```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import seaborn as sns
import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
import warnings
warnings.filterwarnings('ignore')
data = pd.read_csv('/IMDB-Movie-Data.csv')
```

1. Display Top 10 Rows of The Dataset

```
data.head(3)
```

index	Rank	Title	Genre	Description	Director	Actors	Year	Runtime (Minutes)
0	1	Guardians of the Galaxy	Action,Adventure,Sci-Fi	A group of intergalactic criminals are forced to work together to stop a fanatical warrior from taking control of the universe.	James Gunn	Chris Pratt, Vin Diesel, Bradley Cooper, Zoe Saldana	2014	121
1	2	Prometheus	Adventure,Mystery,Sci-Fi	Following clues to the origin of mankind, a team finds a structure on a distant moon, but they soon realize they are not alone.	Ridley Scott	Noomi Rapace, Logan Marshall-Green, Michael Fassbender, Charlize Theron	2012	124
2	3	Split	Horror,Thriller	Three girls are kidnapped by a man with a diagnosed 23 distinct personalities. They must try to escape before the apparent emergence of a frightful new 24th.	M. Night Shyamalan	James McAvoy, Anya Taylor-Joy, Haley Lu Richardson, Jessica Sula	2016	117

2. Check Last 10 Rows of The Dataset

```
data.tail(10)
```

3. Find Shape of Our Dataset (Number of Rows And Number of Columns)

```
data.shape
```

```
(1000, 12)
```

```
print('Number of Rows',data.shape[0])
print('Number of Columns',data.shape[1])
```

```
Number of Rows 1000
Number of Columns 12
```

4. Getting Information About Our Dataset Like Total Number Rows, Total Number of Columns, Datatypes of Each Column And Memory Requirement

data.info()

```
<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
```

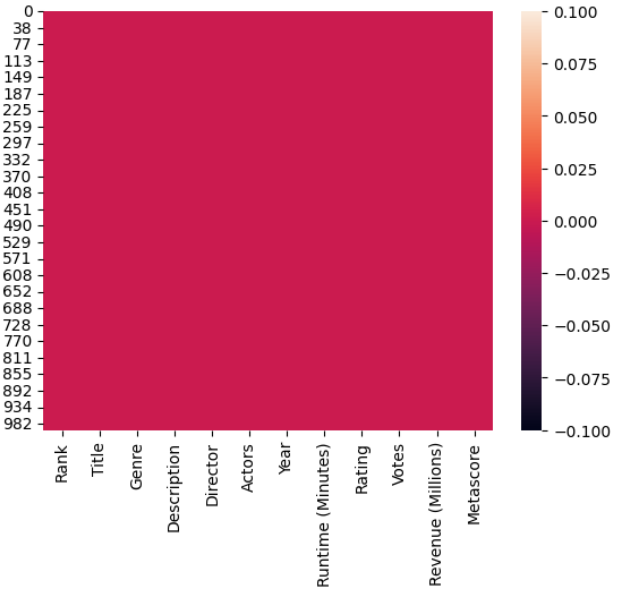
5. Check Null Values In The Dataset

data.isnull().sum()

	0
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

6. Drop All The Missing Values

```
import matplotlib.pyplot as plt
import seaborn as sns
sns.heatmap(data.isnull())
plt.show()
```



7. Check For Duplicate Data

```
dup_data=data.duplicated().any()
print("Are there any duplicated values in data?",dup_data)
```

O/P:

Are there any duplicated values in data? False

8. Get Overall Statistics About The DataFrame

```
data.describe()
```

1 to 8 of 8 entriesFilter

index	Rank	Year	Runtime (Minutes)	Rating	Votes	Revenue (Millions)	Metascore
count	838.0	838.0	838.0	838.0	838.0	838.0	838.0
mean	485.2470167064439	2012.5071599045345	114.63842482100239	6.814319809069212	193230.25178997614	84.5645584725537	59.575178997613364
std	286.5720646344357	3.172359915266028	18.470922051554556	0.8777538418027501	193099.00510393953	104.5202265333547	16.95241649434648
min	1.0	2006.0	66.0	1.9	178.0	0.0	11.0
25%	238.25	2010.0	101.0	6.3	61276.5	13.967500000000001	47.0
50%	475.5	2013.0	112.0	6.9	136879.5	48.150000000000006	60.0
75%	729.75	2015.0	124.0	7.5	271083.0	116.80000000000001	72.0
max	1000.0	2016.0	187.0	9.0	1791916.0	936.63	100.0

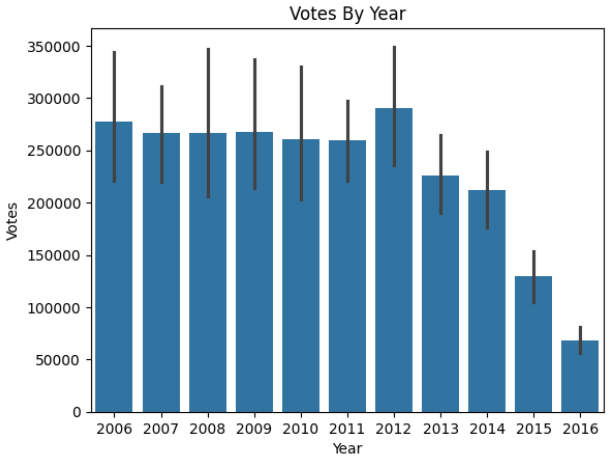
9. Display Title of The Movie Having Runtime >= 180 Minutes

data[data['Runtime (Minutes)']>=180]['Title']

	Title
82	The Wolf of Wall Street
88	The Hateful Eight
311	La vie d'Adèle

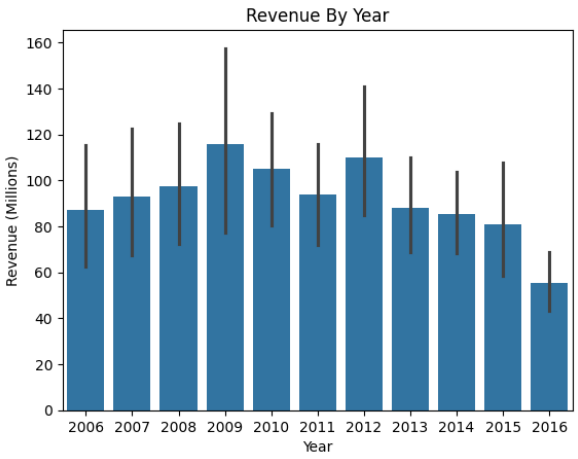
10. In Which Year There Was The Highest Voting?

```
sns.barplot(x='Year',y='Votes',data=data)
plt.title("Votes By Year")
plt.show()
```



11. In Which Year There Was The Highest Revenue?

```
sns.barplot(x='Year',y='Revenue (Millions)',data=data)
plt.title("Revenue By Year")
plt.show()
```



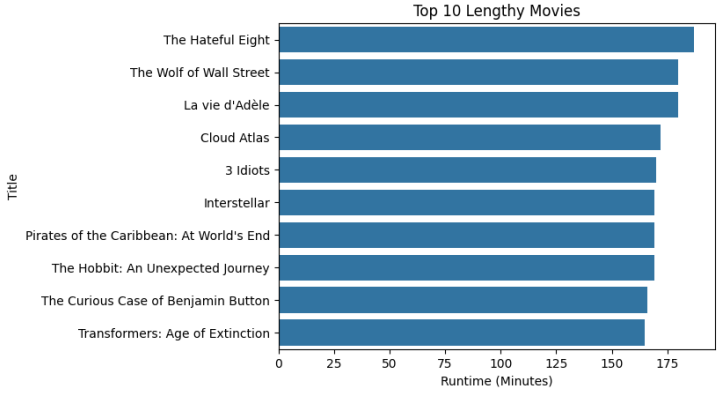
12. Find The Average Rating For Each Director

Director	Rating
Christopher Nolan	8.68
Olivier Nakache	8.60
Makoto Shinkai	8.60
Florian Henckel von Donnersmarck	8.50
Aamir Khan	8.50
...	...
Sam Taylor-Johnson	4.10
Joey Curtis	4.00
George Nolfi	3.90
James Wong	2.70
Jason Friedberg	1.90

524 rows × 1 columns

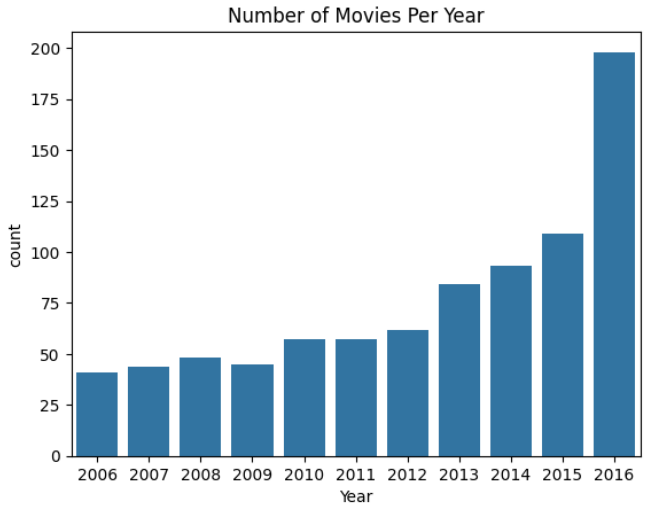
13. Display Top 10 Lengthy Movies Title

```
le =data.nlargest(10,'Runtime (Minutes)')[['Title','Runtime (Minutes)']]. \
set_index('Title')
# Plotting the bar chart
sns.barplot(x='Runtime (Minutes)', y=le.index, data=le.reset_index())
plt.title('Top 10 Lengthy Movies')
plt.show()
```



14. Display Number of Movies Per Year

```
sns.countplot(x='Year',data=data)
plt.title("Number of Movies Per Year")
```



15. Find Most Popular Movie Title (Higest Revenue)

```
data.columns
Index(['Rank', 'Title', 'Genre', 'Description', 'Director', 'Actors', 'Year',
       'Runtime (Minutes)', 'Rating', 'Votes', 'Revenue (Millions)',
       'Metascore', 'rating_cat', 'temp'],
      dtype='object')
data[data['Revenue (Millions)'].max() == data['Revenue (Millions)']]['Title']
```

Title

50 Star Wars: Episode VII - The Force Awakens

dtype: object

16. Display Top 10 Highest Rated Movie Titles And its Directors

```
top_10=data.nlargest(11,'Rating')[['Title','Rating','Director','Revenue (Millions)']].set_index('Title')
top_10
```

Title	Rating	Director
The Dark Knight	9.0	Christopher Nolan
Inception	8.8	Christopher Nolan
Interstellar	8.6	Christopher Nolan
Kimi no na wa	8.6	Makoto Shinkai
The Intouchables	8.6	Olivier Nakache
The Prestige	8.5	Christopher Nolan
The Departed	8.5	Martin Scorsese
The Dark Knight Rises	8.5	Christopher Nolan
Whiplash	8.5	Damien Chazelle
The Lives of Others	8.5	Florian Henckel von Donnersmarck
Taare Zameen Par	8.5	Aamir Khan

```
sns.barplot(top_10['Rating'],top_10.index)
plt.title("Display Top 10 Highest Rated Movie Titles")
```

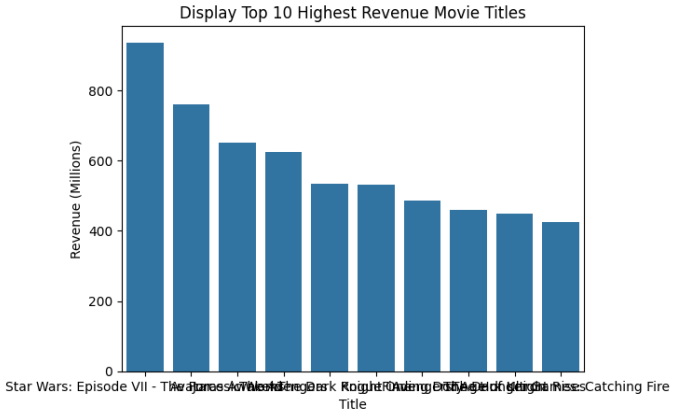
17. Display Top 10 Highest Revenue Movie Titles

```
data.columns
Index(['Rank', 'Title', 'Genre', 'Description', 'Director', 'Actors', 'Year',
       'Runtime (Minutes)', 'Rating', 'Votes', 'Revenue (Millions)',
       'Metascore'],
      dtype='object')
data.sort_values(by='Revenue (Millions)',ascending=False).head(10)
```

Rank	Title	Genre	Description	Director	Actors	Year	Runtime (Minutes)	Rating	Votes	Revenue (Millions)	Metascore	rating_cat	te
50	51	Star Wars: Episode VII - The Force Awakens	Action,Adventure,Fantasy	Three decades after the defeat of the Galactic...	J.J. Abrams	Daisy Ridley, John Boyega, Oscar Isaac, Domhna...	2015	136	8.1	661608	936.63	81.0	Ex
87	88	Avatar	Action,Adventure,Fantasy	A paraplegic marine dispatched to the moon Pan...	James Cameron	Sam Worthington, Zoe Saldana, Sigourney Weaver...	2009	162	7.8	935408	760.51	83.0	Ex
85	86	Jurassic World	Action,Adventure,Sci-Fi	A new theme park, built on the original site o...	Colin Trevorrow	Chris Pratt, Bryce Dallas Howard, Ty Simpkins,...	2015	124	7.0	455169	652.18	59.0	Ex
76	77	The Avengers	Action,Sci-Fi	Earth's mightiest heroes must come together an...	Joss Whedon	Robert Downey Jr., Chris Evans, Scarlett Johan...	2012	143	8.1	1045588	623.28	69.0	Ex
54	55	The Dark Knight	Action,Crime,Drama	When the menace known as the Joker wreaks havo...	Christopher Nolan	Christian Bale, Heath Ledger, Aaron Eckhart,Mi...	2008	152	9.0	1791916	533.32	82.0	Ex

```
top_10 = data.nlargest(10,'Revenue (Millions)')[['Title','Director','Revenue
(Millions)']].set_index('Title')
```

```
sns.barplot(top_10['Revenue (Millions)'])
plt.title("Display Top 10 Highest Revenue Movie Titles")
plt.show()
```



18. Find Average Rating of Movies Year-wise

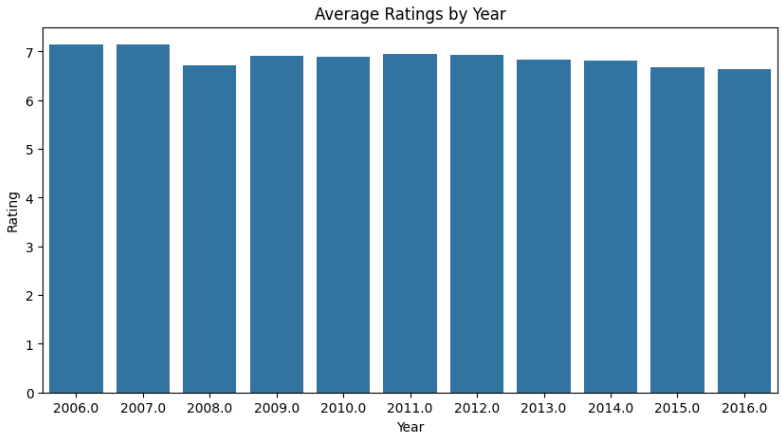
```
data.columns
Index(['Rank', 'Title', 'Genre', 'Description', 'Director', 'Actors', 'Year',
       'Runtime (Minutes)', 'Rating', 'Votes', 'Revenue (Millions)',
       'Metascore'],
      dtype='object')
```

```
data1=data.groupby('Year')[['Year','Rating']].mean().\
sort_values(by='Rating',ascending=False).set_index('Year')
data1
```

1 to 11 of 11 entriesFilter

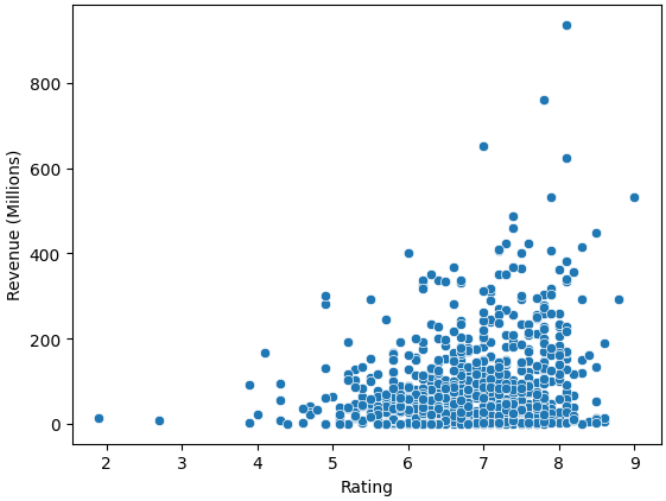
Year	Rating
2006.0	7.14390243902439
2007.0	7.140909090909091
2011.0	6.9456140350877185
2012.0	6.933870967741935
2009.0	6.911111111111111
2010.0	6.894736842105263
2013.0	6.832142857142857
2014.0	6.82258064516129
2008.0	6.708333333333333
2015.0	6.674311926605505
2016.0	6.644444444444444

```
plt.figure(figsize=(10,5))
sns.barplot(x=data1.index,y=data1['Rating'])
plt.title('Average Ratings by Year')
plt.show()
```



19. Does Rating Affect The Revenue?

sns.scatterplot(x='Rating',y='Revenue (Millions)',data=data)



Answer: Yes

20. Classify Movies Based on Ratings [Good,Better and Best]

```
def rating(rating):
    if rating>=7.0:
        return 'Excellent'
    elif rating>=6.0:
        return 'Good'
    else:
        return 'Average'
data['rating_cat']=data['Rating'].apply(rating)

data.head(5)
```

1 to 5 of 5 entriesFilter

index	Rank	Title		Description	Director	Actors		Runtime (Minutes)	Rating	Votes	Revenue (Millions)	Metasc
0	1	Guardians of the Galaxy		A group of intergalactic criminals are forced to work together to stop a fanatical warrior from taking control of the universe.	James Gunn	Chris Pratt, Vin Diesel, Bradley Cooper, Zoe Saldana		121	8.1	757074	333.13	76.0
1	2	Prometheus		Following clues to the origin of mankind, a team finds a structure on a distant moon, but they soon realize they are not alone.	Ridley Scott	Noomi Rapace, Logan Marshall-Green, Michael Fassbender, Charlize Theron		124	7.0	485820	126.46	65.0
2	3	Split		Three girls are kidnapped by a man with a diagnosed 23 distinct personalities. They must try to escape before the apparent emergence of a frightful new 24th.	M. Night Shyamalan	James McAvoy, Anya Taylor-Joy, Haley Lu Richardson, Jessica Sula		117	7.3	157606	138.12	62.0
3	4	Sing		In a city of humanoid animals, a hustling theater impresario's attempt to save his theater with a singing competition becomes grander than he anticipates even as its finalists' find that their lives will never be the same.	Christophe Lourdelet	Matthew McConaughey,Reese Witherspoon, Seth MacFarlane, Scarlett Johansson		108	7.2	60545	270.32	59.0
4	5	Suicide Squad		A secret government agency recruits some of the most dangerous incarcerated super-villains to form a defensive task force. Their first mission: save the world from the apocalypse.	David Ayer	Will Smith, Jared Leto, Margot Robbie, Viola Davis	2016	123	6.2	393727	325.02	40.0

21. Count Number of Action Movies

```
list1=[]
for value in data['Genre']:
    list1.append(value.split(','))
data['temp']=list1
genre=input("Enter Genre you want to count : ").title()
count=0
for value in data['temp']:
    if genre in value:
        count=count+1
print("Total Count is",count)
```

```
Enter Genre you want to count : action
Total Count is 277
```

OR

len(data[data['Genre'].str.contains('action',case=False)])

277

22.Find Most Popular Movie Title (Rating)

data[data['Rating'].max() == data['Rating']]['Title']

Title **54 The Dark Knight** **dtype: object**	

23.top 10 most voted movies by voters

top_movies = data.nlargest(10,'Votes')[['Title', 'Votes', 'Rating']]

print(top_movies)

	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

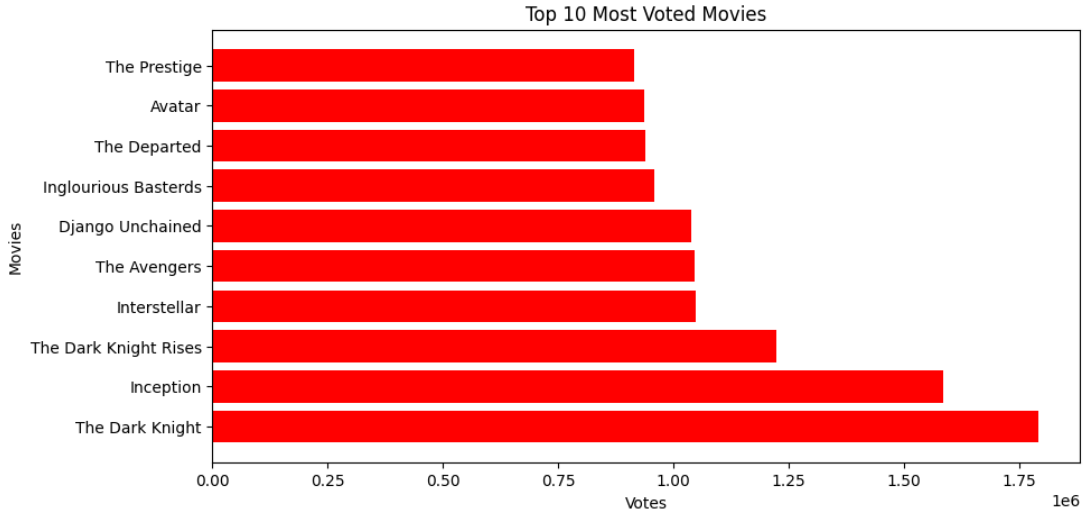
plt.figure(figsize=(10, 5))

plt.barh(top_movies['Title'], top_movies['Votes'], color='Red')

plt.xlabel("Votes")

plt.ylabel("Movies")

plt.title("Top 10 Most Voted Movies")



24.Line plot for revenue trends over the years

plt.figure(figsize=(10, 5))

data.groupby('Year')['Revenue (Millions)'].sum().plot(kind='line', marker='o', color='blue')

plt.xlabel("Year")

plt.ylabel("Total Revenue (Millions)")

plt.title("Yearly Revenue Trends")

plt.show()