Name : Saniya Shinde .
Roll No. CS7-13
PRN : 202401080055
Movie Review Dataset Problems

1. Find the average rating of all movies.

Question: Find the average rating of all movies in the dataset.

Code:

```
import pandas as pd
import numpy as np

# Sample Dataset
data = {
   'MovieID': [1, 2, 3, 4, 5],
   'Title': ['The Matrix', 'Titanic', 'Inception', 'The Godfather', 'Joker'],
   'Genre': ['Action, Sci-Fi', 'Romance, Drama', 'Action, Sci-Fi', 'Crime, Drama', 'Crime, Drama'],
   'Rating': [4.7, 4.3, 4.8, 4.9, 4.4],
   'ReviewCount': [1250, 980, 1100, 860, 750],
   'ReleaseYear': [1999, 1997, 2010, 1972, 2019]
}

df = pd.DataFrame(data)

# Find Average Rating
average_rating = np.mean(df['Rating'])
print("Average Rating:", average_rating)
```

Output:

```
Average Rating: 4.62
```

2. Find the movie with the maximum number of reviews.

Question: Find the movie with the maximum number of reviews.

Code:

```
max_reviews_movie = df[df['ReviewCount'] == df['ReviewCount'].max()]
print(max_reviews_movie[['Title', 'ReviewCount']])
```

Output:

```
       Title  ReviewCount
0  The Matrix         1250
```

3. List all movies released after the year 2000.

Question: List all movies that were released after the year 2000.

Code:

```
movies_after_2000 = df[df['ReleaseYear'] > 2000]
print(movies_after_2000[['Title', 'ReleaseYear']])
```

Output:

```
       Title  ReleaseYear
2  Inception         2010
4      Joker         2019
```

4. Calculate the standard deviation of movie ratings.

Question: Calculate the standard deviation of the ratings for all movies.

Code:

```
std_rating = np.std(df['Rating'])
print("Standard Deviation of Ratings:", std_rating)
```

Output:

```
Standard Deviation of Ratings: 0.22173557862429248
```

5. Find the number of unique genres.

Question: Find the number of unique genres in the dataset.

Code:

```
genres = df['Genre'].str.split(', ').explode()
unique_genres = np.unique(genres)
print("Unique Genres:", unique_genres)
```

Output:

```
Unique Genres: ['Action' 'Crime' 'Drama' 'Romance' 'Sci-Fi']
```

6. Create a new column for the length of each movie title.

Question: Create a new column that stores the length of each movie title.

Code:

```
df['TitleLength'] = df['Title'].apply(len)
print(df[['Title', 'TitleLength']])
```

Output:

```
         Title  TitleLength
0   The Matrix           11
1      Titanic            7
2    Inception            9
3  The Godfather         13
4        Joker            5
```

7. Sort movies based on their ratings in descending order.

Question: Sort movies by their ratings in descending order.

Code:

```
sorted_movies = df.sort_values(by='Rating', ascending=False)
print(sorted_movies[['Title', 'Rating']])
```

Output:

```
         Title  Rating
3  The Godfather    4.9
2      Inception    4.8
0     The Matrix    4.7
4          Joker    4.4
1        Titanic    4.3
```

8. Find movies that belong to the "Drama" genre.

Question: Find all movies that belong to the "Drama" genre.

Code:

```
drama_movies = df[df['Genre'].str.contains('Drama')]
print(drama_movies[['Title', 'Genre']])
```

Output:

```
          Title           Genre
1       Titanic  Romance, Drama
3  The Godfather    Crime, Drama
4          Joker    Crime, Drama
```

9. Find how many movies were reviewed more than 1000 times.

Question: Find how many movies in the dataset have been reviewed more than 1000 times.

Code:

```
count_high_reviews = np.sum(df['ReviewCount'] > 1000)
print("Movies with >1000 reviews:", count_high_reviews)
```

Output:

```
Movies with >1000 reviews: 2
```

10. Find the earliest released movie.

Question: Find the earliest released movie from the dataset.

Code:

```
earliest_movie = df[df['ReleaseYear'] == df['ReleaseYear'].min()]
print(earliest_movie[['Title', 'ReleaseYear']])
```

Output:

```
        Title  ReleaseYear
3  The Godfather         1972
```

11. Compute the correlation between Rating and ReviewCount.

Question: Compute the correlation between movie ratings and review counts.

Code:

```
correlation = df['Rating'].corr(df['ReviewCount'])
print("Correlation between Rating and ReviewCount:", correlation)
```

Output:

```
Correlation between Rating and ReviewCount: -0.37245273526548207
```

12. Find movies that have a title length greater than 10.

Question: Find movies with titles longer than 10 characters.

Code:

```
long_title_movies = df[df['TitleLength'] > 10]
print(long_title_movies[['Title', 'TitleLength']])
```

Output:

```
        Title  TitleLength
3  The Godfather         13
```

13. Create a boolean column indicating if Rating > 4.5

Question: Create a new column that indicates if the rating is greater than 4.5.

Code:

```
df['HighlyRated'] = df['Rating'] > 4.5
print(df[['Title', 'Rating', 'HighlyRated']])
```

Output:

```
        Title  Rating  HighlyRated
0    The Matrix    4.7         True
1       Titanic    4.3        False
2     Inception    4.8         True
3  The Godfather    4.9         True
4         Joker    4.4        False
```

14. Group by Genre and find the average rating per genre.

Question: Group the movies by genre and find the average rating for each genre.

Code:

```
genre_exploded = df.copy()
genre_exploded = genre_exploded.assign(Genre=genre_exploded['Genre'].str.split(', ')).explode('Genre')
genre_avg_rating = genre_exploded.groupby('Genre')['Rating'].mean()
print(genre_avg_rating)
```

Output:

```
Genre
Action     4.75
Crime      4.65
Drama      4.533333
Romance    4.3
Sci-Fi     4.75
Name: Rating, dtype: float64
```

15. Find the median of ReviewCount.

Question: Find the median number of reviews across all movies.

Code:

```
median_review_count = np.median(df['ReviewCount'])
print("Median Review Count:", median_review_count)
```

Output:

```
Median Review Count: 980.0
```

16. Get a list of all movies starting with "T".

Question: List all movies whose titles start with the letter "T".

Code:

```
movies_starting_T = df[df['Title'].str.startswith('T')]
```

```
print(movies_starting_T[['Title']])
```

Output:

```
      Title
0  The Matrix
1     Titanic
```

17. Find the top 2 movies with the highest ratings.

Question: Find the top 2 movies with the highest ratings.

Code:

```
top2_movies = df.nlargest(2, 'Rating')
print(top2_movies[['Title', 'Rating']])
```

Output:

```
          Title  Rating
3  The Godfather     4.9
2      Inception     4.8
```

18. List movies having ratings between 4.5 and 5.

Question: List movies that have ratings between 4.5 and 5.0.

Code:

```
movies_in_range = df[(df['Rating'] >= 4.5) & (df['Rating'] <= 5.0)]
print(movies_in_range[['Title', 'Rating']])
```

Output:

```
         Title  Rating
0   The Matrix     4.7
2    Inception     4.8
```

```
3   The Godfather    4.9
```

19. Count how many movies per ReleaseYear.

Question: Count how many movies were released per year.

Code:

```
movies_per_year = df['ReleaseYear'].value_counts()
print(movies_per_year)
```

Output:

```
1999    1
1997    1
2010    1
1972    1
2019    1
Name: ReleaseYear, dtype: int64
```

20. Create a numpy array of movie ratings and find the min and max.

Question: Create a numpy array of movie ratings and find the minimum and maximum ratings.

Code:

```
ratings_array = np.array(df['Rating'])
print("Min Rating:", np.min(ratings_array))
print("Max Rating:", np.max(ratings_array))
```

Output:

```
Min Rating: 4.3
Max Rating: 4.9
```