Project: Investigate a Dataset - [TMDB 5000 Movie data]

Table of Contents

Introduction

Dataset Description

This dataset was taken from the The movie database (TMDb) (https://www.kaggle.com/datasets/tmdb/themoviedb.org), containing information about 10000 movies released between the years 2015 and 1960.

Number of Rows: 10866

Number of columns: 20

Columns:

- **id** : Identification number
- **imdb_id** : IMDb identification number
- **popularity** : Popularity of the movie in numbers
- **budget** : Budget of the movie
- **revenue** : Revenue of the movie
- **original_title** : Title of the movie
- **cast** : List of actors
- **homepage** : Website homepage of the movie
- **director** : Director of the movie
- **tagline** : The film's advertising slogan
- **keywords** : Keywords related to the movie
- **overview** : Overview of the movie's plot
- **runtime** : Screentime/How long is the movie
- **genres** : Genre of the movie (Action, Comedy, Drama, Romance...)
- **production_companies** : Companies that produced the movie
- **release_date** : Release date of the movie
- **vote_count** : Number of people voting for the movie on IMDb
- **vote_average** : Average vote of people voting for the movie
- **release_year** : Release year of the movie
- **budget_adj** : Budget of the movie with inflation from 2010 in dollars
- **revenue_adj** : Revenue of the movie with inflation from 2010 in dollars

Question(s) for Analysis

After taking a look at the dataset, there are multiple questions that popped out in my mind regarding some feature and the relationships between them.

Question 1: Which genres are the most popular from year to year ?

Question 2: Which movies have the most profit ? and in which year they were released ?

Question 3: What kind of characteristics are associated with movies having high revenues ?

Question 4: Which movies have the highest budgets but are low on vote counts ?

Question 5: Which movies have the lowest budgets but are very high on vote counts ?

Question 6: what are the movies that flopped and topped the most in terms of profit and votes ? and who was their cast/director ?

Question 7: Is the screen runtime related to the success or flopping of the movies ?

Question 8: What is the runtime of the most successful movies between the years 2015 and 1960 ?

Question 9: Which production companies that released the most successful movies each year ?

Question 10: Do movies with more profit have more popularity as well?

Question 11: What is the relationship between Budget and other features in the dataset ?

Question 12: Which cast participated in the most successful/flopped movies ?

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

%matplotlib inline
#to make visualisations be plotted inline within the notebook
```

In [2]:
```
# Upgrade pandas to use dataframe.explode() function.
#!pip install --upgrade pandas==0.25.0
```

Data Wrangling

In [3]:
```
# Load your data and print out a few lines. Perform operations to inspect data
df = pd.read_csv('tmdb-movies.csv', index_col='id')
df.head()
```

Out[3]:

id	imdb_id	popularity	budget	revenue	original_title	cast	
135397	tt0369610	32.985763	150000000	1513528810	Jurassic World	Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi...	http://ww
76341	tt1392190	28.419936	150000000	378436354	Mad Max: Fury Road	Tom Hardy|Charlize Theron|Hugh Keays-Byrne|Nic...	http://www.
262500	tt2908446	13.112507	110000000	295238201	Insurgent	Shailene Woodley|Theo James|Kate Winslet|Ansel...	http://www.thedivergentseri
140607	tt2488496	11.173104	200000000	2068178225	Star Wars: The Force Awakens	Harrison Ford|Mark Hamill|Carrie Fisher|Adam D...	http://www.starwars.
168259	tt2820852	9.335014	190000000	1506249360	Furious 7	Vin Diesel|Paul Walker|Jason Statham|Michelle ...	http:

In [4]:
```
df.tail()
```

Out[4]:

	imdb_id	popularity	budget	revenue	original_title	cast	homepage	director
id								
21	tt0060371	0.080598	0	0	The Endless Summer	Michael Hynson|Robert August|Lord 'Tally Ho' B...	NaN	Bruce Brown
20379	tt0060472	0.065543	0	0	Grand Prix	James Garner|Eva Marie Saint|Yves Montand|Tosh...	NaN	John Frankenheimer
39768	tt0060161	0.065141	0	0	Beregis Avtomobilya	Innokentiy Smoktunovskiy|Oleg Efremov|Georgi Z...	NaN	Eldar Ryazanov
21449	tt0061177	0.064317	0	0	What's Up, Tiger Lily?	Tatsuya Mihashi|Akiko Wakabayashi|Mie Hama|Joh...	NaN	Woody Allen
22293	tt0060666	0.035919	19000	0	Manos: The Hands of Fate	Harold P. Warren|Tom Neyman|John Reynolds|Dian...	NaN	Harold P. Warren

Ir

We can notice that there are some columns having values separated by '|'

In [5]:
```
list_1 = ['cast', 'director', 'keywords', 'genres', 'production_companies']
```

Some information about the data:

In [6]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 10866 entries, 135397 to 22293
Data columns (total 20 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   imdb_id               10856 non-null  object 
 1   popularity            10866 non-null  float64
 2   budget                10866 non-null  int64  
 3   revenue               10866 non-null  int64  
 4   original_title        10866 non-null  object 
 5   cast                  10790 non-null  object 
 6   homepage              2936 non-null   object 
 7   director              10822 non-null  object 
 8   tagline               8042 non-null   object 
 9   keywords              9373 non-null   object 
 10  overview              10862 non-null  object 
 11  runtime               10866 non-null  int64  
 12  genres                10843 non-null  object 
 13  production_companies  9836 non-null   object 
 14  release_date          10866 non-null  object 
 15  vote_count            10866 non-null  int64  
 16  vote_average          10866 non-null  float64
 17  release_year          10866 non-null  int64  
 18  budget_adj            10866 non-null  float64
 19  revenue_adj           10866 non-null  float64
dtypes: float64(4), int64(5), object(11)
memory usage: 1.7+ MB
```

In [7]:
```
df.shape
```

Out[7]: (10866, 20)

The dataset has 10866 rows and 20 columns

In [8]:
```
df.describe()
```

Out[8]:

	popularity	budget	revenue	runtime	vote_count	vote_average	release_year
count	10866.000000	1.086600e+04	1.086600e+04	10866.000000	10866.000000	10866.000000	10866.000000
mean	0.646441	1.462570e+07	3.982332e+07	102.070863	217.389748	5.974922	2001.322658
std	1.000185	3.091321e+07	1.170035e+08	31.381405	575.619058	0.935142	12.812941
min	0.000065	0.000000e+00	0.000000e+00	0.000000	10.000000	1.500000	1960.000000
25%	0.207583	0.000000e+00	0.000000e+00	90.000000	17.000000	5.400000	1995.000000
50%	0.383856	0.000000e+00	0.000000e+00	99.000000	38.000000	6.000000	2006.000000
75%	0.713817	1.500000e+07	2.400000e+07	111.000000	145.750000	6.600000	2011.000000
max	32.985763	4.250000e+08	2.781506e+09	900.000000	9767.000000	9.200000	2015.000000

We can notice that there are some movies having the budget and revenue equals to 0, these can be considered as NaN values.

In [9]:
```
l = df['original_title'].loc[(df['budget']==0.000000e+00) & (df['revenue']==0.0
00000e+00)].values
#select movie names where both the budget and revenue are equals to 0
l
```

Out[9]:
```
array(['Mythica: The Darkspore', 'Me and Earl and the Dying Girl',
       'Mythica: The Necromancer', ..., 'Grand Prix',
       'Beregis Avtomobilya', "What's Up, Tiger Lily?"], dtype=object)
```

Types of features :

In [10]:
```
df.dtypes
```

Out[10]:
```
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
```

We can notice that the release date is an object and not in a date_time type

Checking for duplicate rows:

In [11]:
```
df.duplicated().value_counts()
```

Out[11]:
```
False    10865
True         1
dtype: int64
```

checking for the duplicated row:

In [12]:
```
df[df.duplicated(keep=False)]
```

Out[12]:

id	imdb_id	popularity	budget	revenue	original_title	cast	homepage	director	tagline	
42194	tt0411951	0.59643	30000000	967000	TEKKEN	Jon Foo|Kelly Overton|Cary-Hiroyuki Tagawa|Ian...	NaN	Dwight H. Little	Survival is no game	a
42194	tt0411951	0.59643	30000000	967000	TEKKEN	Jon Foo|Kelly Overton|Cary-Hiroyuki Tagawa|Ian...	NaN	Dwight H. Little	Survival is no game	a

Number of rows with missing data:

In [13]:
```
df.isnull().values.ravel().sum()
```

Out[13]:
```
13434
```

To check if we have missing values for each feature:

In [14]:
```
df.isnull().sum()
```

Out[14]:
```
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
```

Checking for unique values :

In [15]:
```
df.nunique()
```

Out[15]:
```
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
```

In [16]:
```
df['original_title'].unique() #names of unique movie titles
```

Out[16]:
```
array(['Jurassic World', 'Mad Max: Fury Road', 'Insurgent', ...,
       'Beregis Avtomobilya', "What's Up, Tiger Lily?",
       'Manos: The Hands of Fate'], dtype=object)
```

In [17]:
```
df['original_title'].nunique() #number of unique movie titles
```

Out[17]:
```
10571
```

Data Cleaning

I noticed that there are some columns we might not need in further analysis such as : `imdb_id` since we already have an id column, `homepage` , `tagline` , `overview` , and `keywords` since we're not doing a movie recommendation in this project.

In [18]:
```
df.columns
```

Out[18]:
```
Index(['imdb_id', 'popularity', 'budget', 'revenue', 'original_title', 'cast',
       'homepage', 'director', 'tagline', 'keywords', 'overview', 'runtime',
       'genres', 'production_companies', 'release_date', 'vote_count',
       'vote_average', 'release_year', 'budget_adj', 'revenue_adj'],
      dtype='object')
```

In [19]:
```
list_2=['imdb_id','homepage','tagline', 'keywords', 'overview' ]
df.drop(columns= list_2, inplace=True)
```

In [20]:
```
df.head(1)
```

Out[20]:

	popularity	budget	revenue	original_title	cast	director	runtime	g
id								
135397	32.985763	150000000	1513528810	Jurassic World	Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi...	Colin Trevorrow	124	Action|Adventure|Sc Fiction|T

Dropping duplicates:

In [21]:
```
df.drop_duplicates(inplace=True)
```

checking for duplicates again:

In [22]:
```
df.duplicated().any()
```

Out[22]:
```
False
```

Dropping null values:

In [23]:
```
df.dropna(inplace=True)
```

checking for null values again:

In [24]:
```
df.isna().any()
```

Out[24]:
```
popularity              False
budget                  False
revenue                 False
original_title          False
cast                    False
director                False
runtime                 False
genres                  False
production_companies    False
release_date            False
vote_count              False
vote_average            False
release_year            False
budget_adj              False
revenue_adj             False
dtype: bool
```

I noticed in the first part of this project that the release date of each movie is not in date format but in object format. So, we should change it:

In [25]:
```
df['release_date'].dtype
```
Out[25]:
```
dtype('O')
```
In [26]:
```
release_date = pd.to_datetime(df['release_date'])
```
In [27]:
```
release_date
```
Out[27]:
```
id
135397   2015-06-09
76341    2015-05-13
262500   2015-03-18
140607   2015-12-15
168259   2015-04-01
            ...
21       2066-06-15
20379    2066-12-21
39768    2066-01-01
21449    2066-11-02
22293    2066-11-15
Name: release_date, Length: 9772, dtype: datetime64[ns]
```

I encountered a problem where each release date written for example as '11/15/66' is going to be converted to '2066-11-15' instead of 1966 as noted in the release_year

In [28]:
```
#changing the release_date year part with the release_year values :

df['release_date'] = df.apply(lambda x: x.release_date[:-2] + str(x.release_yea
r), axis=1)
df['release_date'] = pd.to_datetime(df['release_date'])
df['release_date']
```
Out[28]:
```
id
135397   2015-06-09
76341    2015-05-13
262500   2015-03-18
140607   2015-12-15
168259   2015-04-01
            ...
21       1966-06-15
20379    1966-12-21
39768    1966-01-01
21449    1966-11-02
22293    1966-11-15
Name: release_date, Length: 9772, dtype: datetime64[ns]
```

as noted in the first part of the notebook, i noticed that some of the budget and revenue values are zeros:

In [29]:
```python
df.loc[(df['budget']==0.000000e+00) & (df['revenue']==0.000000e+00)]
```

Out[29]:

id	popularity	budget	revenue	original_title	cast	director	runtime	
347096	2.165433	0	0	Mythica: The Darkspore	Melanie Stone|Kevin Sorbo|Adam Johnson|Jake St...	Anne K. Black	108	Action|Adver
308369	2.141506	0	0	Me and Earl and the Dying Girl	Thomas Mann|RJ Cyler|Olivia Cooke|Connie Britt...	Alfonso Gomez-Rejon	105	Co
370687	1.876037	0	0	Mythica: The Necromancer	Melanie Stone|Adam Johnson|Kevin Sorbo|Nicola ...	A. Todd Smith	0	Fantasy|Acti
326359	1.724712	0	0	Frozen Fever	Kristen Bell|Idina Menzel|Jonathan Groff|Josh ...	Chris Buck|Jennifer Lee	8	Adventure|Anin
254302	1.661789	0	0	High-Rise	Tom Hiddleston|Sienna Miller|Jeremy Irons|Luke...	Ben Wheatley	119	Action|Dr
...	...	...	...	...	...	...	...	
5060	0.087034	0	0	Carry On Screaming!	Kenneth Williams|Jim Dale|Harry H. Corbett|Joa...	Gerald Thomas	87	
21	0.080598	0	0	The Endless Summer	Michael Hynson|Robert August|Lord 'Tally Ho' B...	Bruce Brown	95	I
20379	0.065543	0	0	Grand Prix	James Garner|Eva Marie Saint|Yves Montand|Tosh...	John Frankenheimer	176	Action|Adve
39768	0.065141	0	0	Beregis Avtomobilya	Innokentiy Smoktunovskiy|Oleg Efremov|Georgi Z...	Eldar Ryazanov	94	Mys
21449	0.064317	0	0	What's Up, Tiger Lily?	Tatsuya Mihashi|Akiko Wakabayashi|Mie Hama|Joh...	Woody Allen	80	Ac

3806 rows × 15 columns

In [30]:
```python
df.loc[df['revenue']==0.000000e+00]
```

Out[30]:

id	popularity	budget	revenue	original_title	cast	director	runtime	
265208	2.932340	30000000	0	Wild Card	Jason Statham|Michael Angarano|Milo Ventimigli...	Simon West	92	Thriller|
334074	2.331636	20000000	0	Survivor	Pierce Brosnan|Milla Jovovich|Dylan McDermott|...	James McTeigue	96	Crime|
347096	2.165433	0	0	Mythica: The Darkspore	Melanie Stone|Kevin Sorbo|Adam Johnson|Jake St...	Anne K. Black	108	Action|Adver
308369	2.141506	0	0	Me and Earl and the Dying Girl	Thomas Mann|RJ Cyler|Olivia Cooke|Connie Britt...	Alfonso Gomez-Rejon	105	Co
370687	1.876037	0	0	Mythica: The Necromancer	Melanie Stone|Adam Johnson|Kevin Sorbo|Nicola ...	A. Todd Smith	0	Fantasy|Acti
...	...	...	...	...	...	...	...	
21	0.080598	0	0	The Endless Summer	Michael Hynson|Robert August|Lord 'Tally Ho' B...	Bruce Brown	95	
20379	0.065543	0	0	Grand Prix	James Garner|Eva Marie Saint|Yves Montand|Tosh...	John Frankenheimer	176	Action|Adve
39768	0.065141	0	0	Beregis Avtomobilya	Innokentiy Smoktunovskiy|Oleg Efremov|Georgi Z...	Eldar Ryazanov	94	Mys
21449	0.064317	0	0	What's Up, Tiger Lily?	Tatsuya Mihashi|Akiko Wakabayashi|Mie Hama|Joh...	Woody Allen	80	Ac
22293	0.035919	19000	0	Manos: The Hands of Fate	Harold P. Warren|Tom Neyman|John Reynolds|Dian...	Harold P. Warren	74	

5022 rows × 15 columns

In [31]:
```
df.loc[df['budget']==0.000000e+00]
```

Out[31]:

id	popularity	budget	revenue	original_title	cast	director	runtime	
280996	3.927333	0	29355203	Mr. Holmes	Ian McKellen|Milo Parker|Laura Linney|Hattie M...	Bill Condon	103	Mys
339527	3.358321	0	22354572	Solace	Abbie Cornish|Jeffrey Dean Morgan|Colin Farrel...	Afonso Poyart	101	Crime|Dra
284289	2.272044	0	45895	Beyond the Reach	Michael Douglas|Jeremy Irvine|Hanna Mangan Law...	Jean-Baptiste LÃ©onetti	95	
347096	2.165433	0	0	Mythica: The Darkspore	Melanie Stone|Kevin Sorbo|Adam Johnson|Jake St...	Anne K. Black	108	Action|Advent
308369	2.141506	0	0	Me and Earl and the Dying Girl	Thomas Mann|RJ Cyler|Olivia Cooke|Connie Britt...	Alfonso Gomez-Rejon	105	Con
...	...	...	...	...	...	...	...	
5060	0.087034	0	0	Carry On Screaming!	Kenneth Williams|Jim Dale|Harry H. Corbett|Joa...	Gerald Thomas	87	
21	0.080598	0	0	The Endless Summer	Michael Hynson|Robert August|Lord 'Tally Ho' B...	Bruce Brown	95	D
20379	0.065543	0	0	Grand Prix	James Garner|Eva Marie Saint|Yves Montand|Tosh...	John Frankenheimer	176	Action|Adver
39768	0.065141	0	0	Beregis Avtomobilya	Innokentiy Smoktunovskiy|Oleg Efremov|Georgi Z...	Eldar Ryazanov	94	Myste
21449	0.064317	0	0	What's Up, Tiger Lily?	Tatsuya Mihashi|Akiko Wakabayashi|Mie Hama|Joh...	Woody Allen	80	Acti

4751 rows × 15 columns

We notice that some of the movies have a revenue but the budget is zero and vice versa! and that is really odd and not realistic. We should change the value of each budget to the mean to make it more realistic, and vice versa

Checking how many zero values are there in each feature :

In [32]:
```
df[['budget', 'revenue']].apply(lambda x: x == 0).sum()
```

Out[32]:
```
budget     4751
revenue    5022
dtype: int64
```

We have to see first if dropping the zero values of budgets and revenues will affect our data, and the information related to each feature:

In [33]:
```
def entropy(Y):
    """
    Also known as Shanon Entropy
    """
    unique, count = np.unique(Y, return_counts=True, axis=0)
    prob = count/len(Y)
    en = np.sum((-1)*prob*np.log2(prob))
    return en
```

In [34]:
```
entropy(df['budget'])
```

Out[34]: 4.59288636847125

In [35]:
```
entropy(df['revenue'])
```

Out[35]: 6.899474856143105

In [36]:
```
df_cleaned = df[(df['budget'] != 0) & (df['revenue'] != 0)]
entropy(df_cleaned['budget'])
```

Out[36]: 6.814872606606914

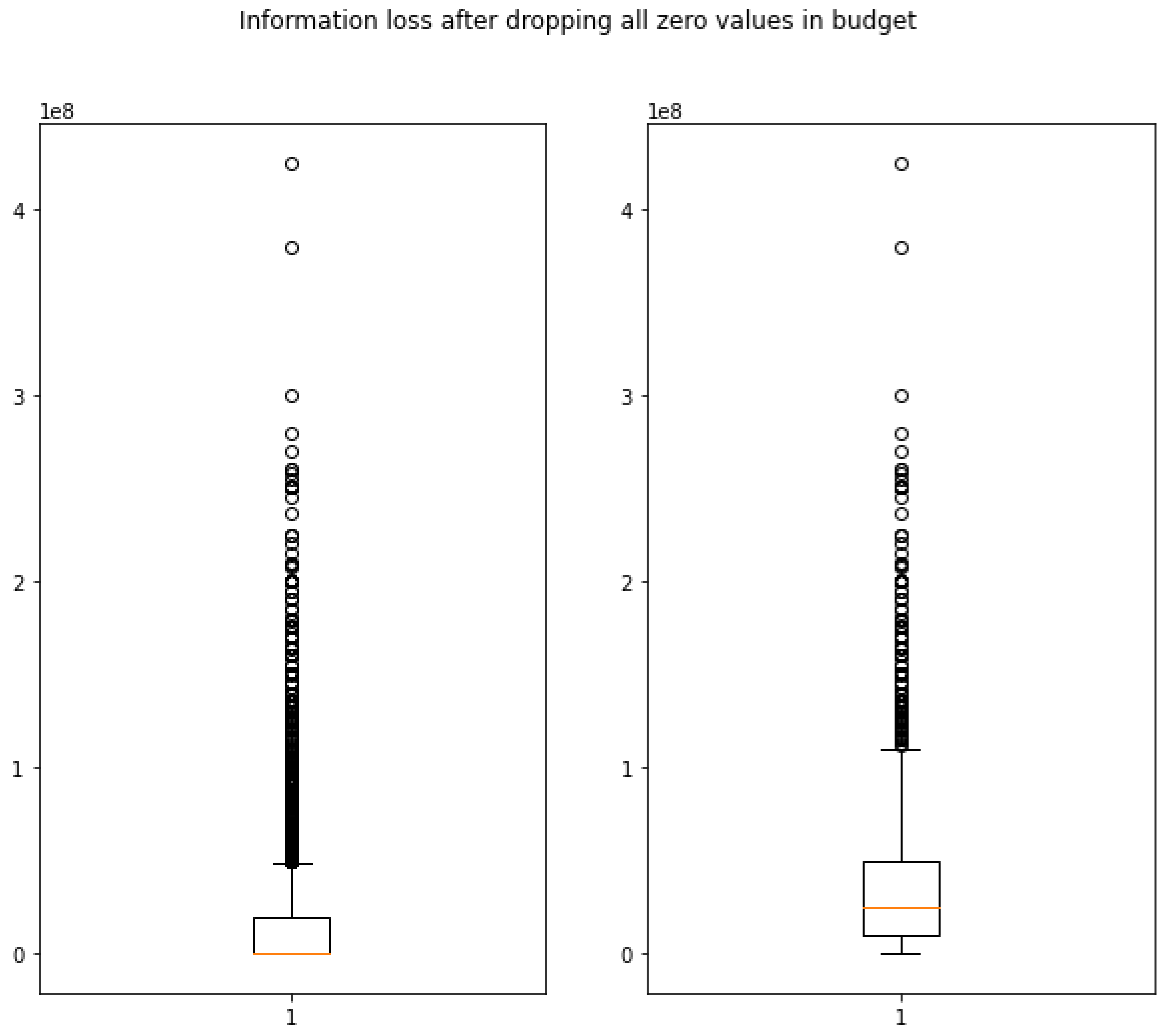
In [37]:
```
entropy(df_cleaned['revenue'])
```

Out[37]: 11.833355099520029

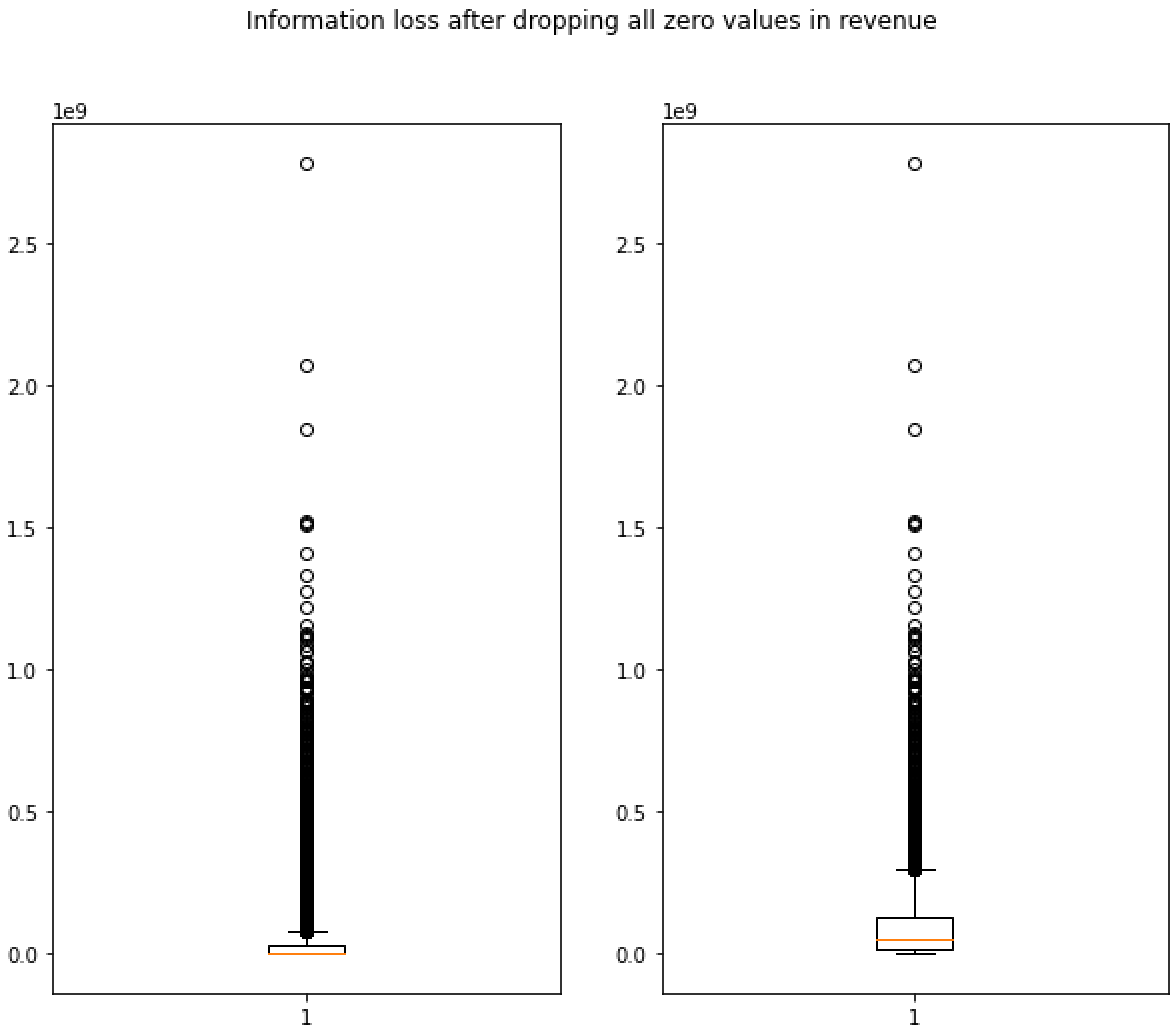
Increasing in shanon's entropy is an indication of information loss : We don't need to drop all the rows containing zeros in budget and revenue columns

we can also see this impact with boxplots:

In [38]:
```
fig, ax = plt.subplots(nrows=1, ncols=2, figsize=(10,8))

ax[0].boxplot(df['budget'])
ax[1].boxplot(df_cleaned['budget'])
plt.suptitle("Information loss after dropping all zero values in budget");
```



In [39]:
```
fig, ax = plt.subplots(nrows=1, ncols=2, figsize=(10,8))

ax[0].boxplot(df['revenue'])
ax[1].boxplot(df_cleaned['revenue'])
plt.suptitle("Information loss after dropping all zero values in revenue");
```



Grouping the budget mean by year:

```
In [40]: mean_budg = df.groupby('release_year')[['budget']].mean()
         mean_budg.head(10)
```

Out[40]:

	budget
release_year	
1960	6.892796e+05
1961	1.537900e+06
1962	1.824071e+06
1963	2.156809e+06
1964	9.630039e+05
1965	2.064856e+06
1966	1.308064e+06
1967	2.795889e+06
1968	1.944297e+06
1969	1.452727e+06

Grouping mean revenue by year :

```
In [41]: mean_rev = df.groupby('release_year')[['revenue']].mean()
         mean_rev.head(10)
```

Out[41]:

	revenue
release_year	
1960	4.531406e+06
1961	1.125734e+07
1962	7.185995e+06
1963	5.511911e+06
1964	8.316629e+06
1965	1.347300e+07
1966	1.925834e+06
1967	2.049541e+07
1968	7.154945e+06
1969	8.412313e+06

Replacing zero values of budget and revenue :

Replacing zero values in budget when revenue isn't null:

```
In [42]: zero_budg = df[(df.budget == 0) & (df.revenue != 0)] #budget equal to zero when
         revenue isn't null
```

In [43]:
```
"""
Function to replace zero values in budget and revenue
"""

def replace_zeros(row : pd.DataFrame, columns : list, df: pd.DataFrame):
    if (row[columns]==0).all():
        row[columns] = df.loc[row.release_year, columns]

    return row
```

In [44]:
```
zero_budg = zero_budg.apply(lambda x: replace_zeros(x, ['budget'], mean_budg),a
xis=1) #replacing budget with mean value
```

In [45]:
```
df[df.index.isin(zero_budg.index)] = zero_budg #replacing budget zero values in
the dataset
```

check again if we filled the zero budget values when the revenue isn't null:

In [46]:
```
df[(df.budget == 0) & (df.revenue != 0)]
```

Out[46]:

	popularity	budget	revenue	original_title	cast	director	runtime	genres	production_companies	releas
id										

Do the same to revenue when budget isn't null but revenue is equal to zero:

In [47]:
```
zero_revenue = df[(df.revenue == 0) & (df.budget != 0)] #revenue zero values wh
en the budget isn't null
```

In [48]:
```
zero_revenue = zero_revenue.apply(lambda x: replace_zeros(x, ['revenue'], mean_
rev),axis=1) #replacing zero value revenues with the mean
```

In [49]:
```
df[df.index.isin(zero_revenue.index)] = zero_revenue #replacing budget zero val
ues in the dataset
```

check again if we filled the zero revenue values when the budget isn't null:

In [50]:
```
df[(df.budget != 0) & (df.revenue == 0)]
```

Out[50]:

	popularity	budget	revenue	original_title	cast	director	runtime	genres	production_companies	releas
id										

I also noticed that the runtime also has some zero values which is unrealistic! to tackle this problem we should replace each zero runtime with the mean value from each year

```python
In [51]: zero_runtime = df[df['runtime'] == 0]
         zero_runtime
```

Out[51]:

	popularity	budget	revenue	original_title	cast	director	runtime	
id								
370687	1.876037	0.0	0.000000e+00	Mythica: The Necromancer	Melanie Stone|Adam Johnson|Kevin Sorbo|Nicola ...	A. Todd Smith	0	Fantasy|A
361931	0.357654	0.0	0.000000e+00	Ronaldo	Cristiano Ronaldo	Anthony Wonke	0	
353345	0.218528	0.0	0.000000e+00	The Exorcism of Molly Hartley	Sarah Lind|Devon Sawa|Gina Holden|Peter MacNei...	Steven R. Monroe	0	
333653	0.176744	0.0	0.000000e+00	If There Be Thorns	Heather Graham|Jason Lewis|Rachael Carpani|Mas...	Nancy Savoca	0	T
286372	0.037459	3250000.0	3.831440e+07	Treehouse	J. Michael Trautmann|Dana Melanie|Daniel Fredr...	Michael G. Bartlett	0	Thriller
286256	0.036904	0.0	0.000000e+00	Tim Maia	Robson Nunes|BabÃº Santana|Alinne Moraes|CauÃ£...	Mauro Lima	0	Documenta
20414	0.082898	0.0	0.000000e+00	Grande, grosso e Verdone	Carlo Verdone|Claudia Gerini|Eva Riccobono|Vit...	Carlo Verdone	0	
289097	0.095583	0.0	0.000000e+00	Cell 213	Bruce Greenwood|Eric Balfour|Michael Rooker|De...	Stephen Kay	0	
158150	0.026459	0.0	0.000000e+00	How to Fall in Love	Brooke D'Orsay|Eric Mabius|Jody Thompson|Gina ...	Mark Griffiths	0	Comec
224815	0.417739	0.0	0.000000e+00	Skinwalker Ranch	Steve Berg|Kyle Davis|Erin Cahill|Jon Gries|De...	Devin McGinn	0	Thriller
248842	0.165765	0.0	0.000000e+00	The Food Guide to Love	Richard Coyle|Leonor Watling|GinÃ©s GarcÃa Mi...	Dominic Harari|Teresa Pelegri	0	Ro
191562	0.147188	0.0	0.000000e+00	Go Goa Gone	Saif Ali Khan|Anand Tiwari|Vir Das|Pooja Gupta	Krishna D.K.|Raj Nidimoru	0	
13713	0.071872	0.0	0.000000e+00	Jean-Philippe	Fabrice Luchini|Johnny Hallyday|Jackie Berroye...	Laurent Tuel	0	

Grouping the runtime by year:

In [52]:
```
mean_runtime = df.groupby('release_year')[['runtime']].mean()
mean_runtime.head(10)
```

Out[52]:

release_year	runtime
1960	110.656250
1961	119.866667
1962	125.833333
1963	111.323529
1964	111.195122
1965	119.294118
1966	108.590909
1967	109.416667
1968	110.540541
1969	110.310345

In [53]:
```
zero_runtime  = zero_runtime .apply(lambda x: replace_zeros(x, ['runtime'], mea
n_runtime),axis=1) #replacing runtime with mean value
df[df.index.isin(zero_runtime.index)] = zero_runtime #replacing runtime zero va
lues in the dataset
```

checking again for zero runtime:

In [54]:
```
df[df['runtime'] == 0]
```

Out[54]:

id	popularity	budget	revenue	original_title	cast	director	runtime	genres	production_companies	releas

I noticed that there is a row where we have two directors at the same time, So I'm going to use the pandas explode function to make two rows for each director

In [55]:
```
df[df.index == 14938].director
```

Out[55]:
```
id
14938    Matt Checkowski|Kurt Mattila
Name: director, dtype: object
```

In [56]:
```
def explode(df: pd.DataFrame, col: str):
    df_new = df.copy()
    df_new[col] = df_new[col].str.split("|")
    df = df_new.explode(col)
    return df
#function to split a row into two based on the name of the director split by
 "|"
```

```
In [57]: explode(df, 'director')
```

Out[57]:

	popularity	budget	revenue	original_title	cast	director	runtime	
id								
135397	32.985763	150000000.0	1.513529e+09	Jurassic World	Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi...	Colin Trevorrow	124.0	Acti
76341	28.419936	150000000.0	3.784364e+08	Mad Max: Fury Road	Tom Hardy|Charlize Theron|Hugh Keays-Byrne|Nic...	George Miller	120.0	Acti
262500	13.112507	110000000.0	2.952382e+08	Insurgent	Shailene Woodley|Theo James|Kate Winslet|Ansel...	Robert Schwentke	119.0	
140607	11.173104	200000000.0	2.068178e+09	Star Wars: The Force Awakens	Harrison Ford|Mark Hamill|Carrie Fisher|Adam D...	J.J. Abrams	136.0	Acti
168259	9.335014	190000000.0	1.506249e+09	Furious 7	Vin Diesel|Paul Walker|Jason Statham|Michelle ...	James Wan	137.0	
...	...	...	...	...	...	...	...	
21	0.080598	0.0	0.000000e+00	The Endless Summer	Michael Hynson|Robert August|Lord 'Tally Ho' B...	Bruce Brown	95.0	
20379	0.065543	0.0	0.000000e+00	Grand Prix	James Garner|Eva Marie Saint|Yves Montand|Tosh...	John Frankenheimer	176.0	Ac
39768	0.065141	0.0	0.000000e+00	Beregis Avtomobilya	Innokentiy Smoktunovskiy|Oleg Efremov|Georgi Z...	Eldar Ryazanov	94.0	
21449	0.064317	0.0	0.000000e+00	What's Up, Tiger Lily?	Tatsuya Mihashi|Akiko Wakabayashi|Mie Hama|Joh...	Woody Allen	80.0	
22293	0.035919	19000.0	1.925834e+06	Manos: The Hands of Fate	Harold P. Warren|Tom Neyman|John Reynolds|Dian...	Harold P. Warren	74.0	

10708 rows × 15 columns

checking for statistical info about the data:

In [58]:
```
df.describe()
```

Out[58]:

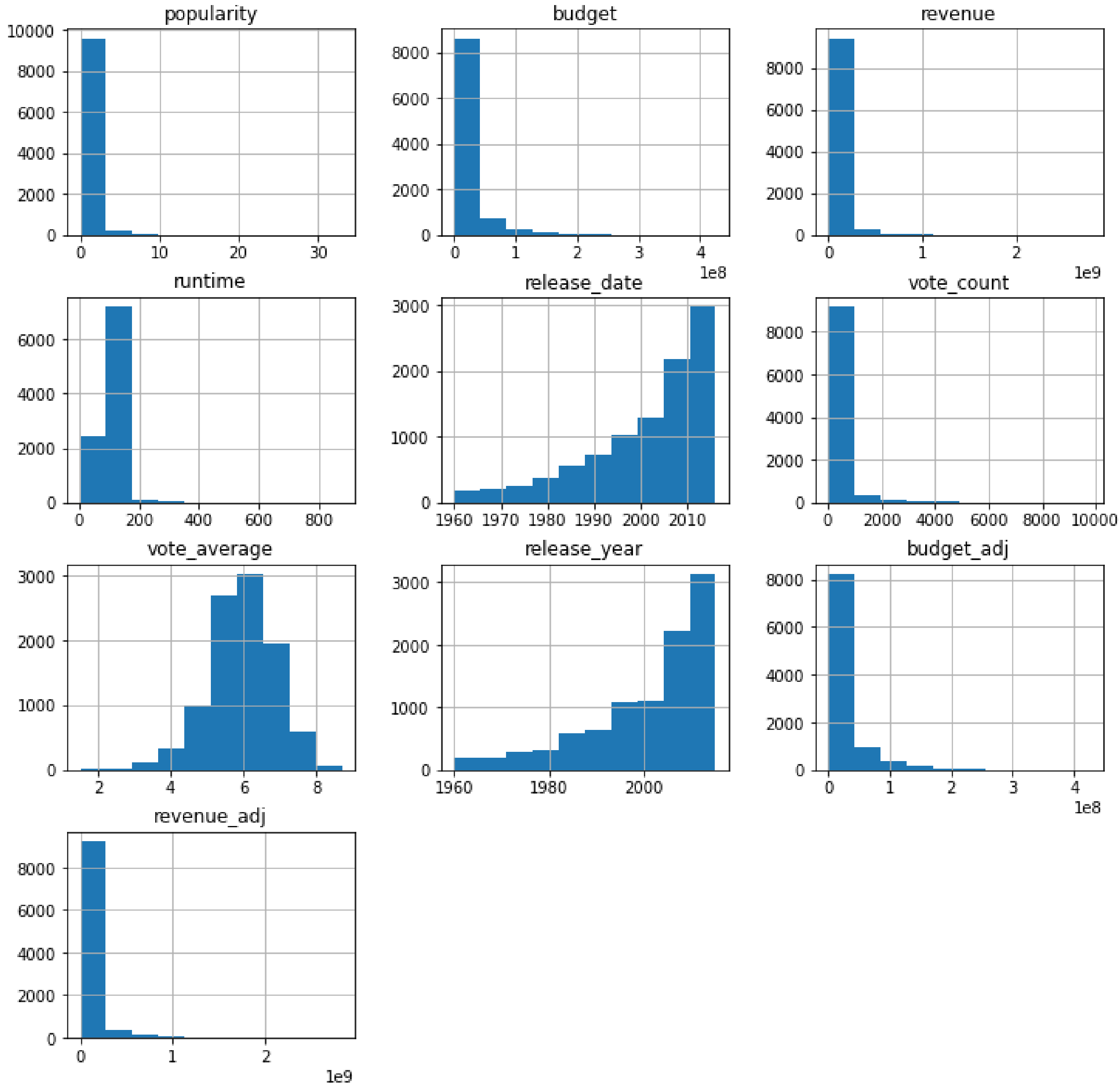
	popularity	budget	revenue	runtime	vote_count	vote_average	release_year	bu
count	9772.000000	9.772000e+03	9.772000e+03	9772.000000	9772.000000	9772.000000	9772.000000	9.77
mean	0.694721	1.753632e+07	4.972249e+07	103.057654	239.312014	5.963528	2000.878428	1.94
std	1.036931	3.185817e+07	1.215553e+08	27.623684	603.011504	0.913174	13.036794	3.56
min	0.000188	0.000000e+00	0.000000e+00	3.000000	10.000000	1.500000	1960.000000	0.00
25%	0.232710	0.000000e+00	0.000000e+00	91.000000	18.000000	5.400000	1994.000000	0.00
50%	0.419762	5.000000e+06	6.865676e+06	100.000000	46.000000	6.000000	2005.000000	3.06
75%	0.776408	2.000000e+07	4.946531e+07	112.000000	173.000000	6.600000	2011.000000	2.46
max	32.985763	4.250000e+08	2.781506e+09	877.000000	9767.000000	8.700000	2015.000000	4.25

Exploratory Data Analysis

Question1: Which genres are the most popular from year to year ?

First i want to see the distribution of each feature :

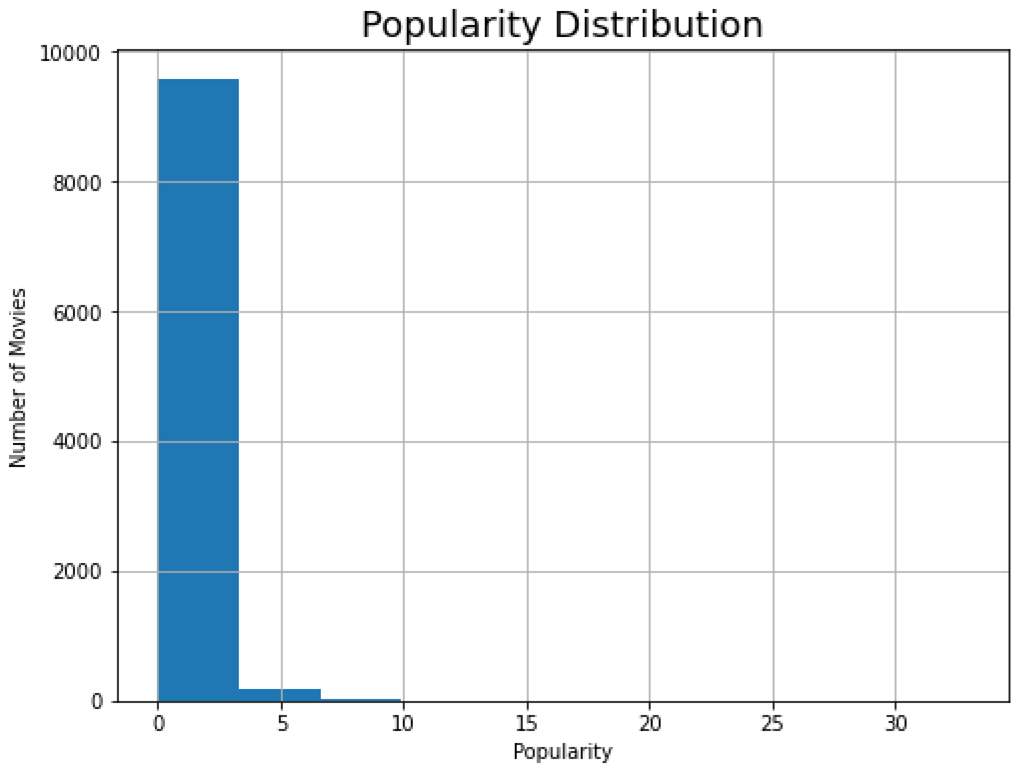
```
In [59]: df.hist(figsize=(12,12));
```



we can see that there are some variables that are skewed either to the right such as : budget and budget_adj , or to the left such as : release_year and release_date. But some variables have a normal distribution such as vote_average

Distribution of popularity:

In [60]:
```
df.popularity.hist(figsize=(8, 6))
plt.title("Popularity Distribution", fontsize=18);
plt.xlabel("Popularity")
plt.ylabel("Number of Movies")
plt.show();
```



Most popular movie of all between the years 2015 and 1960 :

In [61]:
```
df.loc[df['popularity'].idxmax()] #idxmax : returns the index of the first occu
rrence of maximum over requested column
```

Out[61]:
```
popularity                                                    32.985763
budget                                                      150000000.0
revenue                                                    1513528810.0
original_title                                           Jurassic World
cast                    Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi...
director                                                Colin Trevorrow
runtime                                                           124.0
genres                          Action|Adventure|Science Fiction|Thriller
production_companies    Universal Studios|Amblin Entertainment|Legenda...
release_date                                        2015-06-09 00:00:00
vote_count                                                         5562
vote_average                                                        6.5
release_year                                                       2015
budget_adj                                             137999939.280026
revenue_adj                                             1392445892.5238
Name: 135397, dtype: object
```

Most popular movie : **Jurassic World(2015)**

In [62]:
```python
#most popular movies :

most_pop = df.sort_values('popularity', axis=0, ascending=False).head(10)
most_pop
```

Out[62]:

id	popularity	budget	revenue	original_title	cast	director	runtime
135397	32.985763	150000000.0	1.513529e+09	Jurassic World	Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi...	Colin Trevorrow	124.0
76341	28.419936	150000000.0	3.784364e+08	Mad Max: Fury Road	Tom Hardy|Charlize Theron|Hugh Keays-Byrne|Nic...	George Miller	120.0
157336	24.949134	165000000.0	6.217525e+08	Interstellar	Matthew McConaughey|Jessica Chastain|Anne Hath...	Christopher Nolan	169.0
118340	14.311205	170000000.0	7.733124e+08	Guardians of the Galaxy	Chris Pratt|Zoe Saldana|Dave Bautista|Vin Dies...	James Gunn	121.0
262500	13.112507	110000000.0	2.952382e+08	Insurgent	Shailene Woodley|Theo James|Kate Winslet|Ansel...	Robert Schwentke	119.0
100402	12.971027	170000000.0	7.147666e+08	Captain America: The Winter Soldier	Chris Evans|Scarlett Johansson|Sebastian Stan|...	Joe Russo|Anthony Russo	136.0
11	12.037933	11000000.0	7.753980e+08	Star Wars	Mark Hamill|Harrison Ford|Carrie Fisher|Peter ...	George Lucas	121.0
245891	11.422751	20000000.0	7.873990e+07	John Wick	Keanu Reeves|Michael Nyqvist|Alfie Allen|Wille...	Chad Stahelski|David Leitch	101.0
140607	11.173104	200000000.0	2.068178e+09	Star Wars: The Force Awakens	Harrison Ford|Mark Hamill|Carrie Fisher|Adam D...	J.J. Abrams	136.0
131631	10.739009	125000000.0	7.521002e+08	The Hunger Games: Mockingjay - Part 1	Jennifer Lawrence|Josh Hutcherson|Liam Hemswor...	Francis Lawrence	123.0

In [63]:
```python
most_pop_by_year = most_pop.groupby(['release_year'])['genres'].value_counts()
most_pop_by_year
```

Out[63]:
```
release_year  genres
1977          Adventure|Action|Science Fiction               1
2014          Action|Adventure|Science Fiction               1
              Action|Science Fiction|Adventure               1
              Action|Thriller                                1
              Adventure|Drama|Science Fiction                1
              Science Fiction|Adventure|Thriller             1
2015          Action|Adventure|Science Fiction|Thriller      2
              Action|Adventure|Science Fiction|Fantasy       1
              Adventure|Science Fiction|Thriller             1
Name: genres, dtype: int64
```

we can see that most popular genres between 2015 and 1960 are : **Action, Adventure, Science Fiction and Thriller**

Question 2: Which movies have the most profit ? and in which year they were released ?

profit of movies:

In [64]:
```
df['profit'] = df.revenue - df.budget
df.profit.head()
```

Out[64]:
```
id
135397    1.363529e+09
76341     2.284364e+08
262500    1.852382e+08
140607    1.868178e+09
168259    1.316249e+09
Name: profit, dtype: float64
```

In [65]:
```
df.profit.describe()
```

Out[65]:
```
count    9.772000e+03
mean     3.218616e+07
std      1.006611e+08
min     -4.139124e+08
25%      0.000000e+00
50%      0.000000e+00
75%      3.244257e+07
max      2.544506e+09
Name: profit, dtype: float64
```

movies having the most profit (having profit > 75%):

In [66]:
```
successful_movies = df[df.profit >= 3.244257e+07]
successful_movies.sort_values('profit', axis=0, ascending=False).head(5)
```

Out[66]:

	popularity	budget	revenue	original_title	cast	director	runtime	
id								
19995	9.432768	237000000.0	2.781506e+09	Avatar	Sam Worthington|Zoe Saldana|Sigourney Weaver|S...	James Cameron	162.0	Action|Adv
140607	11.173104	200000000.0	2.068178e+09	Star Wars: The Force Awakens	Harrison Ford|Mark Hamill|Carrie Fisher|Adam D...	J.J. Abrams	136.0	A
597	4.355219	200000000.0	1.845034e+09	Titanic	Kate Winslet|Leonardo DiCaprio|Frances Fisher|...	James Cameron	194.0	I
135397	32.985763	150000000.0	1.513529e+09	Jurassic World	Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi...	Colin Trevorrow	124.0	A
168259	9.335014	190000000.0	1.506249e+09	Furious 7	Vin Diesel|Paul Walker|Jason Statham|Michelle ...	James Wan	137.0	

In [67]:
```
successful_movies.groupby('release_year')['original_title'].value_counts()
```

Out[67]:
```
release_year  original_title
1960          Spartacus                        1
1961          One Hundred and One Dalmatians   1
              West Side Story                  1
1962          Dr. No                           1
              How the West Was Won             1
                                              ..
2015          Unfriended                       1
              Vacation                         1
              Vice                             1
              War Room                         1
              Woman in Gold                    1
Name: original_title, Length: 2443, dtype: int64
```

Question 3: What kind of characteristics are associated with movies having high revenues ?

In [68]:
```
df.revenue.describe()
```

Out[68]:
```
count    9.772000e+03
mean     4.972249e+07
std      1.215553e+08
min      0.000000e+00
25%      0.000000e+00
50%      6.865676e+06
75%      4.946531e+07
max      2.781506e+09
Name: revenue, dtype: float64
```

movies having the highest revenues (having revenues > 75%):

In [69]:
```
high_movies = df[df.revenue >= 4.946531e+07]
high_movies.head(5)
```

Out[69]:

id	popularity	budget	revenue	original_title	cast	director	runtime	
135397	32.985763	150000000.0	1.513529e+09	Jurassic World	Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi...	Colin Trevorrow	124.0	Action|Adve
76341	28.419936	150000000.0	3.784364e+08	Mad Max: Fury Road	Tom Hardy|Charlize Theron|Hugh Keays-Byrne|Nic...	George Miller	120.0	Action|Adve
262500	13.112507	110000000.0	2.952382e+08	Insurgent	Shailene Woodley|Theo James|Kate Winslet|Ansel...	Robert Schwentke	119.0	Adve
140607	11.173104	200000000.0	2.068178e+09	Star Wars: The Force Awakens	Harrison Ford|Mark Hamill|Carrie Fisher|Adam D...	J.J. Abrams	136.0	Action|Adve F
168259	9.335014	190000000.0	1.506249e+09	Furious 7	Vin Diesel|Paul Walker|Jason Statham|Michelle ...	James Wan	137.0	Actior

In [70]:
```
high_movies.describe()
```

Out[70]:

	popularity	budget	revenue	runtime	vote_count	vote_average	release_year	bu
count	2389.000000	2.389000e+03	2.389000e+03	2389.000000	2389.000000	2389.000000	2389.000000	2.38
mean	1.469329	4.887598e+07	1.730065e+08	111.191712	720.071578	6.163541	2002.665132	5.55
std	1.750423	4.765504e+07	1.990198e+08	27.111858	1039.334704	0.820476	9.606977	5.03
min	0.010335	3.000000e+00	4.946990e+07	4.000000	10.000000	2.900000	1960.000000	0.00
25%	0.555793	1.500000e+07	6.064831e+07	96.000000	115.000000	5.600000	1998.000000	1.84
50%	1.042281	3.500000e+07	1.011341e+08	107.000000	333.000000	6.200000	2004.000000	4.07
75%	1.765322	6.800000e+07	1.967812e+08	122.000000	852.000000	6.700000	2010.000000	8.00
max	32.985763	3.800000e+08	2.781506e+09	705.000000	9767.000000	8.300000	2015.000000	3.68

In [71]:
```
high_movies[['original_title', 'popularity', 'release_date', 'profit', 'budget'
, 'revenue', 'vote_average']]
```

Out[71]:

id	original_title	popularity	release_date	profit	budget	revenue	vote_average
135397	Jurassic World	32.985763	2015-06-09	1.363529e+09	1.500000e+08	1.513529e+09	6.5
76341	Mad Max: Fury Road	28.419936	2015-05-13	2.284364e+08	1.500000e+08	3.784364e+08	7.1
262500	Insurgent	13.112507	2015-03-18	1.852382e+08	1.100000e+08	2.952382e+08	6.3
140607	Star Wars: The Force Awakens	11.173104	2015-12-15	1.868178e+09	2.000000e+08	2.068178e+09	7.5
168259	Furious 7	9.335014	2015-04-01	1.316249e+09	1.900000e+08	1.506249e+09	7.3
...	...	...	...	...	...	...	...
948	Halloween	1.198849	1978-10-25	6.970000e+07	3.000000e+05	7.000000e+07	7.3
8469	Animal House	1.157930	1978-07-27	1.383000e+08	2.700000e+06	1.410000e+08	6.7
6081	Revenge of the Pink Panther	1.090065	1978-07-19	4.615309e+07	3.426181e+06	4.957927e+07	6.2
11778	The Deer Hunter	0.959754	1978-12-08	3.500000e+07	1.500000e+07	5.000000e+07	7.4
16214	Hooper	0.044675	1978-07-28	7.457382e+07	3.426181e+06	7.800000e+07	6.0

2389 rows × 7 columns

Question 4: Which movies have the highest budgets but are low on vote counts ?

In [72]:
```
m =df.loc[(df['budget'] > df['budget'].quantile(0.75)) & (df['vote_count'] < df
['vote_count'].quantile(0.25))]
m[['original_title', 'release_date', 'budget', 'vote_count']]
```

Out[72]:

id	original_title	release_date	budget	vote_count
254263	The Swan Princess: A Royal Family Tale	2014-02-25	3.000000e+07	14
242166	Red Sky	2014-03-12	2.500000e+07	16
33870	Mao's Last Dancer	2009-10-01	2.500000e+07	16
66193	Sinners and Saints	2010-09-14	2.221868e+07	16
48495	Double Wedding	2010-07-20	1.040024e+08	12
...	...	...	...	...
2071	Shattered	1991-10-11	2.200000e+07	11
6470	Fire Birds	1990-05-25	2.200000e+07	15
22414	Postcards from the Edge	1990-09-12	2.200000e+07	15
46828	Son of the Pink Panther	1993-01-01	2.500000e+07	15
33157	Waterloo	1970-10-26	2.500000e+07	10

95 rows × 4 columns

Question 5: Which movies have the lowest budgets but are very high on vote counts ?

In [73]:
```
m_2 =df.loc[(df['budget'] < df['budget'].quantile(0.25)) & (df['vote_count'] > df['vote_count'].quantile(0.75))]
m_2[['original_title', 'release_date', 'budget', 'vote_count']]
```

Out[73]:

	original_title	release_date	budget	vote_count
id				

it seems to be that there are no low budget movies that have very high vote counts

Question 6: what are the movies that flopped and topped the most in terms of profit and votes ? and who was their cast/director ?

In [74]:
```
#successful movies:
s= successful_movies.where(successful_movies['vote_count']> successful_movies[
'vote_count'].quantile(0.75))
s[['original_title', 'cast', 'director', 'release_date']].head(15)
```

Out[74]:

id	original_title	cast	director	release_date
135397	Jurassic World	Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi...	Colin Trevorrow	2015-06-09
76341	Mad Max: Fury Road	Tom Hardy|Charlize Theron|Hugh Keays-Byrne|Nic...	George Miller	2015-05-13
262500	Insurgent	Shailene Woodley|Theo James|Kate Winslet|Ansel...	Robert Schwentke	2015-03-18
140607	Star Wars: The Force Awakens	Harrison Ford|Mark Hamill|Carrie Fisher|Adam D...	J.J. Abrams	2015-12-15
168259	Furious 7	Vin Diesel|Paul Walker|Jason Statham|Michelle ...	James Wan	2015-04-01
281957	The Revenant	Leonardo DiCaprio|Tom Hardy|Will Poulter|Domhn...	Alejandro GonzÃ¡lez IÃ±Ã¡rritu	2015-12-25
87101	Terminator Genisys	Arnold Schwarzenegger|Jason Clarke|Emilia Clar...	Alan Taylor	2015-06-23
286217	The Martian	Matt Damon|Jessica Chastain|Kristen Wiig|Jeff ...	Ridley Scott	2015-09-30
211672	Minions	Sandra Bullock|Jon Hamm|Michael Keaton|Allison...	Kyle Balda|Pierre Coffin	2015-06-17
150540	Inside Out	Amy Poehler|Phyllis Smith|Richard Kind|Bill Ha...	Pete Docter	2015-06-09
206647	Spectre	Daniel Craig|Christoph Waltz|LÃ©a Seydoux|Ralp...	Sam Mendes	2015-10-26
257344	Pixels	Adam Sandler|Michelle Monaghan|Peter Dinklage|...	Chris Columbus	2015-07-16
99861	Avengers: Age of Ultron	Robert Downey Jr.|Chris Hemsworth|Mark Ruffalo...	Joss Whedon	2015-04-22
273248	The Hateful Eight	Samuel L. Jackson|Kurt Russell|Jennifer Jason ...	Quentin Tarantino	2015-12-25
260346	Taken 3	Liam Neeson|Forest Whitaker|Maggie Grace|Famke...	Olivier Megaton	2015-01-01

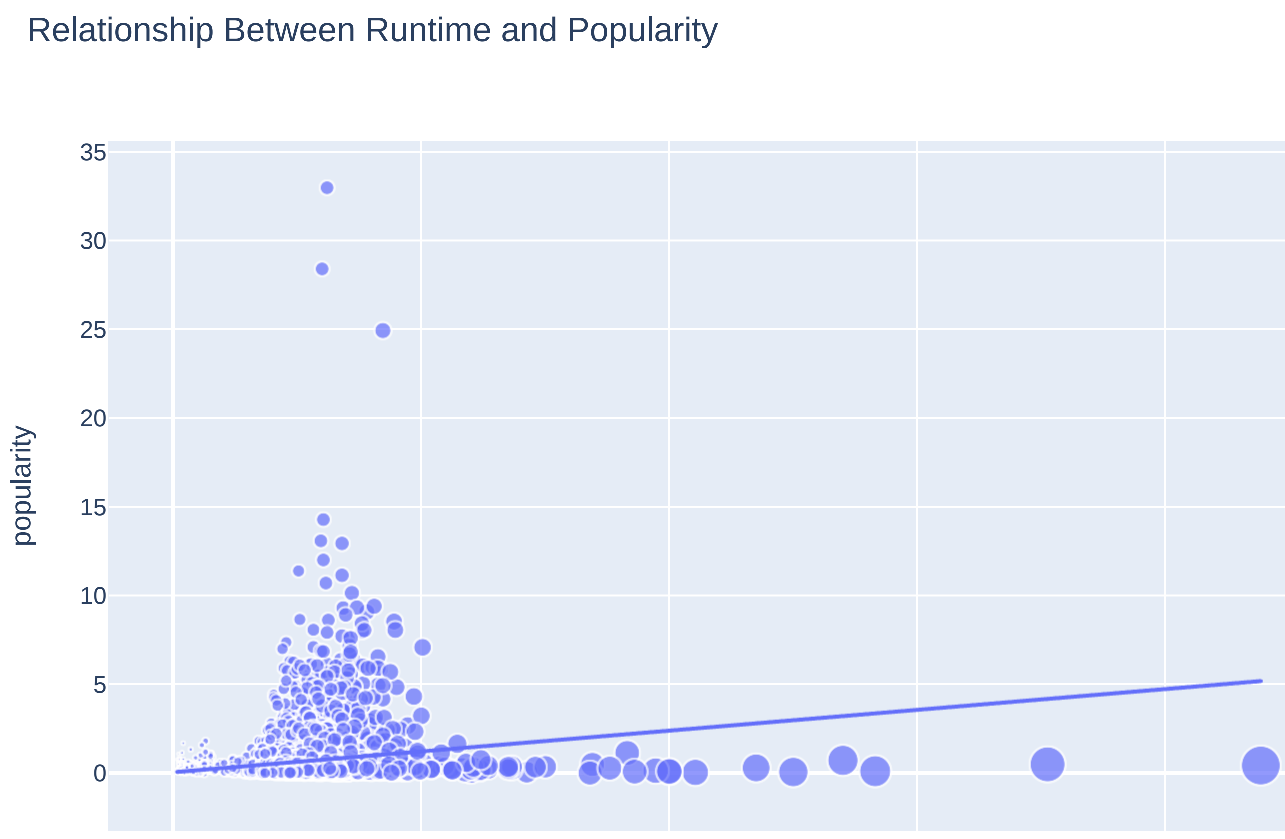
In [75]:
```
#flopped movies:

flopped_movies = df[df.profit < 3.244257e+07]
flopped_movies[['original_title', 'cast', 'director', 'release_date']].head(15)
```

Out[75]:

id	original_title	cast	director	release_date
76757	Jupiter Ascending	Mila Kunis|Channing Tatum|Sean Bean|Eddie Redm...	Lana Wachowski|Lilly Wachowski	2015-02-04
264660	Ex Machina	Domhnall Gleeson|Alicia Vikander|Oscar Isaac|S...	Alex Garland	2015-01-21
158852	Tomorrowland	Britt Robertson|George Clooney|Raffey Cassidy|...	Brad Bird	2015-05-19
280996	Mr. Holmes	Ian McKellen|Milo Parker|Laura Linney|Hattie M...	Bill Condon	2015-06-19
264644	Room	Brie Larson|Jacob Tremblay|Joan Allen|Sean Bri...	Lenny Abrahamson	2015-10-16
339527	Solace	Abbie Cornish|Jeffrey Dean Morgan|Colin Farrel...	Afonso Poyart	2015-09-03
241554	Run All Night	Liam Neeson|Ed Harris|Joel Kinnaman|Boyd Holbr...	Jaume Collet-Serra	2015-03-11
321697	Steve Jobs	Michael Fassbender|Kate Winslet|Seth Rogen|Kat...	Danny Boyle	2015-10-09
293863	The Age of Adaline	Blake Lively|Michiel Huisman|Harrison Ford|Ell...	Lee Toland Krieger	2015-04-16
325348	Hardcore Henry	Sharlto Copley|Haley Bennett|Danila Kozlovskiy...	Ilya Naishuller	2015-09-12
265208	Wild Card	Jason Statham|Michael Angarano|Milo Ventimigli...	Simon West	2015-01-14
254320	The Lobster	Colin Farrell|Rachel Weisz|LÃ©a Seydoux|John C...	Yorgos Lanthimos	2015-10-08
258480	Carol	Cate Blanchett|Rooney Mara|Kyle Chandler|Sarah...	Todd Haynes	2015-11-20
257088	Point Break	Edgar RamÃrez|Luke Bracey|Teresa Palmer|Delro...	Ericson Core	2015-12-03
295964	Burnt	Bradley Cooper|Sienna Miller|Lily James|Alicia...	John Wells	2015-10-02

Question 7: Is the screen runtime related to the success or flopping of the movies ?

In [92]:
```
import plotly.express as px
figure = px.scatter(data_frame = df,
                    x="runtime",
                    y="popularity",
                    size="runtime",
                    trendline="ols",
                    title = "Relationship Between Runtime and Popularity")
figure.show()
```



Runtime and poularity aren't correlated

Question 8: What is the runtime of the most successful movies between the years 2015 and 1960 ?

In [77]:
```
s.sort_values('popularity', axis=0, ascending=False).head(10).groupby(['runtime', 'release_year'])['original_title'].sum()
```

Out[77]:
```
runtime  release_year
101.0    2014.0                                         John Wick
119.0    2015.0                                         Insurgent
120.0    2015.0                                Mad Max: Fury Road
121.0    1977.0                                         Star Wars
         2014.0                           Guardians of the Galaxy
123.0    2014.0          The Hunger Games: Mockingjay - Part 1
124.0    2015.0                                    Jurassic World
136.0    2014.0              Captain America: The Winter Soldier
         2015.0                      Star Wars: The Force Awakens
169.0    2014.0                                      Interstellar
Name: original_title, dtype: object
```

Question 9: Do movies with more profit have more popularity as well?

In [78]:
```python
most_pop['profit'] = df.profit

most_successful = successful_movies.sort_values('profit', axis=0, ascending=False).head(10)
most_successful
```

Out[78]:

	popularity	budget	revenue	original_title	cast	director	runtime	
id								
19995	9.432768	237000000.0	2.781506e+09	Avatar	Sam Worthington|Zoe Saldana|Sigourney Weaver|S...	James Cameron	162.0	Actio
140607	11.173104	200000000.0	2.068178e+09	Star Wars: The Force Awakens	Harrison Ford|Mark Hamill|Carrie Fisher|Adam D...	J.J. Abrams	136.0	
597	4.355219	200000000.0	1.845034e+09	Titanic	Kate Winslet|Leonardo DiCaprio|Frances Fisher|...	James Cameron	194.0	
135397	32.985763	150000000.0	1.513529e+09	Jurassic World	Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi...	Colin Trevorrow	124.0	
168259	9.335014	190000000.0	1.506249e+09	Furious 7	Vin Diesel|Paul Walker|Jason Statham|Michelle ...	James Wan	137.0	
24428	7.637767	220000000.0	1.519558e+09	The Avengers	Robert Downey Jr.|Chris Evans|Mark Ruffalo|Chr...	Joss Whedon	143.0	Sci
12445	5.711315	125000000.0	1.327818e+09	Harry Potter and the Deathly Hallows: Part 2	Daniel Radcliffe|Rupert Grint|Emma Watson|Alan...	David Yates	130.0	
99861	5.944927	280000000.0	1.405036e+09	Avengers: Age of Ultron	Robert Downey Jr.|Chris Hemsworth|Mark Ruffalo...	Joss Whedon	141.0	Acti
109445	6.112766	150000000.0	1.274219e+09	Frozen	Kristen Bell|Idina Menzel|Jonathan Groff|Josh ...	Chris Buck|Jennifer Lee	102.0	
1642	1.136610	22000000.0	1.106280e+09	The Net	Sandra Bullock|Jeremy Northam|Dennis Miller|We...	Irwin Winkler	114.0	Crime|

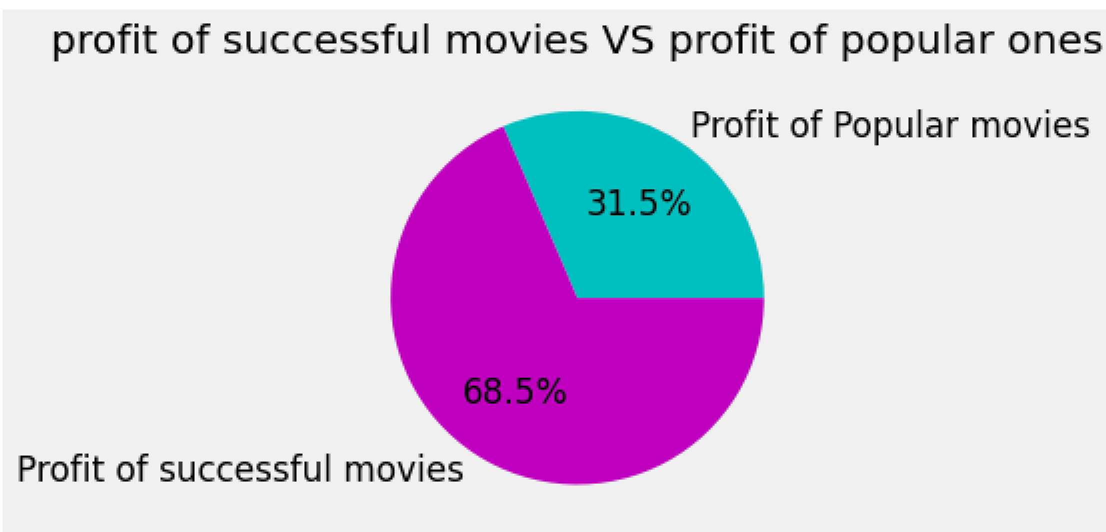
In [79]:
```
most_pop[['original_title', 'popularity', 'release_date', 'profit']]
```

Out[79]:

id	original_title	popularity	release_date	profit
135397	Jurassic World	32.985763	2015-06-09	1.363529e+09
76341	Mad Max: Fury Road	28.419936	2015-05-13	2.284364e+08
157336	Interstellar	24.949134	2014-11-05	4.567525e+08
118340	Guardians of the Galaxy	14.311205	2014-07-30	6.033124e+08
262500	Insurgent	13.112507	2015-03-18	1.852382e+08
100402	Captain America: The Winter Soldier	12.971027	2014-03-20	5.447666e+08
11	Star Wars	12.037933	1977-03-20	7.643980e+08
245891	John Wick	11.422751	2014-10-22	5.873990e+07
140607	Star Wars: The Force Awakens	11.173104	2015-12-15	1.868178e+09
131631	The Hunger Games: Mockingjay - Part 1	10.739009	2014-11-18	6.271002e+08

Pie Chart representing profit of successful movies VS profit of popular ones :

In [98]:
```
plt.pie([most_pop.profit.sum(), most_successful.profit.sum()],
    labels=['Profit of Popular movies', 'Profit of successful movies'],
    colors=['c', 'm'], autopct='%.1f%%',
    textprops={'size': 'large'});
plt.title("profit of successful movies VS profit of popular ones")
plt.show();
```



percentage of popular movies that have highest profit :

In [81]:
```
perc = (most_pop.index.isin(most_successful.index).sum()) / most_pop.shape[0] * 100

#most_pop.shape[0] : number of columns of most_pop
perc
```

Out[81]: 20.0

We can notice that having the most profit doesn't mean the movie is also popular, also we have only 20% of the popular movies having also the highest profit

Question 10: Which production companies that released the most successful movies each year ?

In [82]:
```
s[['original_title', 'production_companies', 'release_date']].head(15)
```

Out[82]:

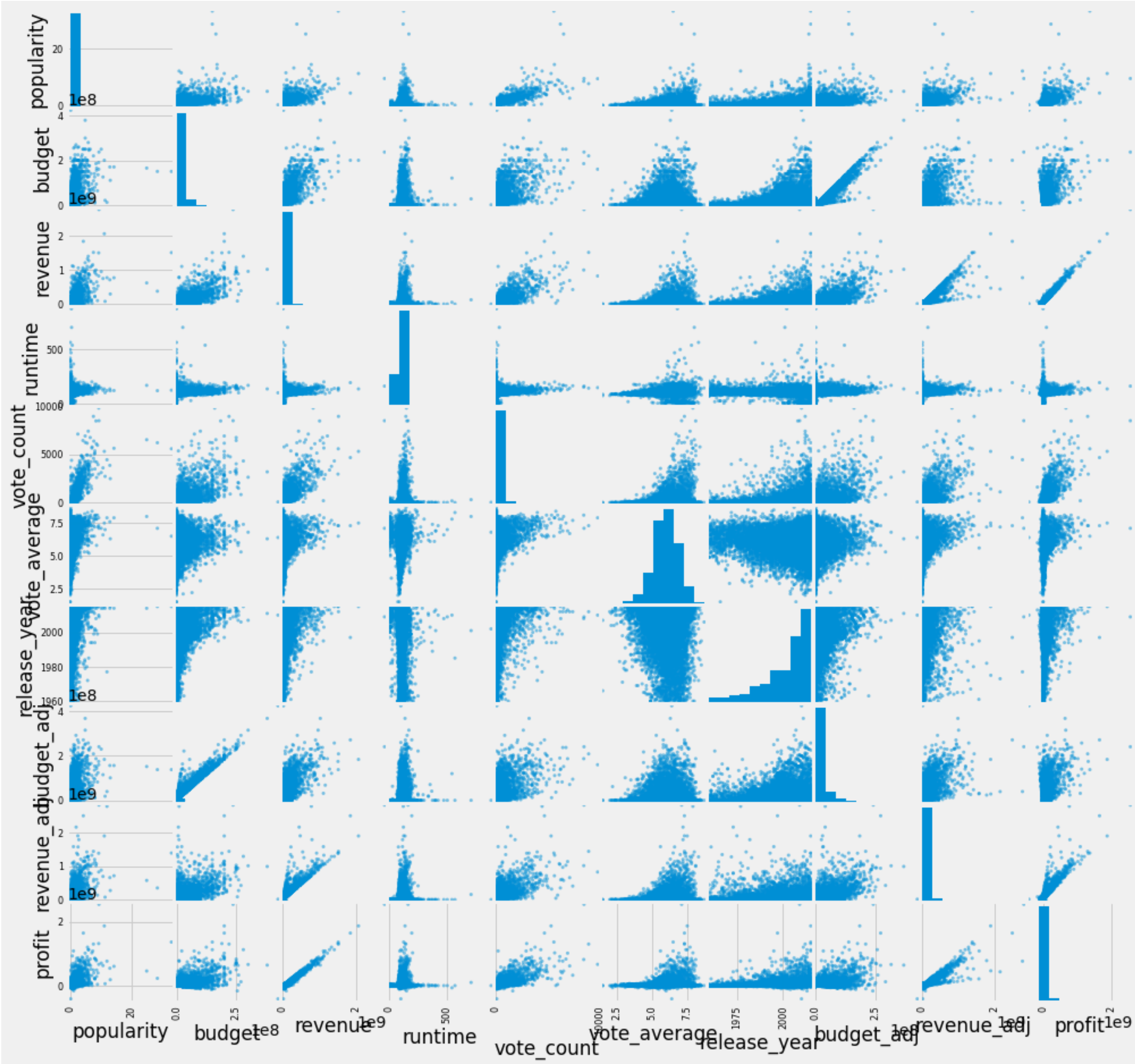
id	original_title	production_companies	release_date
135397	Jurassic World	Universal Studios|Amblin Entertainment|Legenda...	2015-06-09
76341	Mad Max: Fury Road	Village Roadshow Pictures|Kennedy Miller Produ...	2015-05-13
262500	Insurgent	Summit Entertainment|Mandeville Films|Red Wago...	2015-03-18
140607	Star Wars: The Force Awakens	Lucasfilm|Truenorth Productions|Bad Robot	2015-12-15
168259	Furious 7	Universal Pictures|Original Film|Media Rights ...	2015-04-01
281957	The Revenant	Regency Enterprises|Appian Way|CatchPlay|Anony...	2015-12-25
87101	Terminator Genisys	Paramount Pictures|Skydance Productions	2015-06-23
286217	The Martian	Twentieth Century Fox Film Corporation|Scott F...	2015-09-30
211672	Minions	Universal Pictures|Illumination Entertainment	2015-06-17
150540	Inside Out	Walt Disney Pictures|Pixar Animation Studios|W...	2015-06-09
206647	Spectre	Columbia Pictures|Danjaq|B24	2015-10-26
257344	Pixels	Columbia Pictures|Happy Madison Productions	2015-07-16
99861	Avengers: Age of Ultron	Marvel Studios|Prime Focus|Revolution Sun Studios	2015-04-22
273248	The Hateful Eight	Double Feature Films|The Weinstein Company|Fil...	2015-12-25
260346	Taken 3	Twentieth Century Fox Film Corporation|M6 Film...	2015-01-01

In [83]:
```
s.groupby('release_year')['production_companies'].sum()
```

Out[83]:
```
release_year
1961.0                                Walt Disney Productions
1967.0                                   Walt Disney Pictures
1968.0    Stanley Kubrick Productions|Metro-Goldwyn-Maye...
1972.0                Paramount Pictures|Alfran Productions
1973.0                        Warner Bros.|Hoya Productions
1974.0               Paramount Pictures|The Coppola Company
1975.0    Fantasy Films|Warner Bros.Universal Pictures|Z...
1976.0                                       United Artists
1977.0     Lucasfilm|Twentieth Century Fox Film Corporation
1979.0    Twentieth Century-Fox Productions|Brandywine P...
1980.0     Lucasfilm|Twentieth Century Fox Film Corporation
1981.0                         Lucasfilm|Paramount Pictures
1982.0    Universal Pictures|Amblin EntertainmentOrion P...
1983.0    Lucasfilm|Twentieth Century Fox Film Corporati...
1984.0    Orion Pictures|Pacific Western|Hemdale Film|Ci...
1985.0    Universal Pictures|Amblin Entertainment|U-Driv...
1986.0    Twentieth Century Fox Film Corporation|SLM Pro...
1987.0    Twentieth Century Fox Film Corporation|Lawrenc...
1988.0    Twentieth Century Fox Film Corporation|Gordon ...
1989.0    Walt Disney PicturesLucasfilm|Paramount Pictur...
1990.0    TriStar Pictures|Carolco Pictures|Carolco Inte...
1991.0    Walt Disney Pictures|Walt Disney Animation Stu...
1992.0    Walt Disney PicturesTwentieth Century Fox Film...
1993.0    Columbia PicturesWalt Disney Pictures|Tim Burt...
1994.0    Miramax Films|A Band Apart|Jersey FilmsParamou...
1995.0    New Line Cinema|Juno Pix|Cecchi Gori PicturesW...
1996.0    Twentieth Century Fox Film Corporation|Centrop...
1997.0    Paramount Pictures|Twentieth Century Fox Film ...
1998.0    Paramount Pictures|Scott Rudin ProductionsJerr...
1999.0    Regency Enterprises|Fox 2000 Pictures|Taurus F...
2000.0    DreamWorks SKG|Universal Pictures|Scott Free P...
2001.0    WingNut Films|New Line Cinema|The Saul Zaentz ...
2002.0    WingNut Films|New Line Cinema|The Saul Zaentz ...
2003.0    WingNut Films|New Line CinemaLakeshore Enterta...
2004.0    1492 Pictures|Warner Bros.|Heyday Films|P of A...
2005.0    Patalex IV Productions Limited|Warner Bros.|He...
2006.0    Lakeshore Entertainment|Screen GemsWalt Disney...
2007.0    Walt Disney Pictures|Jerry Bruckheimer Films|S...
2008.0    DC Comics|Legendary Pictures|Warner Bros.|Sync...
2009.0    Ingenious Film Partners|Twentieth Century Fox ...
2010.0    Legendary Pictures|Warner Bros.|SyncopyMarvel ...
2011.0    Marvel StudiosBold Films|Marc Platt Production...
2012.0    Marvel StudiosLakeshore Entertainment|Saturn F...
2013.0    Walt Disney Pictures|Walt Disney Animation Stu...
2014.0    Paramount Pictures|Legendary Pictures|Warner B...
2015.0    Universal Studios|Amblin Entertainment|Legenda...
Name: production_companies, dtype: object
```

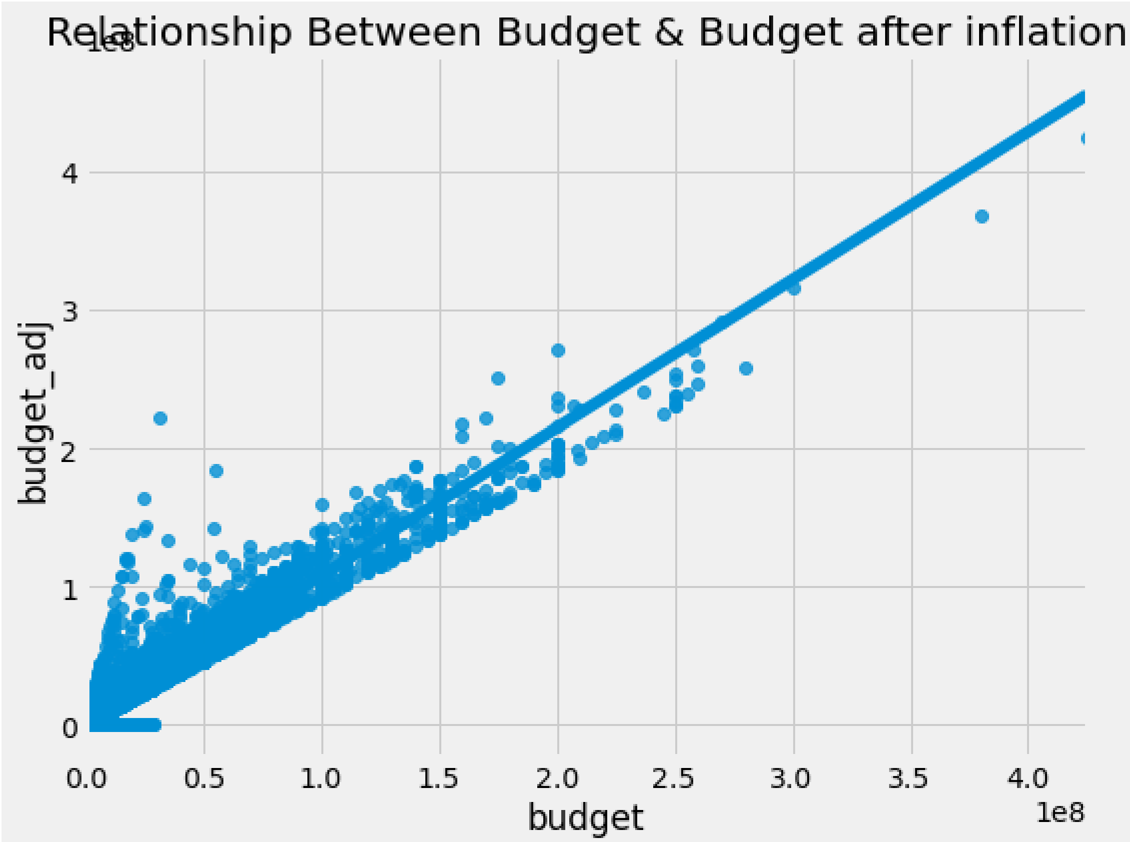
Question 11: What is the relationship between Budget and other features in the dataset ?

In [102]:
```
pd.plotting.scatter_matrix(df, figsize=(15,15));
```

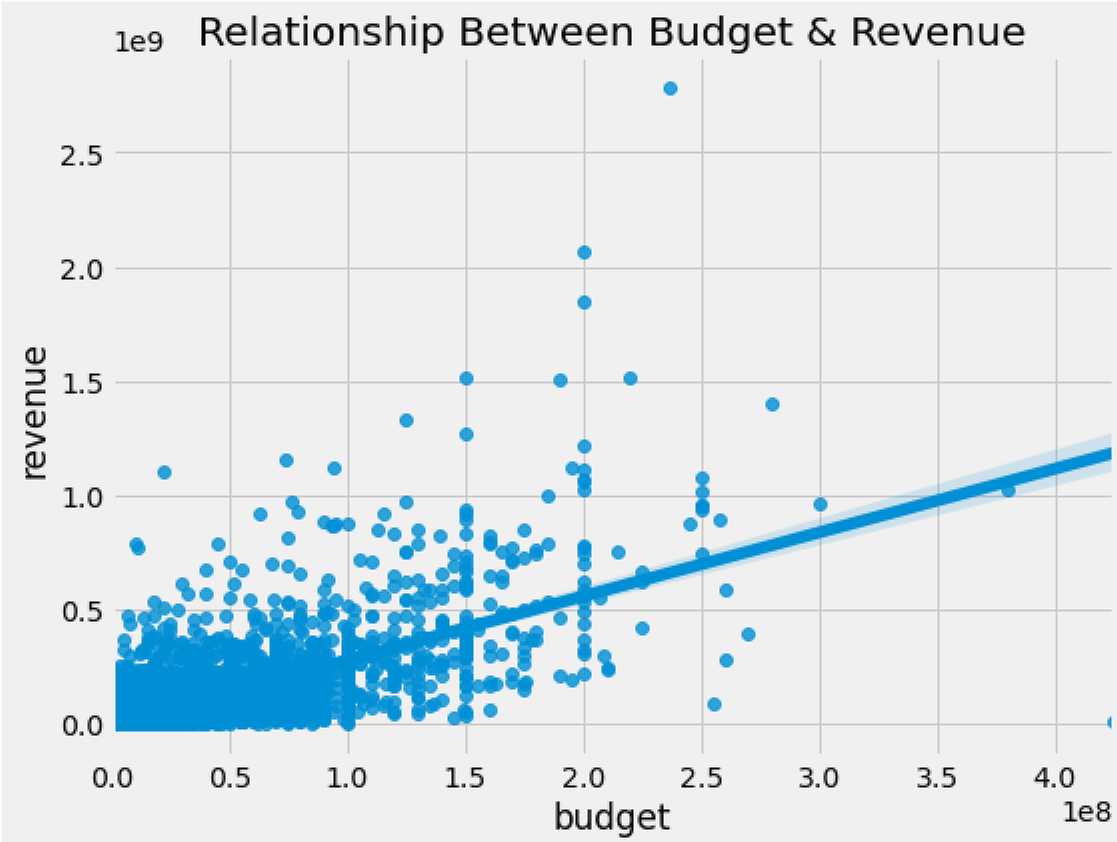


In [107]:

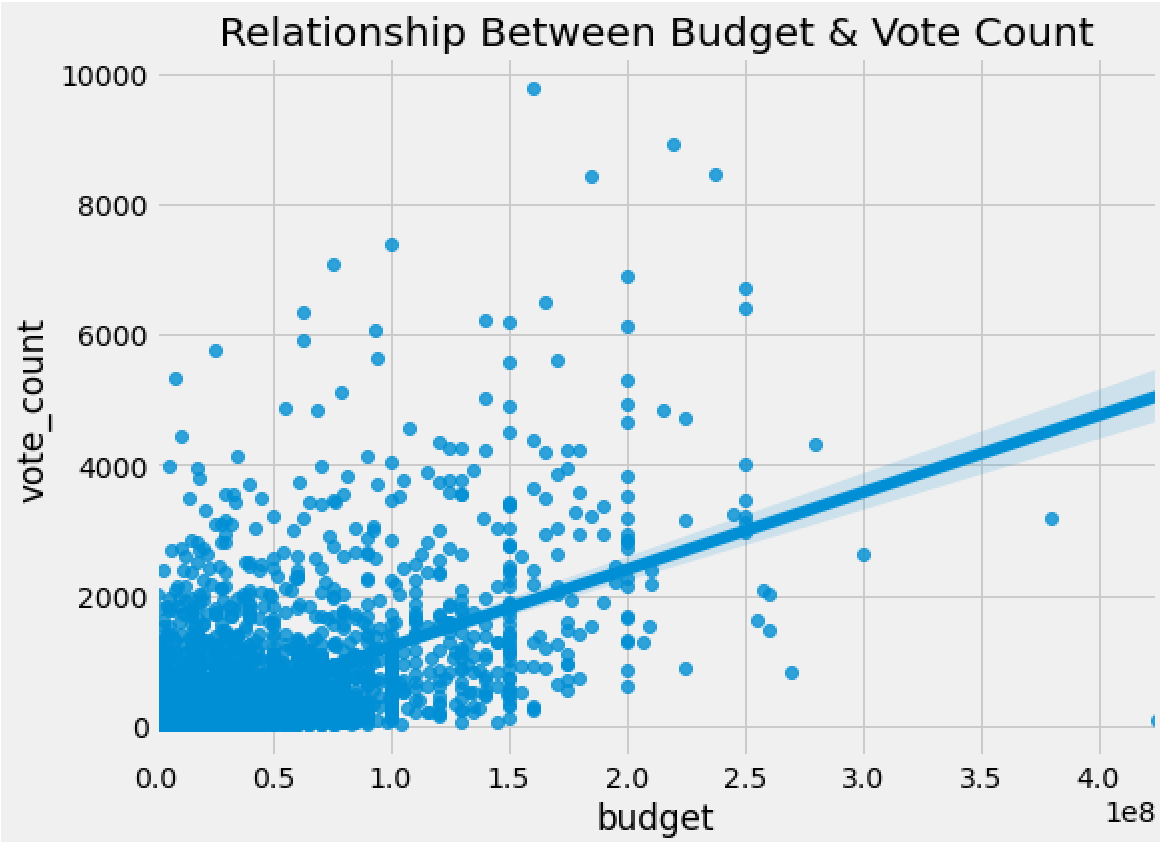
```
plt.figure(figsize=(8, 6))
plt.style.use('fivethirtyeight')
plt.title("Relationship Between Budget & Budget after inflation")
sns.regplot(x="budget", y="budget_adj", data=df)
plt.show()
```



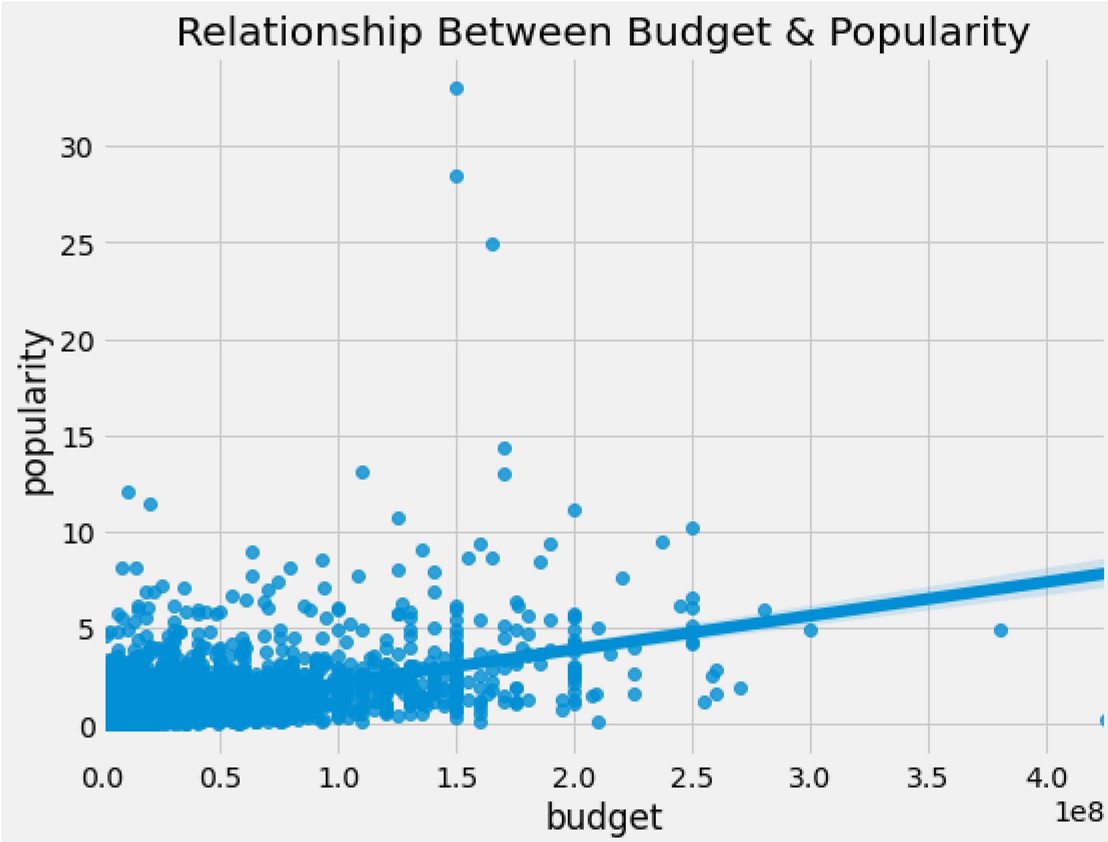
In [108]:
```
plt.figure(figsize=(8, 6))
plt.style.use('fivethirtyeight')
plt.title("Relationship Between Budget & Revenue")
sns.regplot(x="budget", y="revenue", data=df)
plt.show()
```



In [109]:
```
plt.figure(figsize=(8, 6))
plt.style.use('fivethirtyeight')
plt.title("Relationship Between Budget & Vote Count")
sns.regplot(x="budget", y="vote_count", data=df)
plt.show()
```



In [110]:
```
plt.figure(figsize=(8, 6))
plt.style.use('fivethirtyeight')
plt.title("Relationship Between Budget & Popularity")
sns.regplot(x="budget", y="popularity", data=df)
plt.show()
```



Budget is positively correlated with 'budget_adj', 'revenue', 'vote_count' and 'popularity'

Question 12: Which cast participated in the most successful/flopped movies ?

In [85]:
```
#successful movie:
list_3= s.sort_values('popularity', axis=0, ascending=False).head(1).cast.value
s
list_3
```

Out[85]:
```
array(["Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vincent D'Onofrio|Nick Robin
son"],
      dtype=object)
```

name of the movie:

In [86]:
```
s.sort_values('popularity', axis=0, ascending=False).head(1).original_title
```

Out[86]:
```
id
135397    Jurassic World
Name: original_title, dtype: object
```

In [87]:
```
#flopped movie:
list_4= flopped_movies.sort_values('popularity', axis=0, ascending=True).head(1
).cast.values
list_4
```

Out[87]:
```
array(['George C. Scott|Diana Rigg|Richard Dysart|Barnard Hughes|Stephen Elliot
t'],
      dtype=object)
```

name of the movie:

In [88]:
```
flopped_movies.sort_values('popularity', axis=0, ascending=True).head(1).origin
al_title
```

Out[88]:
```
id
32082    The Hospital
Name: original_title, dtype: object
```

Conclusions

In this project, I tried to analyse a dataset taken from The movie database (TMDb) of 10,000 movies.

The cleaning part is what I found the most interesting to do, I noticed that there are some `null values` and `duplicates` in the dataset, as well as `zero values` on some data points such as `budget` , `revenue` and `runtime` .

There are some movies having the budget and revenue equals to 0, these can be considered as NaN values. Normally we can drop these zero values if they don't have an impact on the overall information about the dataset, but since I checked with the help of Shannon's Entropy, I found that We don't need to drop all the rows containing zeros in budget and revenue columns, since increasing in shanon's entropy is an indication of information loss. Also, some of the movies have a revenue but the budget is zero and vice versa! and that is really not realistic. So i changed the value of each budget to the mean overall budget to make it more realistic, and vice versa.

I also noticed that the release date is an `object` and not in a `date_time type` , so I tried to change it but I encountered a problem where each release date written for example as '11/15/66' is going to be converted to '2066-11-15' instead of 1966 as noted in the release_year, so I changed each year part in the release_date to the year noted in the release_year feature to solve the problem.

There were some columns I did not need in further analysis such as : imdb_id since we already have an id column, homepage, tagline, overview, and keywords since we're not doing a movie recommendation in this project, so I dropped them.

I also noticed that the runtime also has some zero values which is unrealistic! to tackle this problem I replaced each zero runtime with the mean overall runtime from each year.

Here are answers to some questions asked earlier:

the most popular genres between 2015 and 1960 are : Action, Adventure, Science Fiction and Thriller

the top 5 movies having the most profit were: Jurassic World(2015) , Mad Max:Fury Road(2015), Insurgent(2015), Star Wars:The Force Awakens(2015) and Furious 7(2015).

the top 5 movies having the highest budget but the lowest vote counts: The Swan Princess: A Royal Family Tale(2014), Red Sky(2014), Mao's Last Dancer(2009), Sinners and Saints(2010) and Double Wedding(2010).

the top 5 movies who flopped in terms of profit: Jupiter Ascending, Ex Machina, Tomorrowland, Mr. Holmes and Room.

We can assume that Runtime and poularity aren't correlated based on graphs, also we can notice that having the most profit doesn't mean the movie is also popular; only 20% of the popular movies have also the highest profit.

Limitations:

The dataset has alot of `zero values` , `null values` and wrong `format` , it makes it a bit hard for me to conduct a proper analysis, nevetherless, I found that an interesting problem to tackle.

Also, having the revenue columns filled with values when the budget columns has alot of `zero values` and vice versa was a non realistic thing and a problem that I found equally interesting.

Another limitaion for me is conducting exploratory analysis with visualizations, I found that I'm more familiar with wrangling the data.

In [89]:
```
from subprocess import call
call(['python', '-m', 'nbconvert', 'Investigate_a_Dataset.ipynb'])
```

Out[89]: 0

Resources:

[geeksforgeeks (https://www.geeksforgeeks.org/selecting-rows-in-pandas-dataframe-based-on-conditions/)](https://www.geeksforgeeks.org/selecting-rows-in-pandas-dataframe-based-on-conditions/)

[pandas.plotting.scatter_matrix (https://pandas.pydata.org/pandas-docs/version/1.2.4/reference/api/pandas.plotting.scatter_matrix.html)](https://pandas.pydata.org/pandas-docs/version/1.2.4/reference/api/pandas.plotting.scatter_matrix.html)

[converting-object-to-datetime-format-in-python (https://stackoverflow.com/questions/38333954/converting-object-to-datetime-format-in-python)](https://stackoverflow.com/questions/38333954/converting-object-to-datetime-format-in-python)

[entropy-calculation-in-python (https://datascience.stackexchange.com/questions/58565/conditional-entropy-calculation-in-python-hyx)](https://datascience.stackexchange.com/questions/58565/conditional-entropy-calculation-in-python-hyx)

[pandas.DataFrame.explode (https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.explode.html)](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.explode.html)

[split-explode-pandas-dataframe-string-entry-to-separate-rows (https://stackoverflow.com/questions/12680754/split-explode-pandas-dataframe-string-entry-to-separate-rows)](https://stackoverflow.com/questions/12680754/split-explode-pandas-dataframe-string-entry-to-separate-rows)

[pandas.DataFrame.plot.pie. (https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.plot.pie.html)](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.plot.pie.html)

In [ ]: