TMDB Movie Database - Analysis

Project Hypothesis : Budget spent on movies And Revenue generated have increased considerably over time are in positive correlation with one another.

Questions Posed:

- What is Growth tendency followed by Movie Budget spent over time?
- Discuss the tendency of revenue generated from movies, has it increased,decreased or stayed in constant proportion over time?
- Does Budget spent on a movie affect or influence Revenue earnings of that movie?
- How are these parameters related to one another? ### Questions based on genre/genres of movies:
- Which genre/genres movies have grossed the most revenue overtime?
- Study the proportion in which movies have been made over time. Considering genre/genres count of movies(frequency) falling into each particular genre.

Highlighting top 5 entities in each of below cases:

- Which genres have grossed most revenue over given time?
- Which genres have most budget spent into making movies over time?
- Which genre/s has/ve gained the most popularity over time ?
- Did you conclude your hypothesis?

Project Outlines and Steps Followed :

```
* Gathering data from provided CSV.
* Data Cleaning and Engineering using Python on Google Colab.
* Dropping Un-neccesary Columns and Dividing columns with heterogeneous da
ta into new individual columns.
* Following our hypothesis. I have further plotted graphs and have mention
ed key take aways from each one of them as we move along this project.
* Python Scripts with Concludding graphs can be easily found below.
```

Importing Required Library and provided DB : TMDB database (CSV file)

In [ ]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from matplotlib.pyplot import figure
figure(figsize=(8, 6), dpi=80)
%matplotlib inline
df=pd.read_csv('/content/drive/MyDrive/tmdb-movies.csv')
df.head(1)
```

Movies DB Gathered till 2015

In [ ]:

```
df.tail(1)
```

Movies DB Gathered from 1966

Initial DB analysis for factors such as MAX, MIN and MEAN of values per given column

In [ ]:

```
df.describe()
```

Acquiring info about Null/Non-Null attributes per column

In [ ]:

```
df.info()
```

Dropping non-required columns from the Database

In [ ]:

```
df.drop(['imdb_id','keywords','budget_adj','revenue_adj','overview','tagline','homepage','cast', 'director','original_title'], axis = 1 , inplace= True)
```

Checking for new Engineered Database with only suitable columns

In [ ]:

```
df.head(1)
```

Splitting 'Genres' (heterogeneous) columns in new individual columns for better Data Analysis and Trend study

In [ ]:

```
df[['genre1','genre2','genre3','genre4','genre5']] = df.genres.str.split('|', expand=True)
```

Splitting 'Production Companies' (heterogeneous) columns in new individual columns for better Data Analysis and Trend study

In [ ]:

```
df[['prod1','prod2','prod3','prod4','prod5']] = df.production_companies.str.split('|', expand=True)
```

Now, Checking for desired columns in our database after splitting above columns.

In [ ]:

```
df.head(1)
```

Further dropping two of columns 'genres' and 'production_companies',

After splitting data into new individual columns and storing in our database.

In [ ]:

```
df.drop(['genres','production_companies'], axis = 1 , inplace= True)
```

In [ ]:

```
df.head(1)
```

Analysing Null / Non-Null values in Engineered Data base.

In [ ]:

```
df.info()
```

Plotting Graphs against Time Period for Trend Analysis.

Runtime VS Time Period Graph Analysis .

As inffered from Graph below, here are my Analysed points :

- Mean Run Time has remained constant for large part of our provided data.
- There have been some outlier values which are evident in the years 1985,2001,2002,2005 and 2011.
- The Mean Run Time per movie can be seen to be around 150 Mins per year.

In [ ]:

```
plt.figure(figsize = (20,10))
plt. bar(df['release_year'],df['runtime'])
plt.xlabel('Time Period (Year)')
plt.ylabel('Run Time in Mins')
plt.title('Movie Run Time Trend')
```

Revenue VS Time Period Graph Analysis .

As inffered from Graph below, here are my Analysed points :

- An Increase in Revenue tendency can be inferred from the graph below.
- The Gross mark of 1 Bn. USD is seen to be crossed only after the year 1995, and significant growth is seen overall.
- Revenue peek is seen in the year : 2010

In [ ]:

```
plt.figure(figsize = (20,10))
plt. bar(df['release_year'],df['revenue'])
plt.xlabel('Time Period (Year)')
plt.ylabel('Revenue spent X 1 Billion USD')
plt.title('Revenue Trend')
```

Budget VS Time Period Graph Analysis .

As inffered from Graph below, here are my Analysed points :

- An Increase in budget tendency can be inferred from the graph below.
- The Gross mark of 1 Bn. USD is seen to be crossed only after the year 1995, and significant growth is seen overall.
- Revenue peek is seen in the year : 2010

In [ ]:

```
plt.figure(figsize = (20,10))
plt. bar(df['release_year'],df['budget'])
plt.xlabel('Time Period (Year)')
plt.ylabel('Budget spent X 100 Million USD')
plt.title('Budget Trend')
```

Conclusion from above Graphs and Analysis :

- Both Budget and Revenue Indices have increased over time and share the same increasing tendency.
- Both Graphs show a spike in the year 2009.
- Above, two graphs prove our hypothesis about Budget,Revenue Correlation and its nature of being directly proportional on one another.

  (Above given conclusions concern certain exceptions, but share above stated results over all.)

In [ ]:

```
df.head(1)
```

Now, we will be adressing scenarios based on Genres in the given time period in Database.

A pie chart inferring, the frequency of genre/genres(overall) with their percentage count in the pie diagram.

Given below we can find these conclusions :

- 'Action, Adventure, Crime, Drama, Thriller' have been the most frequent made genre combination for a movie in the given period (1960 - 2015).
- Above genre combination is followed by these 4 genre combination movies :
  - 'Action, Crime, Drama, Mystery, Thriller'
  - 'Adventure, Animation, Comedy, Family, Fantasy'
  - 'Animation, Adventure, Comedy, Family, Fantasy'
  - 'Action, Adventure, Crime, Drama, Mystery'

In [ ]:

```
plt.figure(figsize = (60, 60))
df[['genre1','genre2','genre3','genre4','genre5']].value_counts().plot.pie()
```

Given Pie charts infer to genres with gross revenue amounts associated.

- Based on below pie chart indices we can clearly see given 'Adventure, Animation, Comedy, Family, Fantasy' genres grossed highest revenue sum through out the periiod.

In [ ]:

```
plt.figure(figsize = (60, 60))
df.groupby(['genre1', 'genre2', 'genre3', 'genre4', 'genre5']).revenue.sum().plo
t.pie()
```

- Based on below pie chart indices we can clearly see given 'Crime, Drama, Mystery, Thriller, Action' genres grossed highest revenue mean through out the periiod.

In [ ]:

```
plt.figure(figsize = (60, 60))
df.groupby(['genre1', 'genre2', 'genre3', 'genre4', 'genre5']).revenue.mean().pl
ot.pie()
```

- Moving ahead, we can see the movie genres per year along with their popularity mean indexed on Y-axis of the below graph. And we can infer to the most popular movies per year from 1960 to 2015

Function used for plotting Popularity and Budget means over year.

In [ ]:

```
def funcplot (x):
  if( x== df.popularity.all ):
    plt.figure(figsize = (60, 10))
    df.groupby(['release_year','genre1', 'genre2', 'genre3', 'genre4', 'genre5'
]).popularity.mean().plot.bar()
  elif (x== df.budget.all(skipna=True) ):
    plt.figure(figsize = (60, 10))
    df.groupby(['release_year','genre1', 'genre2', 'genre3', 'genre4', 'genre5'
]).budget.mean().plot.bar()
```

In [ ]:

```
funcplot(df.popularity.all)
```

- Next, we can find the movies along with their budget means for the movie genres in our database. Further, we can also find maximum mean of the budgets spent per movie genre per year.

In [ ]:

```
funcplot(df.budget.all(skipna=True))
```

Conclusion to our Hypothesis.

Budget and Revenue are positively Correlated and have direct proportional relation between them.

This can finally be verrified using Dataframe.corr() method and checking for correlation between the above two. Now Since here the value is '0.734901' (which is positive), giving a satisfying conclusion to our hypothesis.

In [ ]:

```
df.corr()
```

Sources Reffered :

- Stackoverflow.com
- numpy.org
- geekforgeeks.org
- w3schools.com
- pandas.pydata.com