

# EDS ACTIVITY – 1

Name: Anushka Chavan

Div: CS2

Roll no.:30

Prn: 202401040227

A	B	C	D	E	F
Index	Book Name	Author	Rating	Number of	Score
1	The Invisible	Victoria Scott	4.18	1,184,531	62443
2	The House	T.J. Klune	4.4	718,243	47323
3	Project Hail	Andy Weir	4.51	657,869	45972
4	The Midnight	Matt Haig	3.99	1,967,348	42746
5	I'm Glad M	Jennette M	4.46	1,151,831	37509
6	Fourth Wing	Rebecca Y	4.57	2,071,618	32490
7	The Ballad	Suzanne C	3.96	853,601	24060
8	Piranesi	Susanna Cl	4.23	323,496	23238
9	Lessons in	Bonnie Ga	4.29	1,444,923	23130
10	House of E	Sarah J. M	4.47	815,164	20610
11	Tomorrow	Gabrielle Z	4.15	1,037,972	20547
12	My Dark V	Kate Elizak	4.1	360,089	20223
13	A Court of	Sarah J. M	4.47	1,440,061	19855
14	The Vanish	Brit Benne	4.14	786,310	19749
15	Beach Rea	Emily Henr really		1,304,135	19244
16	The Love F	Ali Hazelw	4.13	1,513,373	18531
17	Babel	R.F. Kuang	4.17	283,448	17606
18	Malibu Ris	Taylor Jenl	4.03	1,118,674	15949
19	The Inherit	Jennifer Ly	4.15	831,277	14214
20	Book Love	Emily Henr	4.13	1,227,947	13253
21	The Four V	Kristin Han	4.3	752,909	12053
22	Under the	T.J. Klune	4.15	255,962	11155
23	Remarkabl	Shelby Var	4.38	747,337	10951
24	Crying in H	Michelle Z	4.26	472,954	10690
25	American I	Jeanine Cu	4.37	606,994	10493

```
import pandas as pd
import numpy as np

# Simulating a mini Goodreads Book Reviews dataset
data = {
    'book_title': [
        'The Great Gatsby', '1984', 'To Kill a Mockingbird', 'Pride and Prejudice',
        'The Catcher in the Rye', 'The Hobbit', 'Fahrenheit 451', 'Jane Eyre',
        'Animal Farm', 'Moby-Dick'
    ],
    'author': [
        'F. Scott Fitzgerald', 'George Orwell', 'Harper Lee', 'Jane Austen',
        'J.D. Salinger', 'J.R.R. Tolkien', 'Ray Bradbury', 'Charlotte Brontë',
        'George Orwell', 'Herman Melville'
    ],
    'rating': [3.9, 4.2, 4.3, 4.3, 3.8, 4.3, 3.9, 4.1, 3.9, 3.5],
    'review_text': [
        'A fascinating classic novel.', 'A chilling depiction of dystopia.',
        'Heartfelt and timeless.', 'Brilliant and witty romance.',
        'A complex coming-of-age story.', 'An adventurous fantasy tale.',
        'Thought-provoking and intense.', 'A passionate narrative.',
        'An allegorical satire.', 'A dense but rewarding read.'
    ]
}

goodreads_df = pd.DataFrame(data)
goodreads_df
```

```
# 1. Total number of books reviewed
goodreads_df.shape[0]
```

Python

```
# 2. List all unique authors
goodreads_df['author'].unique()
```

Python

```
# 3. Count books with rating greater than 4
goodreads_df[goodreads_df['rating'] > 4].shape[0]
```

Python

```
# 4. Find the book with the highest rating
goodreads_df.loc[goodreads_df['rating'].idxmax()]
```

Python

```
# 5. Find the book with the lowest rating
goodreads_df.loc[goodreads_df['rating'].idxmin()]
```

Python

```
# 6. Sort books by rating descending
goodreads_df.sort_values('rating', ascending=False)
```

Python

```
# 7. Find all books written by George Orwell
goodreads_df[goodreads_df['author'] == 'George Orwell']
```

Python

```
# 8. Average book rating
goodreads_df['rating'].mean()
```

Python

```
# 9. Add a column for length of each review
goodreads_df['review_length'] = goodreads_df['review_text'].apply(len)
goodreads_df[['book_title', 'review_length']]
```

Python

```
# 10. Find the book with the longest review
goodreads_df.loc[goodreads_df['review_length'].idxmax()]
```

Python

```
# 11. Create a boolean column 'High Rated' (rating > 4)
goodreads_df['high_rated'] = goodreads_df['rating'] > 4
goodreads_df[['book_title', 'high_rated']]
```

Python

```
# 12. Count how many books are high rated
goodreads_df['high_rated'].sum()
```

Python

```
# 13. Find all reviews containing the word 'classic'
goodreads_df[goodreads_df['review_text'].str.contains('classic', case=False)]
```

Python

```
# 14. Replace 'novel' with 'story' in all reviews
goodreads_df['review_text'] = goodreads_df['review_text'].str.replace('novel', 'story', case=False)
goodreads_df[['book_title', 'review_text']]
```

Python

```
# 15. Find books with 'fantasy' mentioned in review
goodreads_df[goodreads_df['review_text'].str.contains('fantasy', case=False)]
```

Python



```
# 16. Add a new column 'word_count' in reviews
goodreads_df['word_count'] = goodreads_df['review_text'].apply(lambda x: len(x.split()))
goodreads_df[['book_title', 'word_count']]
```

Python

```
# 17. Find the book with the most words in review
goodreads_df.loc[goodreads_df['word_count'].idxmax()]
```

Python

```
# 18. Create a column 'contains_romance' if 'romance' present
goodreads_df['contains_romance'] = goodreads_df['review_text'].str.contains('romance', case=False)
goodreads_df[['book_title', 'contains_romance']]
```

Python

```
# 19. How many reviews talk about 'adventure'?
goodreads_df[goodreads_df['review_text'].str.contains('adventure', case=False)].shape[0]
```

Python

```
# 20. Statistical summary of review lengths
goodreads_df['review_length'].describe()
```

Python

