Importing libraries

In [1]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

Viewing the first 10 rows

In [6]:
```
data=pd.read_csv(r'C:\Users\new user\OneDrive\Documents\data\covid_r\IMDB-Movie-Data.csv')
```

In [7]:
```
data.head(10)
```

Out[7]:

	Rank	Title	Genre	Description	Director	Actors	Year	Runtime (Minutes)	Rating	V
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	757
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	485
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	157
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	60
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	393
5	6	The Great Wall	Action,Adventure,Fantasy	European mercenaries searching for black powde...	Yimou Zhang	Matt Damon, Tian Jing, Willem Dafoe, Andy Lau	2016	103	6.1	56
6	7	La La Land	Comedy,Drama,Music	A jazz pianist falls for an aspiring actress i...	Damien Chazelle	Ryan Gosling, Emma Stone, Rosemarie DeWitt, J....	2016	128	8.3	258
7	8	Mindhorn	Comedy	A has-been actor best known for playing the ti...	Sean Foley	Essie Davis, Andrea Riseborough, Julian Barrat...	2016	89	6.4	2
8	9	The Lost City of Z	Action,Adventure,Biography	A true-life drama, centering on British explor...	James Gray	Charlie Hunnam, Robert Pattinson, Sienna Mille...	2016	141	7.1	7
9	10	Passengers	Adventure,Drama,Romance	A spacecraft traveling to a distant colony pla...	Morten Tyldum	Jennifer Lawrence, Chris Pratt, Michael Sheen,...	2016	116	7.0	192

```
Viewing the last 10 rows
```

In [8]:
```
data.tail(10)
```

Out[8]:

	Rank	Title	Genre	Description	Director	Actors	Year	Runtime (Minutes)	Rating	Vote
990	991	Underworld: Rise of the Lycans	Action,Adventure,Fantasy	An origins story centered on the centuries-old...	Patrick Tatopoulos	Rhona Mitra, Michael Sheen, Bill Nighy, Steven...	2009	92	6.6	12970
991	992	Taare Zameen Par	Drama,Family,Music	An eight-year-old boy is thought to be a lazy ...	Aamir Khan	Darsheel Safary, Aamir Khan, Tanay Chheda, Sac...	2007	165	8.5	10269
992	993	Take Me Home Tonight	Comedy,Drama,Romance	Four years after graduation, an awkward high s...	Michael Dowse	Topher Grace, Anna Faris, Dan Fogler, Teresa P...	2011	97	6.3	4541
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	14090
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	16408
995	996	Secret in Their Eyes	Crime,Drama,Mystery	A tight-knit team of rising investigators, alo...	Billy Ray	Chiwetel Ejiofor, Nicole Kidman, Julia Roberts...	2015	111	6.2	2758
996	997	Hostel: Part II	Horror	Three American college students studying abroa...	Eli Roth	Lauren German, Heather Matarazzo, Bijou Philli...	2007	94	5.5	7315
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	7069
998	999	Search Party	Adventure,Comedy	A pair of friends embark on a mission to reuni...	Scot Armstrong	Adam Pally, T.J. Miller, Thomas Middleditch,Sh...	2014	93	5.6	488
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	1243

shape

In [9]:
```
data.shape
```

Out[9]: (1000, 12)

In [10]:
```
print('Number of rows', data.shape[0])
print('Number of columns', data.shape[1])
```

```
Number of rows 1000
Number of columns 12
```

Getting more information about the dataset

In [12]:
```
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
```

Checking for missing values in the dataset

In [13]:
```
print('Any missing value?', data.isnull(). values.any())
```

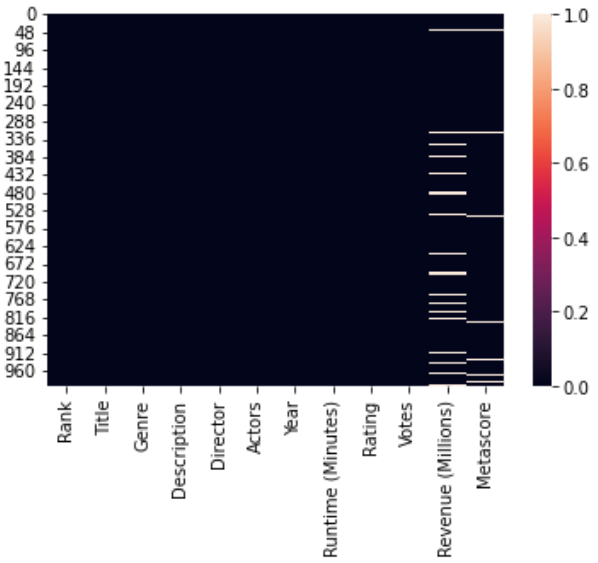
```
Any missing value? True
```

In [15]:
```
data.isnull().sum()
```

Out[15]:
```
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
```

In [16]:
```
sns.heatmap(data.isnull())
```

Out[16]: <AxesSubplot:>



In [17]:
```
per_missing=data.isnull().sum() * 100/ len(data)
```

In [18]:
```
per_missing
```

Out[18]:
```
Rank                   0.0
Title                  0.0
Genre                  0.0
Description            0.0
Director               0.0
Actors                 0.0
Year                   0.0
Runtime (Minutes)      0.0
Rating                 0.0
Votes                  0.0
Revenue (Millions)    12.8
Metascore              6.4
dtype: float64
```

Dropping all missing values

In [22]:
```
data.dropna(axis=0,inplace=True)
```

```
Check for duplicate data
```

In [24]:
```
dup_data=data.duplicated().any()
print('Are there any duplicate values?', dup_data)
```

```
Are there any duplicate values? False
```

Overall statistics

In [27]:
```
data.describe(include='all')
```

Out[27]:

	Rank	Title	Genre	Description	Director	Actors	Year	Runtime (Minutes)	Rating	
count	838.000000	838	838	838	838	838	838.00000	838.000000	838.000000	8
unique	NaN	837	189	838	524	834	NaN	NaN	NaN	
top	NaN	The Host	Action,Adventure,Sci-Fi	A group of intergalactic criminals are forced ...	Ridley Scott	Jennifer Lawrence, Josh Hutcherson, Liam Hemsw...	NaN	NaN	NaN	
freq	NaN	2	50	1	8	2	NaN	NaN	NaN	
mean	485.247017	NaN	NaN	NaN	NaN	NaN	2012.50716	114.638425	6.814320	1
std	286.572065	NaN	NaN	NaN	NaN	NaN	3.17236	18.470922	0.877754	1
min	1.000000	NaN	NaN	NaN	NaN	NaN	2006.00000	66.000000	1.900000	1
25%	238.250000	NaN	NaN	NaN	NaN	NaN	2010.00000	101.000000	6.300000	6
50%	475.500000	NaN	NaN	NaN	NaN	NaN	2013.00000	112.000000	6.900000	1
75%	729.750000	NaN	NaN	NaN	NaN	NaN	2015.00000	124.000000	7.500000	2
max	1000.000000	NaN	NaN	NaN	NaN	NaN	2016.00000	187.000000	9.000000	1

To know title of the movie having runtime >= 180 mins

In [31]:
```
data[data['Runtime (Minutes)']>=180]['Title']
```

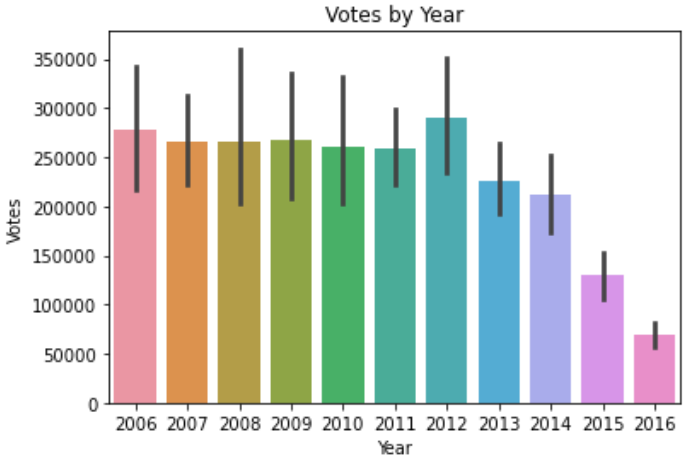
Out[31]:
```
82     The Wolf of Wall Street
88           The Hateful Eight
311             La vie d'Adèle
Name: Title, dtype: object
```

```
In which year was the highest average voting?
```

In [35]:
```
data.groupby('Year')['Votes'].mean().sort_values(ascending= False)
```

Out[35]:
```
Year
2012    290861.483871
2006    277232.219512
2009    267180.577778
2008    266580.145833
2007    266530.704545
2010    261082.929825
2011    259254.736842
2013    225531.892857
2014    211926.881720
2015    129512.651376
2016     68437.823232
Name: Votes, dtype: float64
```

In [37]:
```
sns.barplot(x='Year', y= 'Votes', data=data)
plt.title('Votes by Year')
plt.show()
```

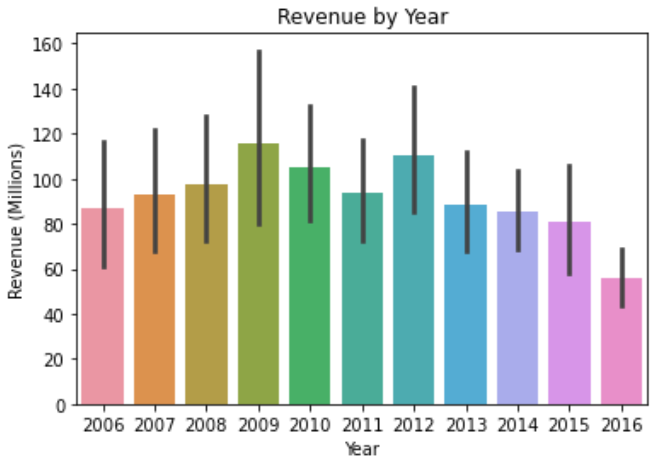


In which Year was the highest average revenue

In [38]:
```
data.groupby('Year')['Revenue (Millions)'].mean().sort_values(ascending= False)
```

Out[38]:
```
Year
2009    115.742000
2012    110.103065
2010    105.081579
2008     97.525417
2011     93.703333
2007     93.074091
2013     88.084643
2006     87.255610
2014     85.433656
2015     80.725596
2016     55.566111
Name: Revenue (Millions), dtype: float64
```

In [39]:
```
sns.barplot(x='Year', y= 'Revenue (Millions)', data=data)
plt.title('Revenue by Year')
plt.show()
```



Averge rating for each director

In [43]:
```
data.groupby('Director')['Rating'].mean().sort_values(ascending=False)
```

Out[43]:
```
Director
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
Name: Rating, Length: 524, dtype: float64
```

Top 10 lengthy movies and runtime

In [48]:
```
top10_len=data.nlargest(10,'Runtime (Minutes)')[['Title', 'Runtime (Minutes)']].set_index('Title'
```

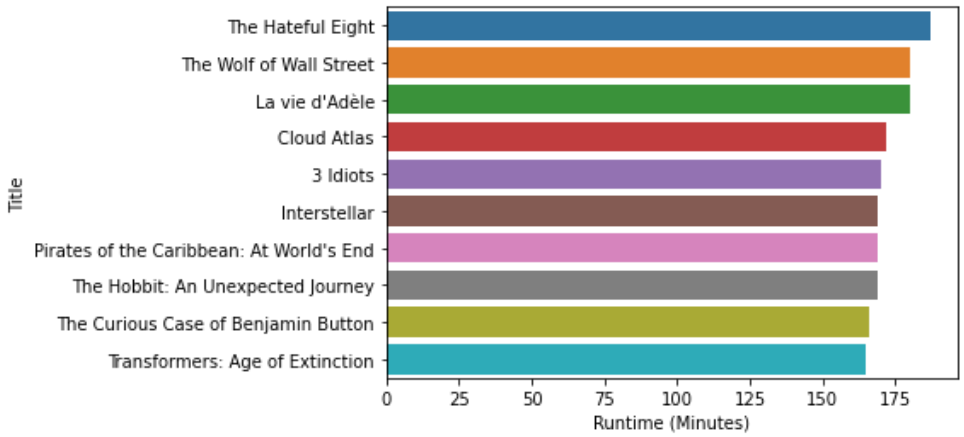
In [49]:
```
top10_len
```

Out[49]:

Title	Runtime (Minutes)
The Hateful Eight	187
The Wolf of Wall Street	180
La vie d'Adèle	180
Cloud Atlas	172
3 Idiots	170
Interstellar	169
Pirates of the Caribbean: At World's End	169
The Hobbit: An Unexpected Journey	169
The Curious Case of Benjamin Button	166
Transformers: Age of Extinction	165

In [51]:
```
sns.barplot(x='Runtime (Minutes)', y=top10_len.index, data=top10_len)
```

Out[51]:
```
<AxesSubplot:xlabel='Runtime (Minutes)', ylabel='Title'>
```

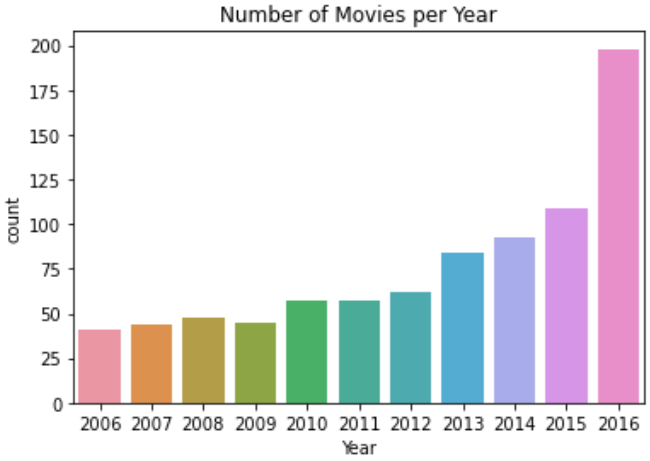


Number of Movies per year

In [52]:
```
data['Year'].value_counts()
```

Out[52]:
```
2016    198
2015    109
2014     93
2013     84
2012     62
2011     57
2010     57
2008     48
2009     45
2007     44
2006     41
Name: Year, dtype: int64
```

In [55]:
```
sns.countplot(x='Year', data=data)
plt.title('Number of Movies per Year')
plt.show()
```



Most popular movie title (Highest Revenue)

In [59]:
```
data[data['Revenue (Millions)']. max() == data['Revenue (Millions)']]['Title']
```

Out[59]:
```
50    Star Wars: Episode VII - The Force Awakens
Name: Title, dtype: object
```

TOP 10 HIGHEST RATED MOVIE TITLES AND ITS DIRECTORS

In [60]:
```
top10_len=data.nlargest(10,'Rating')[['Title', 'Rating','Director']].set_index('Title')
```

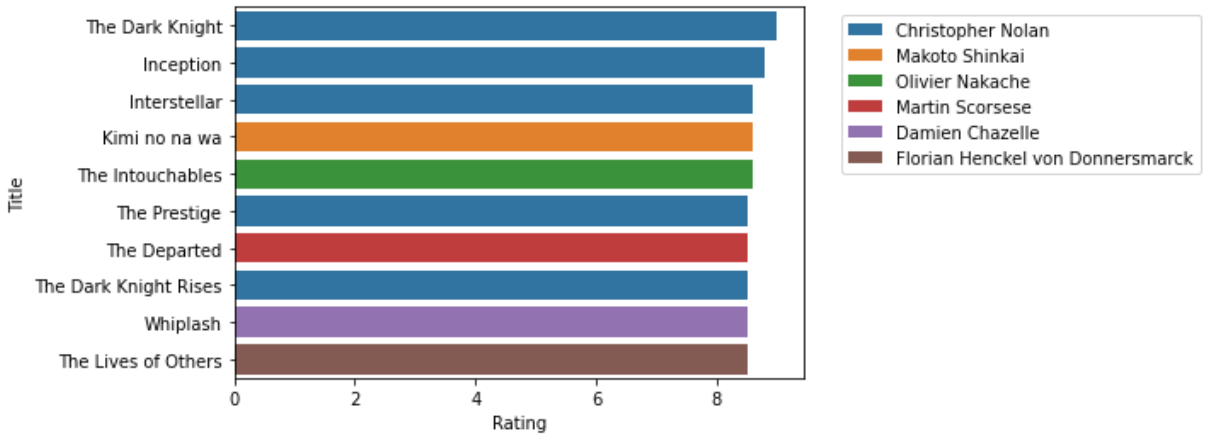
In [61]:
```
top10_len
```

Out[61]:

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

In [70]:
```
sns.barplot(x='Rating', y=top10_len.index, data=top10_len,hue='Director', dodge=False)
plt.legend(bbox_to_anchor=(1.05, 1), loc=2)
```

Out[70]:
```
<matplotlib.legend.Legend at 0x2211d219bb0>
```



Top 10 highest Revenue Movie Titles

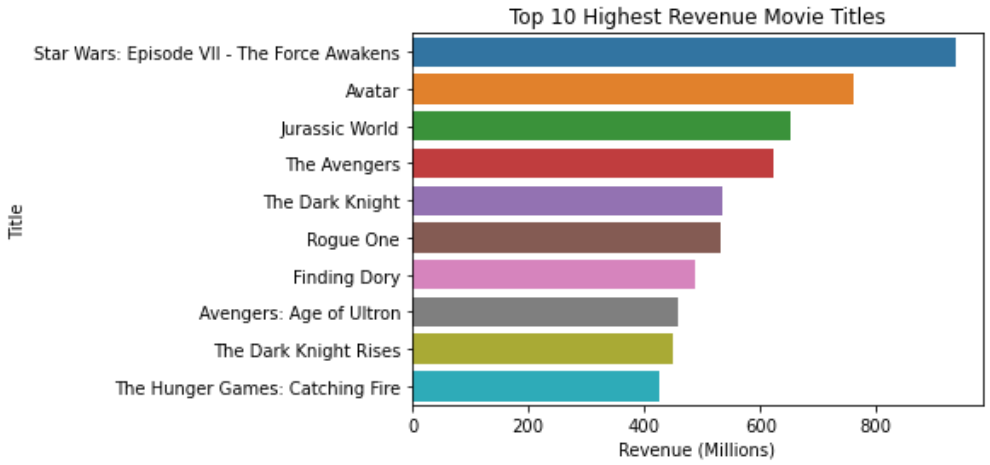
In [71]:
```
top10_len=data.nlargest(10,'Revenue (Millions)')[['Title', 'Revenue (Millions)']].set_index('Titl
```

In [72]:
```
top10_len
```

Out[72]:

Title	Revenue (Millions)
Star Wars: Episode VII - The Force Awakens	936.63
Avatar	760.51
Jurassic World	652.18
The Avengers	623.28
The Dark Knight	533.32
Rogue One	532.17
Finding Dory	486.29
Avengers: Age of Ultron	458.99
The Dark Knight Rises	448.13
The Hunger Games: Catching Fire	424.65

In [75]:
```
sns.barplot(x='Revenue (Millions)', y=top10_len.index, data=top10_len)
plt.title('Top 10 Highest Revenue Movie Titles')
plt.show()
```



Average rating of movies year wise

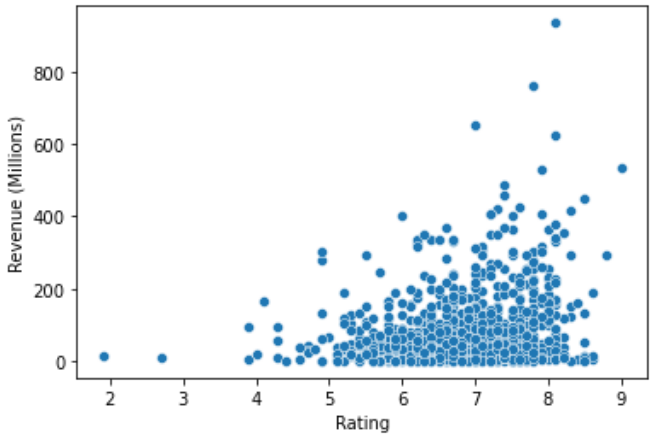
In [78]:
```
data.groupby('Year')['Rating'].mean().sort_values(ascending=False)
```

Out[78]:
```
Year
2006    7.143902
2007    7.140909
2011    6.945614
2012    6.933871
2009    6.911111
2010    6.894737
2013    6.832143
2014    6.822581
2008    6.708333
2015    6.674312
2016    6.644444
Name: Rating, dtype: float64
```

Does rating affect the revenue?

In [79]:
```
sns.scatterplot(x='Rating', y= 'Revenue (Millions)', data = data)
```

Out[79]:
```
<AxesSubplot:xlabel='Rating', ylabel='Revenue (Millions)'>
```



Classiy movies based on ratings [Excellent, Good and Average]

In [82]:
```
def rating(rating):
    if rating >= 7.0:
        return "Excellent"
    elif rating>= 6.0:
        return "Good"
    else:
        return "Average"
```

In [83]:
```
data['rating_cat']= data['Rating']. apply(rating)
```

In [84]:
```
data.head()
```

Out[84]:

	Rank	Title	Genre	Description	Director	Actors	Year	Runtime (Minutes)	Rating	Vot
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	7570
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	4858
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	1576
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	605
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	3937

Count number of action movies

In [88]:
```
len(data[data['Genre'].str.contains('Action', case=False)])
```

Out[88]:
```
277
```

Unique values from genre

In [89]:
```
list1=[]
for value in data['Genre']:
    list1.append(value.split(','))
```

In [90]:
```
list1
```

Out[90]:
```
[['Action', 'Adventure', 'Sci-Fi'],
 ['Adventure', 'Mystery', 'Sci-Fi'],
 ['Horror', 'Thriller'],
 ['Animation', 'Comedy', 'Family'],
 ['Action', 'Adventure', 'Fantasy'],
 ['Action', 'Adventure', 'Fantasy'],
 ['Comedy', 'Drama', 'Music'],
 ['Action', 'Adventure', 'Biography'],
 ['Adventure', 'Drama', 'Romance'],
 ['Adventure', 'Family', 'Fantasy'],
 ['Biography', 'Drama', 'History'],
 ['Action', 'Adventure', 'Sci-Fi'],
 ['Animation', 'Adventure', 'Comedy'],
 ['Action', 'Comedy', 'Drama'],
 ['Animation', 'Adventure', 'Comedy'],
 ['Biography', 'Drama', 'History'],
 ['Action', 'Thriller'],
 ['Biography', 'Drama'],
 ['Drama', 'Mystery', 'Sci-Fi'],
```

In [98]:
```
one_d=[]
for item in list1:
    for item1 in item:
        one_d.append(item1)
```

In [99]:
```
one_d
```

Out[99]:
```
['Action',
 'Adventure',
 'Sci-Fi',
 'Adventure',
 'Mystery',
 'Sci-Fi',
 'Horror',
 'Thriller',
 'Animation',
 'Comedy',
 'Family',
 'Action',
 'Adventure',
 'Fantasy',
 'Action',
 'Adventure',
 'Fantasy',
 'Comedy',
 'Drama',
```

In [101]:
```
uni_list=[]
for item in one_d:
    if item not in uni_list:
        uni_list.append(item)
```

In [102]:
```
uni_list
```

Out[102]:
```
['Action',
 'Adventure',
 'Sci-Fi',
 'Mystery',
 'Horror',
 'Thriller',
 'Animation',
 'Comedy',
 'Family',
 'Fantasy',
 'Drama',
 'Music',
 'Biography',
 'Romance',
 'History',
 'Western',
 'Crime',
 'War',
 'Musical',
 'Sport']
```

How many films of each Genre were made?

In [103]:
```
one_d=[]
for item in list1:
    for item1 in item:
        one_d.append(item1)
```

In [104]:
```
one_d
```

Out[104]:
```
['Action',
 'Adventure',
 'Sci-Fi',
 'Adventure',
 'Mystery',
 'Sci-Fi',
 'Horror',
 'Thriller',
 'Animation',
 'Comedy',
 'Family',
 'Action',
 'Adventure',
 'Fantasy',
 'Action',
 'Adventure',
 'Fantasy',
 'Comedy',
 'Drama',
```

In [105]:
```
from collections import Counter
```

In [106]:
```
Counter(one_d)
```

Out[106]:
```
Counter({'Action': 277,
         'Adventure': 244,
         'Sci-Fi': 107,
         'Mystery': 86,
         'Horror': 87,
         'Thriller': 148,
         'Animation': 45,
         'Comedy': 250,
         'Family': 48,
         'Fantasy': 92,
         'Drama': 419,
         'Music': 15,
         'Biography': 67,
         'Romance': 120,
         'History': 25,
         'Western': 4,
         'Crime': 126,
         'War': 10,
         'Musical': 5,
         'Sport': 15})
```

In [ ]:
```

```