🎓 ACTIVITY NO. 01

🎯 **Topic:** Data Analysis using NumPy & Pandas

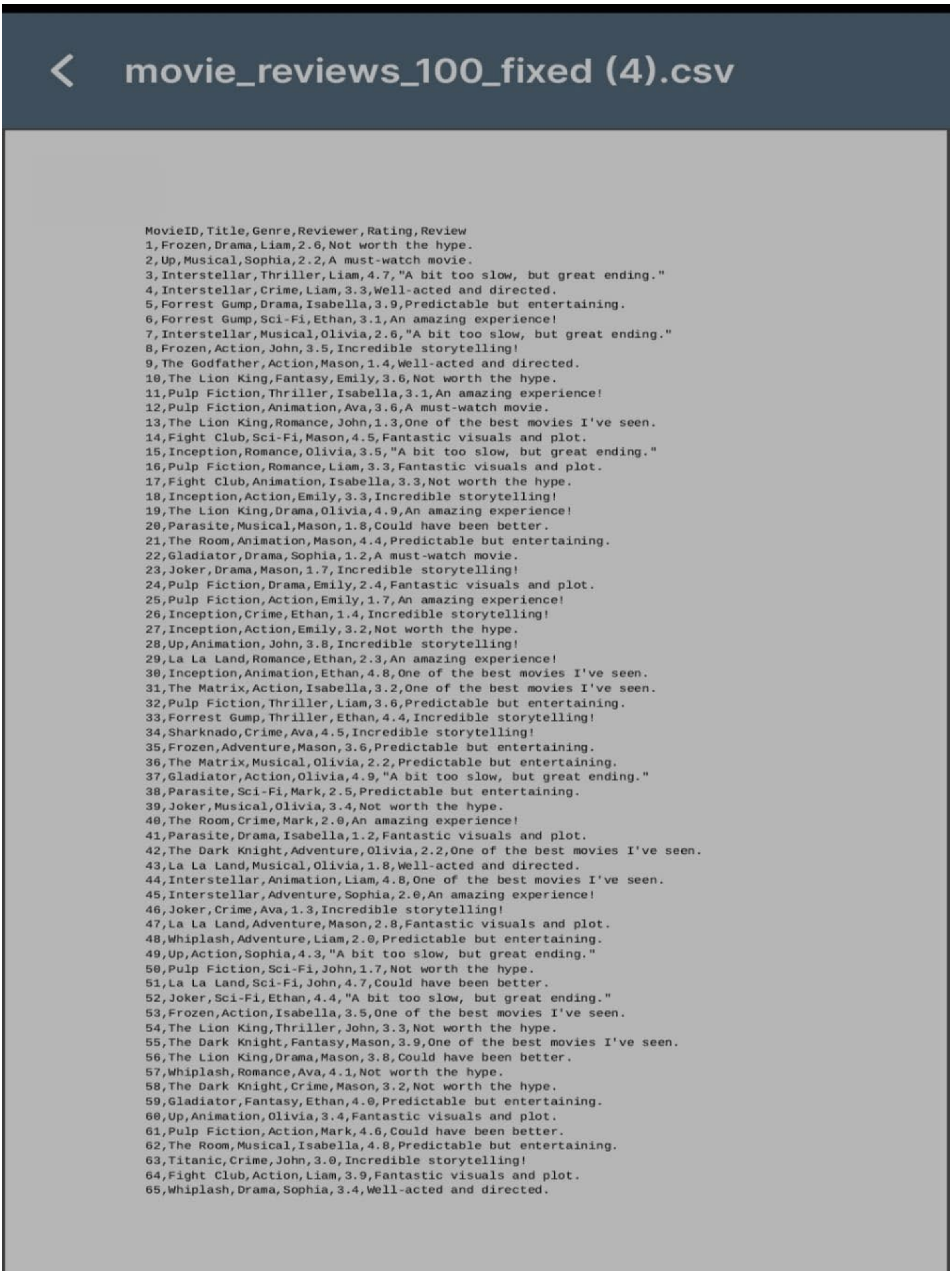
🎯 **Dataset:** IMDB Movie Review Dataset

NAME:AAKANKSHA VIJAY KHARE

PRN:202401040005

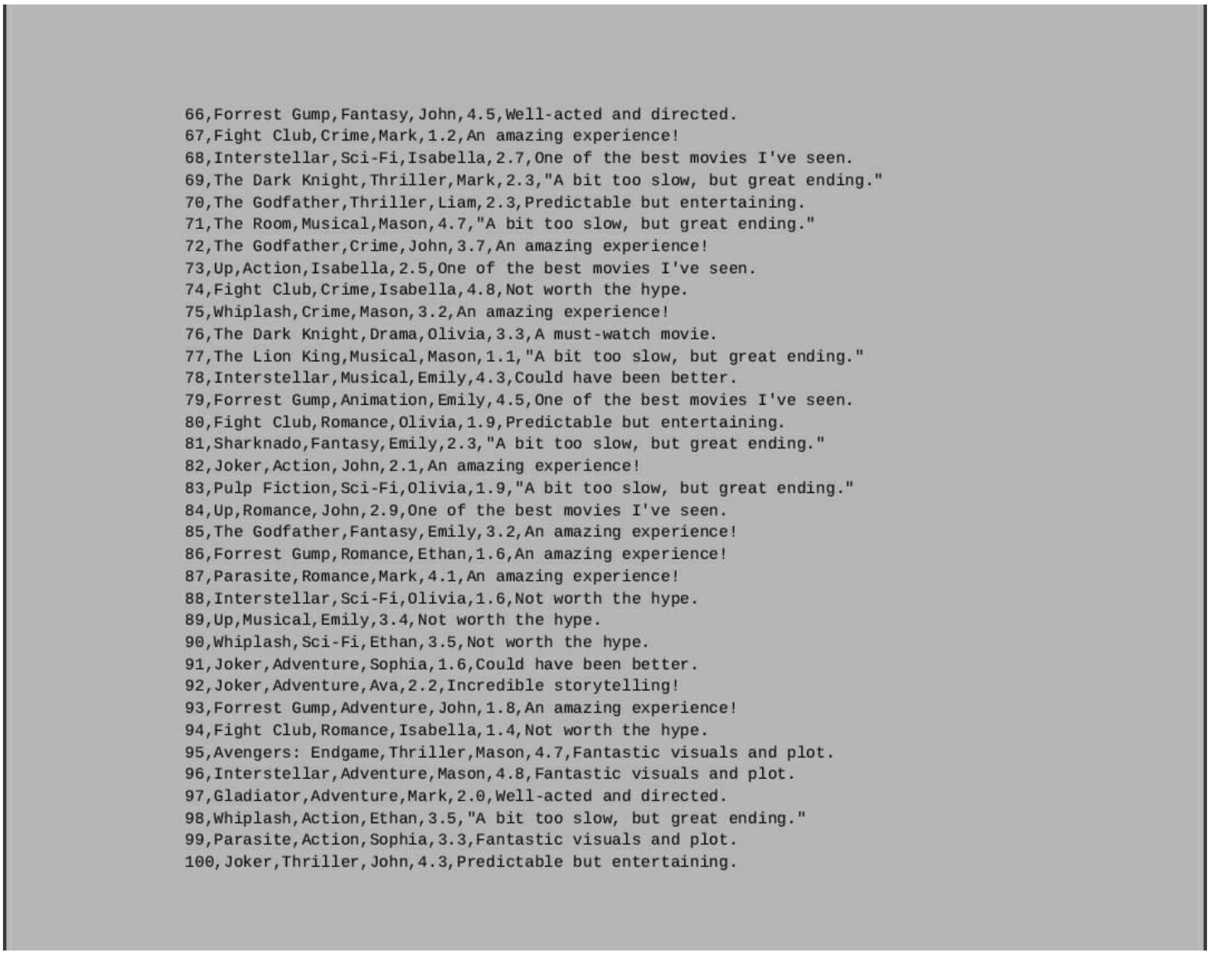
ROLL NO:CS4-13

DATASET:



movie_reviews_100_fixed (4).csv

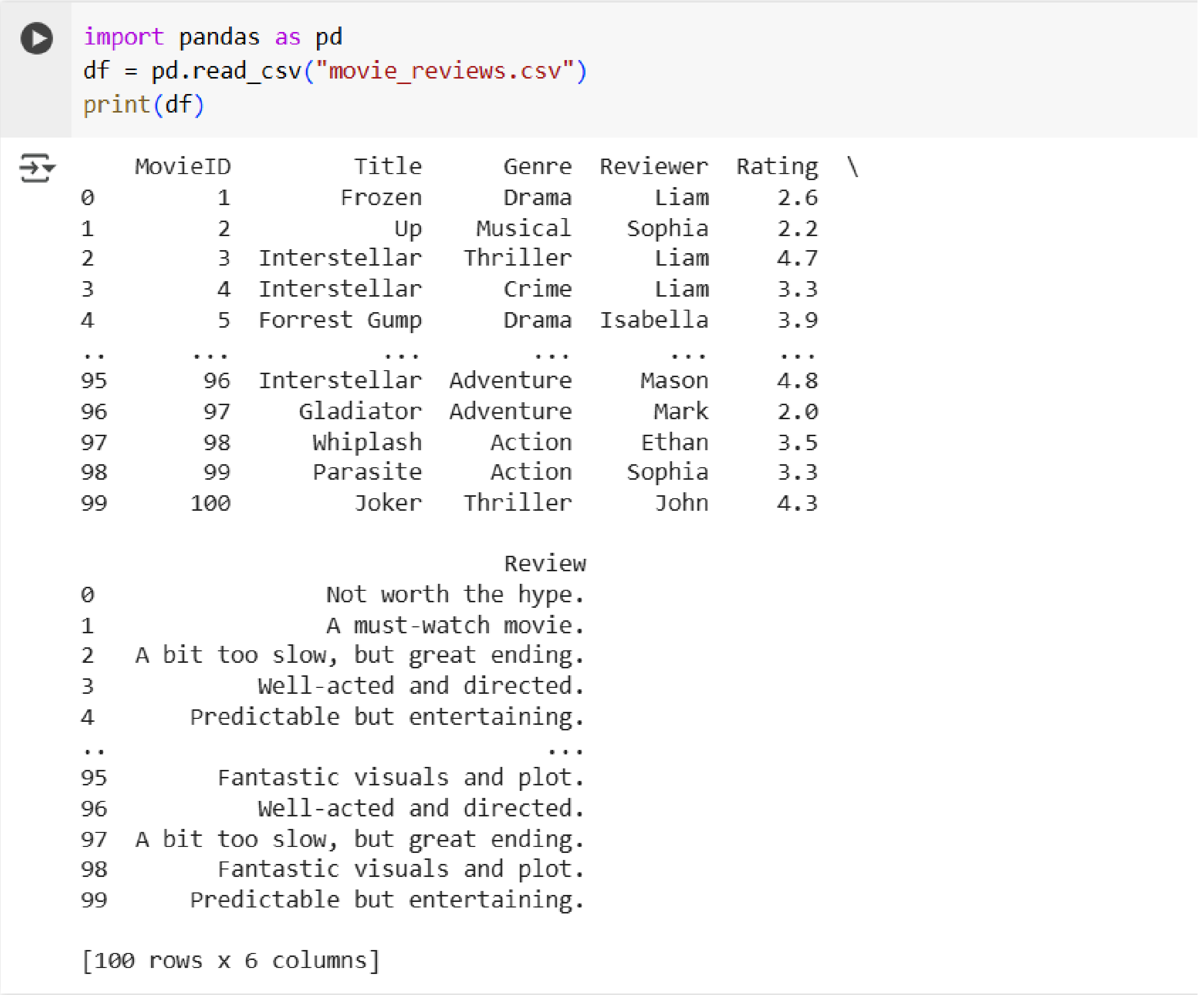
```
MovieID,Title,Genre,Reviewer,Rating,Review
1,Frozen,Drama,Liam,2.6,Not worth the hype.
2,Up,Musical,Sophia,2.2,A must-watch movie.
3,Interstellar,Thriller,Liam,4.7,"A bit too slow, but great ending."
4,Interstellar,Crime,Liam,3.3,Well-acted and directed.
5,Forrest Gump,Drama,Isabella,3.9,Predictable but entertaining.
6,Forrest Gump,Sci-Fi,Ethan,3.1,An amazing experience!
7,Interstellar,Musical,Olivia,2.6,"A bit too slow, but great ending."
8,Frozen,Action,John,3.5,Incredible storytelling!
9,The Godfather,Action,Mason,1.4,Well-acted and directed.
10,The Lion King,Fantasy,Emily,3.6,Not worth the hype.
11,Pulp Fiction,Thriller,Isabella,3.1,An amazing experience!
12,Pulp Fiction,Animation,Ava,3.6,A must-watch movie.
13,The Lion King,Romance,John,1.3,One of the best movies I've seen.
14,Fight Club,Sci-Fi,Mason,4.5,Fantastic visuals and plot.
15,Inception,Romance,Olivia,3.5,"A bit too slow, but great ending."
16,Pulp Fiction,Romance,Liam,3.3,Fantastic visuals and plot.
17,Fight Club,Animation,Isabella,3.3,Not worth the hype.
18,Inception,Action,Emily,3.3,Incredible storytelling!
19,The Lion King,Drama,Olivia,4.9,An amazing experience!
20,Parasite,Musical,Mason,1.8,Could have been better.
21,The Room,Animation,Mason,4.4,Predictable but entertaining.
22,Gladiator,Drama,Sophia,1.2,A must-watch movie.
23,Joker,Drama,Mason,1.7,Incredible storytelling!
24,Pulp Fiction,Drama,Emily,2.4,Fantastic visuals and plot.
25,Pulp Fiction,Action,Emily,1.7,An amazing experience!
26,Inception,Crime,Ethan,1.4,Incredible storytelling!
27,Inception,Action,Emily,3.2,Not worth the hype.
28,Up,Animation,John,3.8,Incredible storytelling!
29,La La Land,Romance,Ethan,2.3,An amazing experience!
30,Inception,Animation,Ethan,4.8,One of the best movies I've seen.
31,The Matrix,Action,Isabella,3.2,One of the best movies I've seen.
32,Pulp Fiction,Thriller,Liam,3.6,Predictable but entertaining.
33,Forrest Gump,Thriller,Ethan,4.4,Incredible storytelling!
34,Sharknado,Crime,Ava,4.5,Incredible storytelling!
35,Frozen,Adventure,Mason,3.6,Predictable but entertaining.
36,The Matrix,Musical,Olivia,2.2,Predictable but entertaining.
37,Gladiator,Action,Olivia,4.9,"A bit too slow, but great ending."
38,Parasite,Sci-Fi,Mark,2.5,Predictable but entertaining.
39,Joker,Musical,Olivia,3.4,Not worth the hype.
40,The Room,Crime,Mark,2.0,An amazing experience!
41,Parasite,Drama,Isabella,1.2,Fantastic visuals and plot.
42,The Dark Knight,Adventure,Olivia,2.2,One of the best movies I've seen.
43,La La Land,Musical,Olivia,1.8,Well-acted and directed.
44,Interstellar,Animation,Liam,4.8,One of the best movies I've seen.
45,Interstellar,Adventure,Sophia,2.0,An amazing experience!
46,Joker,Crime,Ava,1.3,Incredible storytelling!
47,La La Land,Adventure,Mason,2.8,Fantastic visuals and plot.
48,Whiplash,Adventure,Liam,2.0,Predictable but entertaining.
49,Up,Action,Sophia,4.3,"A bit too slow, but great ending."
50,Pulp Fiction,Sci-Fi,John,1.7,Not worth the hype.
51,La La Land,Sci-Fi,John,4.7,Could have been better.
52,Joker,Sci-Fi,Ethan,4.4,"A bit too slow, but great ending."
53,Frozen,Action,Isabella,3.5,One of the best movies I've seen.
54,The Lion King,Thriller,John,3.3,Not worth the hype.
55,The Dark Knight,Fantasy,Mason,3.9,One of the best movies I've seen.
56,The Lion King,Drama,Mason,3.8,Could have been better.
57,Whiplash,Romance,Ava,4.1,Not worth the hype.
58,The Dark Knight,Crime,Mason,3.2,Not worth the hype.
59,Gladiator,Fantasy,Ethan,4.0,Predictable but entertaining.
60,Up,Animation,Olivia,3.4,Fantastic visuals and plot.
61,Pulp Fiction,Action,Mark,4.6,Could have been better.
62,The Room,Musical,Isabella,4.8,Predictable but entertaining.
63,Titanic,Crime,John,3.0,Incredible storytelling!
64,Fight Club,Action,Liam,3.9,Fantastic visuals and plot.
65,Whiplash,Drama,Sophia,3.4,Well-acted and directed.
```



```
66,Forrest Gump,Fantasy,John,4.5,Well-acted and directed.
67,Fight Club,Crime,Mark,1.2,An amazing experience!
68,Interstellar,Sci-Fi,Isabella,2.7,One of the best movies I've seen.
69,The Dark Knight,Thriller,Mark,2.3,"A bit too slow, but great ending."
70,The Godfather,Thriller,Liam,2.3,Predictable but entertaining.
71,The Room,Musical,Mason,4.7,"A bit too slow, but great ending."
72,The Godfather,Crime,John,3.7,An amazing experience!
73,Up,Action,Isabella,2.5,One of the best movies I've seen.
74,Fight Club,Crime,Isabella,4.8,Not worth the hype.
75,Whiplash,Crime,Mason,3.2,An amazing experience!
76,The Dark Knight,Drama,Olivia,3.3,A must-watch movie.
77,The Lion King,Musical,Mason,1.1,"A bit too slow, but great ending."
78,Interstellar,Musical,Emily,4.3,Could have been better.
79,Forrest Gump,Animation,Emily,4.5,One of the best movies I've seen.
80,Fight Club,Romance,Olivia,1.9,Predictable but entertaining.
81,Sharknado,Fantasy,Emily,2.3,"A bit too slow, but great ending."
82,Joker,Action,John,2.1,An amazing experience!
83,Pulp Fiction,Sci-Fi,Olivia,1.9,"A bit too slow, but great ending."
84,Up,Romance,John,2.9,One of the best movies I've seen.
85,The Godfather,Fantasy,Emily,3.2,An amazing experience!
86,Forrest Gump,Romance,Ethan,1.6,An amazing experience!
87,Parasite,Romance,Mark,4.1,An amazing experience!
88,Interstellar,Sci-Fi,Olivia,1.6,Not worth the hype.
89,Up,Musical,Emily,3.4,Not worth the hype.
90,Whiplash,Sci-Fi,Ethan,3.5,Not worth the hype.
91,Joker,Adventure,Sophia,1.6,Could have been better.
92,Joker,Adventure,Ava,2.2,Incredible storytelling!
93,Forrest Gump,Adventure,John,1.8,An amazing experience!
94,Fight Club,Romance,Isabella,1.4,Not worth the hype.
95,Avengers: Endgame,Thriller,Mason,4.7,Fantastic visuals and plot.
96,Interstellar,Adventure,Mason,4.8,Fantastic visuals and plot.
97,Gladiator,Adventure,Mark,2.0,Well-acted and directed.
98,Whiplash,Action,Ethan,3.5,"A bit too slow, but great ending."
99,Parasite,Action,Sophia,3.3,Fantastic visuals and plot.
100,Joker,Thriller,John,4.3,Predictable but entertaining.
```

🔎 PROBLEM STATEMENTS

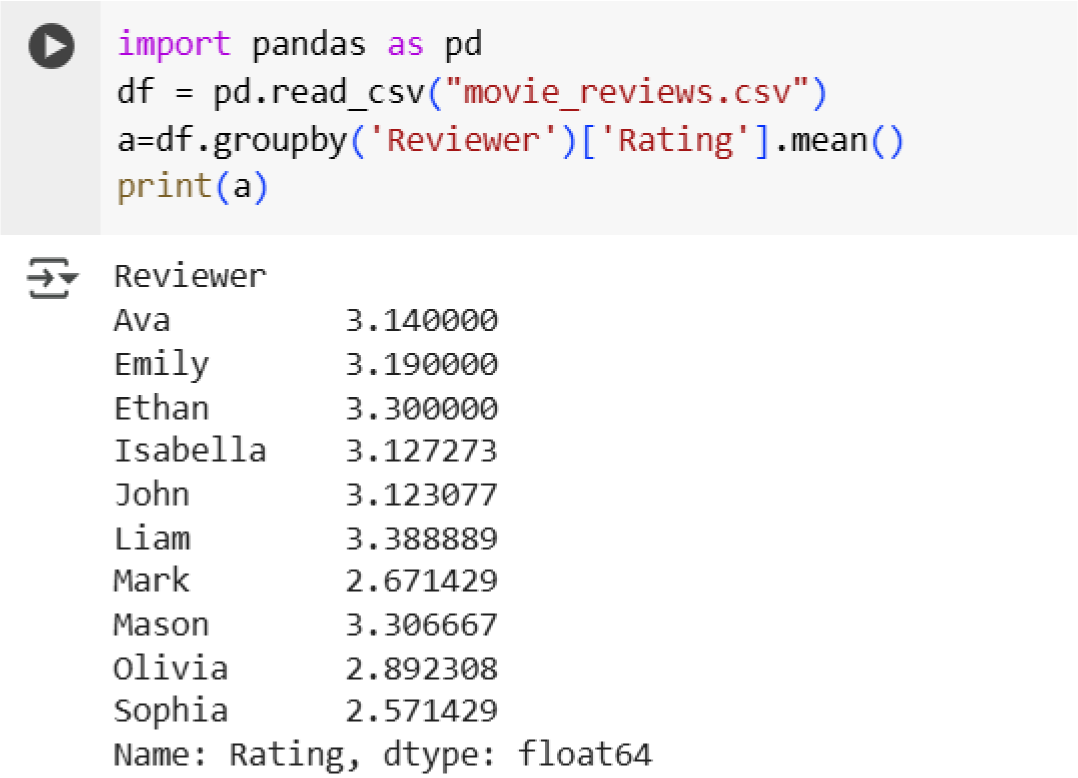
◆ DATA LOADING



```python
import pandas as pd
df = pd.read_csv("movie_reviews.csv")
print(df)
```

```
    MovieID         Title      Genre  Reviewer  Rating  \
0         1        Frozen      Drama      Liam     2.6   
1         2            Up    Musical    Sophia     2.2   
2         3  Interstellar   Thriller      Liam     4.7   
3         4  Interstellar      Crime      Liam     3.3   
4         5  Forrest Gump      Drama  Isabella     3.9   
..      ...           ...        ...       ...     ...   
95       96  Interstellar  Adventure     Mason     4.8   
96       97     Gladiator  Adventure      Mark     2.0   
97       98      Whiplash     Action     Ethan     3.5   
98       99      Parasite     Action    Sophia     3.3   
99      100         Joker   Thriller      John     4.3   

                               Review  
0                 Not worth the hype.  
1                 A must-watch movie.  
2   A bit too slow, but great ending.  
3            Well-acted and directed.  
4       Predictable but entertaining.  
..                                ...  
95        Fantastic visuals and plot.  
96           Well-acted and directed.  
97  A bit too slow, but great ending.  
98        Fantastic visuals and plot.  
99      Predictable but entertaining.  

[100 rows x 6 columns]
```

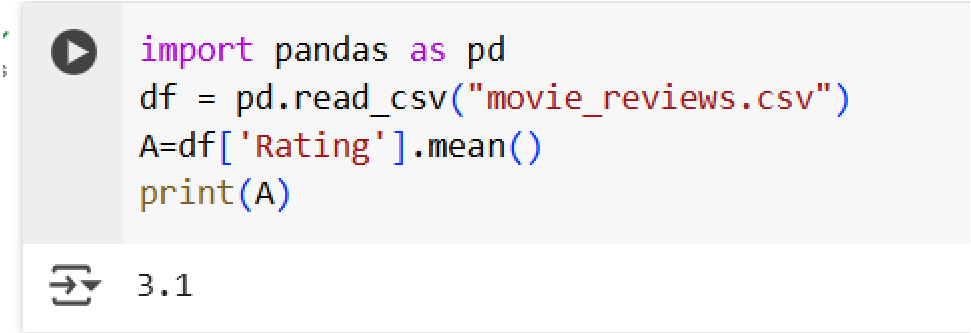
1.Find the average rating given by each reviewer.



```python
import pandas as pd
df = pd.read_csv("movie_reviews.csv")
a=df.groupby('Reviewer')['Rating'].mean()
print(a)
```

```
Reviewer
Ava         3.140000
Emily       3.190000
Ethan       3.300000
Isabella    3.127273
John        3.123077
Liam        3.388889
Mark        2.671429
Mason       3.306667
Olivia      2.892308
Sophia      2.571429
Name: Rating, dtype: float64
```

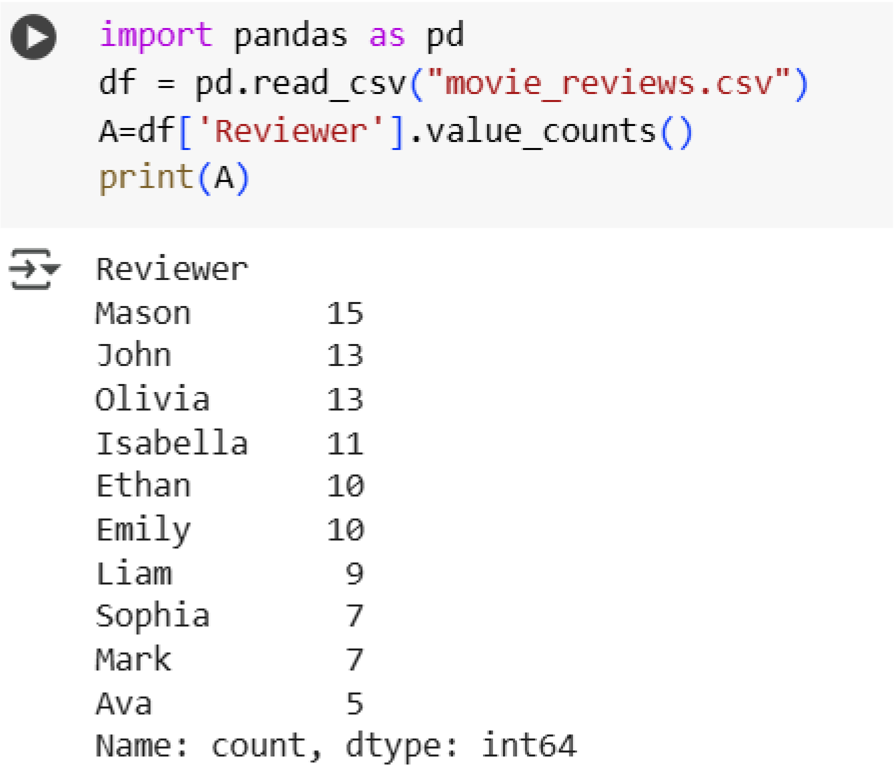
2.Calculate the average rating of all movies.



```python
import pandas as pd
df = pd.read_csv("movie_reviews.csv")
A=df['Rating'].mean()
print(A)
```

```
3.1
```

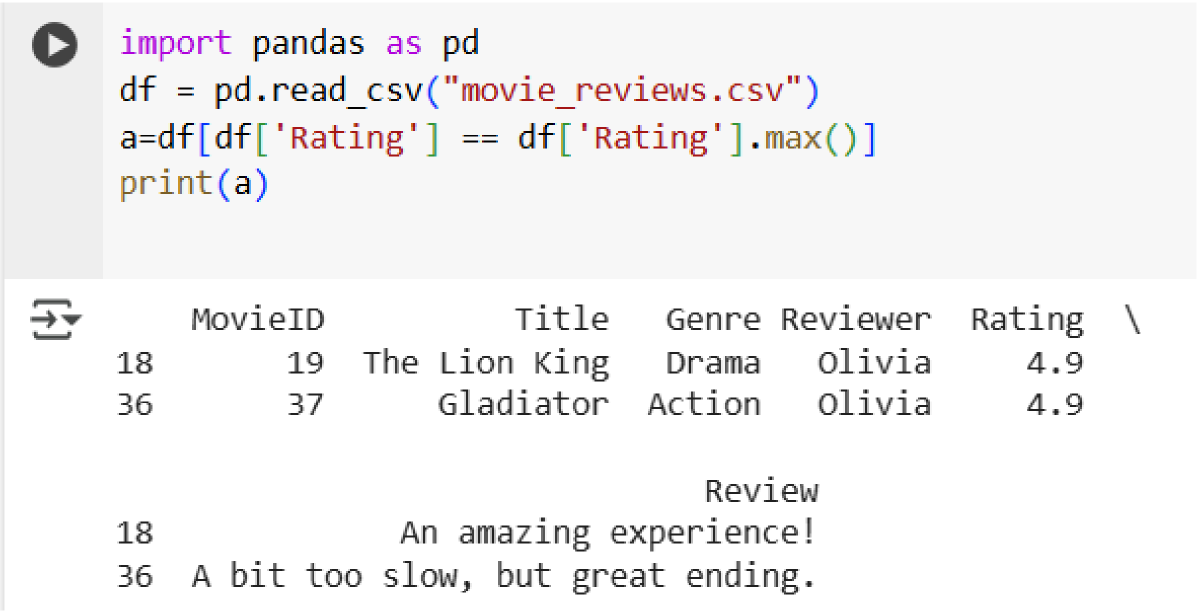
3.Find the number of reviews given by each reviewer.



```
import pandas as pd
df = pd.read_csv("movie_reviews.csv")
A=df['Reviewer'].value_counts()
print(A)
```

```
Reviewer
Mason       15
John        13
Olivia      13
Isabella    11
Ethan       10
Emily       10
Liam         9
Sophia       7
Mark         7
Ava          5
Name: count, dtype: int64
```

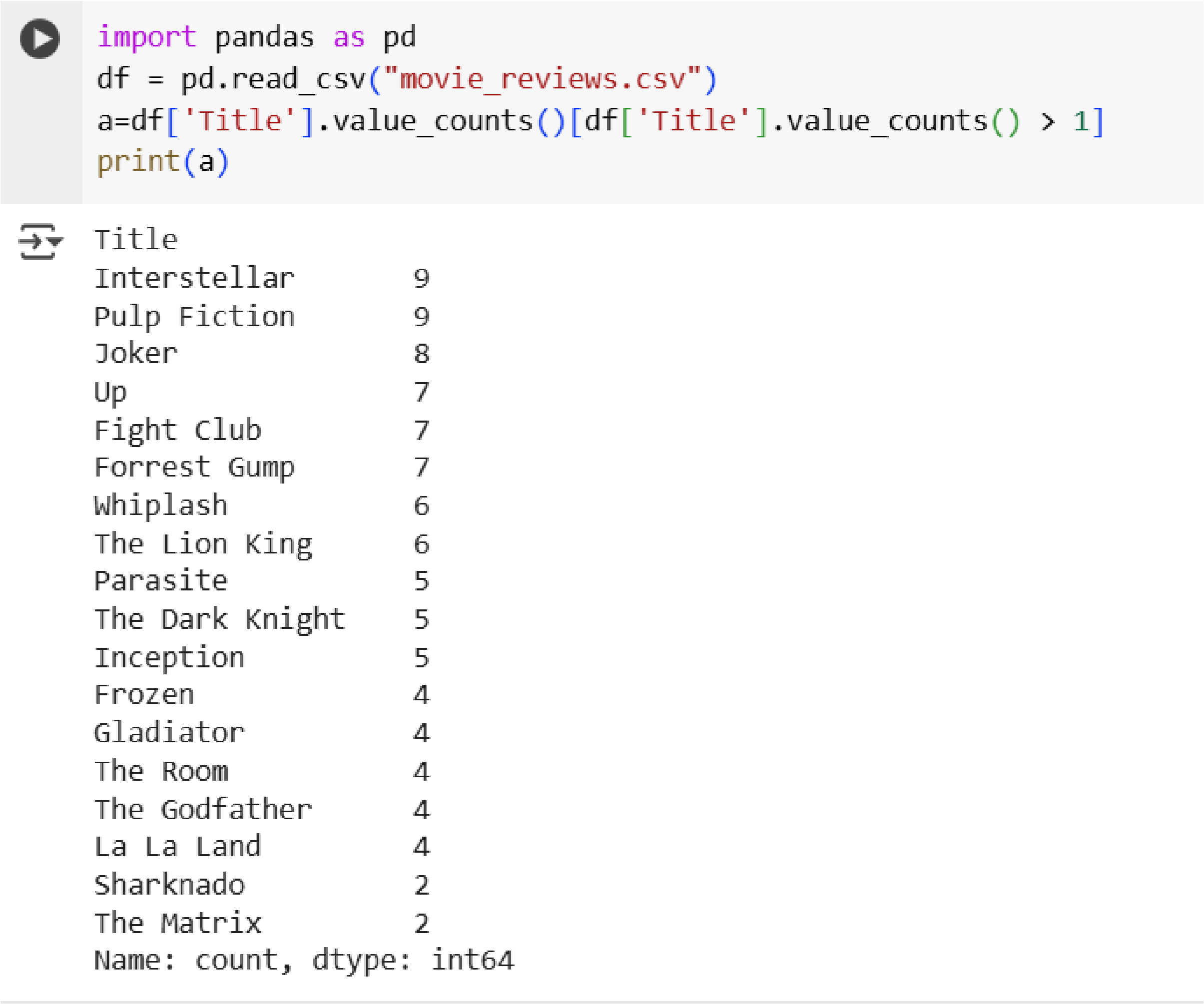
4.Find the highest-rated movie.



```
import pandas as pd
df = pd.read_csv("movie_reviews.csv")
a=df[df['Rating'] == df['Rating'].max()]
print(a)
```

```
    MovieID          Title   Genre Reviewer  Rating  \
18       19  The Lion King   Drama   Olivia     4.9   
36       37      Gladiator  Action   Olivia     4.9   

                               Review  
18             An amazing experience!  
36  A bit too slow, but great ending.  
```

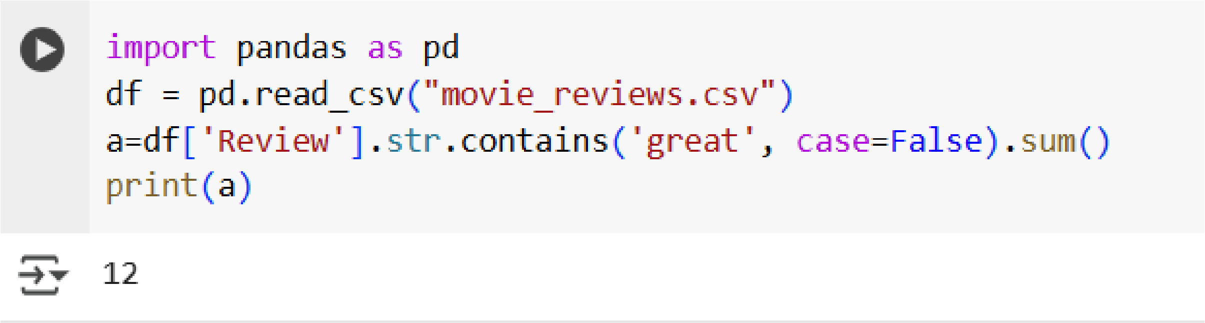
5.Identify movies that have more than one review.



```
import pandas as pd
df = pd.read_csv("movie_reviews.csv")
a=df['Title'].value_counts()[df['Title'].value_counts() > 1]
print(a)
```

```
Title
Interstellar      9
Pulp Fiction      9
Joker             8
Up                7
Fight Club        7
Forrest Gump      7
Whiplash          6
The Lion King     6
Parasite          5
The Dark Knight   5
Inception         5
Frozen            4
Gladiator         4
The Room          4
The Godfather     4
La La Land        4
Sharknado         2
The Matrix        2
Name: count, dtype: int64
```

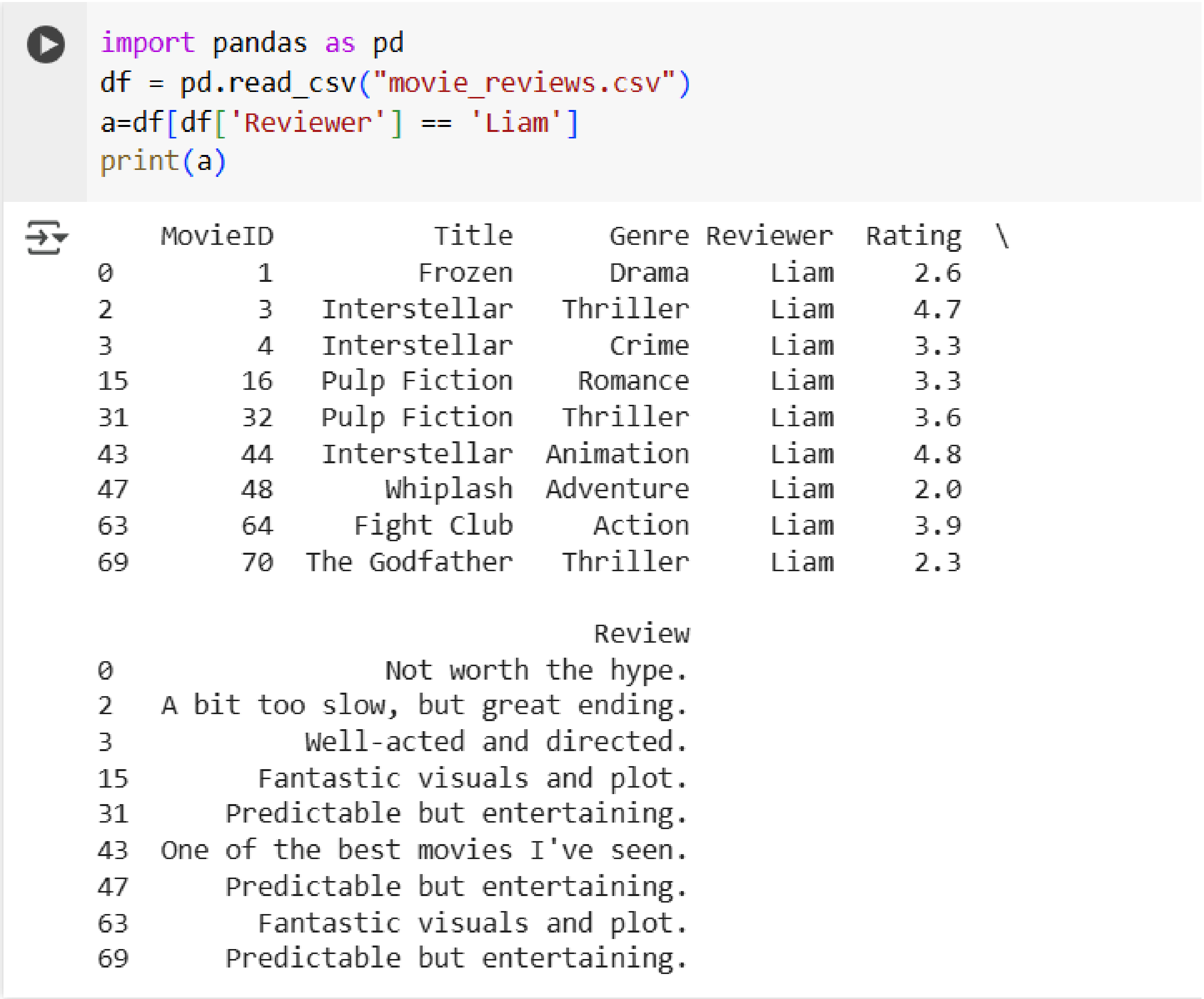
6.Count the number of reviews that contain the word 'great'.



```
import pandas as pd
df = pd.read_csv("movie_reviews.csv")
a=df['Review'].str.contains('great', case=False).sum()
print(a)
```

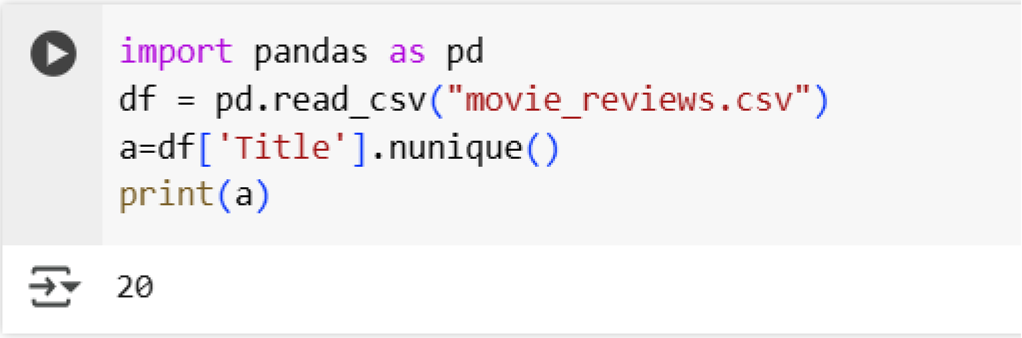
```
12
```

7.List all movies reviewed by 'Liam'.



```
import pandas as pd
df = pd.read_csv("movie_reviews.csv")
a=df[df['Reviewer'] == 'Liam']
print(a)
```

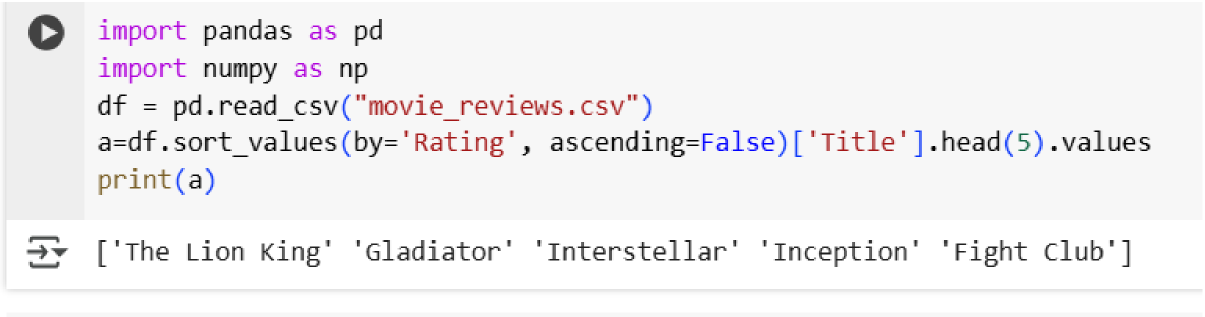
```
    MovieID          Title      Genre Reviewer  Rating  \
0         1         Frozen      Drama     Liam     2.6   
2         3   Interstellar   Thriller     Liam     4.7   
3         4   Interstellar      Crime     Liam     3.3   
15       16   Pulp Fiction    Romance     Liam     3.3   
31       32   Pulp Fiction   Thriller     Liam     3.6   
43       44   Interstellar  Animation     Liam     4.8   
47       48       Whiplash  Adventure     Liam     2.0   
63       64     Fight Club     Action     Liam     3.9   
69       70  The Godfather   Thriller     Liam     2.3   

                               Review  
0                 Not worth the hype.  
2   A bit too slow, but great ending.  
3            Well-acted and directed.  
15        Fantastic visuals and plot.  
31      Predictable but entertaining.  
43  One of the best movies I've seen.  
47      Predictable but entertaining.  
63        Fantastic visuals and plot.  
69      Predictable but entertaining.  
```



```
import pandas as pd
df = pd.read_csv("movie_reviews.csv")
a=df['Title'].nunique()
print(a)
```

```
20
```

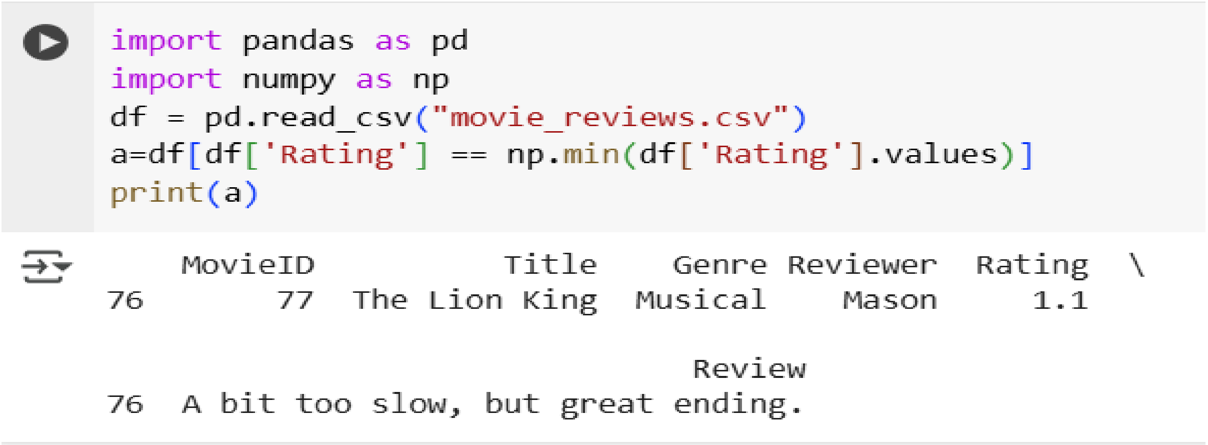
9.Get the titles of top 5 rated movies



```
import pandas as pd
import numpy as np
df = pd.read_csv("movie_reviews.csv")
a=df.sort_values(by='Rating', ascending=False)['Title'].head(5).values
print(a)
```

```
['The Lion King' 'Gladiator' 'Interstellar' 'Inception' 'Fight Club']
```

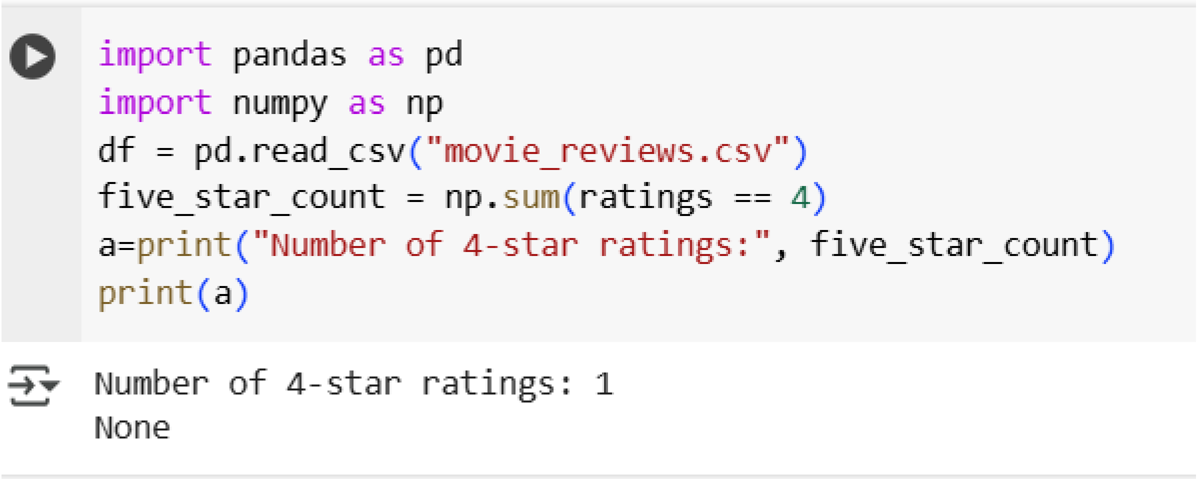
10.Find movies with the minimum rating using NumPy.



```
import pandas as pd
import numpy as np
df = pd.read_csv("movie_reviews.csv")
a=df[df['Rating'] == np.min(df['Rating'].values)]
print(a)
```

```
    MovieID           Title    Genre Reviewer  Rating  \
76       77  The Lion King  Musical    Mason     1.1

                            Review
76  A bit too slow, but great ending.
```

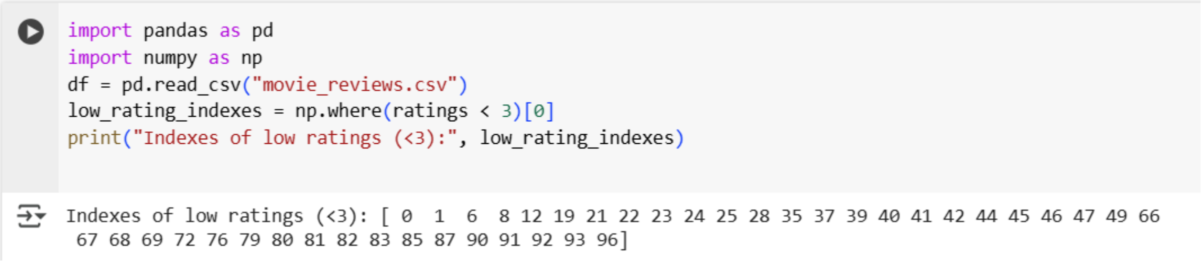
11.Count how many ratings are 4 stars



```
import pandas as pd
import numpy as np
df = pd.read_csv("movie_reviews.csv")
five_star_count = np.sum(ratings == 4)
a=print("Number of 4-star ratings:", five_star_count)
print(a)
```

```
Number of 4-star ratings: 1
None
```

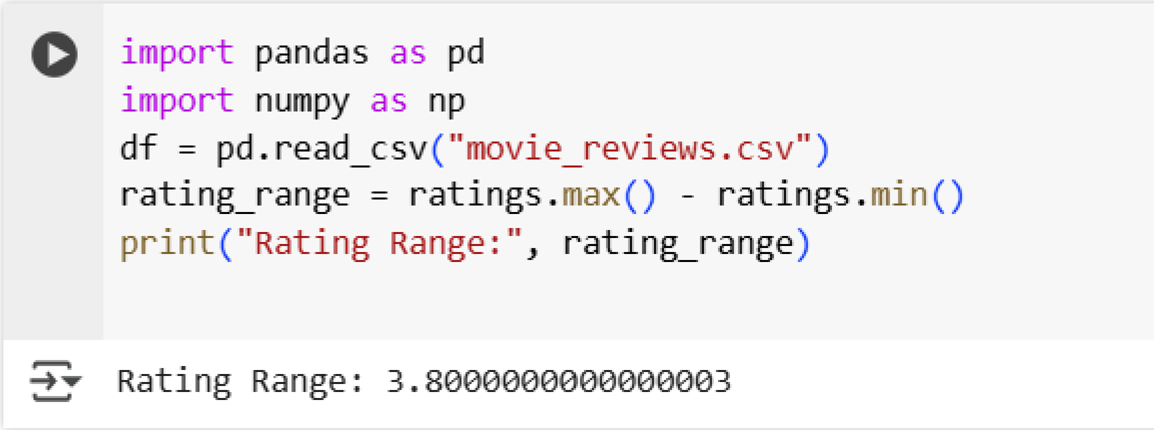
12.Identify indexes of all ratings below 3



```
import pandas as pd
import numpy as np
df = pd.read_csv("movie_reviews.csv")
low_rating_indexes = np.where(ratings < 3)[0]
print("Indexes of low ratings (<3):", low_rating_indexes)
```

```
Indexes of low ratings (<3): [ 0  1  6  8 12 19 21 22 23 24 25 28 35 37 39 40 41 42 44 45 46 47 49 66
 67 68 69 72 76 79 80 81 82 83 85 87 90 91 92 93 96]
```

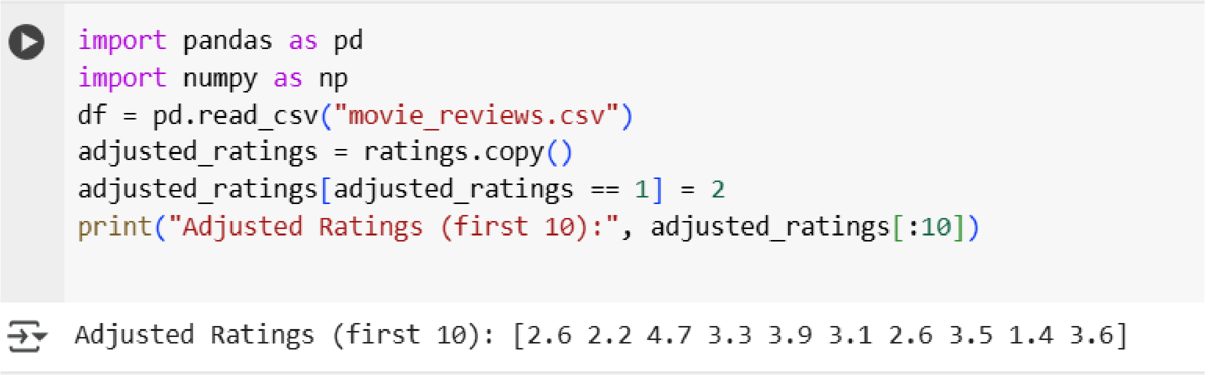
13.Find the difference between the highest and lowest rating using NumPy.



```
import pandas as pd
import numpy as np
df = pd.read_csv("movie_reviews.csv")
rating_range = ratings.max() - ratings.min()
print("Rating Range:", rating_range)
```

```
Rating Range: 3.8000000000000003
```

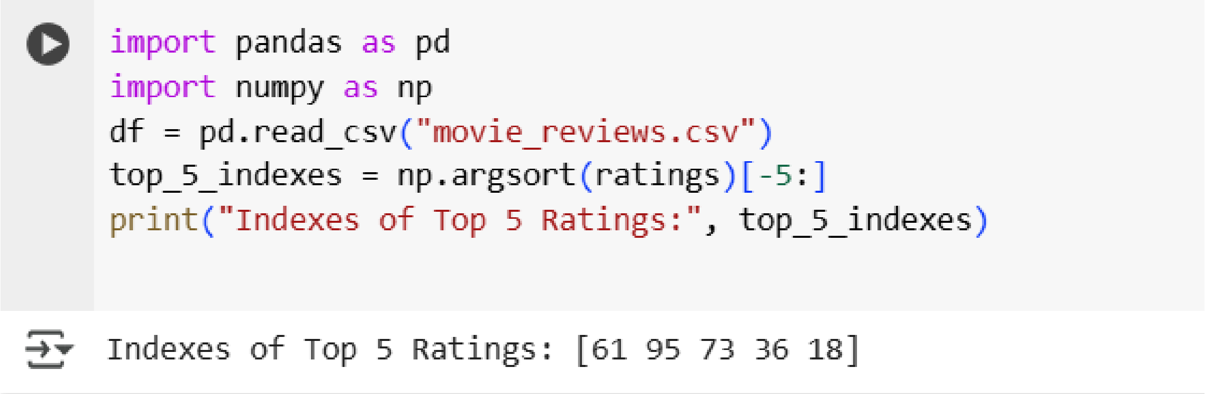
14.Replace all ratings of 1 with 2



```
import pandas as pd
import numpy as np
df = pd.read_csv("movie_reviews.csv")
adjusted_ratings = ratings.copy()
adjusted_ratings[adjusted_ratings == 1] = 2
print("Adjusted Ratings (first 10):", adjusted_ratings[:10])
```

```
Adjusted Ratings (first 10): [2.6 2.2 4.7 3.3 3.9 3.1 2.6 3.5 1.4 3.6]
```

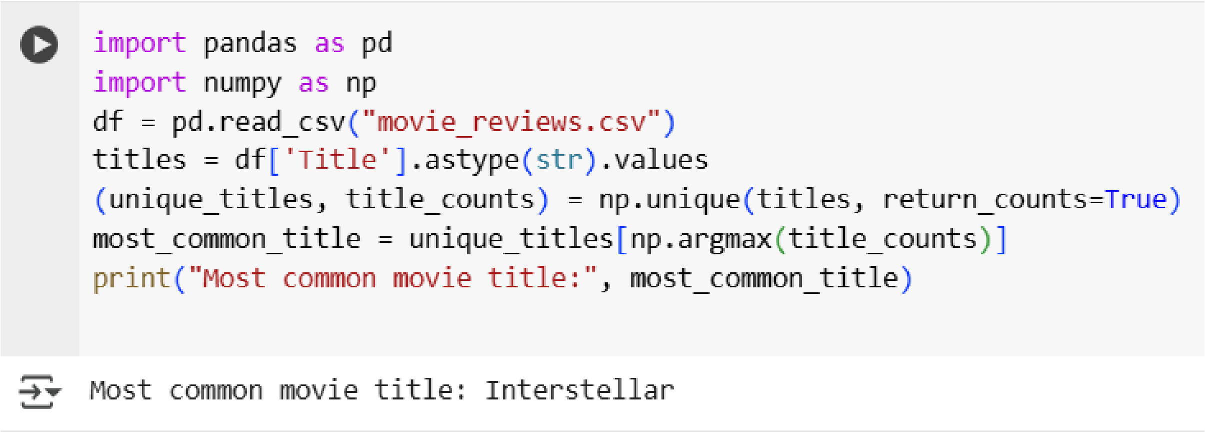
15.Get indices of top 5 ratings



```
import pandas as pd
import numpy as np
df = pd.read_csv("movie_reviews.csv")
top_5_indexes = np.argsort(ratings)[-5:]
print("Indexes of Top 5 Ratings:", top_5_indexes)
```

```
Indexes of Top 5 Ratings: [61 95 73 36 18]
```

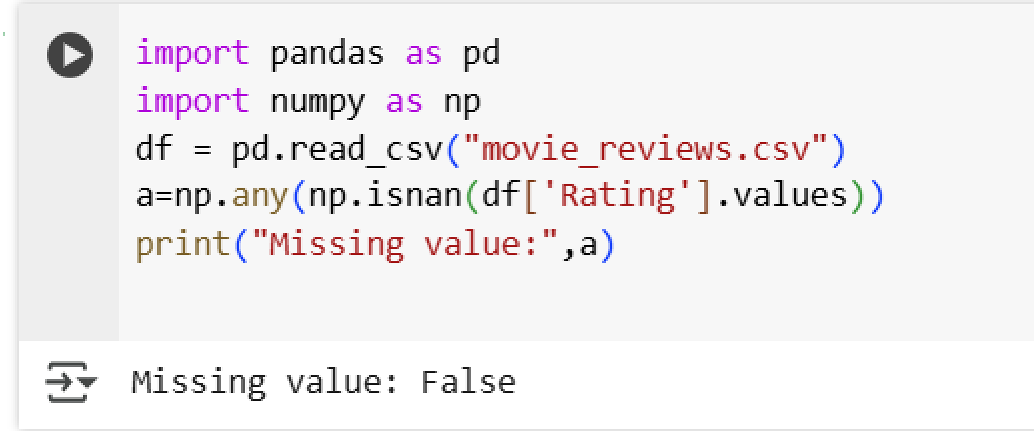
16.Find the most common movie title



```
import pandas as pd
import numpy as np
df = pd.read_csv("movie_reviews.csv")
titles = df['Title'].astype(str).values
(unique_titles, title_counts) = np.unique(titles, return_counts=True)
most_common_title = unique_titles[np.argmax(title_counts)]
print("Most common movie title:", most_common_title)
```

```
Most common movie title: Interstellar
```

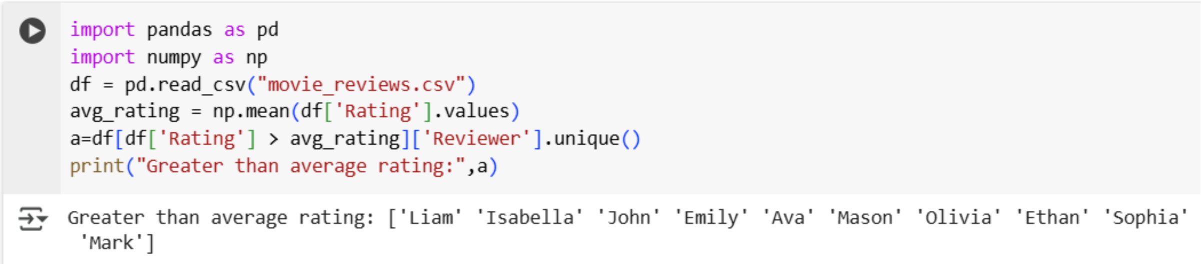
17.Use NumPy to check for missing values in ratings.



```
import pandas as pd
import numpy as np
df = pd.read_csv("movie_reviews.csv")
a=np.any(np.isnan(df['Rating'].values))
print("Missing value:",a)
```

```
Missing value: False
```

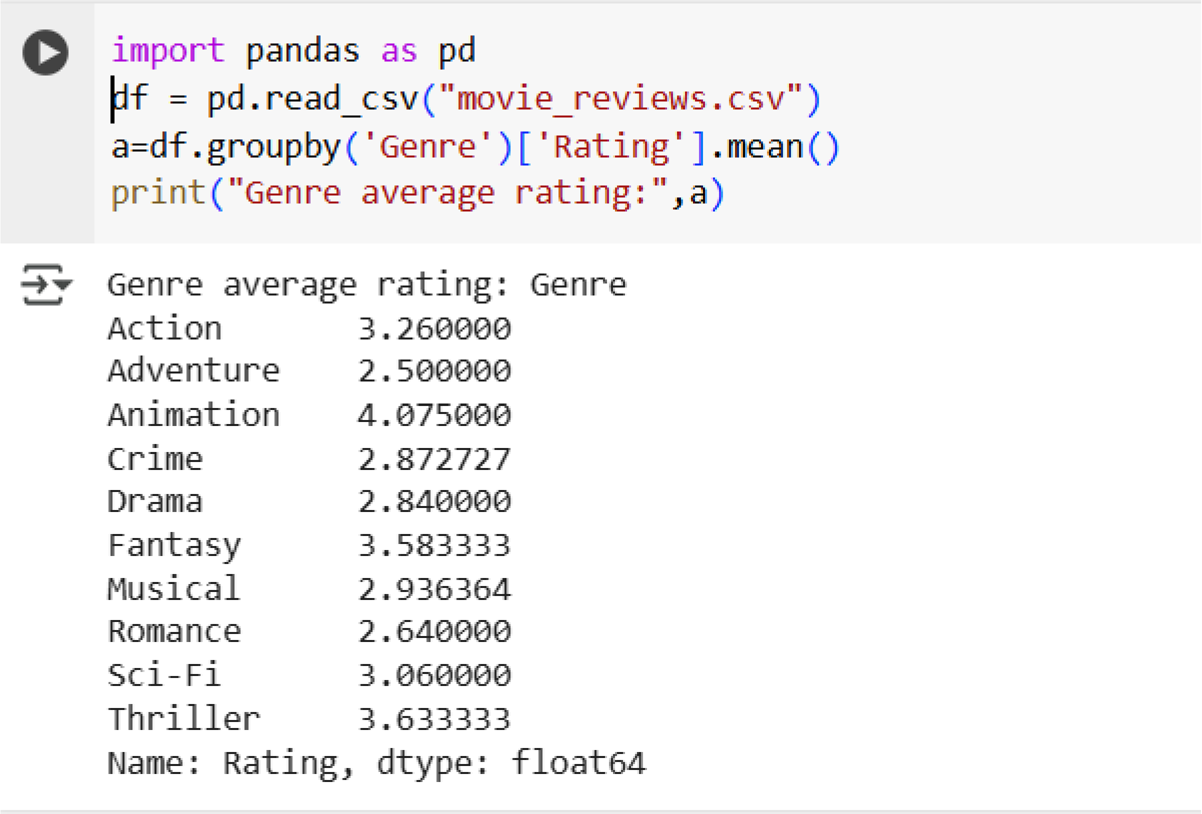
18.Find all reviewers who gave more than the average rating



```
import pandas as pd
import numpy as np
df = pd.read_csv("movie_reviews.csv")
avg_rating = np.mean(df['Rating'].values)
a=df[df['Rating'] > avg_rating]['Reviewer'].unique()
print("Greater than average rating:",a)
```

```
Greater than average rating: ['Liam' 'Isabella' 'John' 'Emily' 'Ava' 'Mason' 'Olivia' 'Ethan' 'Sophia'
 'Mark']
```

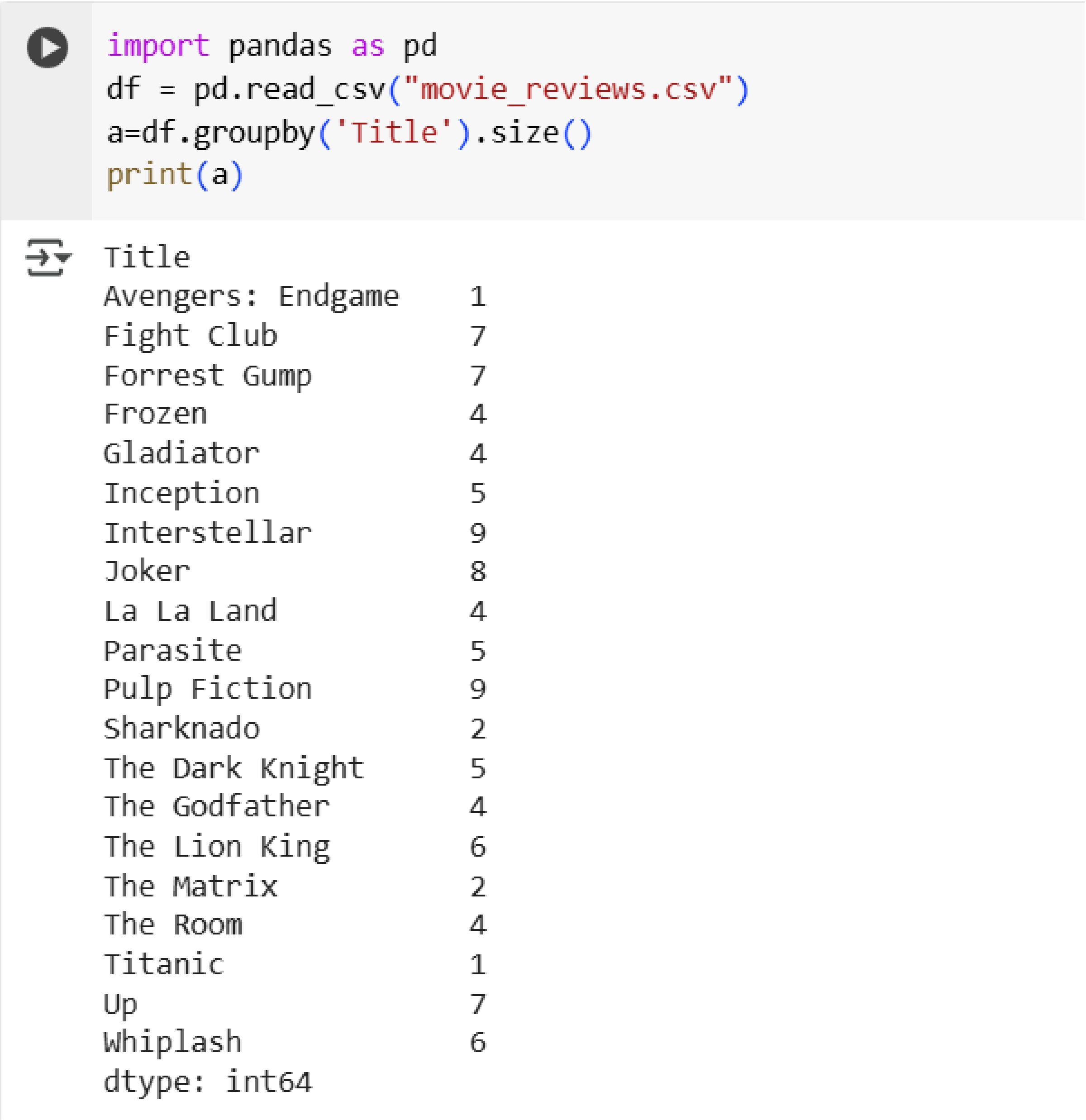
19.Find the average rating per genre using Pandas.



```
import pandas as pd
df = pd.read_csv("movie_reviews.csv")
a=df.groupby('Genre')['Rating'].mean()
print("Genre average rating:",a)
```

```
Genre average rating: Genre
Action       3.260000
Adventure    2.500000
Animation    4.075000
Crime        2.872727
Drama        2.840000
Fantasy      3.583333
Musical      2.936364
Romance      2.640000
Sci-Fi       3.060000
Thriller     3.633333
Name: Rating, dtype: float64
```

20.Group reviews by title and count the number of reviews for each movie.



```python
import pandas as pd
df = pd.read_csv("movie_reviews.csv")
a=df.groupby('Title').size()
print(a)
```

```
Title
Avengers: Endgame    1
Fight Club           7
Forrest Gump         7
Frozen               4
Gladiator            4
Inception            5
Interstellar         9
Joker                8
La La Land           4
Parasite             5
Pulp Fiction         9
Sharknado            2
The Dark Knight      5
The Godfather        4
The Lion King        6
The Matrix           2
The Room             4
Titanic              1
Up                   7
Whiplash             6
dtype: int64
```