READ ME

Data Attributes

Attribute	Description
id	id of film
imdb_id	Engine displacement - the size of an engine in liters
popularity	Each model builds their popularity value slightly differently. For movies: Number of votes for the day, number of views for the day, number of users who marked it as a "favourite" for the day, number of users who added it to their "watchlist" for the day, release date, number of total votes, previous days score
budget	The money spend on production process
revenue	Earned money from the film
original_title	
cast	
homepage	
director	
tagline	
keywords	
overview	
runtime	
genres	
production_companies	
vote_count	Count of Ratings
vote_average	Average Ratings
release_year	
budget_adj	
revenue_adj	

Questions related to this data

- Which movies have the highest revenue or profit?
- Which genres have the highest profit?
- Movie with Highest And Lowest Budget?
- Which movie get the highest or lowest votes (Ratings).
- Is there any relationship between the popularity and the budget?
- Which genres are most popular year by year?

- Which directors directed the most popular movie in the last years?

Import Packages

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

Gathering Data

In [2]:

```
df = pd.read_csv('data/tmdb-movies.csv')
```

Assessing Data

- number of samples in each dataset : *10866*
- number of columns in each dataset : *21*

In [3]:

```
df.shape
```

Out[3]:

```
(10866, 21)
```

- features with missing values : ('imdb_id', 'cast', 'homepage', 'director', 'tagline', 'keywords', 'overview', 'genres', 'production_companies')

In [4]:

```
df.isnull().any()
```

Out[4]:

```
id                      False
imdb_id                  True
popularity              False
budget                  False
revenue                 False
original_title          False
cast                     True
homepage                 True
director                 True
tagline                  True
keywords                 True
overview                 True
runtime                 False
genres                   True
production_companies     True
release_date            False
vote_count              False
vote_average            False
release_year            False
budget_adj              False
revenue_adj             False
dtype: bool
```

In [5]:

```
df.isnull().sum()
```

Out[5]:

```
id                         0
imdb_id                   10
popularity                 0
budget                     0
revenue                    0
original_title             0
cast                      76
homepage                7930
director                  44
tagline                 2824
keywords                1493
overview                   4
runtime                    0
genres                    23
production_companies    1030
release_date               0
vote_count                 0
vote_average               0
release_year               0
budget_adj                 0
revenue_adj                0
dtype: int64
```

- duplicate rows : 1

In [6]:

```
df.duplicated().any()
```

Out[6]:

```
True
```

In [7]:

```
df.duplicated().sum()
```

Out[7]:

```
1
```

- number of non-null unique values for features :

In [8]:

```
df.nunique()
```

Out[8]:

```
id                      10865
imdb_id                 10855
popularity              10814
budget                    557
revenue                  4702
original_title          10571
cast                    10719
homepage                 2896
director                 5067
tagline                  7997
keywords                 8804
overview                10847
runtime                   247
genres                   2039
production_companies     7445
release_date             5909
vote_count               1289
vote_average               72
release_year               56
budget_adj               2614
```

```
revenue_adj                4840
dtype: int64
```

- data types of columns:

In [9]:

```
df.dtypes
```

Out[9]:

```
id                        int64
imdb_id                  object
popularity              float64
budget                    int64
revenue                   int64
original_title           object
cast                     object
homepage                 object
director                 object
tagline                  object
keywords                 object
overview                 object
runtime                   int64
genres                   object
production_companies     object
release_date             object
vote_count                int64
vote_average            float64
release_year              int64
budget_adj              float64
revenue_adj             float64
dtype: object
```

In [10]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10866 entries, 0 to 10865
Data columns (total 21 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   id                    10866 non-null  int64  
 1   imdb_id               10856 non-null  object 
 2   popularity            10866 non-null  float64
 3   budget                10866 non-null  int64  
 4   revenue               10866 non-null  int64  
 5   original_title        10866 non-null  object 
 6   cast                  10790 non-null  object 
 7   homepage              2936 non-null   object 
 8   director              10822 non-null  object 
 9   tagline               8042 non-null   object 
 10  keywords              9373 non-null   object 
 11  overview              10862 non-null  object 
 12  runtime               10866 non-null  int64  
 13  genres                10843 non-null  object 
 14  production_companies  9836 non-null   object 
 15  release_date          10866 non-null  object 
 16  vote_count            10866 non-null  int64  
 17  vote_average          10866 non-null  float64
 18  release_year          10866 non-null  int64  
 19  budget_adj            10866 non-null  float64
 20  revenue_adj           10866 non-null  float64
dtypes: float64(4), int64(6), object(11)
memory usage: 1.7+ MB
```

In [11]:

```
df.original_title
```

Out[11]:

```
0                             Jurassic World
1                         Mad Max: Fury Road
2                                  Insurgent
3               Star Wars: The Force Awakens
4                                  Furious 7
                          ...
10861                     The Endless Summer
10862                             Grand Prix
10863                    Beregis Avtomobilya
10864                 What's Up, Tiger Lily?
10865               Manos: The Hands of Fate
Name: original_title, Length: 10866, dtype: object
```

> Relase years : 1960-2015

In [12]:

```
df.release_year.unique()
```

Out[12]:

```
array([2015, 2014, 1977, 2009, 2010, 1999, 2001, 2008, 2011, 2002, 1994,
       2012, 2003, 1997, 2013, 1985, 2005, 2006, 2004, 1972, 1980, 2007,
       1979, 1984, 1983, 1995, 1992, 1981, 1996, 2000, 1982, 1998, 1989,
       1991, 1988, 1987, 1968, 1974, 1975, 1962, 1964, 1971, 1990, 1961,
       1960, 1976, 1993, 1967, 1963, 1986, 1973, 1970, 1965, 1969, 1978,
       1966])
```

> `profit` is calculated by revenue - budget.

In [13]:

```
df['profit']= df['revenue'] - df['budget']
```

In [14]:

```
df['profit']
```

Out[14]:

```
0        1363528810
1         228436354
2         185238201
3        1868178225
4        1316249360
            ...
10861             0
10862             0
10863             0
10864             0
10865        -19000
Name: profit, Length: 10866, dtype: int64
```

> Checking for the zeros in budget and revenue to prevent inappropriate results.

In [15]:

```
df[df['budget']==0].shape[0]
```

Out[15]:

```
5696
```

In [16]:

```
df[df['revenue']==0].shape[0]
```

Out[16]:

```
6016
```

In [17]:

```
df[df['budget']!=0] #I will use not 0s.
```

Out[17]:

	id	imdb_id	popularity	budget	revenue	original_title	
0	135397	tt0369610	32.985763	150000000	1513528810	Jurassic World	Chris Pratt|B Dallas Howard|I Khar
1	76341	tt1392190	28.419936	150000000	378436354	Mad Max: Fury Road	Tom Hardy|Cha Theron|Hugh Ke Byrne|
2	262500	tt2908446	13.112507	110000000	295238201	Insurgent	Sha Woodley| James Winslet|Ar
3	140607	tt2488496	11.173104	200000000	2068178225	Star Wars: The Force Awakens	Harrison Ford| Hamill|C Fisher|Adar
4	168259	tt2820852	9.335014	190000000	1506249360	Furious 7	Vin Diesel Walker|J Statham|Miche
...	...	...	...	...	...	...	
10835	5923	tt0060934	0.299911	12000000	20000000	The Sand Pebbles	McQueen|Ric Attenborough|Ric
10841	42701	tt0062262	0.264925	75000	0	The Shooting	Will Hutchins| Perkins Nicholson|
10848	2161	tt0060397	0.207257	5115000	12000000	Fantastic Voyage	Stephen Boyd|Ra Welch|Edr O'Brien|Do
10855	13343	tt0059221	0.141026	700000	0	The Ghost & Mr. Chicken	Don Knotts| Staley Redmond Sa
10865	22293	tt0060666	0.035919	19000	0	Manos: The Hands of Fate	Harold P. Warren Neyman| Reynolds|D

5170 rows × 22 columns

In [18]:

```
df[df['revenue']!=0] #I will use not 0s.
```

Out[18]:

	id	imdb_id	popularity	budget	revenue	original_title	
0	135397	tt0369610	32.985763	150000000	1513528810	Jurassic World	Chris Pratt|B Dallas Howard|I Khar
1	76341	tt1392190	28.419936	150000000	378436354	Mad Max: Fury Road	Tom Hardy|Cha Theron|Hugh Ke Byrne|

	id	imdb_id	popularity	budget	revenue	original_title	
2	262500	tt2908446	13.112507	110000000	295238201	Insurgent	Sha Woodley| James Winslet|Ar
3	140607	tt2488496	11.173104	200000000	2068178225	Star Wars: The Force Awakens	Harrison Ford| Hamill|C Fisher|Adar
4	168259	tt2820852	9.335014	190000000	1506249360	Furious 7	Vin Diesel Walker|J Statham|Miche
...	...	...	...	...	...	...	
10822	396	tt0061184	0.670274	7500000	33736689	Who's Afraid of Virginia Woolf?	Eliza Taylor|Ric Burton|Ge Sega
10828	5780	tt0061107	0.402730	3000000	13000000	Torn Curtain	Paul Newman Andrew Kedrova|Hans
10829	6644	tt0061619	0.395668	4653000	6000000	El Dorado	John Wayne|R Mitchum|J Caan|Charle
10835	5923	tt0060934	0.299911	12000000	20000000	The Sand Pebbles	S McQueen|Ric Attenborough|Ric (
10848	2161	tt0060397	0.207257	5115000	12000000	Fantastic Voyage	Stephen Boyd|Ra Welch|Edr O'Brien|Do

4850 rows × 22 columns

Cleaning Column Labels

I will drop the columns I do not need for this analysis which are: (`imdb_id` , `homepage` , `tagline` , `keywords` , `overview` , `runtime` , `production_companies` , `release_date` , `budget_adj` , `revenue_adj`)

In [19]:

```
df.drop(['imdb_id','homepage','tagline','keywords','overview','runtime', 'pro
```

In [20]:

```
df.head()
```

Out[20]:

	id	popularity	budget	revenue	original_title	cast	director	
0	135397	32.985763	150000000	1513528810	Jurassic World	Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi...	Colin Trevorrow	Ac

	id	popularity	budget	revenue	original_title	cast	director	
1	76341	28.419936	150000000	378436354	Mad Max: Fury Road	Tom Hardy|Charlize Theron|Hugh Keays-Byrne|Nic...	George Miller	Ac
2	262500	13.112507	110000000	295238201	Insurgent	Shailene Woodley|Theo James|Kate Winslet|Ansel...	Robert Schwentke	
3	140607	11.173104	200000000	2068178225	Star Wars: The Force Awakens	Harrison Ford|Mark Hamill|Carrie Fisher|Adam D...	J.J. Abrams	Ac
4	168259	9.335014	190000000	1506249360	Furious 7	Vin Diesel|Paul Walker|Jason Statham|Michelle ...	James Wan	

Data Cleaning

Replace Zero Values with Null Values for Budget & Revenue

In [21]:

```
df['budget'] = df['budget'].replace(0, np.NaN)
df['revenue'] = df['revenue'].replace(0, np.NaN)
```

In [22]:

```
df[df['budget']==0].shape[0] #Checking
```

Out[22]:

```
0
```

In [23]:

```
df[df['revenue']==0].shape[0] #Checking
```

Out[23]:

```
0
```

Drop Nulls

In [24]:

```
df = df.dropna(subset=['cast', 'director', 'genres', 'budget', 'revenue'])
```

In [25]:

```
df.isnull().sum().any()
```

Out[25]:

```
False
```

Drop Duplicates

df = df.drop_duplicates()

In [26]:

```
df.duplicated().sum().any()
```

Out[26]:

```
True
```

EDA with Visuals

Q : Is there any relationship between the popularity and the budget?

In [27]:

```
correlation = df["popularity"].corr(df["budget"])

correlation
```

Out[27]:

```
0.4465702124386731
```

A: The correlation between `popularity` and `budget` is 0.4465. It is not close to 1 enough to be in a strong relation. I assume that there is no significant relation between them.

Which movies have the highest profit of all the time?

In [28]:

```
sns.set_theme(style="whitegrid")
```

In [35]:

```
df.columns
```

Out[35]:

```
Index(['id', 'popularity', 'budget', 'revenue', 'original_title', 'cast',
       'director', 'genres', 'vote_count', 'vote_average', 'release_year',
       'profit'],
      dtype='object')
```

In [34]:

```
type(df['profit'])
```

Out[34]:

```
pandas.core.series.Series
```

In [64]:

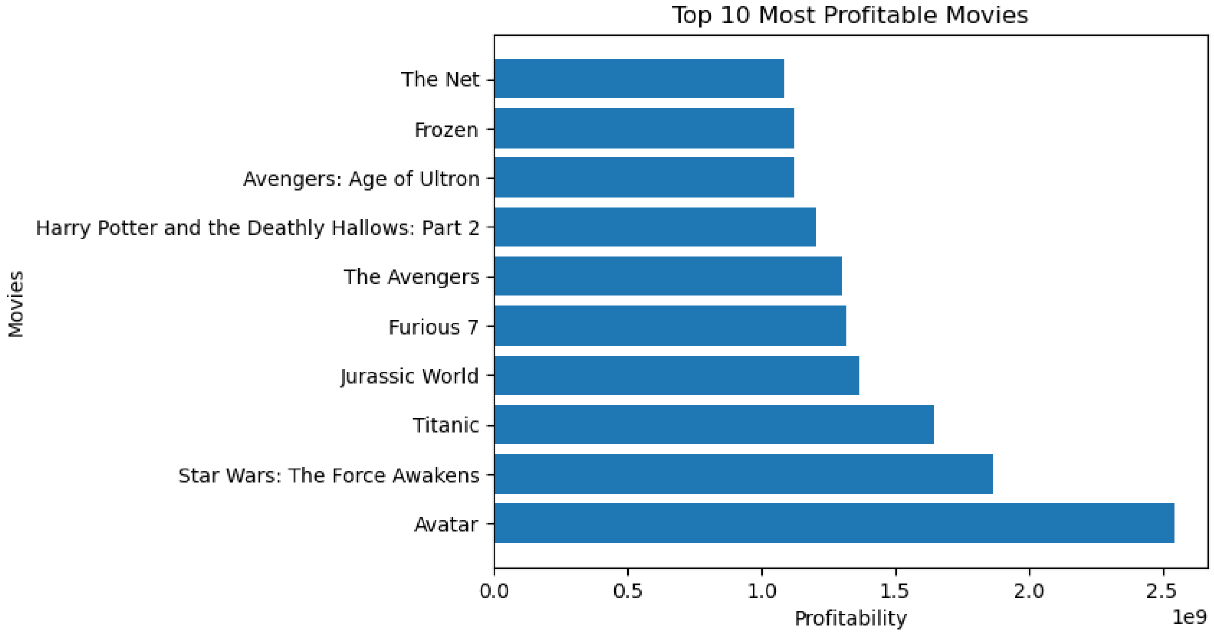
```
df = df.sort_values(by=['profit'], ascending=False)
```

In [65]:

```
x = list(df['original_title'].head(10))
y = list(df['profit'].head(10))
```

In [68]:

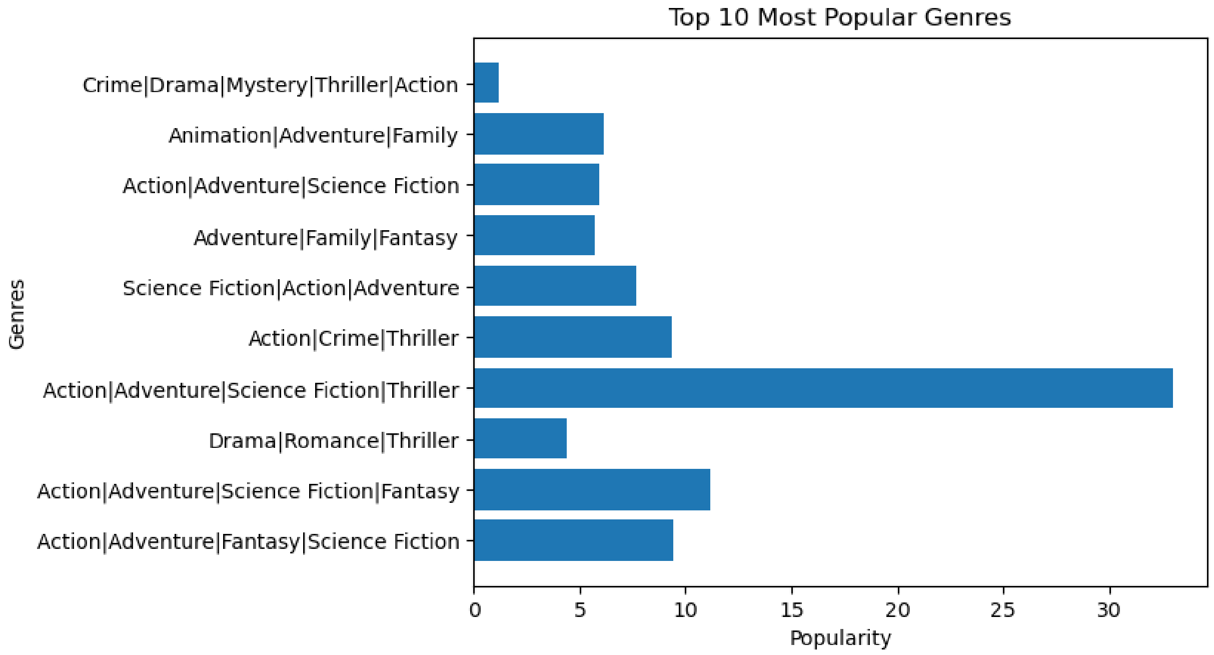
```
plt.barh(x,y)
plt.title('Top 10 Most Profitable Movies')
plt.ylabel('Movies')
plt.xlabel('Profitability')
plt.show()
```



Which genres have the highest profit?

In [69]:

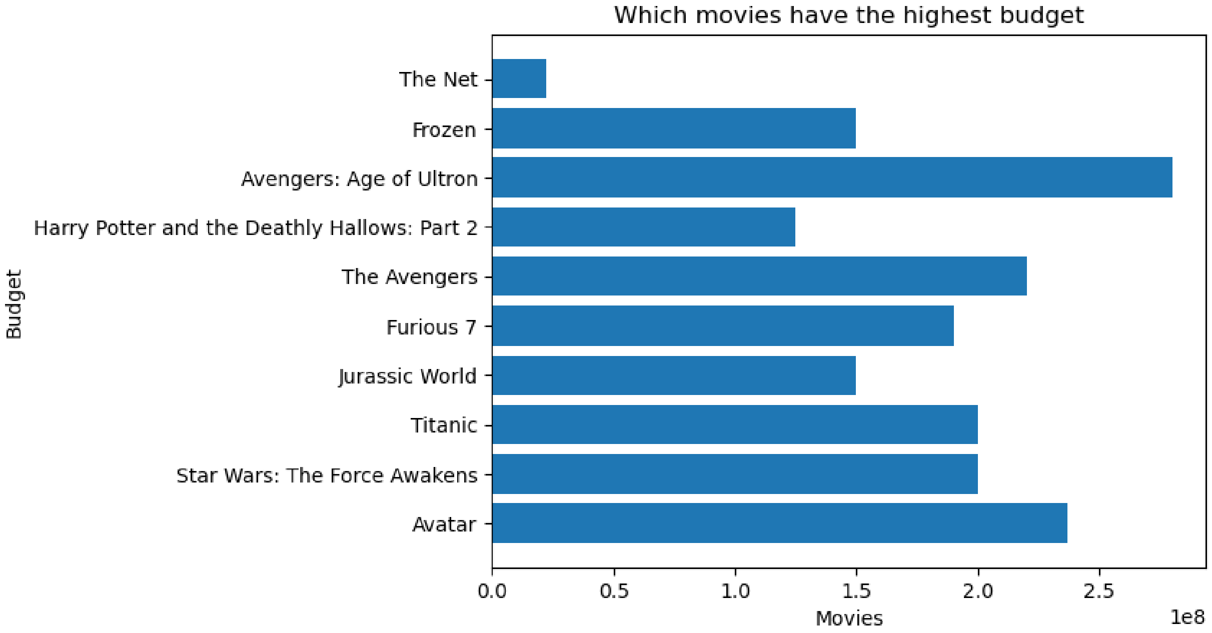
```
x = list(df['genres'].head(10))
y = list(df['popularity'].head(10))
plt.barh(x,y)
plt.title('Top 10 Most Popular Genres')
plt.ylabel('Genres')
plt.xlabel('Popularity')
plt.show()
```



Movie with Highest And Lowest Budget?

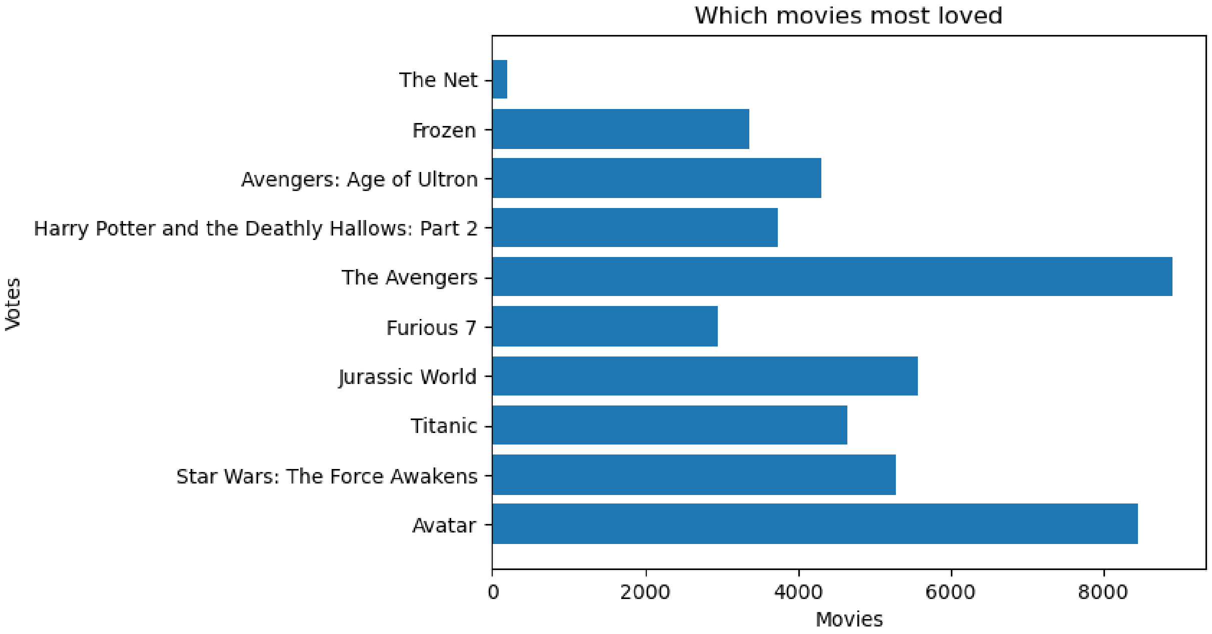
In [70]:

```
x = list(df['original_title'].head(10))
y = list(df['budget'].head(10))
plt.barh(x,y)
plt.title('Which movies have the highest budget')
plt.ylabel('Budget')
plt.xlabel('Movies')
plt.show()
```



Which movie get the highest or lowest votes (Ratings).

```
In [71]: x = list(df['original_title'].head(10))
         y = list(df['vote_count'].head(10))
         plt.barh(x,y)
         plt.title('Which movies most loved')
         plt.ylabel('Votes')
         plt.xlabel('Movies')
         plt.show()
```


Which movies most loved
Votes
The Net
Frozen
Avengers: Age of Ultron
Harry Potter and the Deathly Hallows: Part 2
The Avengers
Furious 7
Jurassic World
Titanic
Star Wars: The Force Awakens
Avatar
Movies