[go.Bar(
6     x=most_books['title'],
7     y=most_books['author'],
8     orientation='h',
9     marker=dict(color=colors.qualitative.Pastel) # Set the bar colors using a qualitative color palette
10 )])
11
12 # Update the layout with dark background and white font
13 fig.update_layout(
14     plot_bgcolor='rgb(17, 17, 17)', # Dark background color
15     paper_bgcolor='rgb(17, 17, 17)', # Dark background color
16     font=dict(color='white'), # White font color
17     title="Top 20 Authors with Most Books",
18     xaxis_title="Total Number of Books",
19     yaxis_title="Authors"
20 )
21
22 # Add labels to the bars
23 for i, val in enumerate(most_books['title']):
24     fig.add_annotation(
25         x=val + 0.3,
26         y=most_books['author'].iloc[i],
27         text=str(val),
28         font=dict(color='black', size=10),
29         showarrow=False
30     )
31
32 # Show the plot
33 fig.show()

```



```

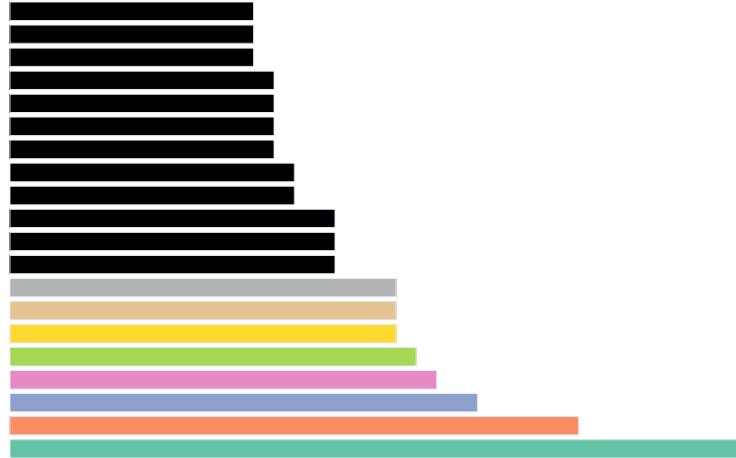
1 ## Top 20 Authors with Highest Rated Books
2
3 highRatedAuthor = data[data['rating'] >= 4.3]
4 highRatedAuthor = highRatedAuthor.groupby('author')['title'].count().reset_index().sort_values('title', ascending=False).head(20)
5
6 # Define the number of bars and obtain a color palette with enough unique colors
7 num_bars = len(highRatedAuthor)
8 color_palette = colors.qualitative.Set2
9
10 # Create a list of distinct colors for each bar

```

```

11 bar_colors = color_palette[:num_bars]
12
13 # Create a Plotly bar plot
14 fig = go.Figure(data=[go.Bar(
15     x=high_rated_author['title'],
16     y=high_rated_author['author'],
17     orientation='h',
18     marker=dict(color=bar_colors) # Set the bar colors using the list of colors
19 )])
20
21 # Update the layout with dark background and white font
22 fig.update_layout(
23     plot_bgcolor='rgb(17, 17, 17)', # Dark background color
24     paper_bgcolor='rgb(17, 17, 17)', # Dark background color
25     font=dict(color='white'), # White font color
26     title="Top 20 Authors with Highest Rated Books",
27     xaxis_title="Number of Books",
28     yaxis_title="Authors"
29 )
30
31 # Show the plot
32 fig.show()

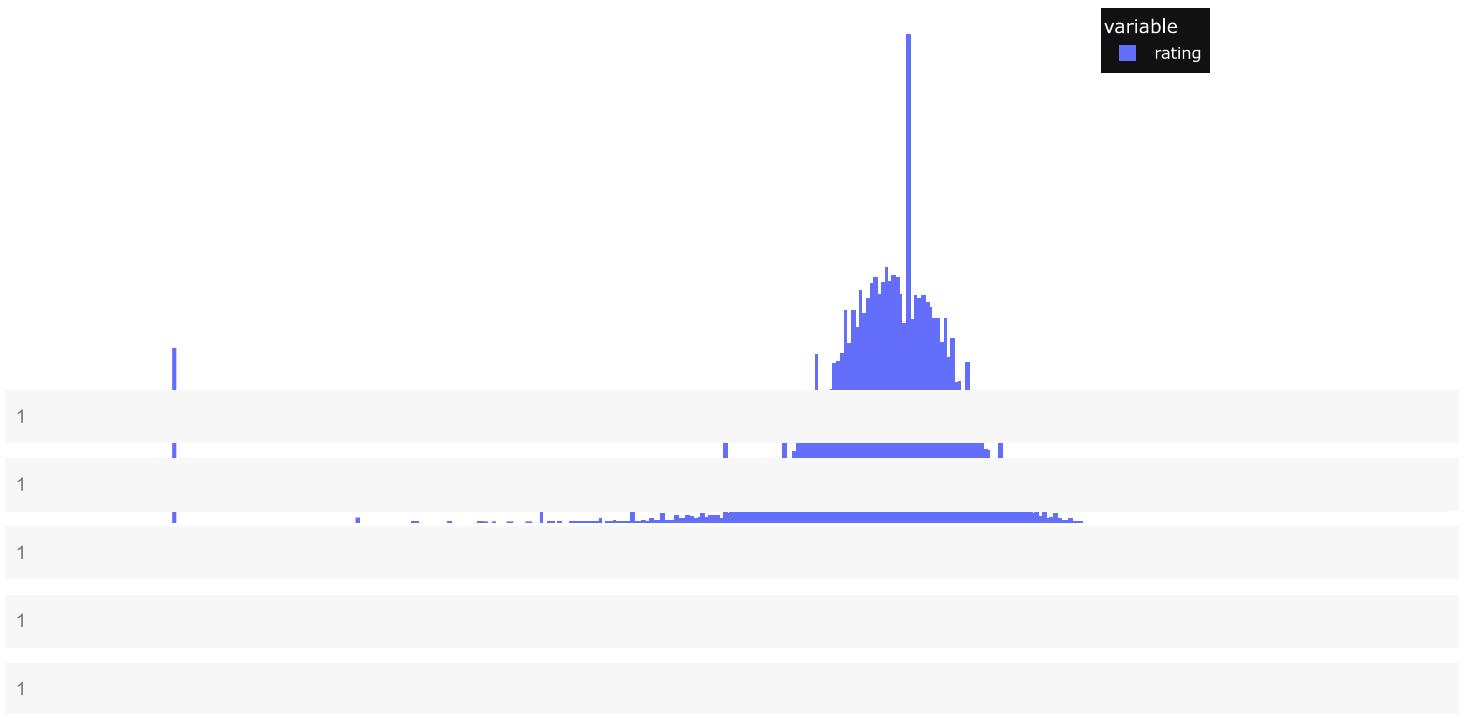
```



```

1 rating = data['rating'].astype(float)
2
3 fig = px.histogram(rating, histnorm='density')
4
5 fig.update_layout(
6     plot_bgcolor='rgb(17, 17, 17)', # Dark background color
7     paper_bgcolor='rgb(17, 17, 17)', # Dark background color
8     font=dict(color='white'), # White font color
9     title="Distribution of Average Ratings",
10    xaxis_title="Average Rating",
11    yaxis_title="Density"
12 )
13
14 fig.show()

```



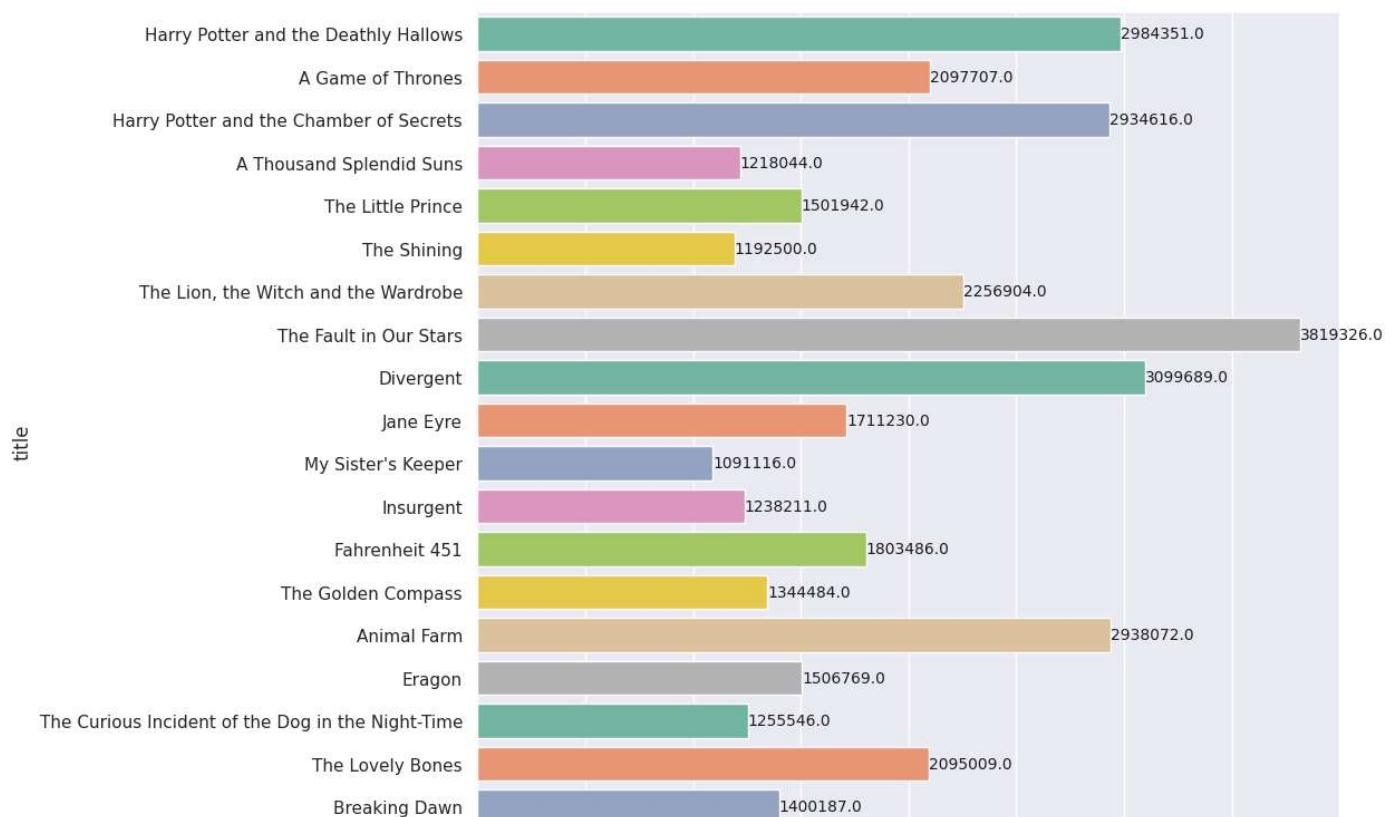
### Top 20 Top Voted Books

```
1 top_vote = top_reviewed_rated.sort_values(by='rating', ascending=False).head(20)
```

```
1 top_vote.head(1)
```

	author	bookformat	desc	genre	img	isbn	isbn13
57742	J.K. Rowling	Hardcover	Harry Potter is leaving Privet Drive for the I...	Fantasy,Young Adult,Fiction,Fantasy,Magic,Chil...	https://i.gr... assets.com/images/S/compressed.ph...	NaN	NaN https://goodreads.com/book/s...

```
1 sns.set(style="darkgrid")
2 plt.figure(figsize=(10, 10))
3
4 color = sns.color_palette("Set2")
5 ax = sns.barplot(x="totalratings", y="title", data=top_vote, palette=color)
6
7 for i in ax.patches:
8     ax.text(i.get_width() + .05, i.get_y() + 0.5, str(i.get_width()), fontsize = 10, color = 'k')
9 plt.show()
10
11
```

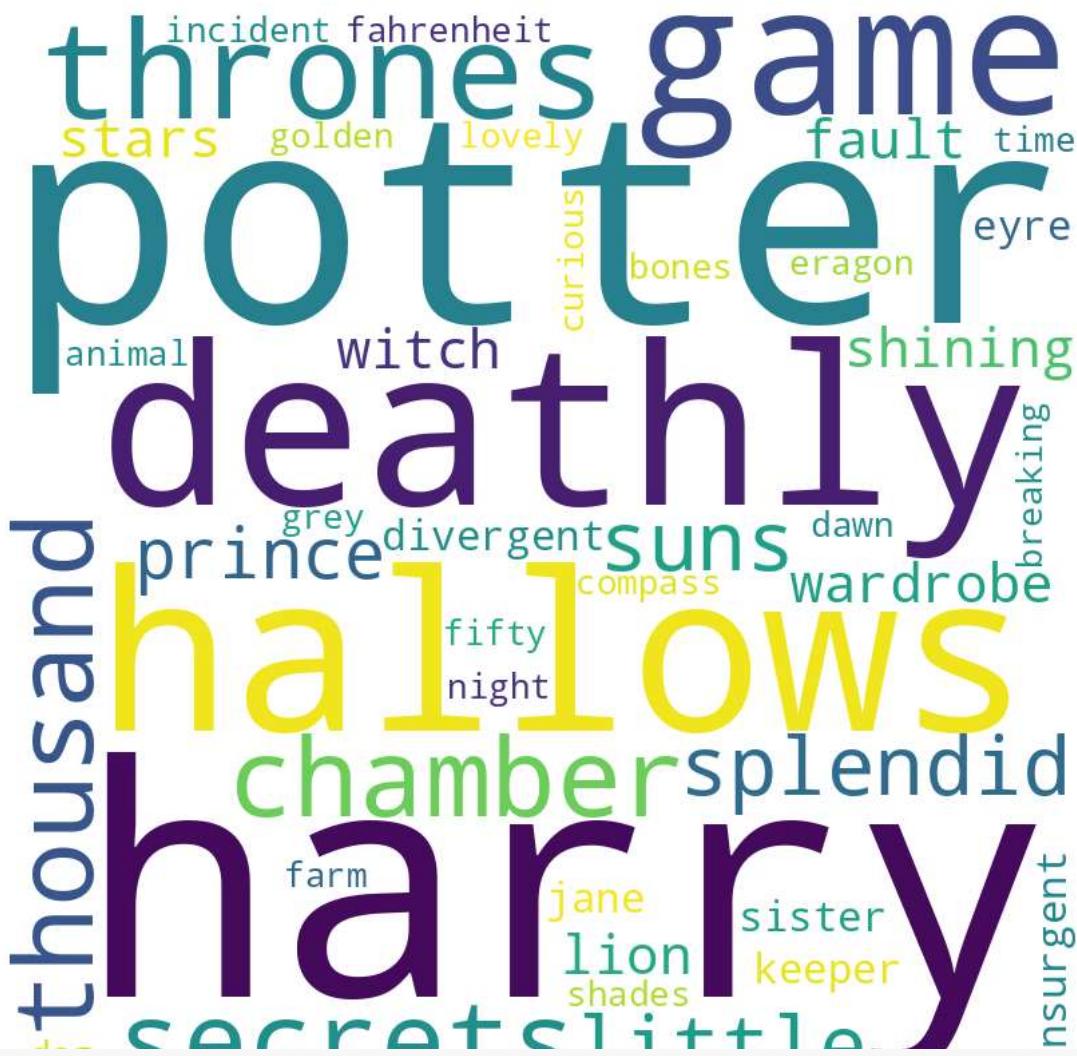


1

```

1 # Python program to generate WordCloud
2
3 # importing all necessary modules
4 # from wordcloud import WordCloud, STOPWORDS
5 # import matplotlib.pyplot as plt
6 # import pandas as pd
7
8 comment_words =
9 stopwords = set(STOPWORDS)
10
11 # iterate through the csv file
12 for title in top_vote.title:
13
14     # typecaste each val to string
15     title = str(title)
16
17     # split the value
18     tokens = title.split()
19
20     # Converts each token into lowercase
21     for i in range(len(tokens)):
22         tokens[i] = tokens[i].lower()
23
24     comment_words += " ".join(tokens)+" "
25
26 wordcloud = WordCloud(width = 800, height = 800,
27                       background_color ='white',
28                       stopwords = stopwords,
29                       min_font_size = 10).generate(comment_words)
30
31 # plot the WordCloud image
32 plt.figure(figsize = (8, 8), facecolor = None)
33 plt.imshow(wordcloud)
34 plt.axis("off")
35 plt.tight_layout(pad = 0)
36
37 plt.show()
38

```

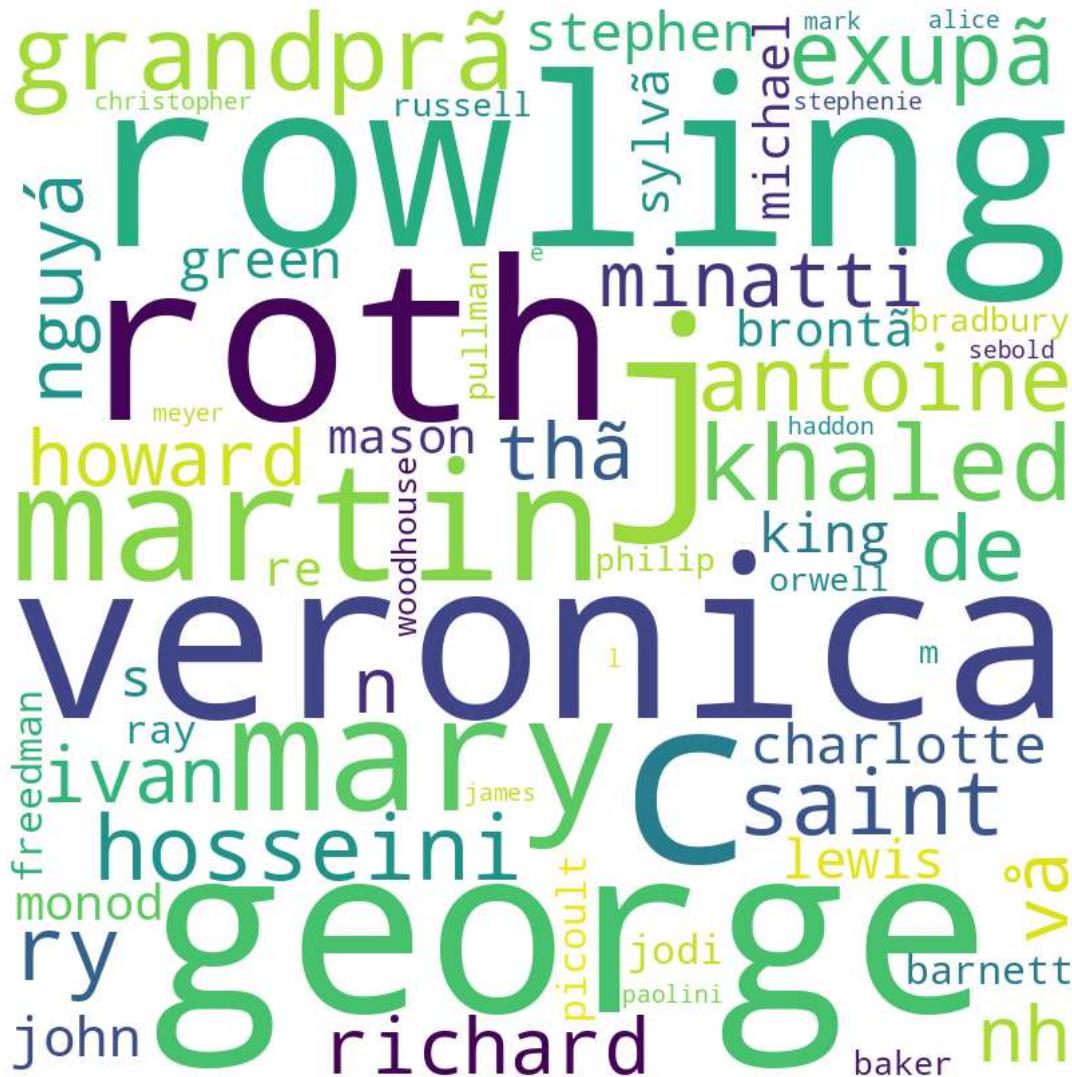


```

1 # Python program to generate WordCloud
2
3 # importing all necessary modules
4 # from wordcloud import WordCloud, STOPWORDS
5 # import matplotlib.pyplot as plt
6 # import pandas as pd
7
8 comment_words = ""
9 stopwords = set(STOPWORDS)
10
11 # iterate through the csv file
12 for author in top_vote.author:
13
14     # typecaste each val to string
15     author = str(author)
16
17     # split the value
18     tokens = author.split()
19
20     # Converts each token into lowercase
21     for i in range(len(tokens)):
22         tokens[i] = tokens[i].lower()
23
24     comment_words += " ".join(tokens)+" "
25
26 wordcloud = WordCloud(width = 800, height = 800,
27                       background_color ='white',
28                       stopwords = stopwords,
29                       min_font_size = 10).generate(comment_words)
30
31 # plot the WordCloud image
32 plt.figure(figsize = (8, 8), facecolor = None)
33 plt.imshow(wordcloud)
34 plt.axis("off")
35 plt.tight_layout(pad = 0)

```

```
36
37 plt.show()
38
```

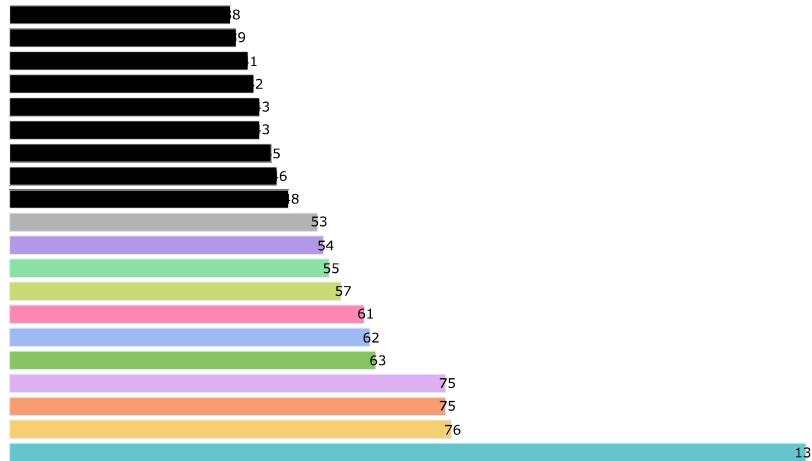


```
1 # Group the dataframe by authors and count the number of titles for each author
2 most_books = data.groupby('author')['title'].count().reset_index().sort_values('title', ascending=False).head(20)
3
4 # Create a Plotly bar plot
5 fig = go.Figure(data=[go.Bar(
6     x=most_books['title'],
7     y=most_books['author'],
8     orientation='h',
9     marker=dict(color=colors.qualitative.Pastel) # Set the bar colors using a qualitative color palette
10 )])
11
12 # Update the layout with dark background and white font
13 fig.update_layout(
14     plot_bgcolor='rgb(17, 17, 17)', # Dark background color
15     paper_bgcolor='rgb(17, 17, 17)', # Dark background color
16     font=dict(color='white'), # White font color
17     title="Top 20 Authors with Most Books",
18     xaxis_title="Total Number of Books",
19     yaxis_title="Authors"
20 )
21
22 # Add labels to the bars
23 for i, val in enumerate(most_books['title']):
24     fig.add_annotation(
25         x=val + 0.3,
26         y=most_books['author'].iloc[i],
27         text=str(val),
28         font=dict(color='black', size=10),
29         showarrow=False
30 )
```

```

30 )
31
32 # Show the plot
33 fig.show()

```



1

1

1

```
1 !pip install plotly==5.16.0
```

```

Collecting plotly==5.16.0
  Downloading plotly-5.16.0-py2.py3-none-any.whl (15.6 MB)
    ━━━━━━━━━━━━━━━━ 15.6/15.6 MB 35.7 MB/s eta 0:00:00
Requirement already satisfied: tenacity>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from plotly==5.16.0) (8.2.3)
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (from plotly==5.16.0) (23.1)
Installing collected packages: plotly
  Attempting uninstall: plotly
    Found existing installation: plotly 5.15.0
    Uninstalling plotly-5.15.0:
      Successfully uninstalled plotly-5.15.0
Successfully installed plotly-5.16.0
WARNING: The following packages were previously imported in this runtime:
[_plotly_utils,plotly]
You must restart the runtime in order to use newly installed versions.

```

[RESTART RUNTIME](#)

```

1 # plotly standard imports
2 import plotly.graph_objs as go
3 #
4
5 # Cufflinks wrapper on plotly
6 import cufflinks
7
8 # Data science imports
9 import pandas as pd
10 import numpy as np
11
12 # Options for pandas
13 pd.options.display.max_columns = 30
14
15 # Display all cell outputs
16 from IPython.core.interactiveshell import InteractiveShell

```

```
17
18 InteractiveShell.ast_node_interactivity = "all"
```

```
1 from plotly.offline import iplot
2
3 cufflinks.go_offline()
4
5 # Set global theme
6 cufflinks.set_config_file(world_readable=True, theme="pearl")
```

```
1 df = top_reviewed.sort_values("rating", ascending = False)
2 #fig = go.Figure(data=go.Scatter(x=df['rating'], y=data['title'], mode='markers', marker=dict(color='red')))
3
4 df.iplot(
5     x="rating",
6     y="title",
7     xTitle="Rating",
8     yTitle="Title",
9     text="title",
10    mode="markers",
11    title="Book Title vs Rating",
12 )
```

1

```
1 df = top_reviewed.sort_values("rating", ascending = False)
```

```
1 #df = top_reviewed.sort_values("Country", ascending = False)
2
3 ax = sns.barplot(x="rating", y="title", data=top_reviewed, palette=color)
4
5 fig = go.Figure(data=go.Scatter(x=df['rating'], y=data['title'], mode='markers', marker=dict(color='red')))
6 #fig = px.Scatter(
7     #df, y='title', x='rating', color='title', size='rating', size_max=20,
8     #color_continuous_scale = px.colors.sequential.RdBu)
9 fig.update_layout(
10     paper_bgcolor="white",
11     plot_bgcolor="white",
12 )
13 # #fig = px.colors.sequential.swatches_continuous()
14 fig.update_yaxes(showgrid=True, gridwidth=1, gridcolor='LightGray')
15 fig.update_xaxes(showgrid=True, gridwidth=1, gridcolor='LightGray')
16 fig.update_layout(height=500, width=1000)
```

```
17 fig.update_coloraxes(colorbar=dict(title='Title'))
18 fig.update_traces(marker=dict(sizeref=0.09))
19 fig.update_yaxes(title="Title")
20 fig.update_xaxes(title='Rating')
21 fig.update_layout(showlegend=True)
22 fig.show()
```





```
1 N = 100000
2 fig = go.Figure(data=go.Scatter(x=data['rating'], y=data['title'], mode='markers', marker=dict(color='red')))
3
4 # fig = go.Figure(data=go.Scatter(
5 #   x=[1, 2, 3, 4],
6 #   y=[10, 11, 12, 13],
7 #   mode='markers',
8 #   marker=dict(size=[40, 60, 80, 100],
9 #             color=[0, 1, 2, 3])
10 # ))
```

11

12.5 minutes

```
1 list(set(top_rated_books['title'].values) - set(top_reviewed_rated['title'].values))
```

[ 'The Puzzling World of Polyhedral Dissections: Hundreds of 3-D Puzzles to Build and Solve',  
 "The Bead Lover's Bible: Techniques For Beadlovers",  
 "The Doctor's Book of Home Remedies for Seniors: An A-to-Z Guide to Staying Physically Active, Mentally Sharp, and Disease-Free",  
 'All the Backyard Birds: East and West',  
 'Luttes Xxx: Inspirations Du Mouvement Des Travailleuses Du Sexe',  
 "L'Album d'Eddy",  
 'The Focke-Wulf FW 190 Dora: Volume One',  
 'Dramaturgi Ru I: Lev Tolstoi, Ivan Andreevici Krilov, Nikolai Gogol, Anton Cehov, Daniil Harms, Ivan Turghenev, Mihail Bulgakov',  
 'Contested Creations in the Book of Job: The-World-As-It-Ought- And-Ought-Not-To-Be',  
 'Advances in Architectural Geometry 2012',  
 'Redemption',  
 'Casey Little, Yo Yo Queen',  
 'Kilimanjaro: the White Roof of Africa',  
 'The Paper Jewelry Collection: Pop Out Artwear',  
 'The Hobbit Tarot',  
 'Federal Taxation of Partnerships & Partners',  
 'Veggie Burger',  
 'The Hot Air Balloon Book: Build and Launch Kongming Lanterns, Solar Tetroons, and More',  
 'Requiem',  
 'Transmutations: Alchemy in Art: Selected Works from the Eddleman and Fisher Collections at the Chemical Heritage Foundation']

The Curious Incident of the Dog in the Night-Time

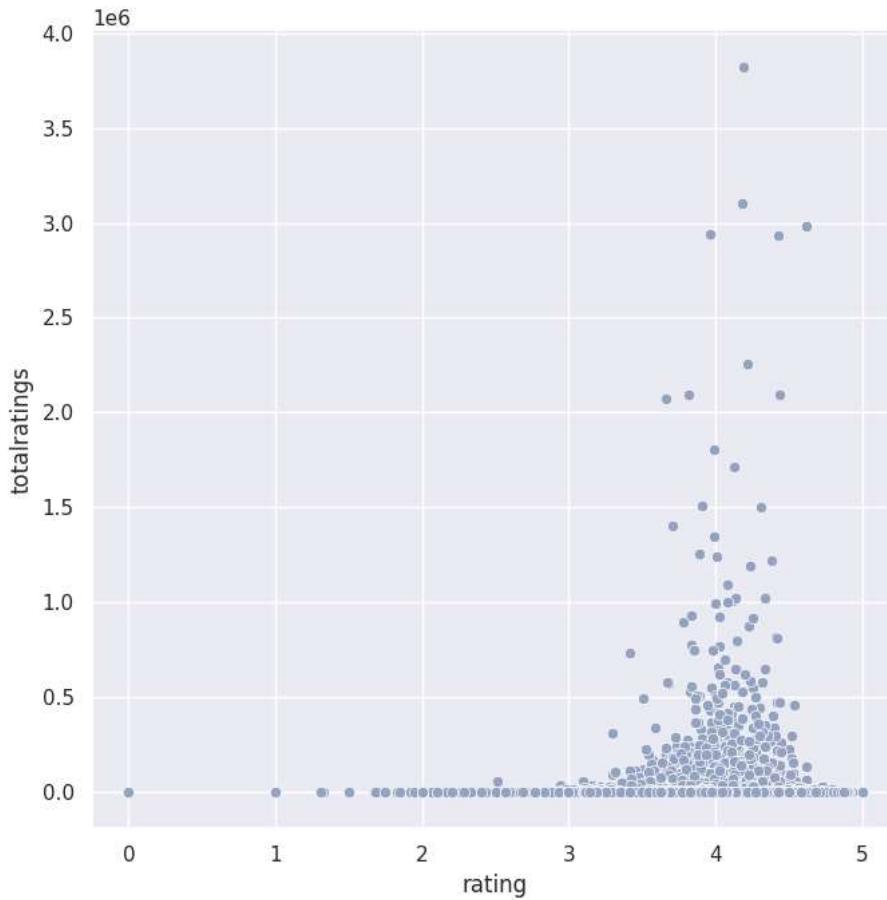
```
1 list(set(top_vote['title'].values) - set(top_rated_books['title'].values))
```

[ "My Sister's Keeper",  
 'A Game of Thrones',  
 'Insurgent',  
 'Breaking Dawn',  
 'The Curious Incident of the Dog in the Night-Time',  
 'Fifty Shades of Grey',  
 'The Fault in Our Stars',  
 'The Little Prince',  
 'Eragon',  
 'Fahrenheit 451',  
 'The Lion, the Witch and the Wardrobe',  
 'Jane Eyre',  
 'A Thousand Splendid Suns',  
 'Harry Potter and the Deathly Hallows',  
 'Divergent',  
 'Animal Farm',  
 'Harry Potter and the Chamber of Secrets',  
 'The Golden Compass',  
 'The Lovely Bones',  
 'The Shining']

```
1 data.columns
```

```
Index(['author', 'bookformat', 'desc', 'genre', 'img', 'isbn', 'isbn13',
       'link', 'pages', 'rating', 'reviews', 'title', 'totalratings'],
      dtype='object')
```

```
1 ax = sns.relplot(data=data, x="rating",y="totalratings" ,color = '#95a3c3', sizes=(100, 200), height=7, marker='o')
```



```
1 new_data = data.copy()
```

```
1 def fun_only_author(text):
2     arlen = text.split('/')
3     return arlen[0]
```

```
1 new_data['only_author'] = new_data['author'].apply(lambda x : fun_only_author(x))
```

```
1 total_rating = new_data.drop_duplicates(subset=['only_author', 'title'], keep='first')
2 total_rating = total_rating.groupby(by=['only_author']).agg({'rating': ['sum']})
3 total_rating.columns = ['totalratings']
4 total_rating.reset_index(inplace=True)
5 total_rating = total_rating.sort_values(by=['totalratings'], ascending=False)
6 total_rating
```

	only_author	totalratings	grid
44920	Mi-Ri Hwang	499.36	grid
52547	R.L. Stine	273.76	grid
67115	Willy Vandersteen	268.30	grid

```

1 total_book = new_data.groupby(by=['only_author']).agg({'title': ['nunique']})
2 total_book.columns = ['total_book']
3 total_book.reset_index(inplace=True)
4 total_book = total_book.sort_values(by=['total_book'], ascending=False)
5 total_book

```

	only_author	total_book	grid
44920	Mi-Ri Hwang	126	grid
67115	Willy Vandersteen	76	grid
52547	R.L. Stine	75	grid
17660	Elinor M. Brent-Dyer	62	grid
40680	Lynn Hagen	62	grid
...	...	...	grid
25072	Horsh Dornbusch	1	grid
25073	Horst A. Friedrichs	1	grid
25074	Horst Bredekamp, John Michael Krois	1	grid
68766	é€Ýç€¬ ç³¼½æÝ'	1	grid
26996	Jacqui Malpass	0	grid

68767 rows × 2 columns

```

1 avg_author = pd.merge(total_book, total_rating, on='only_author', how='outer')
2 avg_author['rating'] = round(avg_author['totalratings'] / avg_author['total_book'], 2)
3 avg_author = avg_author[avg_author['total_book'] > 26]
4 avg_author = avg_author.sort_values(by=['rating'], ascending=False)
5 avg_author
6

```

	only_author	total_book	totalratings	rating	grid icon
33	Takehiko Inoue	28	123.68	4.42	grid icon
37	Brenda Jackson	27	117.65	4.36	grid icon
35	Gosho Aoyama	27	115.23	4.27	grid icon
19	Kristen Ashley	38	162.31	4.27	grid icon
26	Syun Matsuena	31	130.92	4.22	grid icon
36	Radclyffe	27	113.14	4.19	grid icon
15	Jim Davis	43	180.36	4.19	grid icon
5	Anonymous	61	249.39	4.09	grid icon
34	Julie Garwood	28	114.40	4.09	grid icon
13	Yu-Rang Han	44	176.87	4.02	grid icon
21	Nora Roberts	37	148.27	4.01	grid icon
31	Meredes Lackey	28	111.50	3.98	grid icon
6	Louis L'Amour	57	226.70	3.98	grid icon
0	Mi-Ri Hwang	126	499.36	3.96	grid icon
4	Lynn Hagen	62	243.62	3.93	grid icon
23	Stephen King	33	129.59	3.93	grid icon
3	Elinor M. Brent-Dyer	62	243.52	3.93	grid icon
30	Susan Mallery	29	113.65	3.92	grid icon
39	John Montroll	27	105.37	3.90	grid icon
8	Agatha Christie	54	210.20	3.89	grid icon
27	Jean Plaidv	31	120.64	3.89	grid icon

1 new\_data.columns

```
Index(['author', 'bookformat', 'desc', 'genre', 'img', 'isbn', 'isbn13',
       'link', 'pages', 'rating', 'reviews', 'title', 'totalratings',
       'only_author'],
      dtype='object')
```

```
1 total_vote = new_data.drop_duplicates(subset=['only_author', 'title'], keep='first')
2 total_vote.reset_index(inplace=True)
3 total_vote = total_vote[['only_author', 'title', 'rating', 'totalratings']]
4 total_vote
```

	only_author		title	rating	totalratings	grid icon
0	Laurence M. Hauptman		Between Two Fires: American Indians in the Civ...	3.52	33	grid icon
1	Charlotte Fiell,Emmanuelle Dirix		Fashion Sourcebook 1920s	4.51	41	grid icon
2	Andy Anderson		Hungary 56	4.15	26	grid icon
3	Carlotta R. Anderson	All-American Anarchist: Joseph A. Labadie and ...		3.83	6	grid icon
4	Jean Leveille		Les oiseaux gourmands	4.00	1	grid icon
...	...		...	...	...	grid icon
99903	Simon Monk	Make Your Own PCBs with Eagle: From Schematic ...		4.07	56	grid icon
99904	Tracie L. Miller-Nobles,Brenda L. Mattison,Ell...	Horngren's Financial & Managerial Accounting		4.05	38	grid icon
99905	C. John Miller	A Faith Worth Sharing: A Lifetime of Conversat...		4.27	122	grid icon
99906	Albert Marrin	A Volcano Beneath the Snow: John Brown's War A...		3.63	156	grid icon
99907	Marc E. Fitch	Paranormal Nation: Why America Needs Ghosts, U...		3.83	12	grid icon

99908 rows × 4 columns

Average Rating

```
1 C = total_vote.rating.mean()
2 C
```

3.8329021699963963

```
1 m = total_vote.totalratings.quantile(0.9)
2 m
```

3185.0

```
1 k = total_vote.totalratings.quantile(0.8)
2 k
```

1123.6000000000058

```
1 total_vote = total_vote[total_vote['totalratings'] >= m]
2 total_vote.head()
```

	only_author		title	rating	totalratings	
21	Christopher Morley		Parnassus on Wheels	4.03	6107	
24	Robert J. Shiller		Irrational Exuberance	3.98	7019	
37	Ovid,David Malouf,James Michie		The Art of Love	3.81	5154	
41	Joseph Staten		Halo: Contact Harvest	3.98	8033	
67	Eliezer Yudkowsky	Harry Potter and the Methods of Rationality		4.40	14293	

```
1 def weighted_rating(x, m=m, C=C):
2     v = x['totalratings']
3     R = x['rating']
4     return (v/(v+m) * R) + (m/(m+v) * C)
```

```
1 total_vote['score'] = total_vote.apply(weighted_rating, axis=1)
```

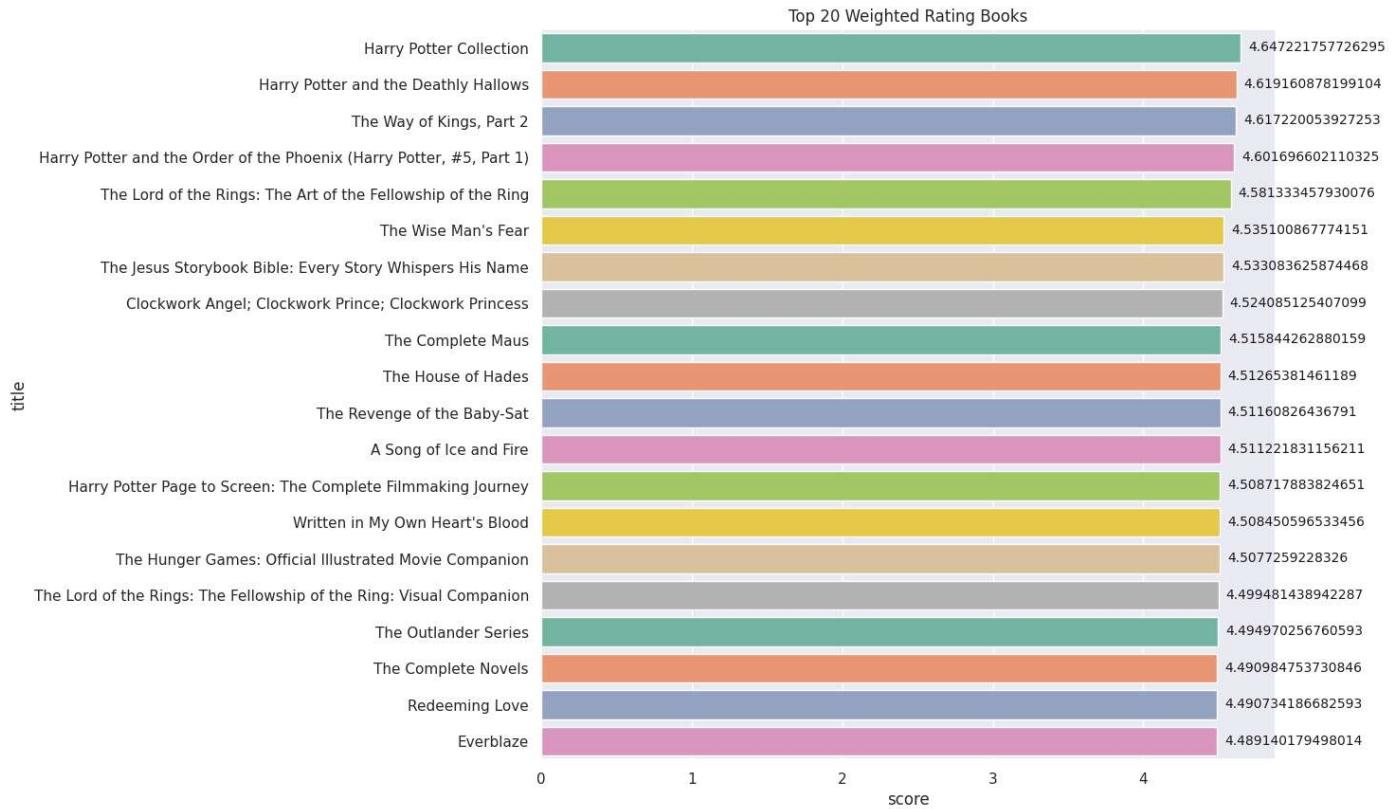
```
1 total_vote = total_vote.sort_values(by='score', ascending=False).head(20)
2 total_vote.head()
```

	only_author		title	rating	totalratings	score	
14779	J.K. Rowling		Harry Potter Collection	4.73	31332	4.647222	
57687	J.K. Rowling		Harry Potter and the Deathly Hallows	4.62	2984351	4.619161	
18094	Brandon Sanderson		The Way of Kings, Part 2	4.79	14458	4.617220	
46518	J.K. Rowling	Harry Potter and the Order of the Phoenix (Har...		4.62	133779	4.601697	
28841	Gary Russell	The Lord of the Rings: The Art of the Fellowsh...		4.62	61649	4.581333	

## Top 20 Weighted Rating Books

```
1 sns.set(style="darkgrid")
2 plt.figure(figsize=(10, 10))
3
4 color = sns.color_palette("Set2")
5 ax = sns.barplot(x="score", y="title", data=total_vote, palette=color)
6
7 for i in ax.patches:
8     ax.text(i.get_width() + .05, i.get_y() + 0.5, str(i.get_width()), fontsize = 10, color = 'k')
9 plt.title("Top 20 Weighted Rating Books")
10 plt.show()
```

```
<Figure size 1000x1000 with 0 Axes>Text(4.697221757726295, 0.0999999999999999, '4.647221757726295')Text(4.669160878199103, 1.1, '4.619160878199104')Text(4.667220053927253, 2.1, '4.617220053927253')Text(4.6516966021103245, 3.1, '4.601696602110325')Text(4.631333457930076, 4.1, '4.581333457930076')Text(4.585100867774151, 5.1, '4.535100867774151')Text(4.583083625874468, 6.1, '4.533083625874468')Text(4.574085125407099, 7.1, '4.524085125407099')Text(4.565844262880159, 8.1, '4.515844262880159')Text(4.56265381461189, 9.1, '4.51265381461189')Text(4.5616082643679094, 10.1, '4.51160826436791')Text(4.561221831156211, 11.1, '4.511221831156211')Text(4.5587178838246505, 12.1, '4.508717883824651')Text(4.558450596533456, 13.1, '4.508450596533456')Text(4.5577259228326, 14.1, '4.5077259228326')Text(4.549481438942287, 15.1, '4.499481438942287')Text(4.5449702567605925, 16.1, '4.494970256760593')Text(4.540984753730846, 17.1, '4.490984753730846')Text(4.5407341866825925, 18.1, '4.490734186682593')Text(4.5391401794980135, 19.1, '4.489140179498014')Text(0.5, 1.0, 'Top 20 Weighted Rating Books')
```



### Most books of an author

```
1 sns.set(style="darkgrid")
2 plt.figure(figsize=(10, 10))
3
4 color = sns.color_palette("Set2")
5 ax = sns.barplot(x="total_book", y="only_author", data=avg_author, palette=color)
6
7 for i in ax.patches:
8     ax.text(i.get_width() + .05, i.get_y() + 0.5, str(i.get_width()), fontsize = 10, color = 'k')
9 plt.show()
```

```
<Figure size 1000x1000 with 0 Axes>Text(28.05, 0.0999999999999999, '28.0')Text(27.05, 1.1, '27.0')Text(27.05, 2.1, '27.0')Text(38.05, 3.1, '38.0')Text(31.05, 4.1, '31.0')Text(27.05, 5.1, '27.0')Text(43.05, 6.1, '43.0')Text(61.05, 7.1, '61.0')Text(28.05, 8.1, '28.0')Text(44.05, 9.1, '44.0')Text(37.05, 10.1, '37.0')Text(28.05, 11.1, '28.0')Text(57.05, 12.1, '57.0')Text(126.05, 13.1, '126.0')Text(62.05, 14.1, '62.0')Text(33.05, 15.1, '33.0')Text(62.05, 16.1, '62.0')Text(29.05, 17.1, '29.0')Text(27.05, 18.1, '27.0')Text(54.05, 19.1, '54.0')Text(31.05, 20.1, '31.0')Text(38.05, 21.1, '30.0')Text(45.05, 22.1, '45.0')Text(29.05, 23.1, '29.0')Text(39.05, 24.1, '39.0')Text(53.05, 25.1, '53.0')Text(28.05, 26.1, '28.0')Text(48.05, 27.1, '48.0')Text(36.05, 28.1, '36.0')Text(27.05, 29.1, '27.0')Text(31.05, 30.1, '31.0')Text(42.05, 31.1, '42.0')Text(46.05, 32.1, '46.0')Text(32.05, 33.1, '32.0')Text(43.05, 34.1, '43.0')Text(75.05, 35.1, '75.0')Text(27.05, 36.1, '27.0')Text(55.05, 37.1, '55.0')Text(38.05, 38.1, '38.0')Text(76.05, 39.1, '76.0')Text(41.05, 40.1, '41.0')
```



```
1 sns.set(style="darkgrid")
2 plt.figure(figsize=(10, 10))
3
4 color = sns.color_palette("Set2")
5 ax = sns.barplot(x="rating", y="only_author", data=avg_author, palette=color)
6
7 for i in ax.patches:
8     ax.text(i.get_width() + .05, i.get_y() + 0.5, str(i.get_width()), fontsize = 10, color = 'k')
9 plt.show()
```

```
<Figure size 1000x1000 with 0 Axes>Text(4.47, 0.0999999999999998, '4.42')Text(4.41, 1.1, '4.36')Text(4.319999999999999, 2.1, '4.27')Text(4.319999999999999, 3.1, '4.27')Text(4.27, 4.1, '4.22')Text(4.24, 5.1, '4.19')Text(4.24, 6.1, '4.19')Text(4.14, 7.1, '4.09')Text(4.14, 8.1, '4.09')Text(4.069999999999999, 9.1, '4.02')Text(4.06, 10.1, '4.01')Text(4.03, 11.1, '3.98')Text(4.03, 12.1, '3.98')Text(4.01, 13.1, '3.96')Text(3.98, 14.1, '3.93')Text(3.98, 15.1, '3.93')Text(3.98, 16.1, '3.93')Text(3.969999999999998, 17.1, '3.92')Text(3.949999999999997, 18.1, '3.9')Text(3.94, 19.1, '3.89')Text(3.94, 20.1, '3.89')Text(3.929999999999997, 21.1, '3.88')Text(3.92, 22.1, '3.87')Text(3.9, 23.1, '3.85')Text(3.889999999999997, 24.1, '3.84')Text(3.88, 25.1, '3.83')Text(3.86, 26.1, '3.81')Text(3.84, 27.1, '3.79')Text(3.82, 28.1, '3.77')Text(3.8, 29.1, '3.75')Text(3.79, 30.1, '3.74')Text(3.79, 31.1, '3.74')Text(3.79, 32.1, '3.74')Text(3.76, 33.1, '3.71')Text(3.71, 34.1, '3.66')Text(3.699999999999997, 35.1, '3.65')Text(3.679999999999997, 36.1, '3.63')Text(3.639999999999997, 37.1, '3.59')Text(3.59, 38.1, '3.54')Text(3.579999999999996, 39.1, '3.53')Text(3.53, 40.1, '3.48')
```



## 1 data.columns

```
Index(['author', 'bookformat', 'desc', 'genre', 'img', 'isbn', 'isbn13',
       'link', 'pages', 'rating', 'reviews', 'title', 'totalratings'],
      dtype='object')
```



```
1 top_pages = data.sort_values(by='pages', ascending=False).head(20)
```

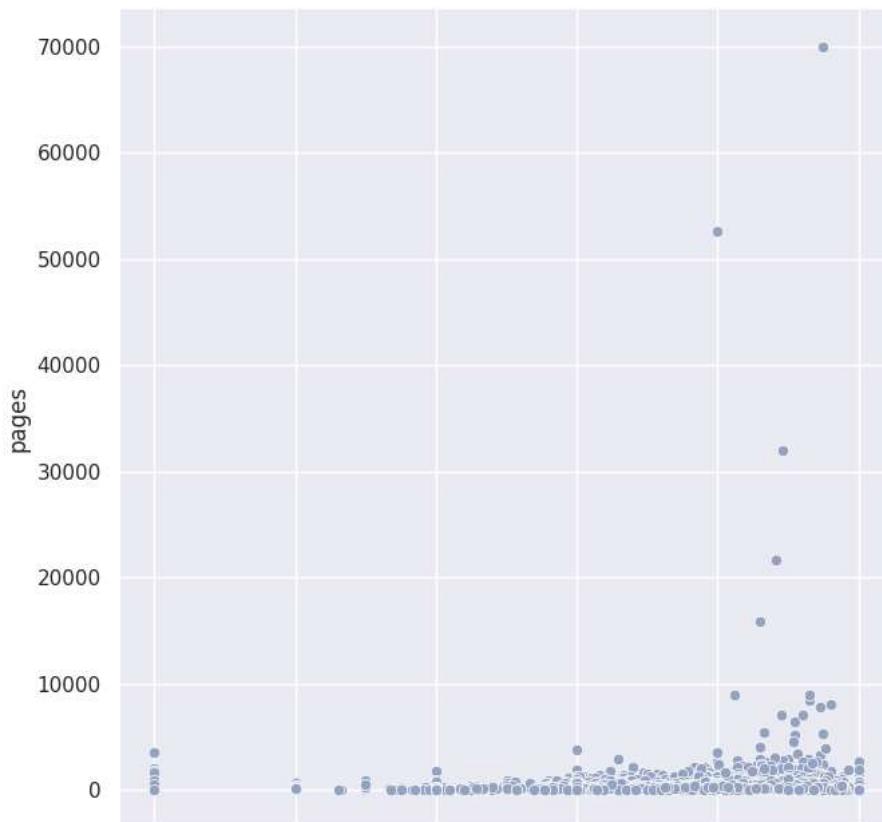
```
1 sns.set(style="darkgrid")
2 plt.figure(figsize=(10, 10))
3
4 color = sns.color_palette("Set2")
5 ax = sns.barplot(x="pages", y="title", data=top_pages, palette=col
6
7 for i in ax.patches:
8     ax.text(i.get_width() + .05, i.get_y() + 0.5, str(i.get_width()), font
9 plt.show()
```

```
<Figure size 1000x1000 with 0 Axes>Text(70000.05, 0.0999999999999998, '70000.0')Text(52646.05, 1.1, '52646.0')Text(32000.05, 2.1, '32000.0')Text(21723.05, 3.1, '21723.0')Text(15918.05, 4.1, '15918.0')Text(9000.05, 5.1, '9000.0')Text(8936.05, 6.1, '8936.0')Text(8448.05, 7.1, '8448.0')Text(8160.05, 8.1, '8160.0')Text(7852.05, 9.1, '7852.0')Text(7144.05, 10.1, '7144.0')Text(7092.05, 11.1, '7092.0')Text(6448.05, 12.1, '6448.0')Text(5448.05, 13.1, '5448.0')Text(5375.05, 14.1, '5375.0')Text(5216.05, 15.1, '5216.0')Text(4528.05, 16.1, '4528.0')Text(4084.05, 17.1, '4084.0')Text(3984.05, 18.1, '3984.0')Text(3808.05, 19.1, '3808.0')
```



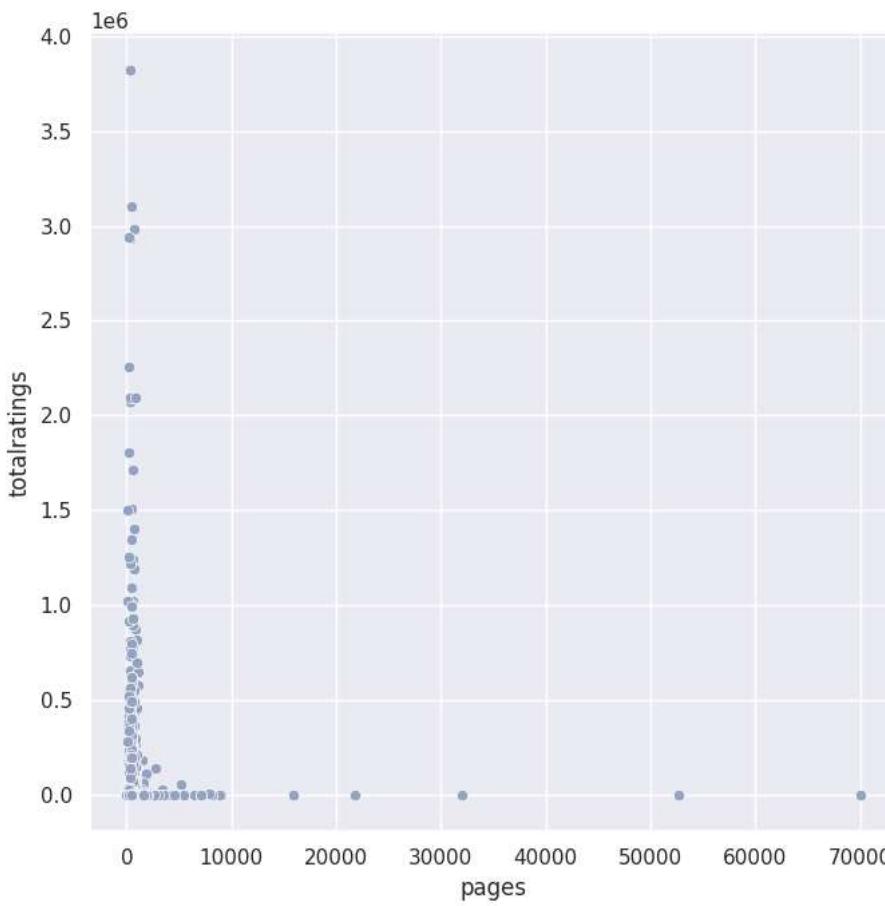
### Relationship between rating and page

```
1 ax = sns.replot(data=data, x="rating", y="pages", color = '#95a3c3', sizes=(100, 200), height=7, marker='o')
```



Relationship between pages and totalratings

```
1 ax = sns.relplot(data=data, x="pages", y="totalratings", color = '#95a3c3', sizes=(100, 200), height=7, marker='o')
```



```
1 title_value = data.title.unique()
```

```
1 from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator
```

```
1 len(new_data.only_author.unique())
```

68767

```
1 new_data.loc[ (new_data['rating'] >= 0) & (new_data['rating'] <= 1), 'rating_between'] = "between_0_to_1"
2 new_data.loc[ (new_data['rating'] > 1) & (new_data['rating'] <= 2), 'rating_between'] = "between_1_to_2"
3 new_data.loc[ (new_data['rating'] > 2) & (new_data['rating'] <= 3), 'rating_between'] = "between_2_to_3"
4 new_data.loc[ (new_data['rating'] > 3) & (new_data['rating'] <= 4), 'rating_between'] = "between_3_to_4"
5 new_data.loc[ (new_data['rating'] > 4) & (new_data['rating'] <= 5), 'rating_between'] = "between_4_to_5"
```

```
1 new_data.head(2)
```

	author	bookformat	desc	genre	img	isbn	isbn13
0	Laurence M. Hauptman	Hardcover	Reveals that several hundred thousand Indians ...	History,Military History,Civil War,American Hi...	<a href="https://i.gr-assets.com/images/S/compressed.ph...">https://i.gr...</a>	002914180X	9.78E+12
1	Charlotte Fiell,Emmanuelle Dirix	Paperback	Fashion Sourcebook - 1920s is the first book i...	Couture,Fashion,Historical,Art,Nonfiction	<a href="https://i.gr-assets.com/images/S/compressed.ph...">https://i.gr...</a>	1906863482	9.78E+12

```
1 trial = new_data[['rating', 'totalratings']]
2 data_model = np.asarray([np.asarray(trial['rating']), np.asarray(trial['totalratings'])]).T
3 data_model
```

```
array([[ 3.52, 33. ],
       [ 4.51, 41. ],
       [ 4.15, 26. ],
       ...,
       [ 4.27, 122. ],
       [ 3.63, 156. ],
       [ 3.83, 12. ]])
```

```
1 from sklearn.cluster import KMeans
```

```
1 # Elbow Method
2
3 score = []
4 x = data_model
5 for cluster in range(1,41):
6     kmeans = KMeans(n_clusters = cluster, init="k-means++", random_state=40)
7     kmeans.fit(x)
8     score.append(kmeans.inertia_)
```

```
    ▾ KMeans
KMeans(n_clusters=1, random_state=40)

    ▾ KMeans
KMeans(n_clusters=2, random_state=40)

    ▾ KMeans
KMeans(n_clusters=3, random_state=40)

    ▾ KMeans
KMeans(n_clusters=4, random_state=40)

    ▾ KMeans
KMeans(n_clusters=5, random_state=40)

    ▾ KMeans
KMeans(n_clusters=6, random_state=40)

    ▾ KMeans
KMeans(n_clusters=7, random_state=40)

    ▾ KMeans
KMeans(random_state=40)

    ▾ KMeans
KMeans(n_clusters=9, random_state=40)

    ▾ KMeans
KMeans(n_clusters=10, random_state=40)

    ▾ KMeans
KMeans(n_clusters=11, random_state=40)

    ▾ KMeans
KMeans(n_clusters=12, random_state=40)

    ▾ KMeans
KMeans(n_clusters=13, random_state=40)

    ▾ KMeans
KMeans(n_clusters=14, random_state=40)

    ▾ KMeans
KMeans(n_clusters=15, random_state=40)

    ▾ KMeans
KMeans(n_clusters=16, random_state=40)

    ▾ KMeans
KMeans(n_clusters=17, random_state=40)

    ▾ KMeans
KMeans(n_clusters=18, random_state=40)

    ▾ KMeans
KMeans(n_clusters=19, random_state=40)

    ▾ KMeans
KMeans(n_clusters=20, random_state=40)

    ▾ KMeans
KMeans(n_clusters=21, random_state=40)

    ▾ KMeans
KMeans(n_clusters=22, random_state=40)

    ▾ KMeans
KMeans(n_clusters=23, random_state=40)

    ▾ KMeans
KMeans(n_clusters=24, random_state=40)

    ▾ KMeans
KMeans(n_clusters=25, random_state=40)

    ▾ KMeans
```

```
KMeans(n_clusters=26, random_state=40)
|   KMeans
KMeans(n_clusters=27, random_state=40)
|   KMeans
KMeans(n_clusters=28, random_state=40)
|   KMeans
KMeans(n_clusters=29, random_state=40)
|   KMeans
KMeans(n_clusters=30, random_state=40)
|   KMeans
KMeans(n_clusters=31, random_state=40)
|   KMeans
KMeans(n_clusters=32, random_state=40)
|   KMeans
KMeans(n_clusters=33, random_state=40)
|   KMeans
```

```
1 # plotting the score
2 plt.figure(figsize=(15, 10))
3 plt.plot(range(1,41), score)
4 plt.title('The Elbow Method')
5 # plt.xlabel('no of clusters')
6 # plt.ylabel('wcss')
7 plt.show()
```

```
<Figure size 1500x1000 with 0 Axes>[<matplotlib.lines.Line2D at 0x7e93bf1d52a0>]Text(0.5, 1.0, 'The Elbow Method')
```

```
1 rating_between_df = new_data['rating_between'].str.get_dummies(sep=',')
2 rating_between_df.head()
```

	between_0_to_1	between_1_to_2	between_2_to_3	between_3_to_4	between_4_to_5	
0	0	0	0	1	0	1
1	0	0	0	0	1	
2	0	0	0	0	1	
3	0	0	0	1	0	
4	0	0	0	1	0	

1

```
1 engine_features = pd.concat([rating_between_df,new_data['rating'], new_data['totalratings']], axis=1)
2 engine_features.head()
```

	between_0_to_1	between_1_to_2	between_2_to_3	between_3_to_4	between_4_to_5	rating	totalratings	
0	0	0	0	1	0	3.52	33	1
1	0	0	0	0	1	4.51	41	
2	0	0	0	0	1	4.15	26	
3	0	0	0	1	0	3.83	6	
4	0	0	0	1	0	4.00	1	

```
1 from sklearn.preprocessing import MinMaxScaler
2 min_max_scaler = MinMaxScaler()
3 engine_features = min_max_scaler.fit_transform(engine_features)
```

0 5 10 15 20 25 30 35 40

```
1 from sklearn import neighbors
```

```
1 engine_model = neighbors.NearestNeighbors(n_neighbors=10, algorithm='ball_tree')
```

```
1 engine_model.fit(engine_features)
```

```
NearestNeighbors
NearestNeighbors(algorithm='ball_tree', n_neighbors=10)
```

```
1 dist, idlist = engine_model.kneighbors(engine_features)
```

```
1 def book_recommendation_engine(book_name):
2     book_list_name = []
3     book_id = new_data[new_data['title'] == book_name].index
4     book_id = book_id[0]
5     # print('book_id', book_id)
6     for newid in idlist[book_id]:
7         # print(newid)
8         book_list_name.append(new_data.loc[newid].title)
9     # print(new_data.loc[newid].title)
10    return book_list_name
```

```
1 book_list_name = book_recommendation_engine('Harry Potter and the Deathly Hallows')
2 book_list_name
```

```
['Harry Potter and the Deathly Hallows',
 'Harry Potter and the Chamber of Secrets',
 'Divergent',
 'The Lion, the Witch and the Wardrobe',
 'A Game of Thrones',
 'The Fault in Our Stars',
 'Jane Eyre',
 'The Little Prince',
```

'A Thousand Splendid Suns',  
'Insurgent']

1

1

1

1

1

1

1

1

1

1

1

1

✓ 0s completed at 1:13 AM

