IMDB Movie Analysis

In [1]:
```python
#Importing required packages
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import plotly.express as px
```

In [2]:
```python
#Importing csv to the dataframe df
df = pd.read_csv('Downloads/IMDB_Movies.csv')
```

In [3]:
```python
#Checking the dataset
df
```

Out[3]:

	color	director_name	num_critic_for_reviews	duration	director_facebook_likes	actor_3_facebook_likes	actor_2_name	actor_1_facebook_likes	gros
0	Color	James Cameron	723.0	178.0	0.0	855.0	Joel David Moore	1000.0	760505847.
1	Color	Gore Verbinski	302.0	169.0	563.0	1000.0	Orlando Bloom	40000.0	309404152.
2	Color	Sam Mendes	602.0	148.0	0.0	161.0	Rory Kinnear	11000.0	200074175.
3	Color	Christopher Nolan	813.0	164.0	22000.0	23000.0	Christian Bale	27000.0	448130642.
4	NaN	Doug Walker	NaN	NaN	131.0	NaN	Rob Walker	131.0	Na
...	...	...	...	...	...	...	...	...	..
5038	Color	Scott Smith	1.0	87.0	2.0	318.0	Daphne Zuniga	637.0	Na
5039	Color	NaN	43.0	43.0	NaN	319.0	Valorie Curry	841.0	Na
5040	Color	Benjamin Roberds	13.0	76.0	0.0	0.0	Maxwell Moody	0.0	Na
5041	Color	Daniel Hsia	14.0	100.0	0.0	489.0	Daniel Henney	946.0	10443.
5042	Color	Jon Gunn	43.0	90.0	16.0	16.0	Brian Herzlinger	86.0	85222.

5043 rows × 28 columns

A.Cleaning the data: This is one of the most important step to perform before moving forward with the analysis. Use your knowledge learned till now to do this. (Dropping columns, removing null values, etc.
Hint: Clean the data

In [4]:
```python
# Checking the columns for analysis
df.columns
```

Out[4]:
```
Index(['color', 'director_name', 'num_critic_for_reviews', 'duration',
       'director_facebook_likes', 'actor_3_facebook_likes', 'actor_2_name',
       'actor_1_facebook_likes', 'gross', 'genres', 'actor_1_name',
       'movie_title', 'num_voted_users', 'cast_total_facebook_likes',
       'actor_3_name', 'facenumber_in_poster', 'plot_keywords',
       'movie_imdb_link', 'num_user_for_reviews', 'language', 'country',
       'content_rating', 'budget', 'title_year', 'actor_2_facebook_likes',
       'imdb_score', 'aspect_ratio', 'movie_facebook_likes'],
      dtype='object')
```

In [5]:
```python
# Dropping unnecessary columns
df = df.drop(['color','director_facebook_likes','actor_1_facebook_likes','actor_2_facebook_likes','actor_3_facebook_likes','actor_2_name','cast_total_facebook_likes','actor_3_name','duration','facenumber_in_poster','content_rating','country','movie_imdb_link','aspect_ratio','plot_keywords'],axis=1)
```

In [6]:
```python
#Finding the number on null values in decreasing order
df.isnull().sum().sort_values(ascending=False)
```

Out[6]:
```
gross                      884
budget                     492
title_year                 108
director_name              104
num_critic_for_reviews      50
num_user_for_reviews        20
language                    12
actor_1_name                 7
genres                       0
movie_title                  0
num_voted_users              0
imdb_score                   0
movie_facebook_likes         0
dtype: int64
```

In [7]:
```python
#Since gross and budget have the most null values I have passed the data frame with not
# null constraint as index to retrieve all not null values
df = df[df['gross'].notnull()]
df = df[df['budget'].notnull()]
```

In [8]:
```python
# Checking after dropping null values in gross and budget
df.isnull().sum(axis=0).sort_values(ascending=False)
```

Out[8]:
```
actor_1_name              3
language                  3
num_critic_for_reviews    1
director_name             0
gross                     0
genres                    0
movie_title               0
num_voted_users           0
num_user_for_reviews      0
budget                    0
title_year                0
imdb_score                0
movie_facebook_likes      0
dtype: int64
```

In [9]:
```python
#Dropping all dupicate records
df.drop_duplicates()
```

Out[9]:

	director_name	num_critic_for_reviews	gross	genres	actor_1_name	movie_title	num_voted_users	num_user_for_review
0	James Cameron	723.0	760505847.0	Action|Adventure|Fantasy|Sci-Fi	CCH Pounder	Avatar	886204	305
1	Gore Verbinski	302.0	309404152.0	Action|Adventure|Fantasy	Johnny Depp	Pirates of the Caribbean: At World's End	471220	123
2	Sam Mendes	602.0	200074175.0	Action|Adventure|Thriller	Christoph Waltz	Spectre	275868	99
3	Christopher Nolan	813.0	448130642.0	Action|Thriller	Tom Hardy	The Dark Knight Rises	1144337	270
5	Andrew Stanton	462.0	73058679.0	Action|Adventure|Sci-Fi	Daryl Sabara	John Carter	212204	73
...	...	...	...	...	...	...	...	..
5033	Shane Carruth	143.0	424760.0	Drama|Sci-Fi|Thriller	Shane Carruth	Primer	72639	37
5034	Neill Dela Llana	35.0	70071.0	Thriller	Ian Gamazon	Cavite	589	3
5035	Robert Rodriguez	56.0	2040920.0	Action|Crime|Drama|Romance|Thriller	Carlos Gallardo	El Mariachi	52055	13
5037	Edward Burns	14.0	4584.0	Comedy|Drama	Kerry Bishé	Newlyweds	1338	1
5042	Jon Gunn	43.0	85222.0	Documentary	John August	My Date with Drew	4285	8

3856 rows × 13 columns

B.Movies with highest profit: Create a new column called profit which contains the difference of the two columns: gross and budget. Sort the column using the profit column as reference. Plot profit (y-axis) vs budget (x- axis) and observe the outliers using the appropriate chart type
Hint: Find the movies with the highest profit?

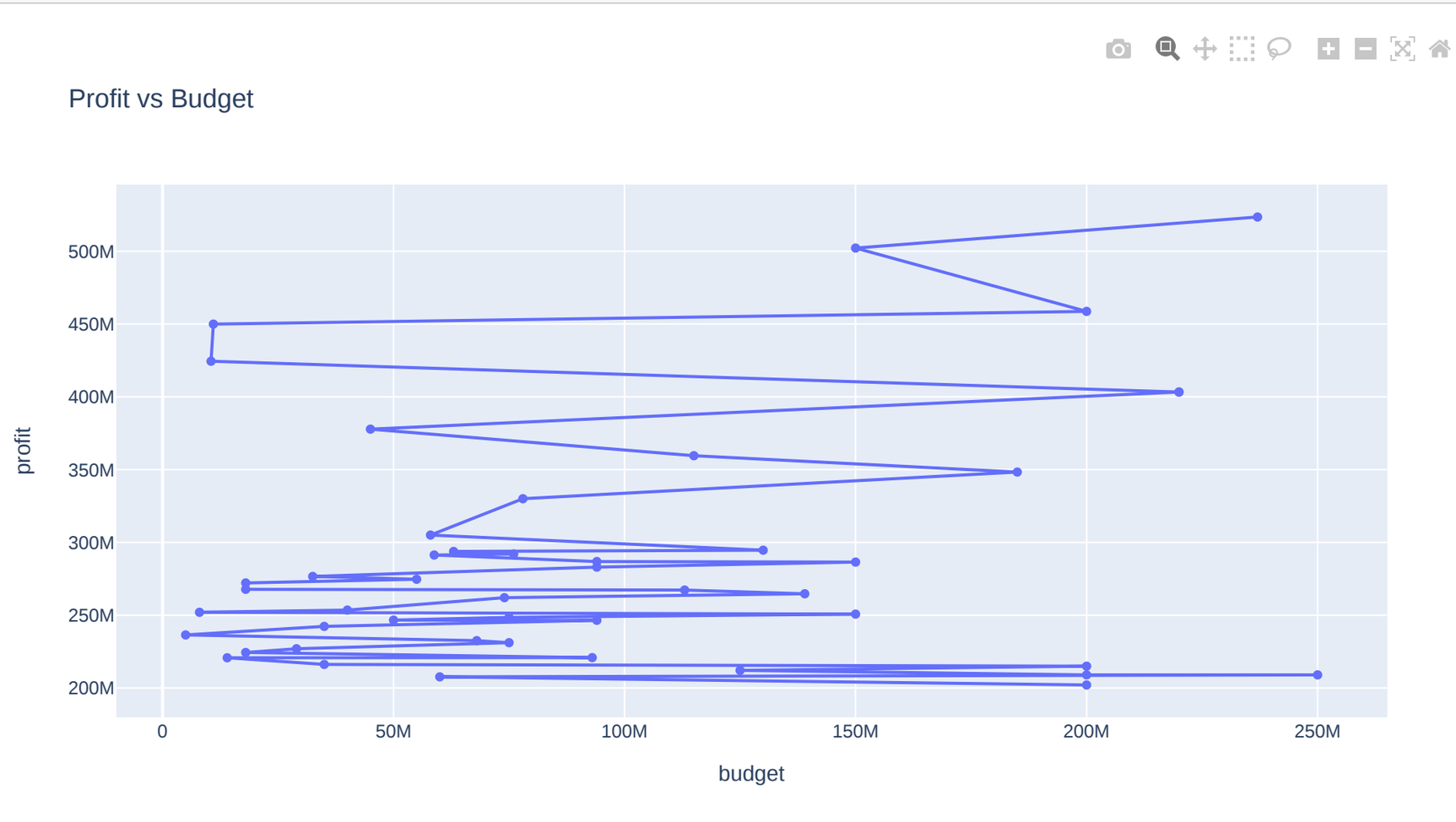
In [10]:
```python
#Creating a new column profit which is difference between gross and budget
df['profit'] = df['gross'] - df['budget']
df.head()
```

Out[10]:

	director_name	num_critic_for_reviews	gross	genres	actor_1_name	movie_title	num_voted_users	num_user_for_reviews	languag
0	James Cameron	723.0	760505847.0	Action|Adventure|Fantasy|Sci-Fi	CCH Pounder	Avatar	886204	3054	Englis
1	Gore Verbinski	302.0	309404152.0	Action|Adventure|Fantasy	Johnny Depp	Pirates of the Caribbean: At World's End	471220	1238	Englis
2	Sam Mendes	602.0	200074175.0	Action|Adventure|Thriller	Christoph Waltz	Spectre	275868	994	Englis
3	Christopher Nolan	813.0	448130642.0	Action|Thriller	Tom Hardy	The Dark Knight Rises	1144337	2701	Englis
5	Andrew Stanton	462.0	73058679.0	Action|Adventure|Sci-Fi	Daryl Sabara	John Carter	212204	738	Englis

In [11]:
```python
#Sorting on basis of profit (Decreasing Order)
df=df.sort_values(by=['profit'],ascending=False)
```

In [12]:
```python
#Plotting Profit Vs Budget of the films having profit greater than 200M
df1 = df[df.profit>200000000]
fig = px.line(df1, x="budget", y="profit", title='Profit vs Budget',markers=True)
fig.show()
```



In [13]:
```python
top10 = df[['movie_title','budget','profit']]
top10.head(10)
```

Out[13]:

	movie_title	budget	profit
0	Avatar	237000000.0	523505847.0
29	Jurassic World	150000000.0	502177271.0
26	Titanic	200000000.0	458672302.0
3024	Star Wars: Episode IV - A New Hope	11000000.0	449935665.0
3080	E.T. the Extra-Terrestrial	10500000.0	424449459.0
794	The Avengers	220000000.0	403279547.0
17	The Avengers	220000000.0	403279547.0
509	The Lion King	45000000.0	377783777.0
240	Star Wars: Episode I - The Phantom Menace	115000000.0	359544677.0
66	The Dark Knight	185000000.0	348316061.0

C.Top 250: Create a new column IMDb_Top_250 and store the top 250 movies with the highest IMDb Rating (corresponding to the column: imdb_score). Also make sure that for all of these movies, the num_voted_users is greater than 25,000. Also add a Rank column containing the values 1 to 250 indicating the ranks of the corresponding films.

Extract all the movies in the IMDb_Top_250 column which are not in the English language and store them in a new column named Top_Foreign_Lang_Film. You can use your own imagination also!

Hint: Find IMDB Top 250

In [14]:
```python
# Creating a dataframe IMDb_Top_250 having voted users greater than 25000
IMDb_Top_250 = df[df['num_voted_users'] > 25000].sort_values(by='imdb_score',ascending=False).head(250)
IMDb_Top_250
```

Out[14]:

	director_name	num_critic_for_reviews	gross	genres	actor_1_name	movie_title	num_voted_users	num_user_for_review
1937	Frank Darabont	199.0	28341469.0	Crime|Drama	Morgan Freeman	The Shawshank Redemption	1689764	414
3466	Francis Ford Coppola	208.0	134821952.0	Crime|Drama	Al Pacino	The Godfather	1155770	223
66	Christopher Nolan	645.0	533316061.0	Action|Crime|Drama|Thriller	Christian Bale	The Dark Knight	1676169	466
2837	Francis Ford Coppola	149.0	57300000.0	Crime|Drama	Robert De Niro	The Godfather: Part II	790926	65
1874	Steven Spielberg	174.0	96067179.0	Biography|Drama|History	Liam Neeson	Schindler's List	865020	127
...	...	...	...	...	...	...	...	
1171	Yimou Zhang	283.0	84961.0	Action|Adventure|History	Jet Li	Hero	149414	84
1748	F. Gary Gray	349.0	161029270.0	Biography|Crime|Drama|History|Music	Aldis Hodge	Straight Outta Compton	119928	33
788	Cameron Crowe	149.0	32522352.0	Adventure|Comedy|Drama|Music	Philip Seymour Hoffman	Almost Famous	207287	82
639	Michael Mann	209.0	28965197.0	Biography|Drama|Thriller	Al Pacino	The Insider	133526	52
525	Alfonso Cuarón	372.0	35286428.0	Drama|Sci-Fi|Thriller	Charlie Hunnam	Children of Men	361767	120

250 rows × 14 columns

In [15]:
```python
# Adding a new column Rank in IMDb_Top_250 as per IMDB Score
IMDb_Top_250['Rank'] = IMDb_Top_250['imdb_score'].rank(method='first',ascending=False)
IMDb_Top_250
```

Out[15]:

	director_name	num_critic_for_reviews	gross	genres	actor_1_name	movie_title	num_voted_users	num_user_for_review
1937	Frank Darabont	199.0	28341469.0	Crime|Drama	Morgan Freeman	The Shawshank Redemption	1689764	414
3466	Francis Ford Coppola	208.0	134821952.0	Crime|Drama	Al Pacino	The Godfather	1155770	223
66	Christopher Nolan	645.0	533316061.0	Action|Crime|Drama|Thriller	Christian Bale	The Dark Knight	1676169	466
2837	Francis Ford Coppola	149.0	57300000.0	Crime|Drama	Robert De Niro	The Godfather: Part II	790926	65
1874	Steven Spielberg	174.0	96067179.0	Biography|Drama|History	Liam Neeson	Schindler's List	865020	127
...	...	...	...	...	...	...	...	
1171	Yimou Zhang	283.0	84961.0	Action|Adventure|History	Jet Li	Hero	149414	84
1748	F. Gary Gray	349.0	161029270.0	Biography|Crime|Drama|History|Music	Aldis Hodge	Straight Outta Compton	119928	33
788	Cameron Crowe	149.0	32522352.0	Adventure|Comedy|Drama|Music	Philip Seymour Hoffman	Almost Famous	207287	82
639	Michael Mann	209.0	28965197.0	Biography|Drama|Thriller	Al Pacino	The Insider	133526	52
525	Alfonso Cuarón	372.0	35286428.0	Drama|Sci-Fi|Thriller	Charlie Hunnam	Children of Men	361767	120

250 rows × 15 columns

In [16]:
```python
# Creating a datafram named Top_Foreign_Lang_Film containing all non english films in top 250
Top_Foreign_Lang_Film = IMDb_Top_250[IMDb_Top_250['language']!='English']
Top_Foreign_Lang_Film
```

Out[16]:

	director_name	num_critic_for_reviews	gross	genres	actor_1_name	movie_title	num_voted_users
4498	Sergio Leone	181.0	6100000.0	Western	Clint Eastwood	The Good, the Bad and the Ugly	503509
4029	Fernando Meirelles	214.0	7563397.0	Crime|Drama	Alice Braga	City of God	533200
4747	Akira Kurosawa	153.0	269061.0	Action|Adventure|Drama	Takashi Shimura	Seven Samurai	229012
2373	Hayao Miyazaki	246.0	10049886.0	Adventure|Animation|Family|Fantasy	Bunta Sugawara	Spirited Away	417971
4259	Florian Henckel von Donnersmarck	215.0	11284657.0	Drama|Thriller	Sebastian Koch	The Lives of Others	259379
4921	Majid Majidi	46.0	925402.0	Drama|Family	Bahare Seddiqi	Children of Heaven	27882
4105	Chan-wook Park	305.0	2181290.0	Drama|Mystery|Thriller	Min-sik Choi	Oldboy	356181
1298	Jean-Pierre Jeunet	242.0	33201661.0	Comedy|Romance	Mathieu Kassovitz	Amélie	534262
2970	Wolfgang Petersen	96.0	11433134.0	Adventure|Drama|Thriller|War	Jürgen Prochnow	Das Boot	168203
4659	Asghar Farhadi	354.0	7098492.0	Drama|Mystery	Shahab Hosseini	A Separation	151812
1329	S.S. Rajamouli	44.0	6498000.0	Action|Adventure|Drama|Fantasy|War	Tamannaah Bhatia	Baahubali: The Beginning	62756
2323	Hayao Miyazaki	174.0	2298191.0	Adventure|Animation|Fantasy	Minnie Driver	Princess Mononoke	221552
4033	Thomas Vinterberg	349.0	610968.0	Drama	Thomas Bo Larsen	The Hunt	170155
2734	Fritz Lang	260.0	26435.0	Drama|Sci-Fi	Brigitte Helm	Metropolis	111841
2829	Oliver Hirschbiegel	192.0	5501940.0	Biography|Drama|History|War	Thomas Kretschmann	Downfall	248354
2551	Guillermo del Toro	406.0	37623143.0	Drama|Fantasy|War	Ivana Baquero	Pan's Labyrinth	467234
3550	Denis Villeneuve	226.0	6857096.0	Drama|Mystery|War	Lubna Azabal	Incendies	80429
4000	Juan José Campanella	262.0	20167424.0	Drama|Mystery|Thriller	Ricardo Darín	The Secret in Their Eyes	131831
2047	Hayao Miyazaki	212.0	4710455.0	Adventure|Animation|Family|Fantasy	Christian Bale	Howl's Moving Castle	214091
2914	Je-kyu Kang	86.0	1110186.0	Action|Drama|War	Min-sik Choi	Tae Guk Gi: The Brotherhood of War	31943
3553	José Padilha	142.0	8060.0	Action|Crime|Drama|Thriller	Wagner Moura	Elite Squad	81644
2830	Alejandro Amenábar	157.0	2086345.0	Biography|Drama|Romance	Belén Rueda	The Sea Inside	64556
4267	Alejandro G. Iñárritu	157.0	5383834.0	Drama|Thriller	Adriana Barraza	Amores Perros	173551
4461	Thomas Vinterberg	98.0	1647780.0	Drama	Ulrich Thomsen	The Celebration	65951
3423	Katsuhiro Ôtomo	150.0	439162.0	Action|Animation|Sci-Fi	Mitsuo Iwata	Akira	106160
4897	Sergio Leone	122.0	3500000.0	Action|Drama|Western	Clint Eastwood	A Fistful of Dollars	147566
3344	Karan Johar	210.0	4018695.0	Adventure|Drama|Thriller	Shah Rukh Khan	My Name Is Khan	69759
4144	Walter Salles	71.0	5595428.0	Drama	Fernanda Montenegro	Central Station	28951
3456	Vincent Paronnaud	242.0	4443403.0	Animation|Biography|Drama|War	Catherine Deneuve	Persepolis	70194
4284	Ari Folman	231.0	2283276.0	Animation|Biography|Documentary|Drama|History|War	Ari Folman	Waltz with Bashir	46107
3264	Michael Haneke	447.0	225377.0	Drama|Romance	Isabelle Huppert	Amour	70382
4415	Fabián Bielinsky	94.0	1221261.0	Crime|Drama|Thriller	Ricardo Darín	Nine Queens	38215
3677	Christophe Barratier	112.0	3629758.0	Drama|Music	Jean-Baptiste Maunier	The Chorus	44151
4640	Cristian Mungiu	233.0	1185783.0	Drama	Anamaria Marinca	4 Months, 3 Weeks and 2 Days	44763
3510	Yash Chopra	29.0	2921738.0	Drama|Musical|Romance	Shah Rukh Khan	Veer-Zaara	34449
2863	Clint Eastwood	251.0	13753931.0	Drama|History|War	Yuki Matsuzaki	Letters from Iwo Jima	132149
2493	Yimou Zhang	283.0	84961.0	Action|Adventure|History	Jet Li	Hero	149414
1171	Yimou Zhang	283.0	84961.0	Action|Adventure|History	Jet Li	Hero	149414

D.Best Directors: TGroup the column using the director_name column.

Find out the top 10 directors for whom the mean of imdb_score is the highest and store them in a new column top10director. In case of a tie in IMDb score between two directors, sort them alphabetically.

Hint: Find the best directors

In [17]:
```python
# Creating a dataframe top10director by using mean on IMDB Score and grouping by director name
top10director = df.groupby('director_name').imdb_score.mean().sort_values(ascending=False).head(10)
top10director
```

Out[17]:
```
director_name
Charles Chaplin          8.600000
Tony Kaye                8.600000
Damien Chazelle          8.500000
Ron Fricke               8.500000
Alfred Hitchcock         8.500000
Majid Majidi             8.500000
Sergio Leone             8.433333
Christopher Nolan        8.425000
Marius A. Markevicius    8.400000
S.S. Rajamouli           8.400000
Name: imdb_score, dtype: float64
```

E.Popular Genres: Perform this step using the knowledge gained while performing previous steps.
Your task: Find popular genres

In [18]:
```python
# Creating a dataframe popular_genres on the same logic as above
popular_genres = df.groupby('genres').imdb_score.mean().sort_values(ascending=False).head(10)
popular_genres
```

Out[18]:
```
genres
Crime|Drama|Fantasy|Mystery                         8.50
Adventure|Animation|Drama|Family|Musical            8.50
Action|Adventure|Drama|Fantasy|War                  8.40
Adventure|Animation|Fantasy                         8.40
Adventure|Drama|Thriller|War                        8.40
Biography|Drama|History|Music                       8.30
Documentary|War                                     8.30
Documentary|Drama|Sport                             8.30
Adventure|Animation|Comedy|Drama|Family|Fantasy     8.30
Adventure|Drama|War                                 8.25
Name: imdb_score, dtype: float64
```

F.Charts: Create three new columns namely, Meryl_Streep, Leo_Caprio, and Brad_Pitt which contain the movies in which the actors: 'Meryl Streep', 'Leonardo DiCaprio', and 'Brad Pitt' are the lead actors. Use only the actor_1_name column for extraction. Also, make sure that you use the names 'Meryl Streep', 'Leonardo DiCaprio', and 'Brad Pitt' for the said extraction.

Append the rows of all these columns and store them in a new column named Combined.

Group the combined column using the actor_1_name column.

Find the mean of the num_critic_for_reviews and num_users_for_review and identify the actors which have the highest mean.

Observe the change in number of voted users over decades using a bar chart. Create a column called decade which represents the decade to which every movie belongs to. For example, the title_year year 1923, 1925 should be stored as 1920s. Sort the column based on the column decade, group it by decade and find the sum of users voted in each decade. Store this in a new data frame called df_by_decade.

Hint: Find the critic-favorite and audience-favorite actors

In [19]:
```python
#Filtering the records as per the 3 actor names and then storing it in combined
Meryl_Streep = df[df['actor_1_name']=='Meryl Streep']
Leo_Caprio = df[df['actor_1_name']=='Leonardo DiCaprio']
Brad_Pitt = df[df['actor_1_name']=='Brad Pitt']
combined = Meryl_Streep.append([Leo_Caprio,Brad_Pitt])
combined
```

```
C:\Users\abhij\AppData\Local\Temp\ipykernel_14236\3565014071.py:5: FutureWarning:

The frame.append method is deprecated and will be removed from pandas in a future version. Use pandas.concat instead.
```

Out[19]:

	director_name	num_critic_for_reviews	gross	genres	actor_1_name	movie_title	num_voted_users
1408	David Frankel	208.0	124732962.0	Comedy|Drama|Romance	Meryl Streep	The Devil Wears Prada	286178
1575	Sydney Pollack	66.0	87100000.0	Biography|Drama|Romance	Meryl Streep	Out of Africa	52339
1204	Nora Ephron	252.0	94125426.0	Biography|Drama|Romance	Meryl Streep	Julie & Julia	79264
1618	David Frankel	234.0	63536011.0	Comedy|Drama|Romance	Meryl Streep	Hope Springs	34258
410	Nancy Meyers	187.0	112703470.0	Comedy|Drama|Romance	Meryl Streep	It's Complicated	69860
2781	Phyllida Lloyd	331.0	29959436.0	Biography|Drama|History	Meryl Streep	The Iron Lady	82327
1925	Stephen Daldry	174.0	41597830.0	Drama|Romance	Meryl Streep	The Hours	102123
3135	Robert Altman	211.0	20338609.0	Comedy|Drama|Music	Meryl Streep	A Prairie Home Companion	19655
1106	Curtis Hanson	42.0	46815748.0	Action|Adventure|Crime|Thriller	Meryl Streep	The River Wild	32544
1674	Carl Franklin	64.0	23209440.0	Drama	Meryl Streep	One True Thing	9283
1483	Robert Redford	227.0	14998070.0	Drama|Thriller|War	Meryl Streep	Lions for Lambs	41170
26	James Cameron	315.0	658672302.0	Drama|Romance	Leonardo DiCaprio	Titanic	793059
97	Christopher Nolan	642.0	292568851.0	Action|Adventure|Sci-Fi|Thriller	Leonardo DiCaprio	Inception	1468200
911	Steven Spielberg	194.0	164435221.0	Biography|Crime|Drama	Leonardo DiCaprio	Catch Me If You Can	525801
296	Quentin Tarantino	765.0	162804648.0	Drama|Western	Leonardo DiCaprio	Django Unchained	955174
179	Alejandro G. Iñárritu	556.0	183635922.0	Adventure|Drama|Thriller|Western	Leonardo DiCaprio	The Revenant	406020
452	Martin Scorsese	490.0	127968405.0	Mystery|Thriller	Leonardo DiCaprio	Shutter Island	786092
361	Martin Scorsese	352.0	132373442.0	Crime|Drama|Thriller	Leonardo DiCaprio	The Departed	873649
50	Baz Luhrmann	490.0	144812796.0	Drama|Romance	Leonardo DiCaprio	The Great Gatsby	362912
3476	Baz Luhrmann	490.0	144812796.0	Drama|Romance	Leonardo DiCaprio	The Great Gatsby	362933
2757	Baz Luhrmann	106.0	46338728.0	Drama|Romance	Leonardo DiCaprio	Romeo + Juliet	167750
1422	Randall Wallace	83.0	56876365.0	Action|Adventure	Leonardo DiCaprio	The Man in the Iron Mask	125219
308	Martin Scorsese	606.0	116866727.0	Biography|Comedy|Crime|Drama	Leonardo DiCaprio	The Wolf of Wall Street	780588
1453	Clint Eastwood	392.0	37304950.0	Biography|Crime|Drama	Leonardo DiCaprio	J. Edgar	102728
257	Martin Scorsese	267.0	102608827.0	Biography|Drama	Leonardo DiCaprio	The Aviator	264318
2067	Jerry Zaks	45.0	12782508.0	Drama	Leonardo DiCaprio	Marvin's Room	20163
990	Danny Boyle	118.0	39778599.0	Adventure|Drama|Thriller	Leonardo DiCaprio	The Beach	176169
1114	Sam Mendes	323.0	22877808.0	Drama|Romance	Leonardo DiCaprio	Revolutionary Road	152591
1560	Sam Raimi	63.0	18636537.0	Action|Thriller|Western	Leonardo DiCaprio	The Quick and the Dead	69197
326	Martin Scorsese	233.0	77679638.0	Crime|Drama	Leonardo DiCaprio	Gangs of New York	314033
641	Ridley Scott	238.0	39380442.0	Action|Drama|Thriller	Leonardo DiCaprio	Body of Lies	174248
307	Edward Zwick	166.0	57366262.0	Adventure|Drama|Thriller	Leonardo DiCaprio	Blood Diamond	400292
400	Steven Soderbergh	186.0	183405771.0	Crime|Thriller	Brad Pitt	Ocean's Eleven	402645
255	Doug Liman	233.0	186336103.0	Action|Comedy|Crime|Romance|Thriller	Brad Pitt	Mr. & Mrs. Smith	348861
940	Neil Jordan	120.0	105264608.0	Drama|Fantasy|Horror	Brad Pitt	Interview with the Vampire: The Vampire Chroni...	239752
470	David Ayer	406.0	85707116.0	Action|Drama|War	Brad Pitt	Fury	303185
254	Steven Soderbergh	198.0	125531634.0	Crime|Thriller	Brad Pitt	Ocean's Twelve	284852
2204	Alejandro G. Iñárritu	285.0	34300771.0	Drama	Brad Pitt	Babel	243799
2682	Andrew Dominik	414.0	14938570.0	Crime|Thriller	Brad Pitt	Killing Them Softly	111625
2898	Tony Scott	122.0	12281500.0	Action|Crime|Drama|Romance|Thriller	Brad Pitt	True Romance	163492
2333	Angelina Jolie Pitt	131.0	531009.0	Drama|Romance	Brad Pitt	By the Sea	7976
1490	Terrence Malick	584.0	13303319.0	Drama|Fantasy	Brad Pitt	The Tree of Life	136367
101	David Fincher	362.0	127490802.0	Drama|Fantasy|Romance	Brad Pitt	The Curious Case of Benjamin Button	459346
683	David Fincher	315.0	37023395.0	Drama	Brad Pitt	Fight Club	1347461
1722	Andrew Dominik	273.0	3904982.0	Biography|Crime|Drama|History|Western	Brad Pitt	The Assassination of Jesse James by the Coward...	136104
611	Jean-Jacques Annaud	76.0	37901509.0	Adventure|Biography|Drama|History|War	Brad Pitt	Seven Years in Tibet	96385
792	Patrick Gilmore	98.0	26288320.0	Adventure|Animation|Comedy|Drama|Family|Fantas...	Brad Pitt	Sinbad: Legend of the Seven Seas	36144
147	Wolfgang Petersen	220.0	133228348.0	Adventure	Brad Pitt	Troy	381672
382	Tony Scott	142.0	26871.0	Action|Crime|Thriller	Brad Pitt	Spy Game	121259

In [20]:
```python
#Mean of critic reviews
combined.groupby('actor_1_name').num_critic_for_reviews.mean()
```

Out[20]:
```
actor_1_name
Brad Pitt            245.000000
Leonardo DiCaprio    330.190476
Meryl Streep         181.454545
Name: num_critic_for_reviews, dtype: float64
```

In [21]:
```python
#Mean of user review
combined.num_user_for_reviews = combined.num_user_for_reviews.astype('int')
combined.groupby('actor_1_name').num_user_for_reviews.mean()
```

Out[21]:
```
actor_1_name
Brad Pitt            742.352941
Leonardo DiCaprio    914.476190
Meryl Streep         297.181818
Name: num_user_for_reviews, dtype: float64
```

In [22]:
```python
#Copying all records to df_by_decade and applying lambda function to convert years into decades, finally adding s at the end
df_by_decade=df.copy(deep=True)
df_by_decade['decade']=df_by_decade['title_year'].apply(lambda x:10*(int(x/10)))
df_by_decade['decade'] = df_by_decade['decade'].astype(str)+'s'
```

In [23]:
```python
#Sorting on decade
df_by_decade=df_by_decade.groupby('decade',as_index=False)['num_voted_users'].sum().sort_values(by="decade")
```

In [24]:
```python
#Plotting the bar graph
sns.barplot(x='decade',y='num_voted_users',data=df_by_decade)
plt.yscale('log')
plt.ylabel("num_voted_users")
plt.xlabel("decade")
plt.show()
```