PROJECT : INVESTIGATE A TMDb MOVIE DATASET

About the Data :

- This dataset contains the details about the movies that are released in various years
- For each movie , the data provides about 21 attributes initially
- There are more than 10,000 Movie details in this data set i.e rows

Findings :

1.Statistical Findings

2.Findings that can be represented with plots

Statistical Findings:

- Movie with highest profit ,its budget and revenue
- Movie with lowest profit , its budget and revenue
- Details of most profit gained movie and most loss gained movie
- Movie with most and least running time
- Number of movies released before year 2000 and number of movies released after year 2000
- Highest votes count , Least votes count , Average votes count

Findings that can be represented with plots:

- Running time of all movies
- Distribution of Genres
- Top 10 movies with most votes
- Top 10 movies with most profits
- Revenue V/s budget
- Revenue V/S popularity
- Revenue V/S Votes

- Revenue V/S profits

Description for Investigation :

- This data contains many attributes for a particular movie.
- All the attributes can't be useful for finding out some results.
- Some of the attributes will directly contribute to the statistical calculations with out any modifications.
- Where as some of them needs to be modified or removed i.e the data needs to be wrangled and cleaned which is an important aspect while performing operations on the data
- Data cleaning may include removal of data , modifying the data , changing the format of the data , extracting useful columns from the old columns etc.
- Data cleaning will help us to flexibly perform operations and obtain the output precisely

Some of the data cleaning process I did :

1. Removed the following columns:

    - **id :** id is not much useful to provide important information
    - **imdb_id :** Though it is imdb id , it does not play an important role in calculating the statistics
    - **budget_adj :** Not so useful
    - **revenue_adj** : Not so useful
    - **homepage :** It is just a website URL which contains simple movie description which is not helpful in statistics
    - **Production companies :** Though it is one of the factor for the movie Cast , it is not important for statistics
    - **Vote average :** Not much useful
    - **taglines :** Not much useful

```
'''REMOVING IRRELEVANT COLUMNS THAT ARE MENTIONED ABOVE'''
df=pd.read_csv('tmdb-movies.csv')
df= df.drop([ 'id', 'imdb_id', 'budget_adj','tagline','revenue_adj', 'homepage','production_companies', 'vote_average'],1)
df.head()
```

Properties of data before removing columns :

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10866 entries, 0 to 10865
Data columns (total 21 columns):
id                      10866 non-null int64
imdb_id                 10856 non-null object
popularity              10866 non-null float64
budget                  10866 non-null int64
revenue                 10866 non-null int64
original_title          10866 non-null object
cast                    10790 non-null object
homepage                2936 non-null object
director                10822 non-null object
tagline                 8042 non-null object
keywords                9373 non-null object
overview                10862 non-null object
runtime                 10866 non-null int64
genres                  10843 non-null object
production_companies    9836 non-null object
release_date            10866 non-null object
vote_count              10866 non-null int64
vote_average            10866 non-null float64
release_year            10866 non-null int64
budget_adj              10866 non-null float64
revenue_adj             10866 non-null float64
dtypes: float64(4), int64(6), object(11)
memory usage: 1.7+ MB
```

Properties of data after removing the irrelevant columns :

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10866 entries, 0 to 10865
Data columns (total 13 columns):
popularity        10866 non-null float64
budget            10866 non-null int64
revenue           10866 non-null int64
original_title    10866 non-null object
cast              10790 non-null object
director          10822 non-null object
keywords          9373 non-null object
overview          10862 non-null object
runtime           10866 non-null int64
genres            10843 non-null object
release_date      10866 non-null object
vote_count        10866 non-null int64
release_year      10866 non-null int64
dtypes: float64(1), int64(5), object(7)
memory usage: 1.1+ MB
```

2.Changing the date format using to_date function :

Before changing the format :

release_date
6/9/15
5/13/15
3/18/15
12/15/15

After changing the format :

releasedate
2015-06-09
2015-05-13
2015-03-18

3.Removing the duplicate values

Checking duplicate values and removing the duplicates

```
'''Checking and Removing the duplicate rows'''
a=df.shape[0]
df=df.drop_duplicates()
b=df.shape[0]
print("Number of columns before removing duplicates is ",a)
print("Number of columns after removing duplicates is ",b)
```

```
Number of columns before removing duplicates is  10866
Number of columns after removing duplicates is  10865
```

(It has 1 duplicate value since 10866-10865 =1)

4.Removing NULL's and Zero's from revenue , budget and runtime column

```
'''REMOVING NAN AND ZERO VALUES '''
df = df[df.budget !=0]# using  boolean technique ,respective rows which has budget column value zero is removed
df=df[df.revenue !=0]# using  boolean technique ,respective rows which has revenue column value zero is removed
df=df[df.runtime!=0]
df.dropna()#removing  null values if any
print(df.shape)#Updated rows and columns
df
```

Finally the shape of data is :

```
print("The shape is ")
print(df.shape)#Updated rows and columns
```

```
The shape is
(3855, 13)
```

Results for Findings :

1. Movie with highest profit , its budget and revenue

```
'''Movie with highest profit , its budget and revenue'''
df['profit']=df['revenue']-df['budget']
high_profit_movie=df.loc[df['profit'].idxmax()]['original_title']
high_profit_revenue=df.loc[df['profit'].idxmax()]['revenue']
high_profit_budget=df.loc[df['profit'].idxmax()]['budget']
high_profit=df.loc[df['profit'].idxmax()]['profit']
print("The movie with highest profit is ",high_profit_movie)
print("The profit is : ",high_profit)
print("The budget is : ",high_profit_budget)
print("The revenue is : ",high_profit_revenue)
```

```
The movie with highest profit is  Avatar
The profit is :  2544505847
The budget is :  237000000
The revenue is :  2781505847
```

2. Movie with lowest profit , its budget and revenue :

```
'''Movie with least profit , its budget and revenue'''
least_profit=df.loc[df['profit'].idxmin()]['profit']
least_profit_movie=df.loc[df['profit'].idxmin()]['original_title']
least_profit_revenue=df.loc[df['profit'].idxmin()]['revenue']
least_profit_budget=df.loc[df['profit'].idxmin()]['budget']
print("The movie with least profit is ",least_profit_movie)
print("The profit is ",least_profit)
print("The budget is : ",least_profit_budget)
print("The revenue is : ",least_profit_revenue)
```

```
The movie with least profit is  The Warrior's Way
The profit is  -413912431
The budget is :  425000000
The revenue is :  11087569
```

3. Details of most profit gained movie and most loss gained movie:

```
'''The details of the most profit movie'''
high_profit_details=pd.DataFrame(df.loc[df['profit'].idxmax()])
display(high_profit_details)
```

	1386
popularity	9.43277
budget	237000000
revenue	2781505847
original_title	Avatar
cast	Sam Worthington|Zoe Saldana|Sigourney Weaver|S...
director	James Cameron
keywords	culture clash|future|space war|space colony|so...
overview	In the 22nd century, a paraplegic Marine is di...
runtime	162
genres	Action|Adventure|Fantasy|Science Fiction
vote_count	8458
release_year	2009
releasedate	2009-12-10 00:00:00
profit	2544505847

```
'''The details of the least profit movie'''
least_profit_details=pd.DataFrame(df.loc[df['profit'].idxmin()])
display(least_profit_details)
```

	2244
popularity	0.25054
budget	425000000
revenue	11087569
original_title	The Warrior's Way
cast	Kate Bosworth|Jang Dong-gun|Geoffrey Rush|Dann...
director	Sngmoo Lee
keywords	assassin|small town|revenge|deception|super speed
overview	An Asian assassin (Dong-gun Jang) is forced to...
runtime	100
genres	Adventure|Fantasy|Action|Western|Thriller
vote_count	74
release_year	2010
releasedate	2010-12-02 00:00:00
profit	-413912431

4. Movie with most and least running time

```
'''Movie's running time'''
print("Highest Running time movie : ",df.loc[df['runtime'].idxmax()]['original_title'])
print("Less Running time movie : ",df.loc[df['runtime'].idxmin()]['original_title'])
```

```
Most Running time movie :  Carlos
Less Running time movie :  Kid's Story
```

5. Number of movies released before year 2000 and number of movies released after year 2000

```
'''Movies released before 2000 and movies released after 2000'''
x=df.loc[df['release_year']>=2000].shape[0]
y=df.loc[df['release_year']<2000].shape[0]
print("Number of movies released after year 2000 is ",x)
print("Number of movies released before year 2000 is ",y)
```

```
Number of movies released after year 2000 is  2500
Number of movies released before year 2000 is  1354
```

6. Highest votes count , Least votes count , Average votes count

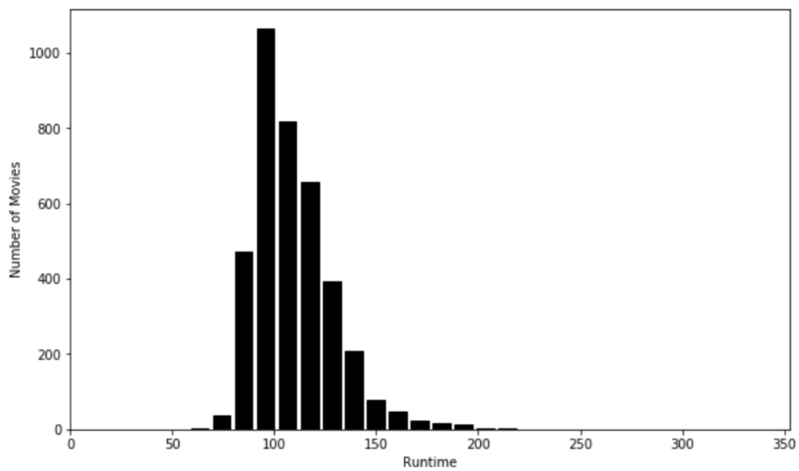
```python
'''Votes statistics'''
print("Highest Votes : ",df['vote_count'].max())
print("Movie : ",df.loc[df['vote_count'].idxmax()]['original_title'])
print("Lowest Votes : ",df['vote_count'].min())
print("Movie : ",df.loc[df['vote_count'].idxmin()]['original_title'])
```

```
Highest Votes :  9767
Movie :  Inception
Lowest Votes :  10
Movie :  Beautiful
```

Run time of all Movies :

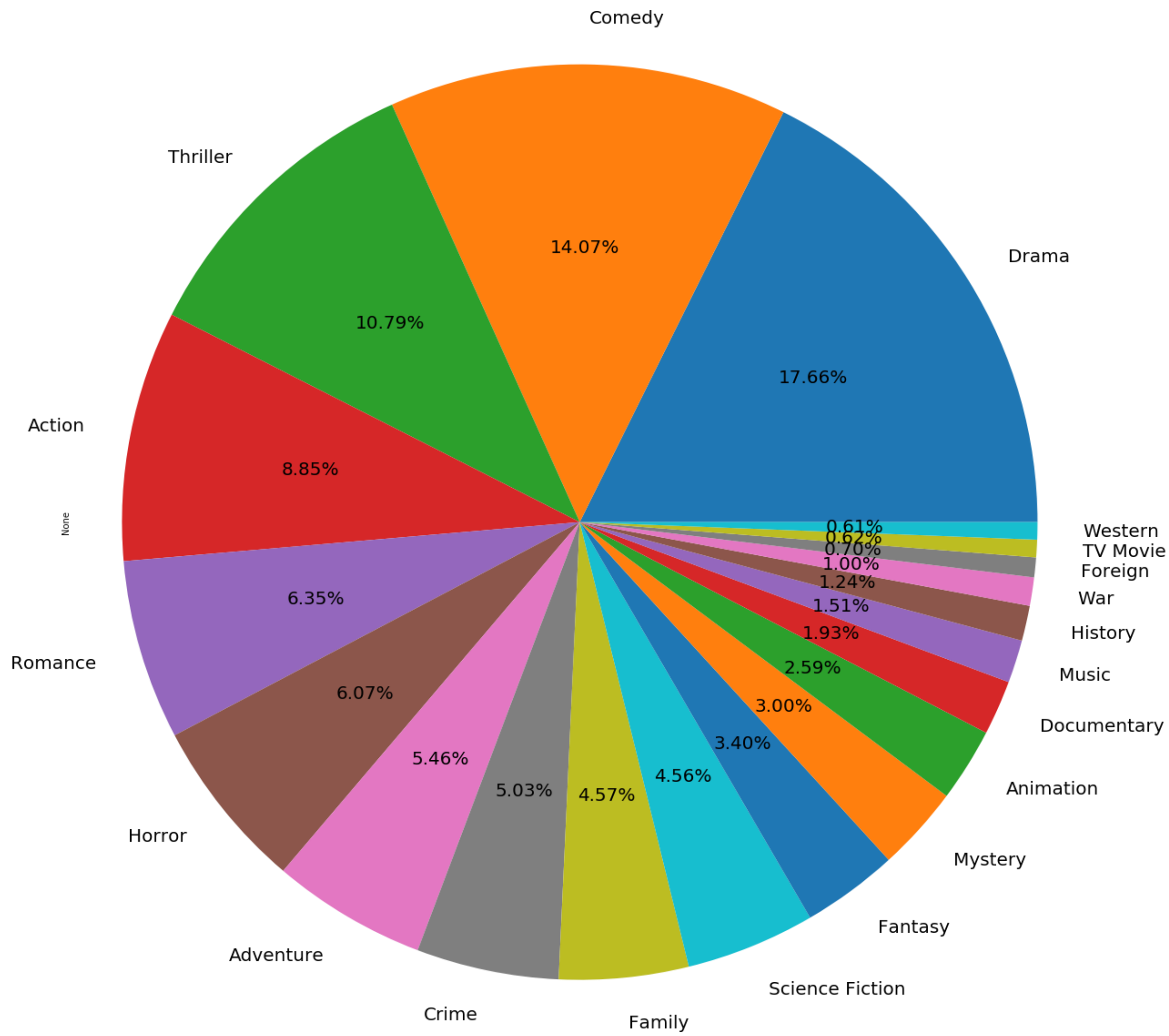
```python
'''Running time of All movies representing with a histogram'''

plt.figure(figsize=(10,6))#Setting the figure size
plt.xlabel('Runtime')#Labelling the x variable
plt.ylabel('Number of Movies')
plt.hist(df['runtime'],rwidth=0.8,bins=30,color = "black", ec="black")
plt.show()
```

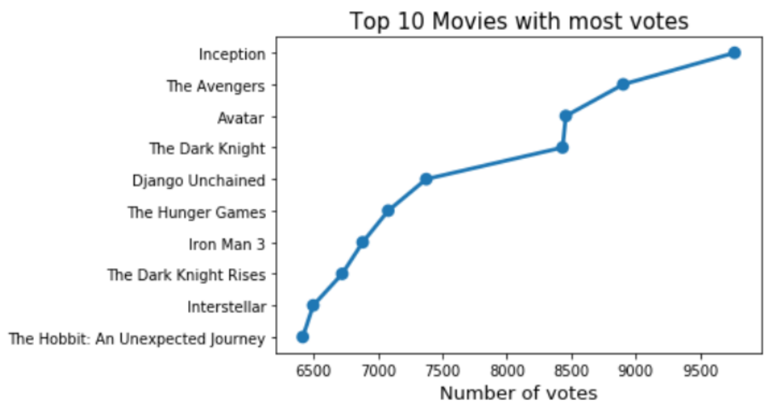


Distribution of Genres :

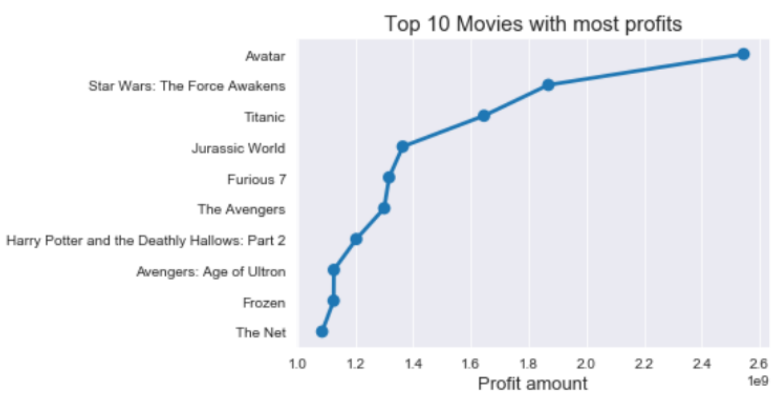
```python
plt.figure(figsize=(18,18))#Setting the figure size
x=pd.Series(df['genres'].str.cat(sep = '|').split('|'))
x=x.value_counts()
x.plot.pie(
       autopct='%2.2f%%', textprops={'fontsize': 20})
plt.axis('equal')
plt.tight_layout()
plt.show()
```



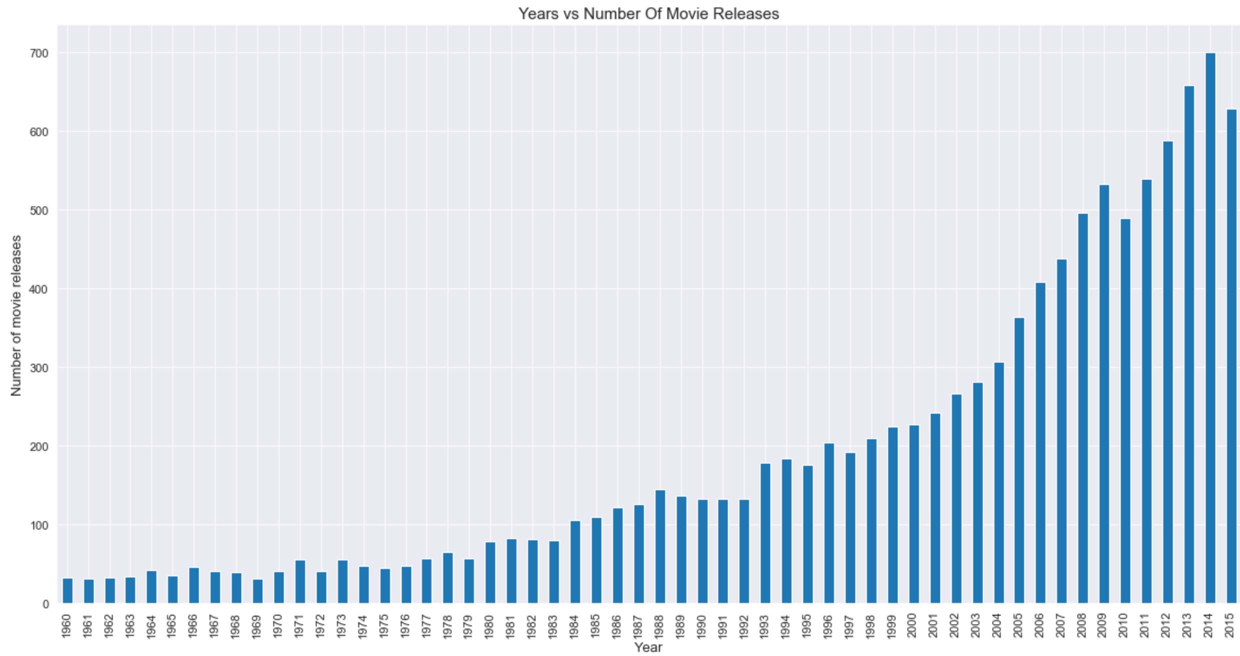
Top 10 movies with most votes :



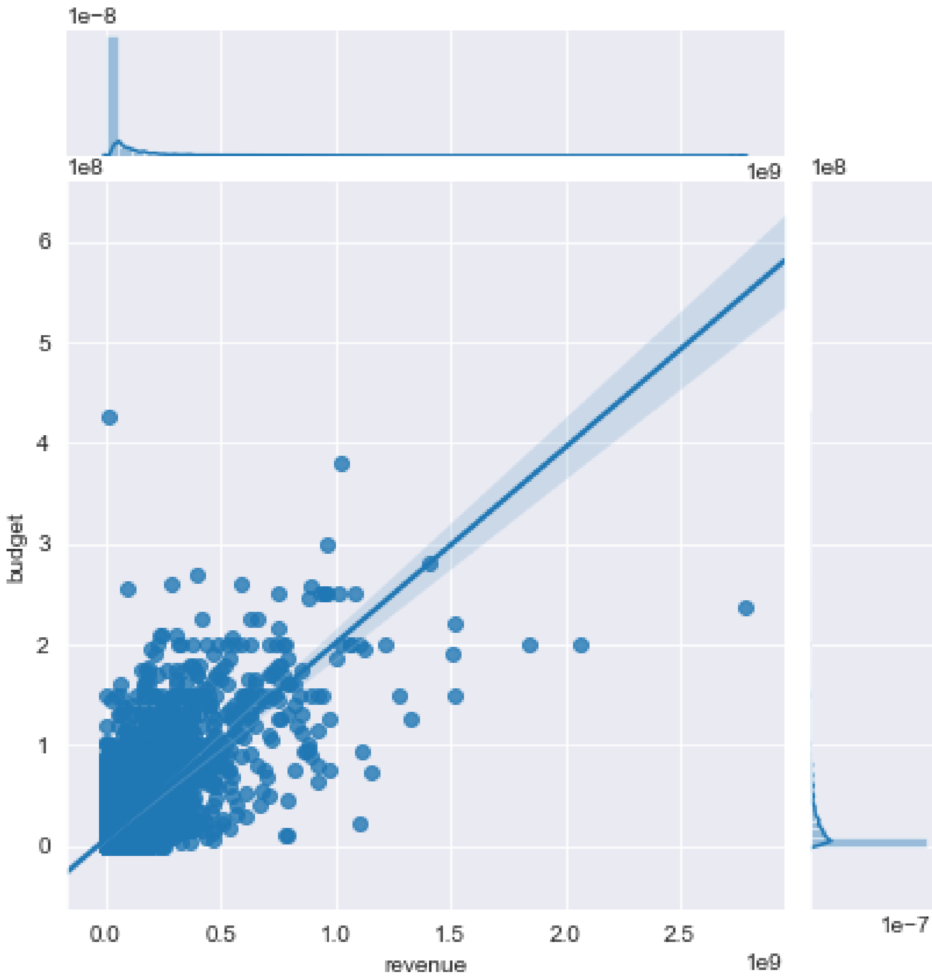
Top 10 movies with most profits :



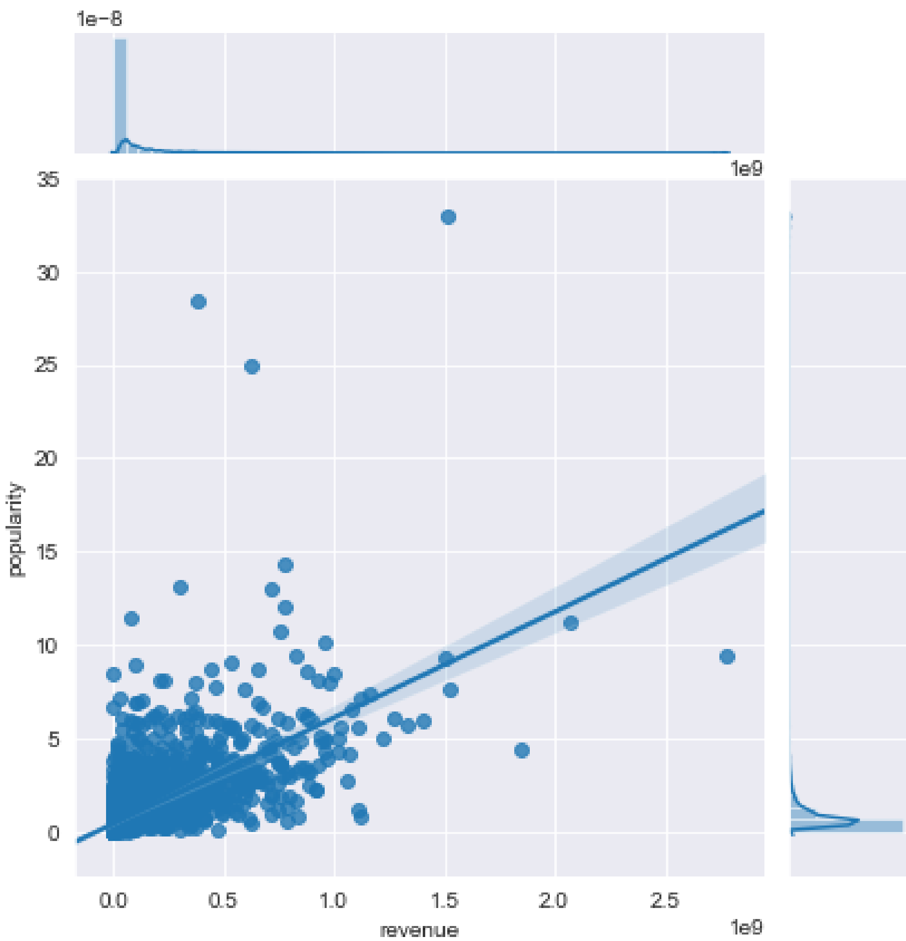
Number of Movie releases per an year :



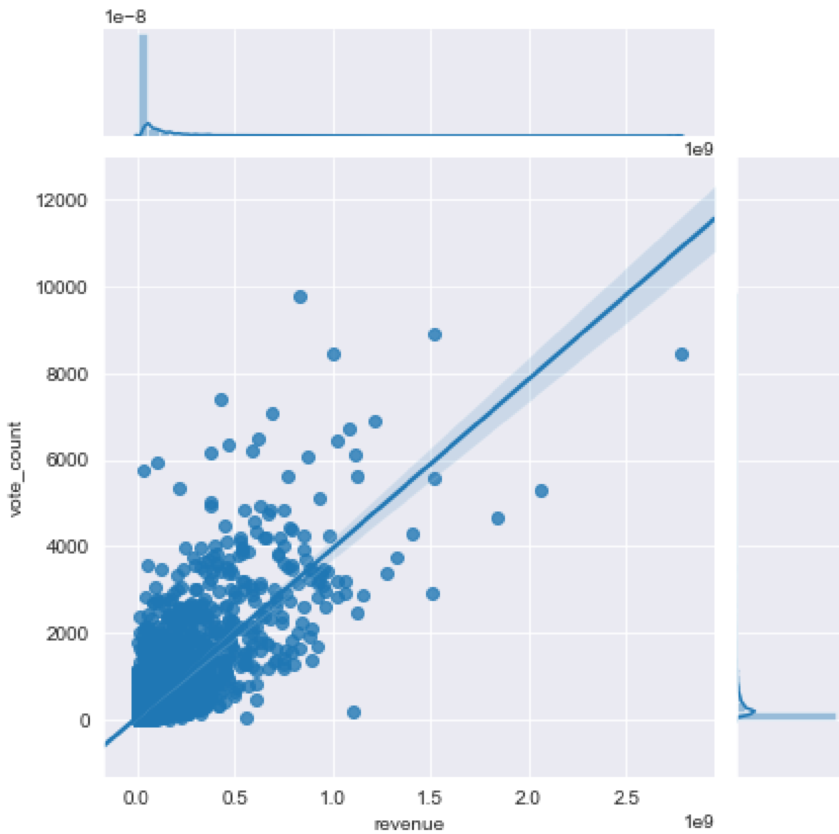
Revenue V/S Budget :



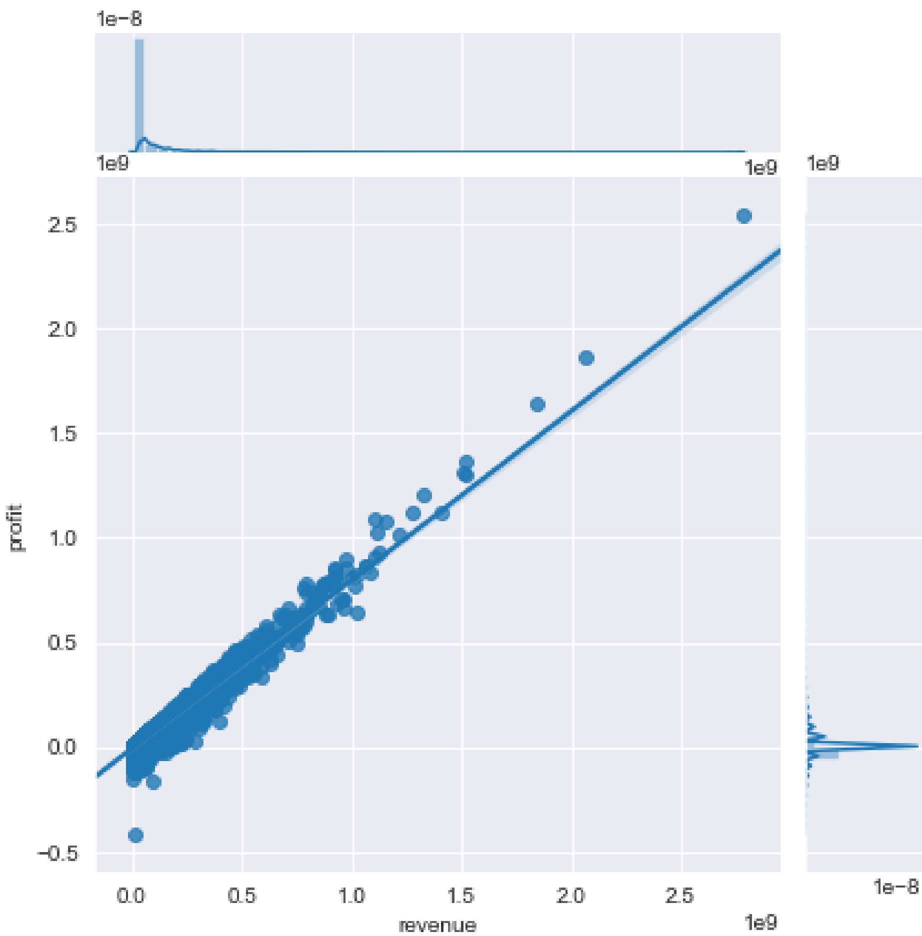
Revenue V/S popularity :



Revenue V/S Votes :



Revenue V/S profit :



Conclusions:

- Avatar movie has most profits
- Inception movie has most number of votes
- Among all the Genres , Drama is most popular
- The year in which more number of movies are released is 2014
- The movies with runtime around 100 are more in number
- Revenue is directly proportional to budget
- The Genre that is hardly used is Western