IMDB op 250 movies analysis

https://www.kaggle.com/datasets/karkavelrajaj/imdb-top-250-movies?select=movies.csv\ (https://www.kaggle.com/datasets/karkavelrajaj/imdb-top-250-movies?select=movies.csv%5C) mentioned above is the link to a dataset.

1. plot a graph to show the distribution of genre in the top 250 movies
2. find out which movie has the maximum number of votes and which genre it belongs to and its duration.
3. find out which movie has the minimum number of votes and which genre it belongs to and its duration.
4. find out movies of each genre which has maximum number of votes.

About Dataset This dataset is having the data of the top 250 Movies as per their IMDB rating listed on the official website of IMDB In this EDA projects we will be perfroming IMDB top 250 movies database analysis which is present on kaggle platform itself, we will be using multiple libraries in this project which will help us in analysing and visualizing our dataset.

Visualizing is important as it helps people/developers see, interact with, and better understand the data

Libraries to be used during this project

In [48]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

```

Loading and Summarizing the Dataset

In [49]:
```
df = pd.read_csv('C:\\Users\\User\\Desktop\\movies.csv')
```

In [50]:

```
df.head()
```

Out[50]:

	rank	movie_id	title	year	link	imbd_votes	imbd_rat
0	1	tt0111161	The Shawshank Redemption	1994	https://www.imdb.com/title/tt0111161	2,711,075	
1	2	tt0068646	The Godfather	1972	https://www.imdb.com/title/tt0068646	1,882,829	
2	3	tt0468569	The Dark Knight	2008	https://www.imdb.com/title/tt0468569	2,684,051	
3	4	tt0071562	The Godfather Part II	1974	https://www.imdb.com/title/tt0071562	1,285,350	
4	5	tt0050083	12 Angry Men	1957	https://www.imdb.com/title/tt0050083	800,954	

5 rows × 22 columns

In [51]:
```
1 df.tail()      #shows last five rows of dataset
```

Out[51]:

	rank	movie_id	title	year	link	imbd_votes	imbd_ratin
245	246	tt0071411	Dersu Uzala	1975	https://www.imdb.com/title/tt0071411	31,167	8.
246	247	tt1454029	The Help	2011	https://www.imdb.com/title/tt1454029	466,011	8.
247	248	tt0103639	Aladdin	1992	https://www.imdb.com/title/tt0103639	429,219	8.
248	249	tt0083987	Gandhi	1982	https://www.imdb.com/title/tt0083987	234,688	8.
249	250	tt0099348	Dances with Wolves	1990	https://www.imdb.com/title/tt0099348	271,823	8.

5 rows × 22 columns

In [8]:
```
1 df.columns
2
```

Out[8]:
```
Index(['rank', 'movie_id', 'title', 'year', 'link', 'imbd_votes',
       'imbd_rating', 'certificate', 'duration', 'genre', 'cast_id',
       'cast_name', 'director_id', 'director_name', 'writer_id', 'writer_n
ame',
       'storyline', 'user_id', 'user_name', 'review_id', 'review_title',
       'review_content'],
      dtype='object')
```

In [9]:
```
1 df.info()
2
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 250 entries, 0 to 249
Data columns (total 22 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   rank            250 non-null    int64
 1   movie_id        250 non-null    object
 2   title           250 non-null    object
 3   year            250 non-null    int64
 4   link            250 non-null    object
 5   imbd_votes      250 non-null    object
 6   imbd_rating     250 non-null    float64
 7   certificate     249 non-null    object
 8   duration        250 non-null    object
 9   genre           250 non-null    object
 10  cast_id         250 non-null    object
 11  cast_name       250 non-null    object
 12  director_id     250 non-null    object
 13  director_name   250 non-null    object
 14  writer_id       250 non-null    object
 15  writer_name     250 non-null    object
 16  storyline       250 non-null    object
 17  user_id         250 non-null    object
 18  user_name       250 non-null    object
 19  review_id       250 non-null    object
 20  review_title    250 non-null    object
 21  review_content  250 non-null    object
dtypes: float64(1), int64(2), object(19)
memory usage: 43.1+ KB
```

In [52]:
```
1 df.describe()
```

Out[52]:

	rank	year	imbd_rating
count	250.000000	250.000000	250.000000
mean	125.500000	1986.360000	8.306800
std	72.312977	25.125356	0.229006
min	1.000000	1921.000000	8.000000
25%	63.250000	1966.250000	8.100000
50%	125.500000	1994.000000	8.200000
75%	187.750000	2006.000000	8.400000
max	250.000000	2022.000000	9.300000

Data Preparation -Dropping irrelevant columns

In [44]:
```
df = df[['rank',
         #'movie_id',
         'title', 'year',
         #'link',
         'imbd_votes','imbd_rating', 'certificate', 'duration', 'genre'
         #'cast_id','cast_name', 'director_id',
         'director_name',
         #'writer_id', 'writer_name',
         #'storyline', 'user_id', 'user_name', 'review_id', 'review_tit
         #'review_content'
        ]]
```

In [45]:
```
df.head()
```

Out[45]:

	title	year	imbd_votes	imbd_rating	certificate	duration	genre	direct
0	The Shawshank Redemption	1994	2711075	9.3	R	142	Drama	
1	The Godfather	1972	1882829	9.2	R	175	Crime,Drama	Fra
2	The Dark Knight	2008	2684051	9.0	PG-13	152	Action,Crime,Drama	Cl
3	The Godfather Part II	1974	1285350	9.0	R	202	Crime,Drama	Fra
4	12 Angry Men	1957	800954	9.0	Approved	96	Crime,Drama	Sidr

In [46]:
```
df.shape
```

Out[46]:
```
(249, 8)
```

Renaming column names

In [55]:
```
df.columns = df.columns.str.title()     # First letter of every word wi
df.head(2)
```

Out[55]:

	Rank	Movie_Id	Title	Year	Link	Imbd_Votes	Imbd_R
0	1	tt0111161	The Shawshank Redemption	1994	https://www.imdb.com/title/tt0111161	2,711,075	
1	2	tt0068646	The Godfather	1972	https://www.imdb.com/title/tt0068646	1,882,829	

2 rows × 22 columns

Checking for Null Values in the Dataset

In [12]:
```
1 df.isnull().sum()
```

Out[12]:
```
rank              0
movie_id          0
title             0
year              0
link              0
imbd_votes        0
imbd_rating       0
certificate       1
duration          0
genre             0
cast_id           0
cast_name         0
director_id       0
director_name     0
writer_id         0
writer_name       0
storyline         0
user_id           0
user_name         0
review_id         0
review_title      0
review_content    0
dtype: int64
```

In [ ]:
```
1 # only 1 null value is found . we can use bfill  which will replace the
2 #df['Certificate'] = df['Certificate'].bfill()
3 #df['Certificate'].isnull().sum()
```

Handling the Null Values in the Dataset

In [14]:
```
1 df = df.dropna()
```

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

Out[15]:
```
rank             0
movie_id         0
title            0
year             0
link             0
imbd_votes       0
imbd_rating      0
certificate      0
duration         0
genre            0
cast_id          0
cast_name        0
director_id      0
director_name    0
writer_id        0
writer_name      0
storyline        0
user_id          0
user_name        0
review_id        0
review_title     0
review_content   0
dtype: int64
```

In [47]:
```
#Finding duplicates if any
df.duplicated().sum()
```

Out[47]:
```
0
```

Replacing Commas(,) in Imbd_Votes so we can use it as integer

In [56]:
```
df['Imbd_Votes']
```

Out[56]:
```
0      2,711,075
1      1,882,829
2      2,684,051
3      1,285,350
4        800,954
         ...
245       31,167
246      466,011
247      429,219
248      234,688
249      271,823
Name: Imbd_Votes, Length: 250, dtype: object
```

In [18]:
```
imbd_votes = []
for i in df['imbd_votes'].values:
    imbd_votes.append(int(i.replace(',','')))
```

In [58]:
```
df['Imbd_Votes'] = df['Imbd_Votes'].str.replace(',','')
```

In [59]:
```
1 df.head()
```

Out[59]:

	Rank	Movie_Id	Title	Year	Link	Imbd_Votes	Imbd_R
0	1	tt0111161	The Shawshank Redemption	1994	https://www.imdb.com/title/tt0111161	2711075	
1	2	tt0068646	The Godfather	1972	https://www.imdb.com/title/tt0068646	1882829	
2	3	tt0468569	The Dark Knight	2008	https://www.imdb.com/title/tt0468569	2684051	
3	4	tt0071562	The Godfather Part II	1974	https://www.imdb.com/title/tt0071562	1285350	
4	5	tt0050083	12 Angry Men	1957	https://www.imdb.com/title/tt0050083	800954	

5 rows × 22 columns

In [60]:
```
1 df['Imbd_Votes']
```

Out[60]:
```
0      2711075
1      1882829
2      2684051
3      1285350
4       800954
        ...
245      31167
246     466011
247     429219
248     234688
249     271823
Name: Imbd_Votes, Length: 250, dtype: object
```

Changing dtype if required

In [61]:
```
1 df.dtypes
```

Out[61]:
```
Rank              int64
Movie_Id         object
Title            object
Year              int64
Link             object
Imbd_Votes       object
Imbd_Rating     float64
Certificate      object
Duration         object
Genre            object
Cast_Id          object
Cast_Name        object
Director_Id      object
Director_Name    object
Writer_Id        object
Writer_Name      object
Storyline        object
User_Id          object
User_Name        object
Review_Id        object
Review_Title     object
Review_Content   object
dtype: object
```

In [62]:
```
1 df['Imbd_Votes'] = pd.to_numeric(df['Imbd_Votes'])
```

In [63]:
```
1 df.dtypes
```

Out[63]:
```
Rank              int64
Movie_Id         object
Title            object
Year              int64
Link             object
Imbd_Votes        int64
Imbd_Rating     float64
Certificate      object
Duration         object
Genre            object
Cast_Id          object
Cast_Name        object
Director_Id      object
Director_Name    object
Writer_Id        object
Writer_Name      object
Storyline        object
User_Id          object
User_Name        object
Review_Id        object
Review_Title     object
Review_Content   object
dtype: object
```

Feature understanding

In [64]:
```
1 df.head(2)
```

Out[64]:

	Rank	Movie_Id	Title	Year	Link	Imbd_Votes	Imbd_R
0	1	tt0111161	The Shawshank Redemption	1994	https://www.imdb.com/title/tt0111161	2711075	
1	2	tt0068646	The Godfather	1972	https://www.imdb.com/title/tt0068646	1882829	

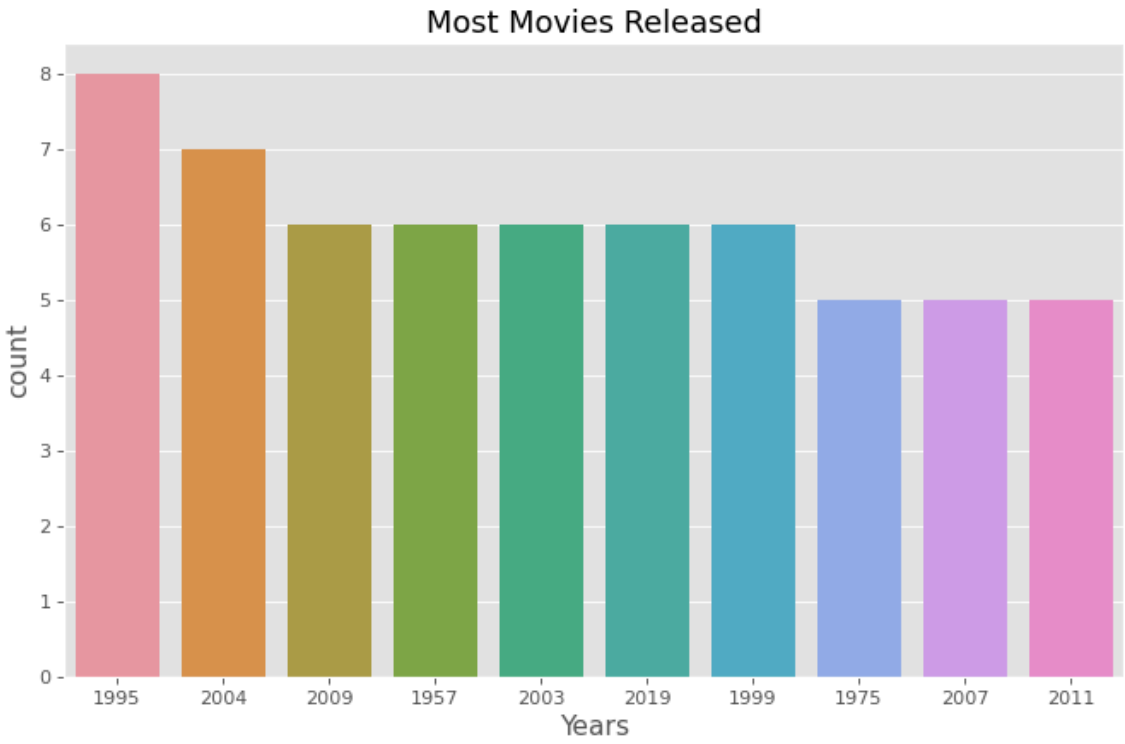
2 rows × 22 columns

In which year most movies released? (Top 10)

In [66]:
```
1 ax = df.Year.value_counts().head(10)
2 ax
```

Out[66]:
```
1995    8
2004    7
2009    6
1957    6
2003    6
2019    6
1999    6
1975    5
2007    5
2011    5
Name: Year, dtype: int64
```

In [67]:
```
plt.figure(figsize = (10,6),dpi=80)
sns.countplot(x='Year', data=df , order=ax.index)
plt.title('Most Movies Released',fontsize=16)
plt.xlabel('Years',fontsize = 14)
plt.ylabel('count',fontsize = 14)
plt.show()
```



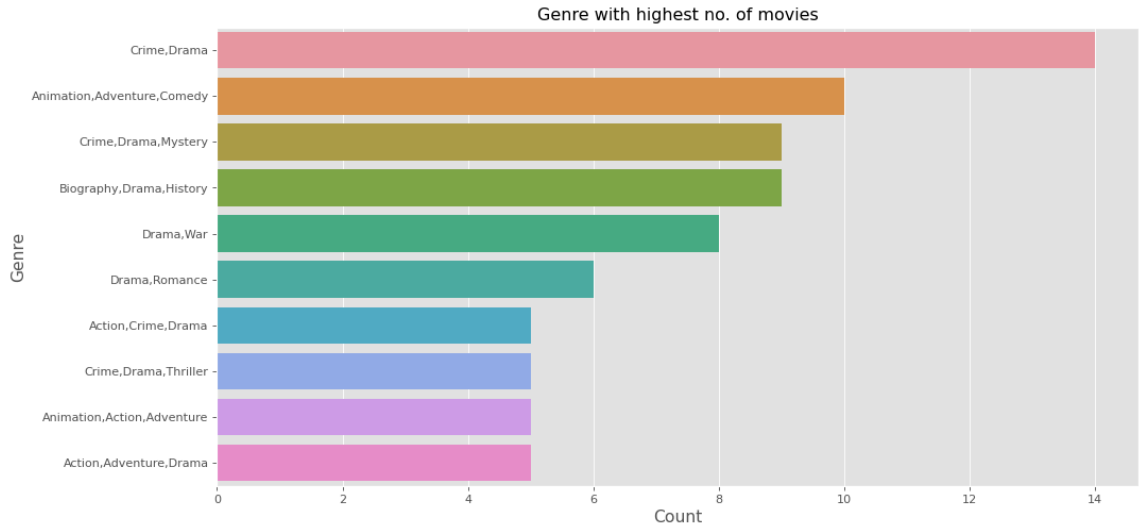
Most movies released in a year is 1995

Which Genre has highest number of movies? (Top10)

In [68]:
```
ax = df.query('Genre!="Drama"')['Genre'].value_counts().head(10)
ax
```

Out[68]:
```
Crime,Drama                    14
Animation,Adventure,Comedy     10
Crime,Drama,Mystery             9
Biography,Drama,History         9
Drama,War                       8
Drama,Romance                   6
Action,Crime,Drama              5
Crime,Drama,Thriller            5
Animation,Action,Adventure      5
Action,Adventure,Drama          5
Name: Genre, dtype: int64
```

In [69]:
```
plt.figure(figsize=(14,7),dpi=80)
sns.countplot(y='Genre', data=df, order=ax.index)
plt.title('Genre with highest no. of movies')
plt.xlabel('Count',fontsize=14)
plt.ylabel('Genre',fontsize=14)
plt.show()
```



Genre which has produced most no. of movies is Crime,Drama

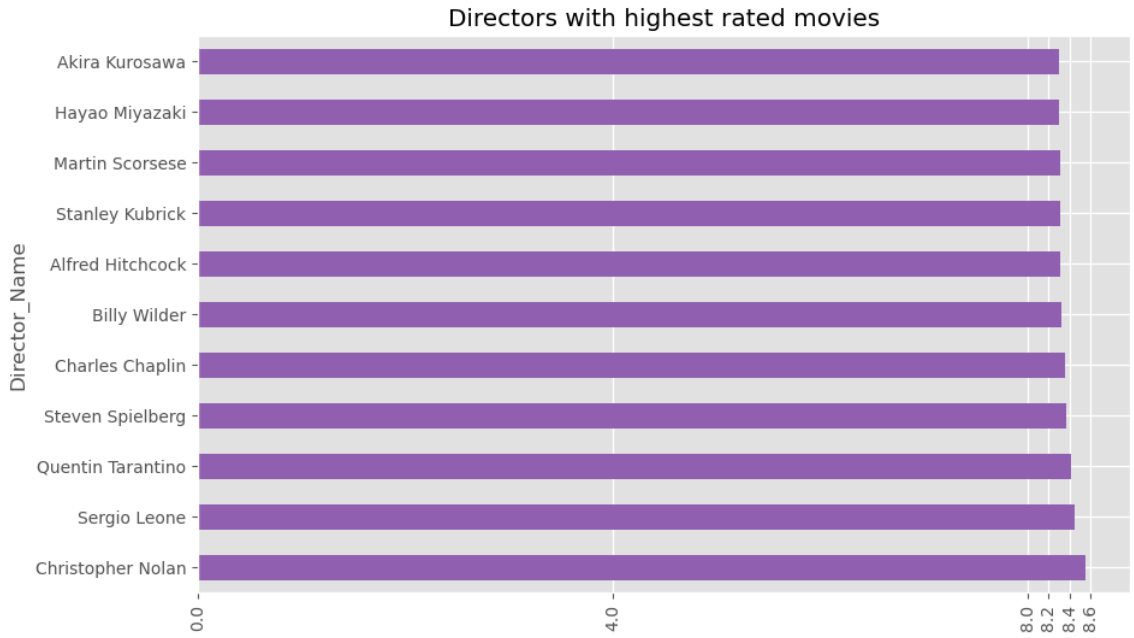
Which director has produced highest rated movies on an average ? MIN MOVIES = 4

In [71]:
```
ax = df.groupby('Director_Name')['Imbd_Rating']\
.agg(['count','mean'])\
.query("count>=4")\
.sort_values('mean',ascending=False)['mean']
ax
```

Out[71]:
```
Director_Name
Christopher Nolan    8.557143
Sergio Leone         8.450000
Quentin Tarantino    8.420000
Steven Spielberg     8.371429
Charles Chaplin      8.360000
Billy Wilder         8.320000
Alfred Hitchcock     8.316667
Stanley Kubrick      8.314286
Martin Scorsese      8.314286
Hayao Miyazaki       8.300000
Akira Kurosawa       8.300000
Name: mean, dtype: float64
```

In [72]:

```
ax.plot(kind='barh',figsize=(10,6),color='#9460b3')
plt.xticks([0,4,8,8.2,8.4,8.6],rotation='vertical')
plt.title('Directors with highest rated movies')
plt.show()
```



Christopher Nolan is the director which has produced highest rated movies

In [73]:

```
df.head(2)
```

Out[73]:

	Rank	Movie_Id	Title	Year	Link	Imbd_Votes	Imbd_R
0	1	tt0111161	The Shawshank Redemption	1994	https://www.imdb.com/title/tt0111161	2711075	
1	2	tt0068646	The Godfather	1972	https://www.imdb.com/title/tt0068646	1882829	

2 rows × 22 columns

Genre with highest imbd votes?

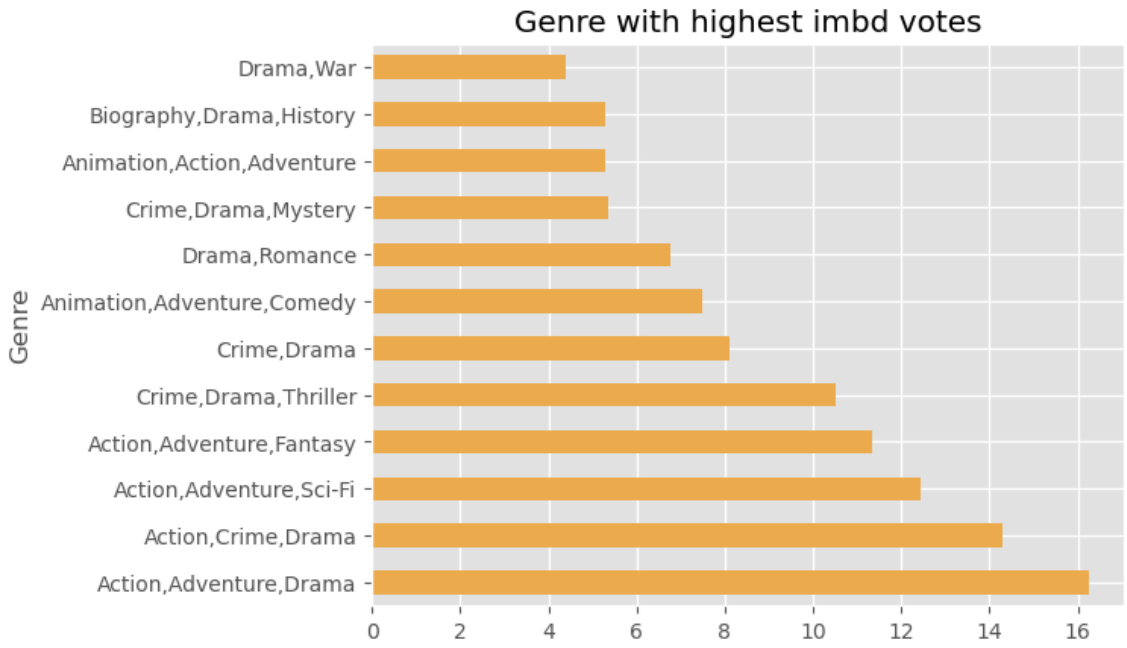
In [74]:
```
df['Imbd_Votes'] = df['Imbd_Votes']/100000
df['Imbd_Votes']
```

Out[74]:
```
0      27.11075
1      18.82829
2      26.84051
3      12.85350
4       8.00954
         ...
245     0.31167
246     4.66011
247     4.29219
248     2.34688
249     2.71823
Name: Imbd_Votes, Length: 250, dtype: float64
```

In [75]:
```
ax = df.query('Genre!="Drama"').groupby("Genre")['Imbd_Votes']\
.agg(['count','mean'])\
.query('count>=5')\
.sort_values('mean',ascending=False)['mean']
ax
```

Out[75]:
```
Genre
Action,Adventure,Drama        16.236516
Action,Crime,Drama            14.277402
Action,Adventure,Sci-Fi       12.454394
Action,Adventure,Fantasy      11.331398
Crime,Drama,Thriller          10.508566
Crime,Drama                    8.094168
Animation,Adventure,Comedy     7.474754
Drama,Romance                  6.755375
Crime,Drama,Mystery            5.366801
Animation,Action,Adventure     5.304332
Biography,Drama,History        5.303087
Drama,War                      4.393323
Name: mean, dtype: float64
```

In [76]:
```
ax.plot(kind='barh',color='#edae4e')
plt.title('Genre with highest imbd votes')
plt.show()
```



Summary

The IMDb Top 250 Movies dataset is a collection of the 250 highest rated movies according to IMDb (Internet Movie Database), as rated by the website's users. The dataset includes information about each movie, such as its title, year of release, length, genre, IMDb rating, and number of ratings. It also includes information about the cast and crew of each movie, such as the director, actors, and writers.

The dataset includes a variety of movies, ranging from classic films to more recent releases. It includes both fictional and non-fictional movies, and covers a wide range of genres, including action, adventure, comedy, drama, fantasy, horror, mystery, romance, science fiction, and more.

In this analysis, we have learned about

1. Most movies released in year - 1995 was the year
2. Genre with highest no. of movies - Crime,Drama
3. Director with highest imbd rating = Christopher Nolan
4. Genre with highest imbd votes = Action,Adventure,Drama

In [ ]:
```
1
```

In [ ]:
```
1
```