TMDB MOVIE DATA ANALYSIS

December 2, 2019

1 The Following Analysis Analyses TMDB MOVIE DATA.

[ ]:
```
# At first we will import all the required packages
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
% matplotlib inline
```

We will now be providing certain information regarding the dataset we are going to analyse.

The data set we are going to work on is TMDB MOVIE DATASET. The dataset contains information about **10866** movies and different attributes regarding those (such as their **runtime, cast, budget, revenue etc**.) The following analysis will be based on this dataset.

[7]:
```
# We will now load the dataset required
df = pd.read_csv('tmdb-movies (1).csv')
df.head(1)
```

[7]:
```
       id    imdb_id  popularity     budget     revenue original_title  \
0  135397  tt0369610   32.985763  150000000  1513528810 Jurassic World

                                                cast  \
0  Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi…

                        homepage          director            tagline  …  \
0  http://www.jurassicworld.com/  Colin Trevorrow  The park is open.  …

                                            overview runtime  \
0  Twenty-two years after the events of Jurassic …     124

                                      genres  \
0  Action|Adventure|Science Fiction|Thriller

                                production_companies release_date vote_count  \
0  Universal Studios|Amblin Entertainment|Legenda…       6/9/15       5562
```

```
   vote_average  release_year    budget_adj   revenue_adj
0           6.5          2015  1.379999e+08  1.392446e+09

[1 rows x 21 columns]
```

2 QUESTIONS

Here we will be posing the questions, which we will try to address in the following analysis.

Q1. Are movies with greater budget more popular?

Q2. Are movies generating greater revenue more popular?

Q3. Is runtime having any correlation with the popularity of the movie?

3 EXPLORING THE DATASET

At first we will **explore** our dataset and see what all it reveals. We will look at **different attributes of the dataset** and will see how to clean it for the further analysis

[8]:
```
# We first see different attributes of different columns in the dataset.
df.info()
```

```
<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
```

The main information important here is that columns are having many **null values** and different columns are having **different data types**(Which would not require much change), where **object and float are the most prevalent**. Let us see how many null values these columns are having.

[11]:
```
# Showcasing number of null values columns are having.
df.isnull().sum()
```

[11]:
```
id                         0
imdb_id                   10
popularity                 0
budget                     0
revenue                    0
original_title             0
cast                      76
homepage                7930
director                  44
tagline                 2824
keywords                1493
overview                   4
runtime                    0
genres                    23
production_companies    1030
release_date               0
vote_count                 0
vote_average               0
release_year               0
budget_adj                 0
revenue_adj                0
dtype: int64
```

We see that columns like **Homepage, Tagline, Keywords, Production Companies** are having **many null values**. But we also know that these columns are not very important for our analysis. So we will drop these columns and take care of remaining null values further when we clean the data.

[12]:
```
# Now we will showcase different statistical attributes of the dataset.
df.describe()
```

[12]:
```
                 id    popularity        budget       revenue       runtime  \
count  10866.000000  10866.000000  1.086600e+04  1.086600e+04  10866.000000
mean   66064.177434      0.646441  1.462570e+07  3.982332e+07    102.070863
std    92130.136561      1.000185  3.091321e+07  1.170035e+08     31.381405
min        5.000000      0.000065  0.000000e+00  0.000000e+00      0.000000
25%    10596.250000      0.207583  0.000000e+00  0.000000e+00     90.000000
50%    20669.000000      0.383856  0.000000e+00  0.000000e+00     99.000000
```

```
75%     75610.000000      0.713817  1.500000e+07  2.400000e+07    111.000000
max    417859.000000     32.985763  4.250000e+08  2.781506e+09    900.000000

         vote_count  vote_average  release_year    budget_adj   revenue_adj
count  10866.000000  10866.000000  10866.000000  1.086600e+04  1.086600e+04
mean     217.389748      5.974922   2001.322658  1.755104e+07  5.136436e+07
std      575.619058      0.935142     12.812941  3.430616e+07  1.446325e+08
min       10.000000      1.500000   1960.000000  0.000000e+00  0.000000e+00
25%       17.000000      5.400000   1995.000000  0.000000e+00  0.000000e+00
50%       38.000000      6.000000   2006.000000  0.000000e+00  0.000000e+00
75%      145.750000      6.600000   2011.000000  2.085325e+07  3.369710e+07
max     9767.000000      9.200000   2015.000000  4.250000e+08  2.827124e+09
```

The above chart reveals that there is a **lot of difference in the min and max popularity , budget, revenue, runtime, release year etc.** Also the dataset entails data regarding **10866 movies**. The data is **suited well for our analyses**.

[13]:
```python
# Seeing how many duplicate rows are there in the dataset
sum(df.duplicated())
```

[13]:
```
1
```

So there is **one duplicate row**, which we will be dropping while cleaning the data.

4 CLEANING THE DATASET

We will now go about cleaning the dataset to **make our analysis simpler and easy to communicate and understand**.

[37]:
```python
# Let us first drop the columns that our not required for our analysis
df.drop(['id', 'imdb_id','keywords','tagline', 'cast', 'homepage','overview',␣
 ↪'genres', 'production_companies', 'release_date', 'vote_average',␣
 ↪'budget_adj', 'revenue_adj'], axis = 'columns', inplace = True)
```

[39]:
```python
df.head()
```

[39]:
```
   popularity     budget     revenue                original_title  \
0   32.985763  150000000  1513528810                Jurassic World
1   28.419936  150000000   378436354            Mad Max: Fury Road
2   13.112507  110000000   295238201                     Insurgent
3   11.173104  200000000  2068178225  Star Wars: The Force Awakens
4    9.335014  190000000  1506249360                     Furious 7

          director  runtime  vote_count  release_year
0  Colin Trevorrow      124        5562          2015
1    George Miller      120        6185          2015
2 Robert Schwentke      119        2480          2015
3      J.J. Abrams      136        5292          2015
```

```
4          James Wan      137         2947          2015
```

[40]:
```
# Let us now see how many null values are there in each column.
df.isnull().sum()
```

[40]:
```
popularity         0
budget             0
revenue            0
original_title     0
director          44
runtime            0
vote_count         0
release_year       0
dtype: int64
```

Only director column is having null values which is again not a problem as far as our analysis is concerned so we will leave it as it is.

[45]:
```
# Now we will take a look at duplicate rows.
sum(df.duplicated())
```

[45]:
```
1
```

[47]:
```
# Let us now drop this row.
df.drop_duplicates(inplace = True)
```

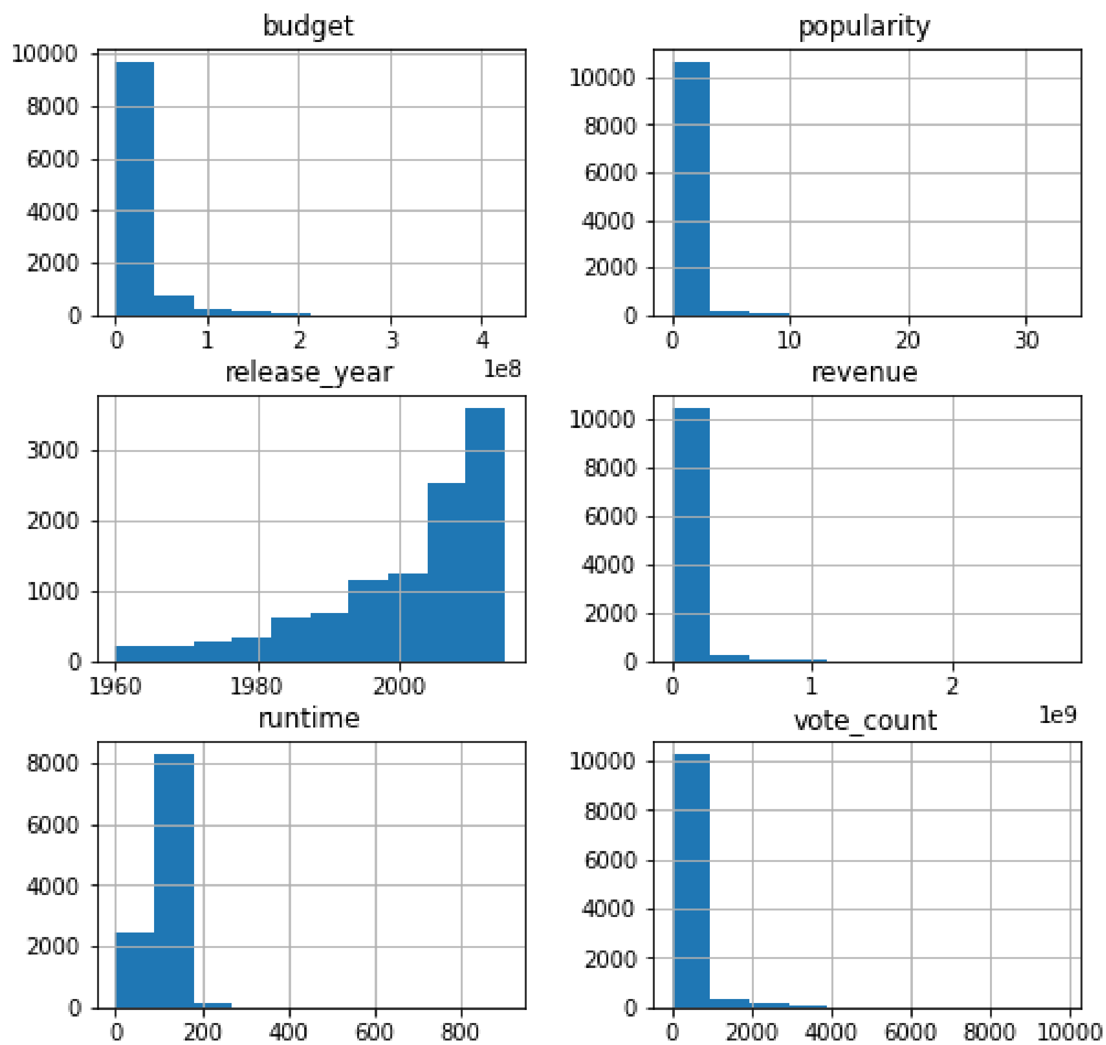
[48]:
```
sum(df.duplicated())
```

[48]:
```
0
```

The **duplicate row has been dropped**.

The **data now looks clean for further analysis**.

[50]:
```
df.hist(figsize = (8,8));
```



We see that **histograms of budget, popularity, revenue, runtime and vote_count are skewed to the right**. This reveals that **lesser number of movies generated greater budget, popularity, revenue and vote_count, greater number of movies have a runtime of 0 to 200. The histogram of release_year is skewed to the left, which reveals that more number of movies were released in 21st century**.

NOW OUR DATA IS READY FOR THE ANALYSIS

Population is taken to be the **dependent variable**. So we will change the **population into a categorical variable**. This would make our **analysis** much **easier** and would lead to a **much better representation** and a **clearer answering** to our questions.

[53]:
```
# Changing Population into a categorical variable.
bin_edges = [0, 10, 20, 30, 40]
bin_names = ['low', 'medium', 'high', 'very high']
df['popularity_n'] = pd.cut(df['popularity'], bin_edges, labels = bin_names)
```

[54]:
```
df.head()
```

[54]:
```
   popularity     budget     revenue                original_title  \
0   32.985763  150000000  1513528810                Jurassic World
1   28.419936  150000000   378436354            Mad Max: Fury Road
2   13.112507  110000000   295238201                     Insurgent
3   11.173104  200000000  2068178225  Star Wars: The Force Awakens
4    9.335014  190000000  1506249360                     Furious 7

           director  runtime  vote_count  release_year popularity_n
0   Colin Trevorrow      124        5562          2015    very high
1     George Miller      120        6185          2015         high
2  Robert Schwentke      119        2480          2015       medium
3       J.J. Abrams      136        5292          2015       medium
4         James Wan      137        2947          2015          low
```

Here we have formed a new column **popularity_n** which is a **categorical variable of the column popularity**. **It categorizes popularity as low, medium, high and very high**.

5 NOW WE ARE ALL SET TO ADDRESS OUR QUESTIONS :-

Q1. ARE MOVIES WITH GREATER BUDGET MORE POPULAR?

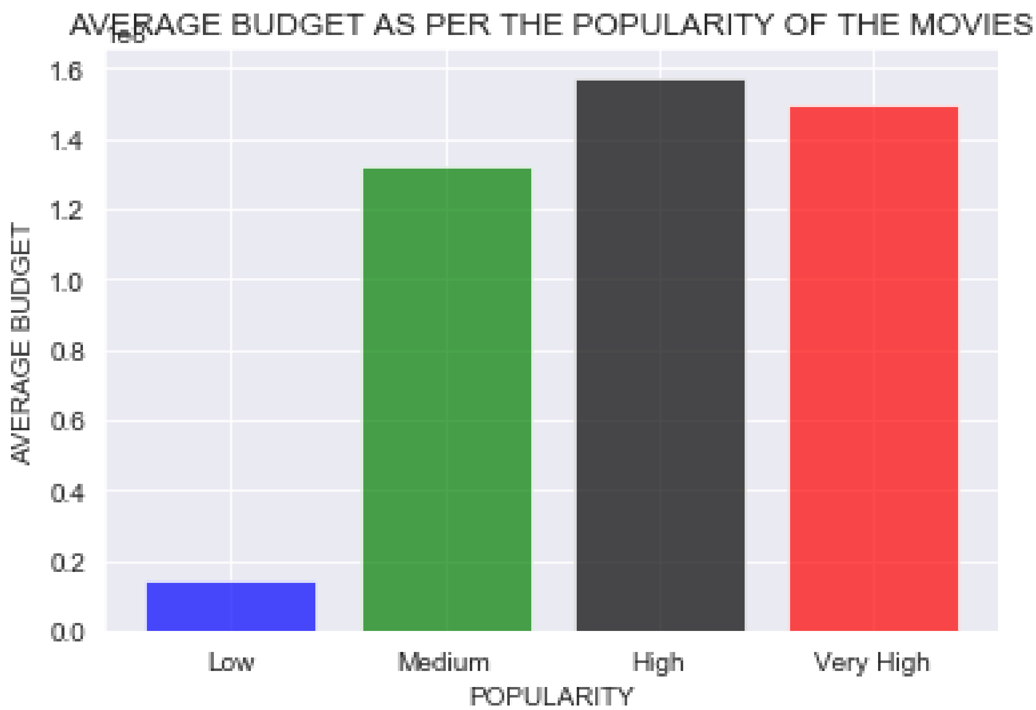
In order to address this question, we will use **pandas groupby funtion to group the dataframe by popularity**, and will keep average budget alongside.

[55]:
```
df.groupby('popularity_n')['budget'].mean()
```

[55]:
```
popularity_n
low            1.449897e+07
medium         1.320000e+08
high           1.575000e+08
very high      1.500000e+08
Name: budget, dtype: float64
```

We will now **visualise these results** with a **bar graph** using **Matplotlib**.

[67]:
```
sns.set_style('darkgrid')
names = ['Low', 'Medium', 'High', 'Very High']
values = [1.449897e+07, 1.320000e+08, 1.575000e+08, 1.500000e+08]
plt.bar([1, 2, 3, 4], values, tick_label = names, color = ['blue', 'green',␣
 ↪'black', 'red'], alpha = 0.7)
plt.xlabel('POPULARITY')
plt.ylabel('AVERAGE BUDGET')
plt.title('AVERAGE BUDGET AS PER THE POPULARITY OF THE MOVIES');
```



Our above analysis reveals that ***movies with high and very high popularity have required greater average budget to be made**.

So the answer to our question is that , movies with higher popularity require greater budget.

[ ]:

```

```

Q2. ARE MOVIES GENERATING GREATER REVENUE MORE POPULAR?

In order to address this question, we will use **pandas groupby funtion to group the dataframe by popularity**, and will keep average revenue alongside.

[68]:

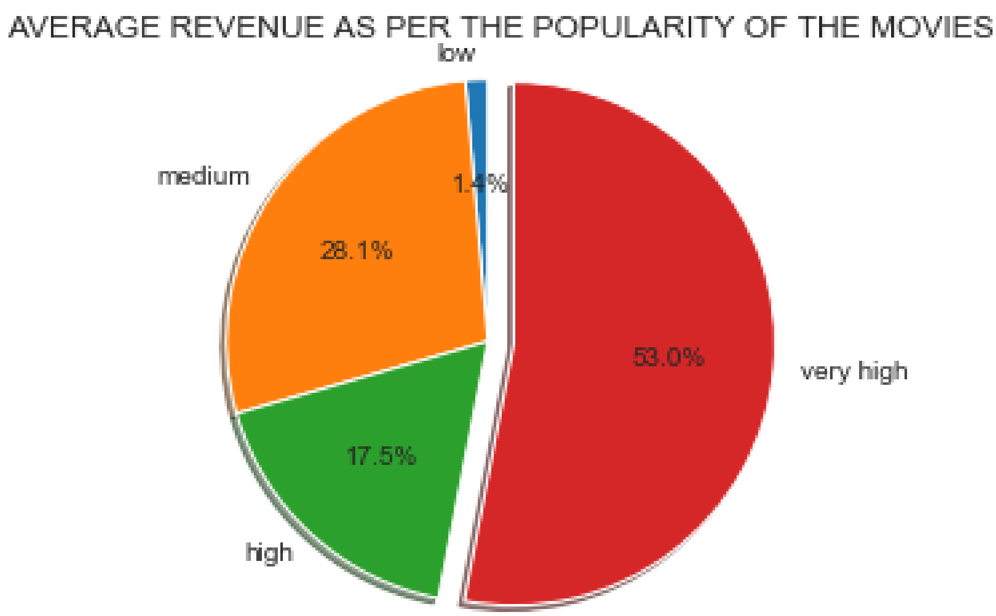
```
df.groupby('popularity_n')['revenue'].mean()
```

[68]:

```
popularity_n
low          3.904484e+07
medium       8.016067e+08
high         5.000944e+08
very high    1.513529e+09
Name: revenue, dtype: float64
```

We will now **visualise these results** using a **pie chart** with **Matplotlib**.

[74]:
```
sizes = [3.904484e+07, 8.016067e+08, 5.000944e+08, 1.513529e+09]
labels =['low', 'medium', 'high', 'very high']
explode = [0, 0, 0, 0.1]
plt.pie(sizes, labels= labels, explode= explode, autopct = '%1.1f%%', shadow =␣
 ↪True, startangle = 90)
plt.title('AVERAGE REVENUE AS PER THE POPULARITY OF THE MOVIES')
plt.axis('equal');
```



Our analysis reveals that, movies with **very high popularity** have earned the **largest average revenue** followed by movies with medium and high popularity.

So the answer to our question is that Yes, movies with very high popularity have generated greater revenue but movies with medium popularity have generated greater revenue than movies with high popularity(A possible reason could be a later hike in the popularity which went unrecorded or the music album of those movies became very popular, generating greater revenues).

[ ]:
```

```

Q3. IS RUNTIME HAVING ANY CORRELATION WITH THE POPULARITY OF THE MOVIES?

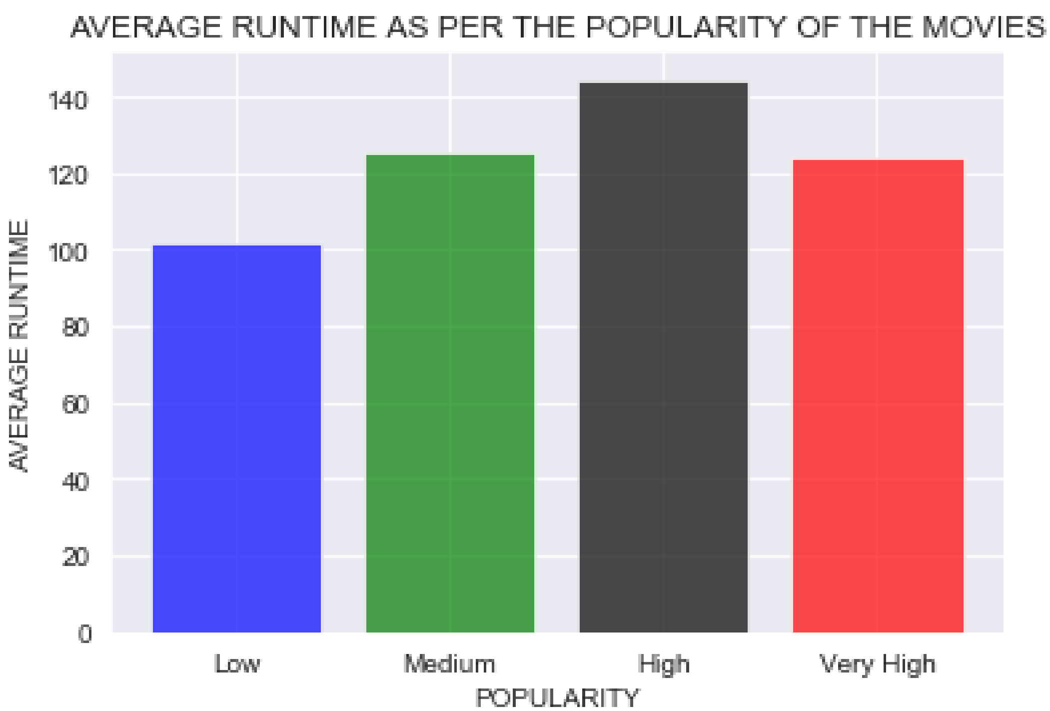
In order to address this question, we will use **pandas groupby funtion to group the dataframe by popularity**, and will keep average runtime alongside.

[77]:
```
df.groupby('popularity_n')['runtime'].mean()
```

[77]:
```
popularity_n
low          102.04496
medium       125.12500
high         144.50000
very high    124.00000
Name: runtime, dtype: float64
```

We will again **visualise these results** with a **bar graph** using **Matplotlib**.

[78]:
```
sns.set_style('darkgrid')
names = ['Low', 'Medium', 'High', 'Very High']
values = [102.04496, 125.12500, 144.50000, 124.00000]
plt.bar([1, 2, 3, 4], values, tick_label = names, color = ['blue', 'green',␣
↪'black', 'red'], alpha = 0.7)
plt.xlabel('POPULARITY')
plt.ylabel('AVERAGE RUNTIME')
plt.title('AVERAGE RUNTIME AS PER THE POPULARITY OF THE MOVIES');
```



The above analysis reveals that **height of the bars first increases and then decreases** showing that **there is no visible correlation between the popularity and average runtime of the movies**, but the movies generally with a higher runtime are more popular.

So the answer to our question is that there is no specific correlation between popularity and runtime of the movies.

6 CONCLUSION

Our analysis is now complete. **Analysis reveals that** :-

1. Movies with **higher popularity require a greater budget to be made and also generate greater revenues**. On the basis of this observation, following things can be concluded :-
    - People prefer movies with great cinematic shots, a good and popular cast and the movies that are shot on good locations (all these require a greater budget).
    - People prefer to see highly popular movies in theatre and they popularise it among there fellow mates, which attracts greater population to see the movie, hence generating greater revenues.
2. There is **no correlation between popularity and runtime of the movies, but the movies generally with a higher runtime are more popular**. This observation concludes that :-
    - People do not mind sitting behind the screen for long hours if they find a movie to be good :).

7 LIMITATIONS

The following are the **limitations to the analysis** :- 1. We have **dropped the columns revenue_adj** and budget_adj, i.e. **we have not accounted for inflation**. Though inflation inclusion is outside the scope of our analysis but it could have revealed interesting things. 2. We also **did not take into account the vote_count column** in our analysis. It could also have revealed certain things about the dataset.