Film Industry Analysis

Questions addressed in the analysis:

- How has the no. of movies made and the monetary trend of the industry changed in the past 55 years?
- What are the most profitable movies of the data collected?
- Do movies of a particular Genre gain more profit?
- What are the movies that incurred huge loss?
- Are movies of a particular Genre incurring more loss?
- Average values related to profitable and non-profitable movies.
- Is there a relation between average vote and profit gained by a movie? Do movies with high average vote always gain high profit?

Data Wrangling

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
import seaborn as sns
sns.set_style('darkgrid')
```

In [2]:

```
df=pd.read_csv("tmdb-movies.csv")
```

In [3]:

```
df.release_year.min()
```

Out[3]:

```
1960
```

In [4]:

```
df.release_year.max()
```

Out[4]:

```
2015
```

The data has been collected from 1960-2015.

In [5]:

```
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
```

In [6]:

```
#checking for duplicate rows
df.duplicated().sum()
```

Out[6]:

```
1
```

Data Cleaning

Issues to be addressed:

- Duplicate row present
- Budget and revenue not having int datatype
- There are many columns which will not be useful for the desired analysis.
- Rows having 0 in budget or revenue.

In [7]:

```
# drop duplicate row
df.drop_duplicates(inplace=True)
df.duplicated().sum()
```

Out[7]:

```
0
```

In [8]:

```
#dropping undesired columns
delete=['imdb_id','homepage','tagline','keywords','overview','vote_count','budget_adj','rev
df.drop(delete,axis=1,inplace=True)
```

In [9]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 10865 entries, 0 to 10865
Data columns (total 13 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   id                    10865 non-null  int64
 1   popularity            10865 non-null  float64
 2   budget                10865 non-null  int64
 3   revenue               10865 non-null  int64
 4   original_title        10865 non-null  object
 5   cast                  10789 non-null  object
 6   director              10821 non-null  object
 7   runtime               10865 non-null  int64
 8   genres                10842 non-null  object
 9   production_companies  9835 non-null   object
 10  release_date          10865 non-null  object
 11  vote_average          10865 non-null  float64
 12  release_year          10865 non-null  int64
dtypes: float64(2), int64(5), object(6)
memory usage: 933.7+ KB
```

In [10]:

```
#dropping rows with budget or revenue as 0
df['budget']=df['budget'].replace(0,np.NAN)
df['revenue']=df['revenue'].replace(0,np.NAN)
df.dropna(subset=['budget','revenue'], inplace = True)
```

In [11]:

```
#changing data type :
change_type=['budget', 'revenue']
df[change_type]=df[change_type].applymap(np.int64)
```

In [12]:

```
df.dtypes
```

Out[12]:

```
id                        int64
popularity              float64
budget                    int64
revenue                   int64
original_title           object
cast                     object
director                 object
runtime                   int64
genres                   object
production_companies     object
release_date             object
vote_average            float64
release_year              int64
dtype: object
```

Exploratory Data Analysis

Monetary trend and No.of movies made in past 55 years (1960-2015)

In [13]:

```
df.insert(4,'profit',df['revenue']-df['budget'])
```

In [14]:

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

Out[14]:

	id	popularity	budget	revenue	profit	original_title	cast	direct
0	135397	32.985763	150000000	1513528810	1363528810	Jurassic World	Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi...	Co Trevorro
1	76341	28.419936	150000000	378436354	228436354	Mad Max: Fury Road	Tom Hardy|Charlize Theron|Hugh Keays-Byrne|Nic...	Geor Mil

In [15]:

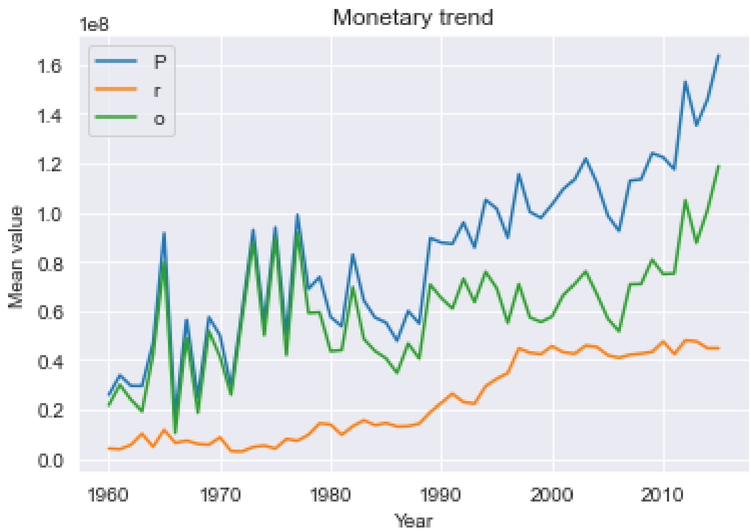
```
profits = df.groupby('release_year').mean()['profit']
budgets = df.groupby('release_year').mean()['budget']
revenues = df.groupby('release_year').mean()['revenue']
```

In [16]:

```
year=df['release_year'].unique().tolist()
year.sort(reverse=False)
```

In [17]:

```
plt.plot(year,revenues,label="Revenues")
plt.plot(year,budgets,label="Budgets")
plt.plot(year,profits,label="Profits")
plt.xlabel('Year')
plt.ylabel('Mean value')
plt.title('Monetary trend')
plt.legend('Profit')
plt.show()
```



In [18]:

```
#No. of movies made per year
x=df['release_year'].value_counts()
x
```

Out[18]:

```
2011    199
2013    180
2010    178
2009    174
2006    169
2008    167
2014    165
2007    165
2005    163
2015    160
2012    158
2004    147
2002    127
2003    121
2001    121
1999    116
2000    106
1998     92
1997     90
1996     86
1995     81
1993     72
1994     62
1988     57
1990     53
1992     53
1989     51
1991     50
1986     48
1987     46
1984     42
1985     41
1983     31
1981     30
1982     26
1980     23
1977     19
1978     17
1979     16
1976     15
1971     13
1967     13
1974     13
1973     12
1970     11
1961     10
1975      9
1968      9
1972      8
1964      7
1962      7
1963      6
```

```
1965      5
1966      5
1960      5
1969      4
Name: release_year, dtype: int64
```

General function to find top movies as per category

In [19]:

```python
def percentile(col):
    x=np.percentile(df[col],75)
    df_high=df[df[col] > x]
    df_high.sort_values(by=[col],inplace=True,ascending=False)
    return df_high
```

Most profitable movies:

In [20]:

```python
df_pro=percentile('profit')
df_pro
```

Out[20]:

	id	popularity	budget	revenue	profit	original_title	cast	director	ru
1386	19995	9.432768	237000000	2781505847	2544505847	Avatar	Sam Worthington|Zoe Saldana|Sigourney Weaver|S...	James Cameron	
3	140607	11.173104	200000000	2068178225	1868178225	Star Wars: The Force Awakens	Harrison Ford|Mark Hamill|Carrie Fisher|Adam D...	J.J. Abrams	
5231	597	4.355219	200000000	1845034188	1645034188	Titanic	Kate Winslet|Leonardo DiCaprio|Frances Fisher|...	James Cameron	
							Chris Pratt|Bryce		

In [21]:

```
# renaming columns of most profitable movies dataframe
df_pro.rename(columns=lambda x: x[:] + "_pro", inplace=True)

#confirming changes
df_pro
```

cast_pro	director_pro	runtime_pro	genres_pro	production_companies_pro	release_da
Sam 'orthington|Zoe ana|Sigourney Weaver|S...	James Cameron	162	Action|Adventure|Fantasy|Science Fiction	Ingenious Film Partners|Twentieth Century Fox ...	1
Harrison Ford|Mark Hamill|Carrie sher|Adam D...	J.J. Abrams	136	Action|Adventure|Science Fiction|Fantasy	Lucasfilm|Truenorth Productions|Bad Robot	1
Kate nslet|Leonardo Caprio|Frances Fisher|...	James Cameron	194	Drama|Romance|Thriller	Paramount Pictures|Twentieth Century Fox Film ...	1
ıris Pratt|Bryce					

Do movies of a particular Genre gain more profit?

In [22]:

```
drama=df_pro['genres_pro'].str.contains('Drama').sum()
action=df_pro['genres_pro'].str.contains('Action').sum()
adventure=df_pro['genres_pro'].str.contains('Adventure').sum()
romance=df_pro['genres_pro'].str.contains('Romance').sum()
thriller=df_pro['genres_pro'].str.contains('Thriller').sum()
scifi=df_pro['genres_pro'].str.contains('Science Fiction').sum()
family=df_pro['genres_pro'].str.contains('Family').sum()
war=df_pro['genres_pro'].str.contains('War').sum()
animation=df_pro['genres_pro'].str.contains('Animation').sum()
fantasy=df_pro['genres_pro'].str.contains('Fantasy').sum()
crime=df_pro['genres_pro'].str.contains('Crime').sum()
comedy=df_pro['genres_pro'].str.contains('Comedy').sum()
western=df_pro['genres_pro'].str.contains('Western').sum()
li=[drama,action,adventure,romance,thriller,scifi,family,war,comedy,fantasy,western,animati
li_names=["drama","action","adventure","romance","thriller","scifi","family","war","comedy"
li
```

Out[22]:

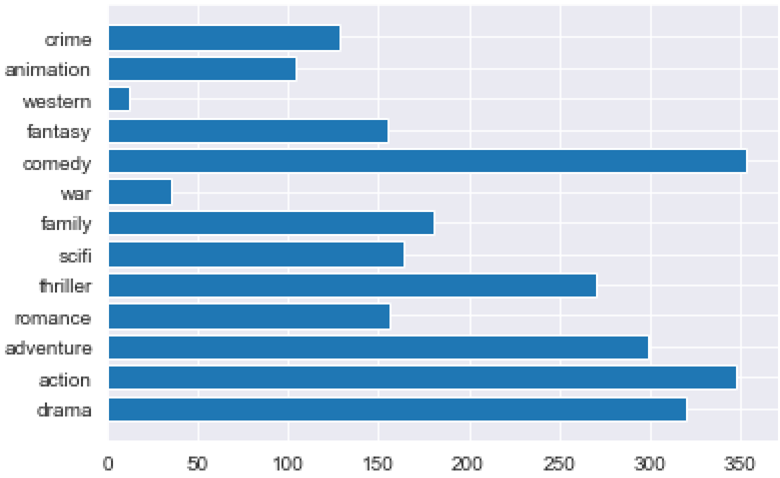
```
[320, 348, 299, 156, 270, 164, 180, 35, 353, 155, 12, 104, 129]
```

In [23]:

```
plt.barh(li_names,li)
```

Out[23]:

```
<BarContainer object of 13 artists>
```



General function to find average values related to profitable movies:

In [24]:

```
def avg(col):
    return df_pro[col].mean()
```

In [25]:

```
#Average budget of profitable movies:
avg('budget_pro')
```

Out[25]:

```
69666148.45746888
```

In [26]:

```
#Average vote of profitable movies:
avg('vote_average_pro')
```

Out[26]:

```
6.44740663900415
```

In [27]:

```
##Average runtime of profitable movies:
avg('runtime_pro')
```

Out[27]:

115.21473029045643

General function to find lowest movies as per category

In [28]:

```
def low_per(col):
    x=np.percentile(df[col],25)
    df_low=df[df[col] < x]
    df_low.sort_values(by=[col],inplace=True,ascending=False)
    return df_low
```

Movies that incurred huge loss:

In [29]:

```
df_low=low_per('profit')
df
```

Out[29]:

	id	popularity	budget	revenue	profit	original_title	cast	directo
0	135397	32.985763	150000000	1513528810	1363528810	Jurassic World	Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi...	Colin Trevorrov
1	76341	28.419936	150000000	378436354	228436354	Mad Max: Fury Road	Tom Hardy|Charlize Theron|Hugh Keays-Byrne|Nic...	George Mille
2	262500	13.112507	110000000	295238201	185238201	Insurgent	Shailene Woodley|Theo James|Kate Winslet|Ansel...	Rober Schwentke
3	140607	11.173104	200000000	2068178225	1868178225	Star Wars: The Force Awakens	Harrison Ford|Mark Hamill|Carrie Fisher|Adam D...	J.J Abrams

In [30]:

```
# renaming columns of most profitable movies dataframe
df_low.rename(columns=lambda x: x[:] + "_low", inplace=True)

# confirm changes
df_low
```

Out[30]:

	id_low	popularity_low	budget_low	revenue_low	profit_low	original_title_low	cast_low	di
5932	209901	0.411515	3000000	1675381	-1324619	Nothing Left to Fear	Clancy Brown|James Tupper|Anne Heche|Ethan Pec...	
7776	1961	0.422526	1500000	173066	-1326934	My Name Is Bruce	Bruce Campbell|Grace Thorsen|Ted Raimi|Adam Bo...	
5133	321	0.276911	4361898	3031801	-1330097	Mambo Italiano	Luke Kirby|Ginette Reno|Paul Sorvino|Mary Wals...	
							Dennis	

Are movies of a particular Genre incurring more loss?

In [31]:

```
drama=df_low['genres_low'].str.contains('Drama').sum()
action=df_low['genres_low'].str.contains('Action').sum()
adventure=df_low['genres_low'].str.contains('Adventure').sum()
romance=df_low['genres_low'].str.contains('Romance').sum()
thriller=df_low['genres_low'].str.contains('Thriller').sum()
scifi=df_low['genres_low'].str.contains('Science Fiction').sum()
family=df_low['genres_low'].str.contains('Family').sum()
war=df_low['genres_low'].str.contains('War').sum()
animation=df_low['genres_low'].str.contains('Animation').sum()
fantasy=df_low['genres_low'].str.contains('Fantasy').sum()
crime=df_low['genres_low'].str.contains('Crime').sum()
comedy=df_low['genres_low'].str.contains('Comedy').sum()
western=df_low['genres_low'].str.contains('Western').sum()
li=[drama,action,adventure,romance,thriller,scifi,family,war,comedy,fantasy,western,animati
li_names=["drama","action","adventure","romance","thriller","scifi","family","war","comedy"
li
```

Out[31]:

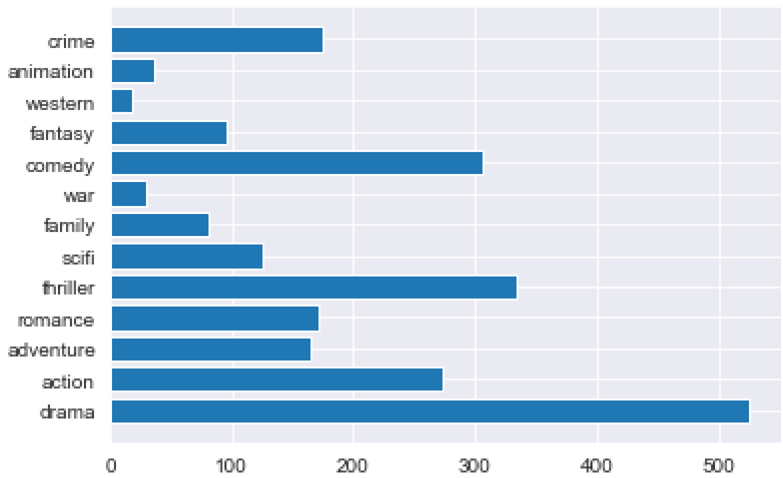
```
[525, 274, 165, 172, 335, 125, 81, 30, 307, 95, 18, 36, 175]
```

In [32]:

```
plt.barh(li_names,li)
```

Out[32]:

```
<BarContainer object of 13 artists>
```



In [33]:

```
#### General function to find average values related to unprofitable movies:
```

In [34]:

```
def avg(col):
    return df_low[col].mean()
```

In [35]:

```
#Average budget of unprofitable movies:
avg('budget_low')
```

Out[35]:

```
31623838.817427386
```

In [36]:

```
#Average vote of unprofitable movies
avg('vote_average_low')
```

Out[36]:

```
5.842323651452282
```

In [37]:

```
##Average runtime of unprofitable movies:
avg('runtime_low')
```

Out[37]:

```
107.42116182572614
```

Relation between Average Vote and Profit Gained

Do movies with high average vote always gain high profit?

In [38]:

```
df['profit'].corr(df['vote_average'])
```

Out[38]:

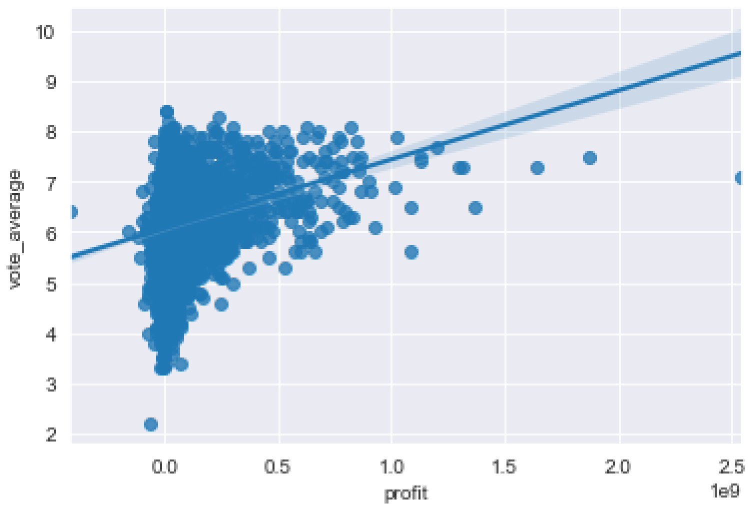
```
0.25943499037670154
```

In [39]:

```
sns.regplot('profit','vote_average',data=df)
```

Out[39]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1b849f28>
```



This implies a weak positive correlation between average vote and profit gained.

Conclusions:

- The budget, revenue and profit have been increasing since 1960.It is apparently because the no. of movies made during 1960 (5 movies) to 2015 (160 movies) has increased a lot.
- Comedy and action movies have gained more profit than any other genres.
- The average budget of profitable movies ia much higher than unprofitable ones.

- Drama movies have incurred way too much loss as compared to other genres.
- Movies with high average vote could have gained profit or might have incurred loss.Movies with high average vote do not imply that they would have gained a lot.

Limitations:

- Many rows have been dropped as budget or revenue were 0.
- The currency of budget and revenue have not been mentioned.Different countries differ in the value of the currency used. *