Assignment/Task 5

Pandas - Data Analysis of IMDB movies data

As we have a basic understanding of the different data structures in Pandas, let’s explore the fun and interesting ‘IMDB-movies-dataset’ and get our hands dirty by performing practical data analysis on real data. ¶

It is an open-source dataset and you can download it from this link.

https://www.kaggle.com/PromptCloudHQ/imdb-data (https://www.kaggle.com/PromptCloudHQ/imdb-data)

We will read the data from the .csv file and perform the following basic operations on movies data

1. Load the IMDb Dataset and read

In [8]:
```
import pandas as pd

# Read data from .csv file
data = pd.read_csv('IMDB-Movie-Data.csv')
print(data)
```

```
     Rank                    Title  ... Revenue (Millions) Metascore
0       1  Guardians of the Galaxy  ...             333.13      76.0
1       2               Prometheus  ...             126.46      65.0
2       3                    Split  ...             138.12      62.0
3       4                     Sing  ...             270.32      59.0
4       5            Suicide Squad  ...             325.02      40.0
..    ...                      ...  ...                ...       ...
995   996     Secret in Their Eyes  ...                NaN      45.0
996   997          Hostel: Part II  ...              17.54      46.0
997   998   Step Up 2: The Streets  ...              58.01      50.0
998   999             Search Party  ...                NaN      22.0
999  1000               Nine Lives  ...              19.64      11.0

[1000 rows x 12 columns]
```

2. View the dataset

In [14]:
```
# Preview top 5 rows using head()
print(f"Printing Top 5 rows:\n {data.head()}")

# Preview below 5 rows using tail()
print(f"Printing below 5 rows:\n {data.tail()}")
```

```
Printing Top 5 rows:
    Rank                    Title  ... Revenue (Millions) Metascore
0     1  Guardians of the Galaxy  ...             333.13      76.0
1     2               Prometheus  ...             126.46      65.0
2     3                    Split  ...             138.12      62.0
3     4                     Sing  ...             270.32      59.0
4     5            Suicide Squad  ...             325.02      40.0

[5 rows x 12 columns]
Printing below 5 rows:
      Rank                   Title  ... Revenue (Millions) Metascore
995   996    Secret in Their Eyes  ...                NaN      45.0
996   997         Hostel: Part II  ...              17.54      46.0
997   998  Step Up 2: The Streets  ...              58.01      50.0
998   999            Search Party  ...                NaN      22.0
999  1000              Nine Lives  ...              19.64      11.0

[5 rows x 12 columns]
```

3. Understand some basic information about the dataset and Inspect the dataframe Inspect the dataframe's columns, shapes, variable types etc.

In [35]:
```
# info() is one of my favorite methods that gives all necessary information about different columns in a dataframe.
print("Printing basic information about this data:\n")
data.info()

# columns gives us the list of columns in the dataframe
print("\n\nPrinting columns of this data:\n",data.columns)

# shape can be used to get the shape of dataframe
# This function tells us that there are 1000 rows and 12 columns in the dataset
print("\n\nPrinting shapes of this data:",data.shape)

# describe() method gives the basic statistical summaries of all numerical attributes in the dataframe.
print("\n\nPrinting basic statistical summaries of this data:")
data.describe()
```

```
Printing basic information about this data:

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 12 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   Rank                1000 non-null   int64  
 1   Title               1000 non-null   object 
 2   Genre               1000 non-null   object 
 3   Description         1000 non-null   object 
 4   Director            1000 non-null   object 
 5   Actors              1000 non-null   object 
 6   Year                1000 non-null   int64  
 7   Runtime (Minutes)   1000 non-null   int64  
 8   Rating              1000 non-null   float64
 9   Votes               1000 non-null   int64  
 10  Revenue (Millions)  872 non-null    float64
 11  Metascore           936 non-null    float64
dtypes: float64(3), int64(4), object(5)
memory usage: 93.9+ KB


Printing columns of this data:
 Index(['Rank', 'Title', 'Genre', 'Description', 'Director', 'Actors', 'Year',
       'Runtime (Minutes)', 'Rating', 'Votes', 'Revenue (Millions)',
       'Metascore'],
      dtype='object')


Printing shapes of this data: (1000, 12)


Printing basic statistical summaries of this data:
```

Out[35]:

	Rank	Year	Runtime (Minutes)	Rating	Votes	Revenue (Millions)	Metascore
count	1000.000000	1000.000000	1000.000000	1000.000000	1.000000e+03	872.000000	936.000000
mean	500.500000	2012.783000	113.172000	6.723200	1.698083e+05	82.956376	58.985043
std	288.819436	3.205962	18.810908	0.945429	1.887626e+05	103.253540	17.194757
min	1.000000	2006.000000	66.000000	1.900000	6.100000e+01	0.000000	11.000000
25%	250.750000	2010.000000	100.000000	6.200000	3.630900e+04	13.270000	47.000000
50%	500.500000	2014.000000	111.000000	6.800000	1.107990e+05	47.985000	59.500000
75%	750.250000	2016.000000	123.000000	7.400000	2.399098e+05	113.715000	72.000000
max	1000.000000	2016.000000	191.000000	9.000000	1.791916e+06	936.630000	100.000000

4. Data Selection – Indexing and Slicing data

In [44]:
```
# Extract data as series
genre = data['Genre']
print(f"Extract data as series:\n{genre}")

# Extract data as dataframe
print(f"\n\nExtract data as dataframe:\n{data[['Genre']]}")

# If we want to extract multiple columns from the data, simply add the column names to the list.
some_cols = data[['Title','Genre','Actors','Director','Rating']]
print(f"\n\nExtract multiple columns from the data:\n{some_cols}")

data.iloc[10:15][['Title','Rating','Revenue (Millions)']]
```

```
Extract data as series:
0       Action,Adventure,Sci-Fi
1      Adventure,Mystery,Sci-Fi
2               Horror,Thriller
3       Animation,Comedy,Family
4      Action,Adventure,Fantasy
                 ...
995         Crime,Drama,Mystery
996                      Horror
997         Drama,Music,Romance
998            Adventure,Comedy
999       Comedy,Family,Fantasy
Name: Genre, Length: 1000, dtype: object


Extract data as dataframe:
                        Genre
0     Action,Adventure,Sci-Fi
1    Adventure,Mystery,Sci-Fi
2             Horror,Thriller
3     Animation,Comedy,Family
4    Action,Adventure,Fantasy
..                        ...
995       Crime,Drama,Mystery
996                    Horror
997       Drama,Music,Romance
998          Adventure,Comedy
999     Comedy,Family,Fantasy

[1000 rows x 1 columns]


Extract multiple columns from the data:
                       Title  ... Rating
0    Guardians of the Galaxy  ...    8.1
1                 Prometheus  ...    7.0
2                      Split  ...    7.3
3                       Sing  ...    7.2
4              Suicide Squad  ...    6.2
..                       ...  ...    ...
995     Secret in Their Eyes  ...    6.2
996          Hostel: Part II  ...    5.5
997   Step Up 2: The Streets  ...    6.2
998             Search Party  ...    5.6
999               Nine Lives  ...    5.3

[1000 rows x 5 columns]
```

Out[44]:

	Title	Rating	Revenue (Millions)
10	Fantastic Beasts and Where to Find Them	7.5	234.02
11	Hidden Figures	7.8	169.27
12	Rogue One	7.9	532.17
13	Moana	7.7	248.75
14	Colossal	6.4	2.87

5. Data Selection – Based on Conditional Filtering

In [45]:
```
data[((data['Year'] >= 2010) & (data['Year'] <= 2016))
      & (data['Rating'] < 6.0)
      & (data['Revenue (Millions)'] > data['Revenue (Millions)'].quantile(0.95))]
```

Out[45]:

	Rank	Title	Genre	Description	Director	Actors	Year	Runtime (Minutes)	Rating	Votes	Revenue (Millions)	Metascore
941	942	The Twilight Saga: Eclipse	Adventure,Drama,Fantasy	As a string of mysterious killings grips Seatt...	David Slade	Kristen Stewart, Robert Pattinson, Taylor Laut...	2010	124	4.9	192740	300.52	58.0

6. Groupby operation

In [46]:
```
data.groupby('Director')[['Rating']].mean().head()
```

Out[46]:

Director	Rating
Aamir Khan	8.5
Abdellatif Kechiche	7.8
Adam Leon	6.5
Adam McKay	7.0
Adam Shankman	6.3

7. Sorting operation

In [47]:
```
data.groupby('Director')[['Rating']].mean().sort_values(['Rating'], ascending=False).head()
```

Out[47]:

Director	Rating
Nitesh Tiwari	8.80
Christopher Nolan	8.68
Makoto Shinkai	8.60
Olivier Nakache	8.60
Florian Henckel von Donnersmarck	8.50

8. Dealing with missing values

In [48]:
```
# To check null values row-wise
data.isnull().sum()
```

Out[48]:
```
Rank                    0
Title                   0
Genre                   0
Description             0
Director                0
Actors                  0
Year                    0
Runtime (Minutes)       0
Rating                  0
Votes                   0
Revenue (Millions)    128
Metascore              64
dtype: int64
```

9. Dropping columns and null values

In [49]:
```
# Drops all rows containing missing data
data.dropna()
```

Out[49]:

	Rank	Title	Genre	Description	Director	Actors	Year	Runtime (Minutes)	Rating	Votes	Revenue (Millions)	Metasc
0	1	Guardians of the Galaxy	Action,Adventure,Sci-Fi	A group of intergalactic criminals are forced ...	James Gunn	Chris Pratt, Vin Diesel, Bradley Cooper, Zoe S...	2014	121	8.1	757074	333.13	7
1	2	Prometheus	Adventure,Mystery,Sci-Fi	Following clues to the origin of mankind, a te...	Ridley Scott	Noomi Rapace, Logan Marshall-Green, Michael Fa...	2012	124	7.0	485820	126.46	6
2	3	Split	Horror,Thriller	Three girls are kidnapped by a man with a diag...	M. Night Shyamalan	James McAvoy, Anya Taylor-Joy, Haley Lu Richar...	2016	117	7.3	157606	138.12	6
3	4	Sing	Animation,Comedy,Family	In a city of humanoid animals, a hustling thea...	Christophe Lourdelet	Matthew McConaughey,Reese Witherspoon, Seth Ma...	2016	108	7.2	60545	270.32	5
4	5	Suicide Squad	Action,Adventure,Fantasy	A secret government agency recruits some of th...	David Ayer	Will Smith, Jared Leto, Margot Robbie, Viola D...	2016	123	6.2	393727	325.02	4
...	...	...	...	...	...	...	...	...	...	...	...	
993	994	Resident Evil: Afterlife	Action,Adventure,Horror	While still out to destroy the evil Umbrella C...	Paul W.S. Anderson	Milla Jovovich, Ali Larter, Wentworth Miller,K...	2010	97	5.9	140900	60.13	3
994	995	Project X	Comedy	3 high school seniors throw a birthday party t...	Nima Nourizadeh	Thomas Mann, Oliver Cooper, Jonathan Daniel Br...	2012	88	6.7	164088	54.72	4
996	997	Hostel: Part II	Horror	Three American college students studying abroa...	Eli Roth	Lauren German, Heather Matarazzo, Bijou Philli...	2007	94	5.5	73152	17.54	4
997	998	Step Up 2: The Streets	Drama,Music,Romance	Romantic sparks occur between two dance studen...	Jon M. Chu	Robert Hoffman, Briana Evigan, Cassie Ventura,...	2008	98	6.2	70699	58.01	5
999	1000	Nine Lives	Comedy,Family,Fantasy	A stuffy businessman finds himself trapped ins...	Barry Sonnenfeld	Kevin Spacey, Jennifer Garner, Robbie Amell,Ch...	2016	87	5.3	12435	19.64	1

838 rows × 12 columns

10. Apply( ) function

In [51]:
```
# Classify movies based on ratings
def rating_group(rating):
    if rating >= 7.5:
        return 'Good'
    elif rating >= 6.0:
        return 'Average'
    else:
        return 'Bad'

# Lets apply this function on our movies data
# creating a new variable in the dataset to hold the rating category
data['Rating_category'] = data['Rating'].apply(rating_group)

data[['Title','Director','Rating','Rating_category']].head(5)
```

Out[51]:

	Title	Director	Rating	Rating_category
0	Guardians of the Galaxy	James Gunn	8.1	Good
1	Prometheus	Ridley Scott	7.0	Average
2	Split	M. Night Shyamalan	7.3	Average
3	Sing	Christophe Lourdelet	7.2	Average
4	Suicide Squad	David Ayer	6.2	Average

In [ ]: