Movie analysis

September 30, 2025

```
[5]: import pandas as pd
     import numpy as np
     import matplotlib.pyplot as plt
     import seaborn as sns
     import warnings
     warnings.filterwarnings("ignore")

     df = pd.read_csv('imdb_movie_dataset.csv')
     df.head()
```

```
[5]:    Rank                    Title                     Genre  \
     0     1  Guardians of the Galaxy   Action,Adventure,Sci-Fi
     1     2               Prometheus  Adventure,Mystery,Sci-Fi
     2     3                    Split           Horror,Thriller
     3     4                     Sing   Animation,Comedy,Family
     4     5            Suicide Squad  Action,Adventure,Fantasy

                                              Description              Director  \
     0  A group of intergalactic criminals are forced …            James Gunn
     1  Following clues to the origin of mankind, a te…          Ridley Scott
     2  Three girls are kidnapped by a man with a diag…    M. Night Shyamalan
     3  In a city of humanoid animals, a hustling thea…  Christophe Lourdelet
     4  A secret government agency recruits some of th…            David Ayer

                                               Actors  Year  Runtime (Minutes)  \
     0  Chris Pratt, Vin Diesel, Bradley Cooper, Zoe S…  2014                121
     1  Noomi Rapace, Logan Marshall-Green, Michael Fa…  2012                124
     2  James McAvoy, Anya Taylor-Joy, Haley Lu Richar…  2016                117
     3  Matthew McConaughey,Reese Witherspoon, Seth Ma…  2016                108
     4  Will Smith, Jared Leto, Margot Robbie, Viola D…  2016                123

        Rating   Votes  Revenue (Millions)  Metascore
     0     8.1  757074              333.13       76.0
     1     7.0  485820              126.46       65.0
     2     7.3  157606              138.12       62.0
     3     7.2   60545              270.32       59.0
     4     6.2  393727              325.02       40.0
```

0.1 Overview of dataset

[15]:
```python
print('Shape of dataset:',df.shape,end='\n\n')
print('Columns of dataset:\n',df.columns,end='\n\n')
print('properties of dataset:\n',df.info(),end='\n\n')
print('Properties of attributes:\n',df.describe())
```

```
Shape of dataset: (1000, 12)

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

<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
properties of dataset:
 None

Properties of attributes:
               Rank         Year  Runtime (Minutes)       Rating         Votes
\
count  1000.000000  1000.000000        1000.000000  1000.000000  1.000000e+03
mean    500.500000  2012.783000         113.172000     6.723200  1.698083e+05
std     288.819436     3.205962          18.810908     0.945429  1.887626e+05
min       1.000000  2006.000000          66.000000     1.900000  6.100000e+01
25%     250.750000  2010.000000         100.000000     6.200000  3.630900e+04
50%     500.500000  2014.000000         111.000000     6.800000  1.107990e+05
75%     750.250000  2016.000000         123.000000     7.400000  2.399098e+05
max    1000.000000  2016.000000         191.000000     9.000000  1.791916e+06
```

```
       Revenue (Millions)   Metascore
count          872.000000  936.000000
mean            82.956376   58.985043
std            103.253540   17.194757
min              0.000000   11.000000
25%             13.270000   47.000000
50%             47.985000   59.500000
75%            113.715000   72.000000
max            936.630000  100.000000
```

0.2 Cleaning of dataset

[19]:
```
print('Missing value in each column:\n',df.isnull().sum())
print('\n\nTotal missing value:',df.isnull().sum().sum())
```

```
Missing value in each column:
 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


Total missing value: 192
```

[21]:
```
# checking missing value containing row
df[df.isnull().any(axis=1)]
```

[21]:
```
     Rank                  Title                     Genre  \
7       8               Mindhorn                    Comedy
22     23         Hounds of Love       Crime,Drama,Horror
25     26        Paris pieds nus                    Comedy
26     27  Bahubali: The Beginning  Action,Adventure,Drama
27     28             Dead Awake           Horror,Thriller
..    …                      …                       …
988   989                Martyrs                    Horror
989   990                  Selma  Biography,Drama,History
992   993   Take Me Home Tonight      Comedy,Drama,Romance
995   996   Secret in Their Eyes       Crime,Drama,Mystery
```

```
998   999              Search Party         Adventure,Comedy

                                          Description        Director  \
7    A has-been actor best known for playing the ti…      Sean Foley
22   A cold-blooded predatory couple while cruising…       Ben Young
25   Fiona visits Paris for the first time to assis…  Dominique Abel
26   In ancient India, an adventurous and daring ma…  S.S. Rajamouli
27   A young woman must save herself and her friend…  Phillip Guzman
..                                                …               …
988  A young woman's quest for revenge against the …  Pascal Laugier
989  A chronicle of Martin Luther King's campaign t…    Ava DuVernay
992  Four years after graduation, an awkward high s…   Michael Dowse
995  A tight-knit team of rising investigators, alo…       Billy Ray
998  A pair of friends embark on a mission to reuni…  Scot Armstrong

                                               Actors  Year  \
7    Essie Davis, Andrea Riseborough, Julian Barrat…  2016
22   Emma Booth, Ashleigh Cummings, Stephen Curry,S…  2016
25   Fiona Gordon, Dominique Abel,Emmanuelle Riva, …  2016
26   Prabhas, Rana Daggubati, Anushka Shetty,Tamann…  2015
27   Jocelin Donahue, Jesse Bradford, Jesse Borrego…  2016
..                                                …   …
988  Morjana Alaoui, Mylène Jampanoï, Catherine Bég…  2008
989  David Oyelowo, Carmen Ejogo, Tim Roth, Lorrain…  2014
992  Topher Grace, Anna Faris, Dan Fogler, Teresa P…  2011
995  Chiwetel Ejiofor, Nicole Kidman, Julia Roberts…  2015
998  Adam Pally, T.J. Miller, Thomas Middleditch,Sh…  2014

     Runtime (Minutes)  Rating  Votes  Revenue (Millions)  Metascore
7                   89     6.4   2490                 NaN       71.0
22                 108     6.7   1115                 NaN       72.0
25                  83     6.8    222                 NaN        NaN
26                 159     8.3  76193                6.50        NaN
27                  99     4.7    523                0.01        NaN
..                 …      …     …                 …        …
988                 99     7.1  63785                 NaN       89.0
989                128     7.5  67637               52.07        NaN
992                 97     6.3  45419                6.92        NaN
995                111     6.2  27585                 NaN       45.0
998                 93     5.6   4881                 NaN       22.0

[162 rows x 12 columns]
```

```
[151]: # heari 100 rows+ containes the null values, so we cannot drop it. Hear we are␣
        ↪filling this null values which can be fill with mean values.
       df['Revenue (Millions)'] = df['Revenue (Millions)'].fillna(df['Revenue␣
        ↪(Millions)'].mean())
```

```python
df['Metascore'] = df['Metascore'].fillna(df['Metascore'].mean())

print('Now missing value in each column:\n',df.isnull().sum())
df
```

```
Now missing value in each column:
 Rank                  0
Title                 0
Genre                 0
Description           0
Director              0
Actors                0
Year                  0
Runtime (Minutes)     0
Rating                0
Votes                 0
Revenue (Millions)    0
Metascore             0
dtype: int64
```

[151]:
```
     Rank                    Title                     Genre  \
0       1  Guardians of the Galaxy   Action,Adventure,Sci-Fi
1       2               Prometheus  Adventure,Mystery,Sci-Fi
2       3                    Split           Horror,Thriller
3       4                     Sing   Animation,Comedy,Family
4       5            Suicide Squad  Action,Adventure,Fantasy
..    …                        …                         …
995   996     Secret in Their Eyes       Crime,Drama,Mystery
996   997          Hostel: Part II                    Horror
997   998   Step Up 2: The Streets       Drama,Music,Romance
998   999             Search Party          Adventure,Comedy
999  1000               Nine Lives     Comedy,Family,Fantasy

                                           Description             Director  \
0    A group of intergalactic criminals are forced …           James Gunn
1    Following clues to the origin of mankind, a te…         Ridley Scott
2    Three girls are kidnapped by a man with a diag…   M. Night Shyamalan
3    In a city of humanoid animals, a hustling thea…  Christophe Lourdelet
4    A secret government agency recruits some of th…           David Ayer
..                                                 …                    …
995  A tight-knit team of rising investigators, alo…            Billy Ray
996  Three American college students studying abroa…             Eli Roth
997  Romantic sparks occur between two dance studen…           Jon M. Chu
998  A pair of friends embark on a mission to reuni…       Scot Armstrong
999  A stuffy businessman finds himself trapped ins…     Barry Sonnenfeld

                                                Actors  Year  \
0    Chris Pratt, Vin Diesel, Bradley Cooper, Zoe S…  2014
1    Noomi Rapace, Logan Marshall-Green, Michael Fa…  2012
2    James McAvoy, Anya Taylor-Joy, Haley Lu Richar…  2016
3    Matthew McConaughey,Reese Witherspoon, Seth Ma…  2016
4    Will Smith, Jared Leto, Margot Robbie, Viola D…  2016
..                                                …    …
995  Chiwetel Ejiofor, Nicole Kidman, Julia Roberts…  2015
996  Lauren German, Heather Matarazzo, Bijou Philli…  2007
997  Robert Hoffman, Briana Evigan, Cassie Ventura,…  2008
998  Adam Pally, T.J. Miller, Thomas Middleditch,Sh…  2014
999  Kevin Spacey, Jennifer Garner, Robbie Amell,Ch…  2016

     Runtime (Minutes)  Rating   Votes  Revenue (Millions)  Metascore
0                  121     8.1  757074          333.130000       76.0
1                  124     7.0  485820          126.460000       65.0
2                  117     7.3  157606          138.120000       62.0
3                  108     7.2   60545          270.320000       59.0
4                  123     6.2  393727          325.020000       40.0
..                   …     …     …                   …        …
995                111     6.2   27585           82.956376       45.0
996                 94     5.5   73152           17.540000       46.0
997                 98     6.2   70699           58.010000       50.0
998                 93     5.6    4881           82.956376       22.0
999                 87     5.3   12435           19.640000       11.0

[1000 rows x 12 columns]
```

[152]:
```python
# now check duplicate values
dupli = df.duplicated().sum()
print('Total duplicat values:',dupli)
if(dupli > 0):
    df = df.drop_duplicate()
    print('Dropped duplicate')
else:
    print('No duplicates')
```

```
Total duplicat values: 0
No duplicates
```

0.3 EDA

[134]:
```python
df.describe()
```

[134]:
```
              Rank         Year  Runtime (Minutes)       Rating         Votes  \
count  1000.000000  1000.000000        1000.000000  1000.000000  1.000000e+03
mean    500.500000  2012.783000         113.172000     6.723200  1.698083e+05
std     288.819436     3.205962          18.810908     0.945429  1.887626e+05
min       1.000000  2006.000000          66.000000     1.900000  6.100000e+01
```

```
25%      250.750000  2010.000000          100.000000     6.200000  3.630900e+04
50%      500.500000  2014.000000          111.000000     6.800000  1.107990e+05
75%      750.250000  2016.000000          123.000000     7.400000  2.399098e+05
max     1000.000000  2016.000000          191.000000     9.000000  1.791916e+06

       Revenue (Millions)    Metascore
count         1000.000000  1000.000000
mean            82.956376    58.985043
std             96.412043    16.634858
min              0.000000    11.000000
25%             17.442500    47.750000
50%             60.375000    58.985043
75%             99.177500    71.000000
max            936.630000   100.000000
```

[135]:

```
# listing unique year
print("Year which are listed in the data set:\n",df['Year'].unique())
```

```
Year which are listed in the data set:
 [2014 2012 2016 2015 2007 2011 2008 2006 2009 2010 2013]
```

[136]:

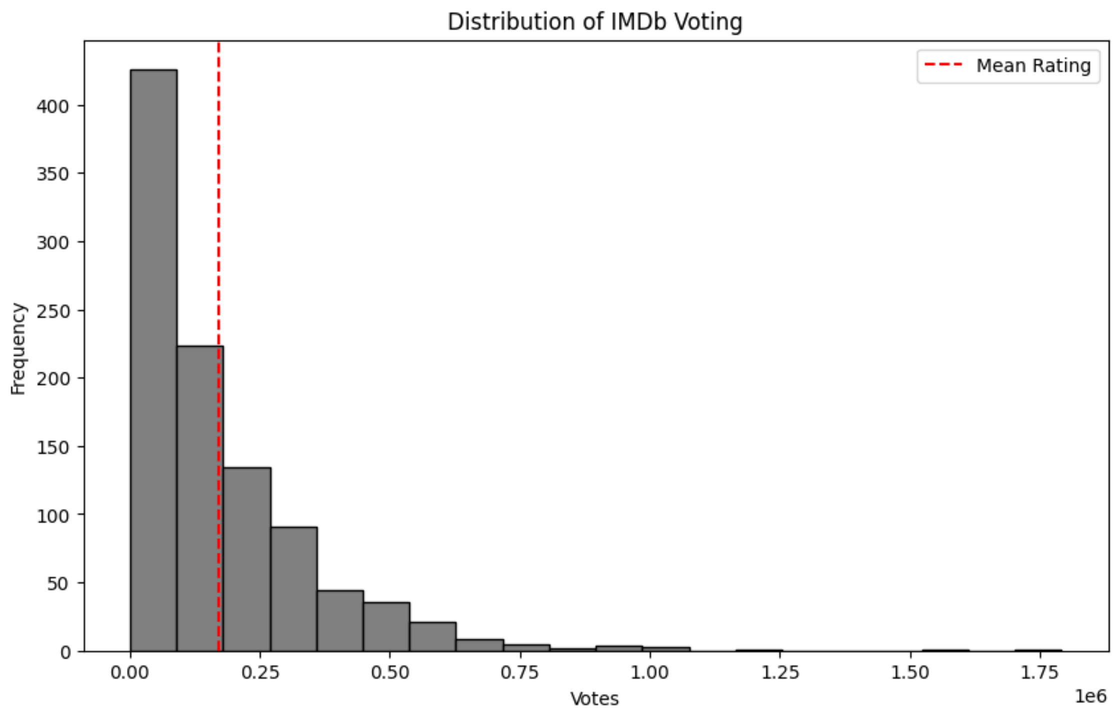
```
# average votes on the movies
avg_vote = df['Votes'].mean()
print('Average votes: ',avg_vote)
print('Number of movies has greater votes than avg votes:',(df['Votes'] >␣
 ↪avg_vote).sum())
```

```
Average votes:  169808.255
Number of movies has greater votes than avg votes: 367
```

0.3.1 Votes Distribution

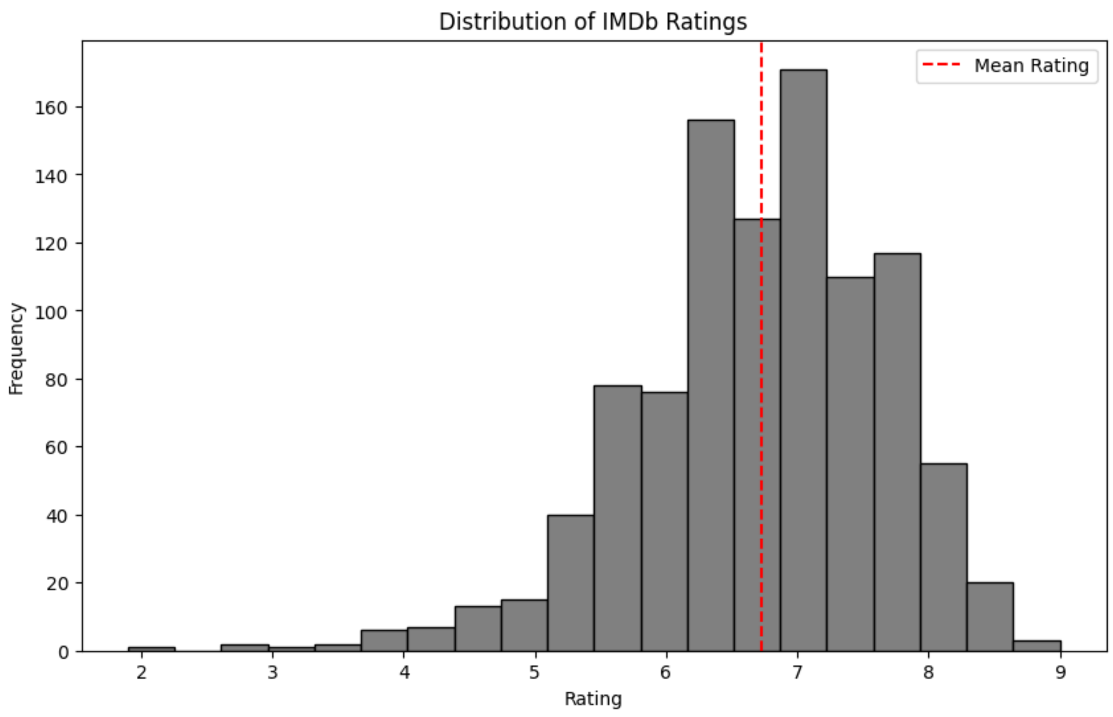
[153]:

```
plt.figure(figsize=(10, 6))
plt.hist(df['Votes'], bins=20, color='grey',edgecolor='black')
plt.title('Distribution of IMDb Voting')
plt.xlabel('Votes')
plt.ylabel('Frequency')
plt.axvline(df['Votes'].mean(), color='red', linestyle='--', label='Mean␣
 ↪Rating')
plt.legend()
plt.show()
```



0.3.2 Rating distribution

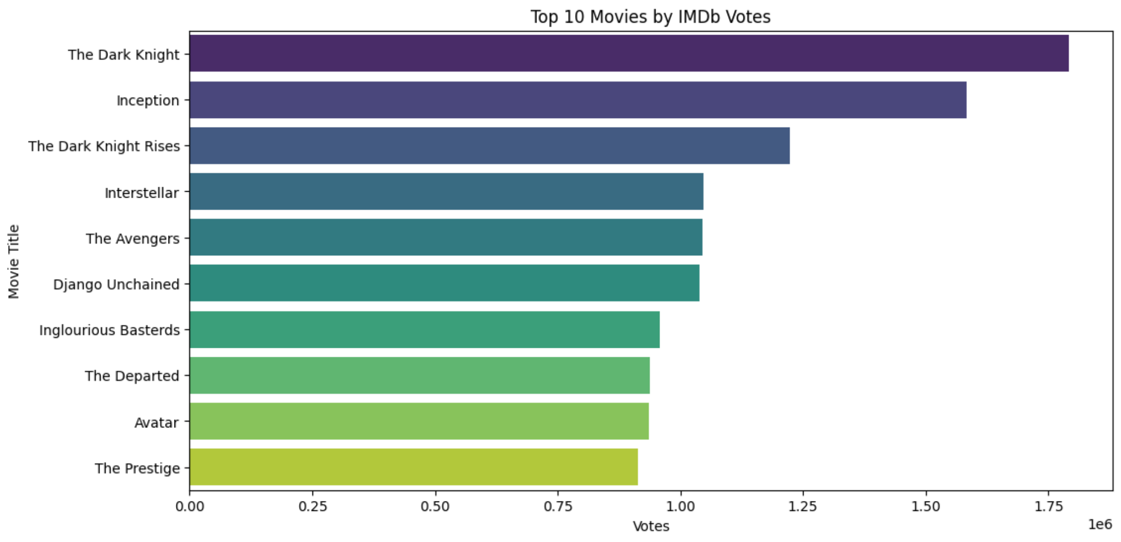
[154]:
```
plt.figure(figsize=(10, 6))
plt.hist(df['Rating'], bins=20, color='grey',edgecolor='black')
plt.title('Distribution of IMDb Ratings')
plt.xlabel('Rating')
plt.ylabel('Frequency')
plt.axvline(df['Rating'].mean(), color='red', linestyle='--', label='Mean␣
  ↪Rating')
plt.legend()
plt.show()
```



0.3.3 top voted movies

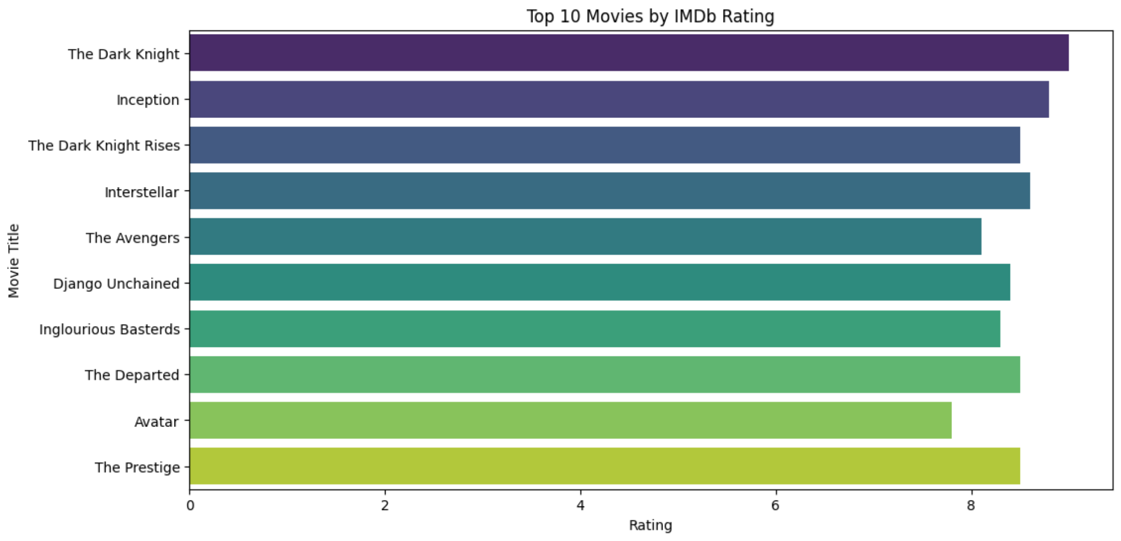
```
[155]: top10_voted = df.loc[:,['Title','Votes','Rating']].
        ↪sort_values(by='Votes',ascending=False)[0:10]
       print(top10_voted)

       plt.figure(figsize=(12, 6))
       sns.barplot(x='Votes', y='Title', data=top10_voted, palette='viridis')
       plt.title('Top 10 Movies by IMDb Votes')
       plt.xlabel('Votes')
       plt.ylabel('Movie Title')
       plt.show()
```

```
                       Title    Votes  Rating
54           The Dark Knight  1791916     9.0
80                 Inception  1583625     8.8
124    The Dark Knight Rises  1222645     8.5
36              Interstellar  1047747     8.6
76              The Avengers  1045588     8.1
144         Django Unchained  1039115     8.4
77      Inglourious Basterds   959065     8.3
99              The Departed   937414     8.5
87                    Avatar   935408     7.8
64              The Prestige   913152     8.5
```



0.3.4 top rated movies

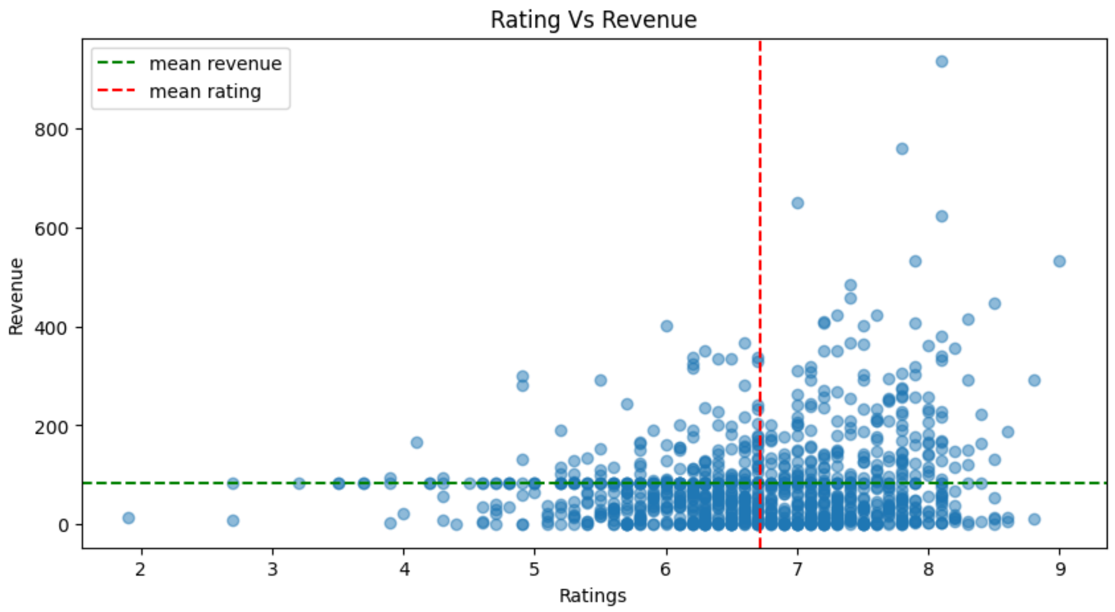
```
[156]: plt.figure(figsize=(12, 6))
sns.barplot(x='Rating', y='Title', data=top10_voted, palette='viridis')
plt.title('Top 10 Movies by IMDb Rating')
plt.xlabel('Rating')
plt.ylabel('Movie Title')
plt.show()
```



0.3.5 Rating Vs Revenue Overview

[157]:
```
print('Mean Revenue:',df['Revenue (Millions)'].mean())
print('Mean Rating:',df['Rating'].mean())
plt.figure(figsize=(10,5))

plt.scatter(df['Rating'],df['Revenue (Millions)'],alpha=0.5)
plt.title('Rating Vs Revenue')
plt.xlabel('Ratings')
plt.ylabel('Revenue')
plt.axhline(df['Revenue (Millions)'].mean(), color='green', linestyle='--',␣
 ↪label='mean revenue')
plt.axvline(df['Rating'].mean(), color='red',linestyle='--',label='mean rating')
plt.legend()
plt.show()
```

```
Mean Revenue: 82.95637614678898
Mean Rating: 6.723199999999999
```



0.3.6 Extracting main genre

[158]:
```
# unique genre
print(df['Genre'].unique())
```

```
['Action,Adventure,Sci-Fi' 'Adventure,Mystery,Sci-Fi' 'Horror,Thriller'
 'Animation,Comedy,Family' 'Action,Adventure,Fantasy' 'Comedy,Drama,Music'
 'Comedy' 'Action,Adventure,Biography' 'Adventure,Drama,Romance'
 'Adventure,Family,Fantasy' 'Biography,Drama,History'
```

```
'Animation,Adventure,Comedy' 'Action,Comedy,Drama' 'Action,Thriller'
'Biography,Drama' 'Drama,Mystery,Sci-Fi' 'Adventure,Drama,Thriller'
'Drama' 'Crime,Drama,Horror' 'Action,Adventure,Drama' 'Drama,Thriller'
'Action,Adventure,Comedy' 'Action,Horror,Sci-Fi' 'Adventure,Drama,Sci-Fi'
'Action,Adventure,Western' 'Comedy,Drama' 'Horror'
'Adventure,Drama,Fantasy' 'Action,Crime,Thriller' 'Action,Crime,Drama'
'Adventure,Drama,History' 'Crime,Horror,Thriller' 'Drama,Romance'
'Comedy,Drama,Romance' 'Horror,Mystery,Thriller' 'Crime,Drama,Mystery'
'Drama,Romance,Thriller' 'Drama,History,Thriller' 'Action,Drama,Thriller'
'Drama,History' 'Action,Drama,Romance' 'Drama,Fantasy' 'Action,Sci-Fi'
'Adventure,Drama,War' 'Action,Comedy,Fantasy' 'Biography,Comedy,Crime'
'Crime,Drama' 'Comedy,Crime,Drama' 'Action,Comedy,Crime'
'Animation,Drama,Fantasy' 'Horror,Mystery,Sci-Fi'
'Drama,Mystery,Thriller' 'Crime,Drama,Thriller' 'Biography,Crime,Drama'
'Crime,Mystery,Thriller' 'Action,Horror,Thriller' 'Romance,Sci-Fi'
'Action,Fantasy,War' 'Action,Biography,Drama' 'Drama,Horror,Mystery'
'Adventure,Drama,Family' 'Adventure,Comedy,Romance' 'Action'
'Adventure,Crime,Mystery' 'Comedy,Family,Musical'
'Adventure,Comedy,Drama' 'Drama,Horror,Thriller' 'Drama,Music'
'Mystery,Thriller' 'Mystery,Thriller,Western' 'Comedy,Family'
'Biography,Comedy,Drama' 'Drama,Western' 'Drama,Mystery,Romance'
'Action,Drama,Mystery' 'Action,Adventure,Crime'
'Adventure,Sci-Fi,Thriller' 'Action,Comedy,Mystery' 'Thriller,War'
'Action,Adventure,Thriller' 'Drama,Fantasy,Romance'
'Action,Drama,History' 'Animation,Adventure,Family' 'Adventure,Horror'
'Drama,Romance,Sci-Fi' 'Action,Adventure,Family' 'Action,Comedy'
'Comedy,Romance' 'Horror,Mystery' 'Drama,Family,Fantasy' 'Sci-Fi'
'Drama,War' 'Drama,Fantasy,Horror' 'Crime,Drama,History'
'Horror,Sci-Fi,Thriller' 'Action,Drama,Sport' 'Adventure,Biography,Drama'
'Biography,Drama,Thriller' 'Action,Adventure,Mystery' 'Drama,Horror'
'Comedy,Crime' 'Drama,Fantasy,War' 'Action,Adventure,Romance'
'Action,Drama,War' 'Drama,Musical,Romance' 'Drama,Sci-Fi,Thriller'
'Action,Drama,Sci-Fi' 'Drama,Sci-Fi' 'Adventure,Fantasy' 'Thriller'
'Biography,Drama,Romance' 'Action,Adventure' 'Action,Fantasy'
'Action,Drama,Horror' 'Comedy,Music,Romance' 'Biography,Drama,Sport'
'Action,Horror' 'Comedy,Horror,Thriller' 'Crime,Drama,Music'
'Action,Sci-Fi,Thriller' 'Drama,Horror,Sci-Fi' 'Drama,Sport'
'Comedy,Horror' 'Comedy,Fantasy,Romance' 'Comedy,Fantasy'
'Comedy,Drama,Fantasy' 'Adventure,Comedy,Horror' 'Comedy,Mystery'
'Action,Mystery,Sci-Fi' 'Action,Crime,Fantasy' 'Comedy,Fantasy,Horror'
'Animation,Action,Adventure' 'Action,Comedy,Family' 'Comedy,Sci-Fi'
'Action,Biography,Crime' 'Adventure,Comedy' 'Comedy,Music'
'Comedy,Drama,Horror' 'Action,Horror,Romance' 'Action,Drama,Fantasy'
'Action,Mystery,Thriller' 'Action,Adventure,Horror'
'Animation,Family,Fantasy' 'Adventure,Horror,Mystery'
'Action,Horror,Mystery' 'Adventure,Comedy,Family' 'Action,Crime,Mystery'
'Comedy,Drama,Family' 'Action,Crime,Sport' 'Mystery,Sci-Fi,Thriller'
'Sci-Fi,Thriller' 'Adventure,Drama,Horror' 'Biography,History,Thriller'
```

```
'Adventure,Comedy,Sci-Fi' 'Fantasy,Horror' 'Action,Fantasy,Thriller'
'Comedy,Romance,Sport' 'Animation,Action,Comedy' 'Drama,Fantasy,Thriller'
'Action,Comedy,Romance' 'Action,Fantasy,Horror' 'Mystery,Romance,Sci-Fi'
'Comedy,Drama,Thriller' 'Comedy,Western' 'Drama,History,War'
'Fantasy,Horror,Thriller' 'Drama,Horror,Musical' 'Drama,Family'
'Romance,Sci-Fi,Thriller' 'Animation,Fantasy' 'Drama,Mystery,War'
'Action,Drama,Family' 'Adventure,Drama,Western' 'Drama,Music,Romance'
'Comedy,Romance,Western' 'Adventure,Drama' 'Drama,Thriller,War'
'Drama,Fantasy,Mystery' 'Comedy,Crime,Thriller' 'Animation,Comedy,Drama'
'Action,Comedy,Sci-Fi' 'Drama,Romance,War' 'Adventure,Fantasy,Mystery'
'Mystery,Romance,Thriller' 'Biography,Drama,Mystery'
'Animation,Drama,Romance' 'Comedy,Horror,Romance' 'Action,Thriller,War'
'Action,Comedy,Horror' 'Action,Crime,Sci-Fi' 'Crime,Thriller'
'Comedy,Horror,Sci-Fi' 'Crime,Drama,Fantasy' 'Drama,Fantasy,Music'
'Action,Comedy,Sport' 'Fantasy,Mystery,Thriller' 'Adventure'
'Adventure,Biography' 'Adventure,Biography,Crime' 'Comedy,Drama,Musical'
'Comedy,Family,Romance' 'Biography,Drama,Family' 'Drama,Fantasy,Musical'
'Adventure,Family' 'Adventure,Comedy,Fantasy' 'Drama,Family,Music'
'Comedy,Family,Fantasy']
```

[174]:
```python
genre = df['Genre']

for i in range(1000):
    g = genre.loc[i].split(',')
    df.loc[i,['Main Genre']] = g[0]
# here we have split all the genre of each movie into different column.
df
```

[174]:
```
     Rank                    Title                     Genre  \
0       1  Guardians of the Galaxy   Action,Adventure,Sci-Fi
1       2               Prometheus  Adventure,Mystery,Sci-Fi
2       3                    Split           Horror,Thriller
3       4                     Sing   Animation,Comedy,Family
4       5            Suicide Squad  Action,Adventure,Fantasy
..    …                        …                         …
995   996     Secret in Their Eyes       Crime,Drama,Mystery
996   997          Hostel: Part II                    Horror
997   998   Step Up 2: The Streets       Drama,Music,Romance
998   999             Search Party          Adventure,Comedy
999  1000               Nine Lives     Comedy,Family,Fantasy

                                           Description              Director  \
0    A group of intergalactic criminals are forced …            James Gunn
1    Following clues to the origin of mankind, a te…          Ridley Scott
2    Three girls are kidnapped by a man with a diag…    M. Night Shyamalan
3    In a city of humanoid animals, a hustling thea…  Christophe Lourdelet
4    A secret government agency recruits some of th…            David Ayer
```

```
..                                                 …                     …
995  A tight-knit team of rising investigators, alo…             Billy Ray
996  Three American college students studying abroa…              Eli Roth
997  Romantic sparks occur between two dance studen…           Jon M. Chu
998  A pair of friends embark on a mission to reuni…        Scot Armstrong
999  A stuffy businessman finds himself trapped ins…     Barry Sonnenfeld

                                                Actors  Year  \
0    Chris Pratt, Vin Diesel, Bradley Cooper, Zoe S…  2014
1    Noomi Rapace, Logan Marshall-Green, Michael Fa…  2012
2    James McAvoy, Anya Taylor-Joy, Haley Lu Richar…  2016
3    Matthew McConaughey,Reese Witherspoon, Seth Ma…  2016
4    Will Smith, Jared Leto, Margot Robbie, Viola D…  2016
..                                                 …   …
995  Chiwetel Ejiofor, Nicole Kidman, Julia Roberts…  2015
996  Lauren German, Heather Matarazzo, Bijou Philli…  2007
997  Robert Hoffman, Briana Evigan, Cassie Ventura,…  2008
998  Adam Pally, T.J. Miller, Thomas Middleditch,Sh…  2014
999  Kevin Spacey, Jennifer Garner, Robbie Amell,Ch…  2016

     Runtime (Minutes)  Rating   Votes  Revenue (Millions)  Metascore  \
0                  121     8.1  757074          333.130000       76.0
1                  124     7.0  485820          126.460000       65.0
2                  117     7.3  157606          138.120000       62.0
3                  108     7.2   60545          270.320000       59.0
4                  123     6.2  393727          325.020000       40.0
..                   …       …       …                   …          …
995                111     6.2   27585           82.956376       45.0
996                 94     5.5   73152           17.540000       46.0
997                 98     6.2   70699           58.010000       50.0
998                 93     5.6    4881           82.956376       22.0
999                 87     5.3   12435           19.640000       11.0

    Main Genre
0       Action
1    Adventure
2       Horror
3    Animation
4       Action
..         …
995      Crime
996     Horror
997      Drama
998  Adventure
999     Comedy

[1000 rows x 13 columns]
```

[178]:
```
print(df['Main Genre'].nunique())
df['Main Genre'].unique()
```

```
13
```

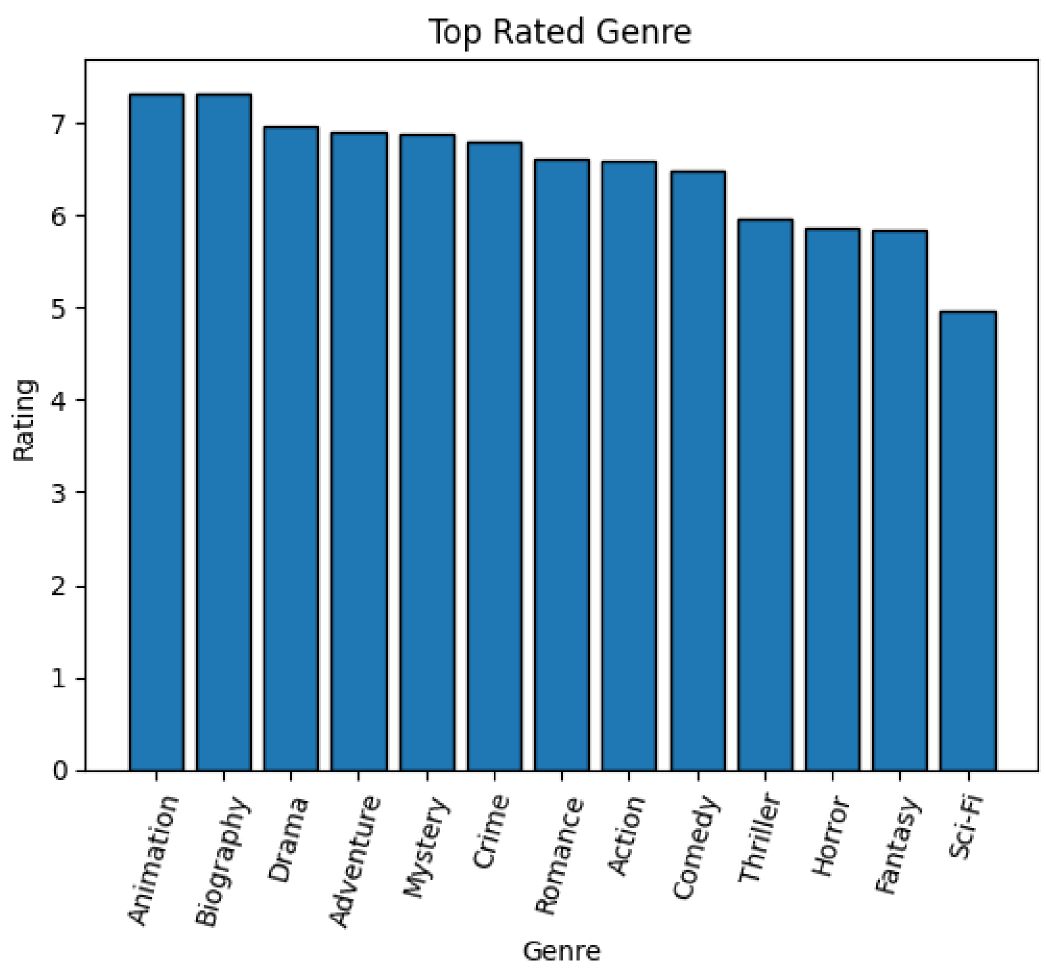
[178]:
```
array(['Action', 'Adventure', 'Horror', 'Animation', 'Comedy',
       'Biography', 'Drama', 'Crime', 'Romance', 'Mystery', 'Thriller',
       'Sci-Fi', 'Fantasy'], dtype=object)
```

0.3.7 Top rated genre

[182]:
```
top_genre = df.groupby('Main Genre')['Rating'].mean().
 ↪sort_values(ascending=False)
print(top_genre)
```

```
Main Genre
Animation    7.324490
Biography    7.318750
Drama        6.954872
Adventure    6.908000
Mystery      6.876923
Crime        6.807042
Romance      6.600000
Action       6.592491
Comedy       6.493143
Thriller     5.960000
Horror       5.867391
Fantasy      5.850000
Sci-Fi       4.966667
Name: Rating, dtype: float64
```

[184]:
```
plt.bar(top_genre.index, top_genre.values, edgecolor='black')
plt.title('Top Rated Genre')
plt.xlabel('Genre')
plt.ylabel('Rating')
plt.xticks(rotation=75)
plt.show()
```



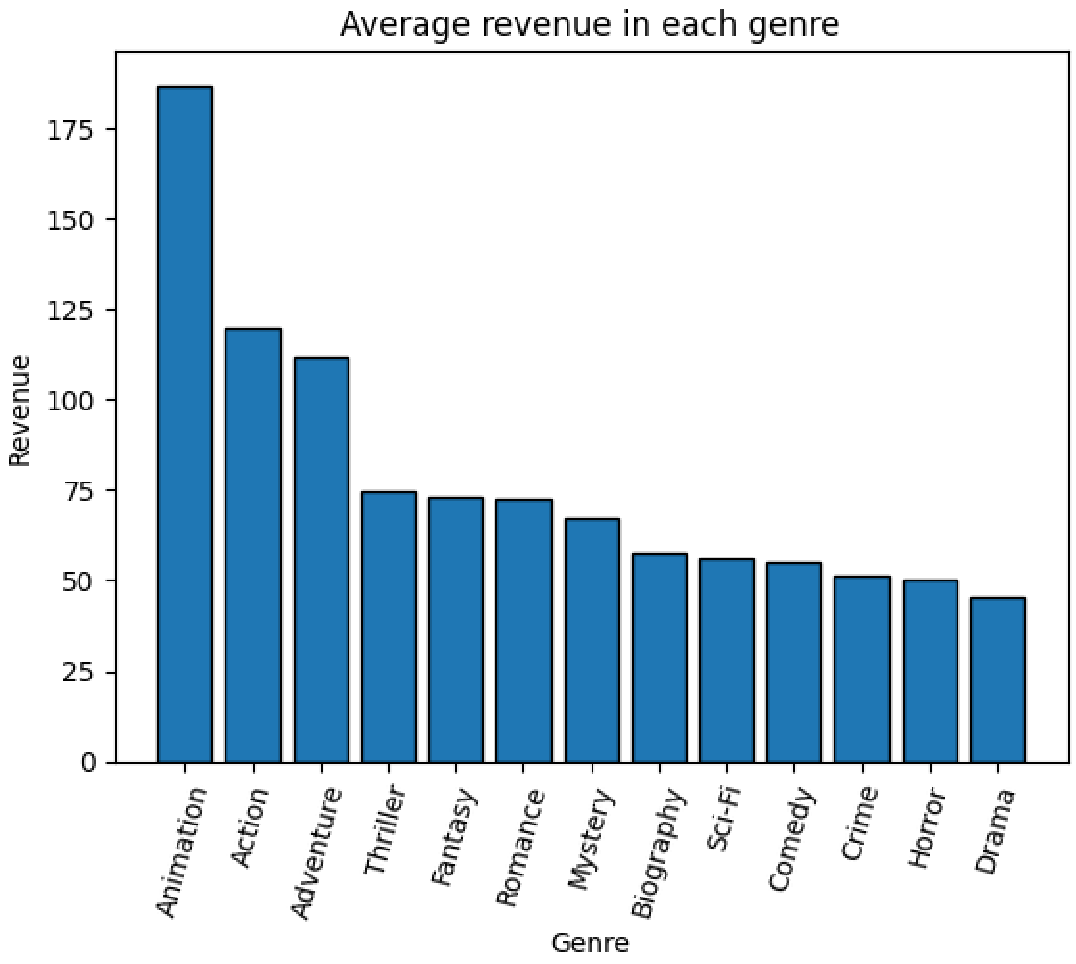
0.3.8 Average revenue in each genre

[186]:
```
revenue_genre = df.groupby('Main Genre')['Revenue (Millions)'].mean().
  ↪sort_values(ascending=False)
print(revenue_genre)
```

```
Main Genre
Animation    186.804342
Action       119.822793
Adventure    111.827007
Thriller      74.692739
Fantasy       73.033188
Romance       72.703188
Mystery       67.237135
Biography     57.642117
Sci-Fi        56.075459
Comedy        54.988578
```

```
Crime          51.078991
Horror         50.231742
Drama          45.290865
Name: Revenue (Millions), dtype: float64
```

[187]:
```
plt.bar(revenue_genre.index, revenue_genre.values, edgecolor='black')
plt.title('Average revenue in each genre')
plt.xlabel('Genre')
plt.ylabel('Revenue')
plt.xticks(rotation=75)
plt.show()
```


Average revenue in each genre
Revenue
Animation
Action
Adventure
Thriller
Fantasy
Romance
Mystery
Biography
Sci-Fi
Comedy
Crime
Horror
Drama
Genre