Investigate_a_Dataset

August 23, 2021

1 Project : Investigate Movies Dataset

1.1 Table of Contents

Introduction
    Data Wrangling
    Exploratory Data Analysis
    Conclusions
    ## Introduction

```
This investigaton is about TMDb dataset which contains more than 10000 movies to make a compariz
```

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

    ## Data Wrangling

1.1.1 General Properties

```
In [2]: # Load movies data and print out a few lines.
        df = pd.read_csv('tmdb-movies.csv')
        df.shape
```

```
Out[2]: (10866, 21)
```

```
In [3]: #check the information about db
        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
```

```
In [4]: #nmber of duplicated
        sum(df.duplicated())
```

```
Out[4]: 1
```

```
In [5]: #define column with null values
        df.columns[df.isnull().any()]
```

```
Out[5]: Index(['imdb_id', 'cast', 'homepage', 'director', 'tagline', 'keywords',
               'overview', 'genres', 'production_companies'],
              dtype='object')
```

1.1.2 Data Cleaning

Drop unwanted columns

```
In [6]: #drop all the specified columns
        df.drop(['imdb_id', 'homepage', 'tagline', 'keywords','production_companies','overview',
        #check data again
        df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10866 entries, 0 to 10865
Data columns (total 14 columns):
id                10866 non-null int64
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
runtime           10866 non-null int64
genres            10843 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(4)
memory usage: 1.2+ MB
```

Drop nulls

```
In [7]: #drop null values
        df.dropna(inplace=True)
```

```
In [8]: #check again if any column has any null values
        df.columns[df.isnull().any()]
```

```
Out[8]: Index([], dtype='object')
```

Drop duplicated data

```
In [9]: #drop the duplication
        df.drop_duplicates(inplace=True)
        sum(df.duplicated())
```

```
Out[9]: 0
```

```
In [10]: # create a function to make a new genres column with a single genre for each movie row
         def genres():
             #split data in each row in genres column
             df['genres'] = df['genres'].str.split('|').apply(lambda genres: np.array(genres))
             genres = df.genres
             # create a new list to append-only the first genre from each row
             new_genres_list = []
             for list in genres:
                 new_genres_list.append(list[0])
             #drop the old genres column
             old_genres_list = df.columns[8]
             df.drop(old_genres_list, axis = 1, inplace = True)
             #replace the old column with the new one
             df[old_genres_list] = new_genres_list
             unique_genres_list = np.unique(new_genres_list)
             return unique_genres_list
         genres()
```

```
Out[10]: array(['Action', 'Adventure', 'Animation', 'Comedy', 'Crime',
                'Documentary', 'Drama', 'Family', 'Fantasy', 'Foreign', 'History',
                'Horror', 'Music', 'Mystery', 'Romance', 'Science Fiction',
                'TV Movie', 'Thriller', 'War', 'Western'],
               dtype='<U15')

In [11]: #check the new genres column
         df['genres']

Out[11]: 0                  Action
         1                  Action
         2               Adventure
         3                  Action
         4                  Action
         5                 Western
         6         Science Fiction
         7                   Drama
         8                  Family
         9                  Comedy
         10                 Action
         11        Science Fiction
         12                  Drama
         13                 Action
         14                 Action
         15                  Crime
         16                  Crime
         17        Science Fiction
         18                Romance
         19                    War
         20                 Action
         21                 Action
         22                 Action
         23                  Drama
         24                 Comedy
         25                 Action
         26                 Comedy
         27                  Crime
         28                  Drama
         29                 Action
                           ...
         10836              Comedy
         10837                 War
         10838              Action
         10839              Family
         10840            Thriller
         10841             Western
         10842           Animation
         10843           Adventure
```

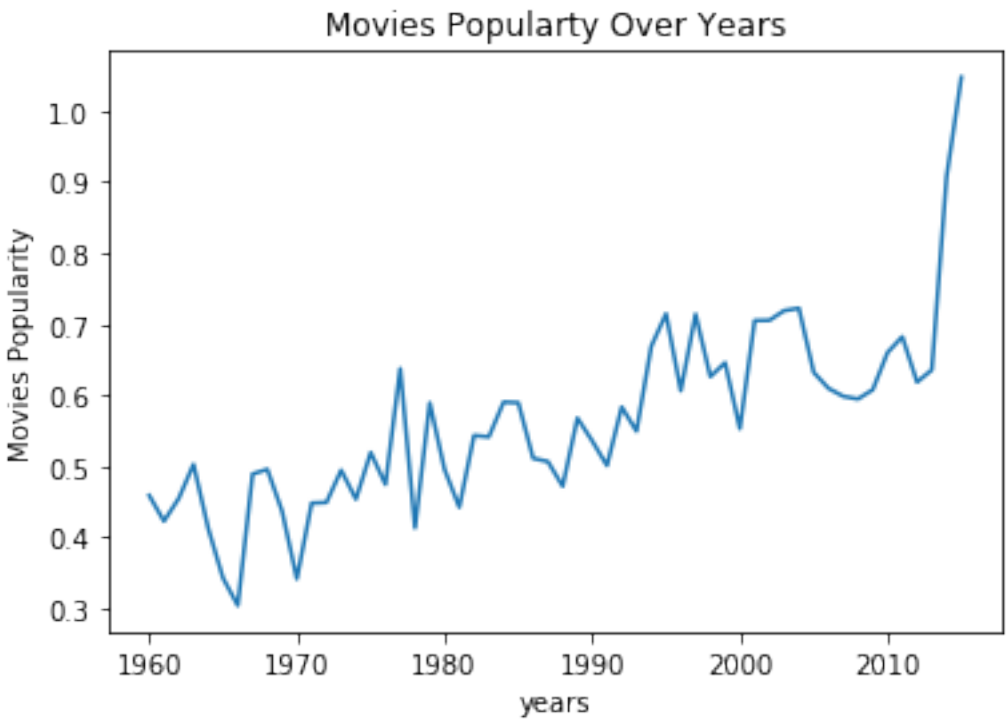
```
10844          Adventure
10845             Comedy
10846             Horror
10847    Science Fiction
10848          Adventure
10849             Action
10850             Action
10851          Adventure
10852            Western
10853             Comedy
10854           Thriller
10855             Comedy
10856             Comedy
10857             Action
10858             Comedy
10859            Mystery
10860             Comedy
10861        Documentary
10862             Action
10863            Mystery
10864             Action
10865             Horror
Name: genres, Length: 10731, dtype: object
```

At the end of the wrangling section, what has been done is cleaning the data from null values, duplication and drop all the columns that will not be used in this investigation. ## Exploratory Data Analysis

1.1.3 Research Question 1 - Which year has released most populare movies ?

Movies' popularity over years

```
In [12]: popularity = df.groupby('release_year')['popularity'].mean()
         plt.plot(popularity)
         plt.title('Movies Popularty Over Years')
         plt.xlabel('years')
         plt.ylabel('Movies Popularity');
```

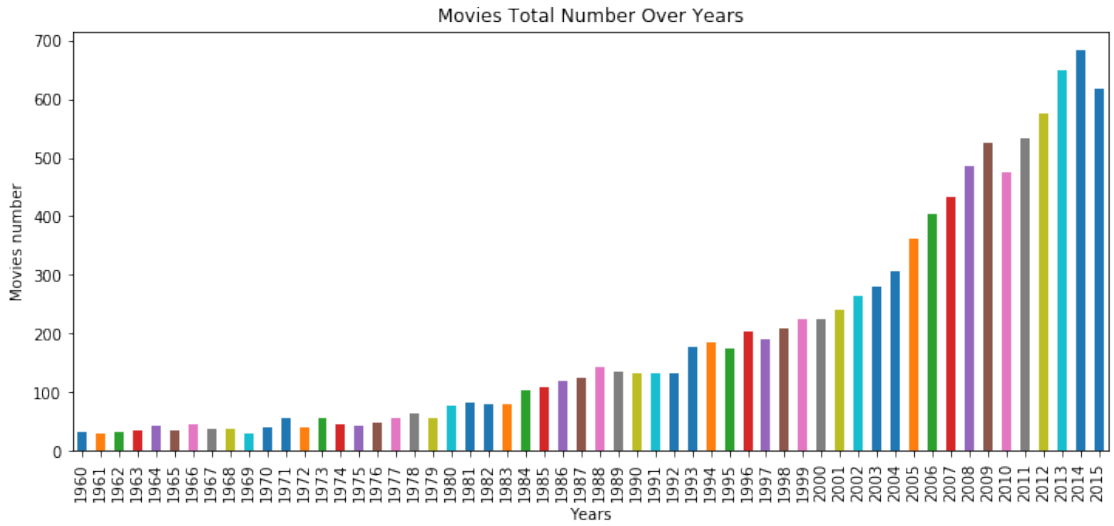


To get the answer to this question, I have grouped the movies into their released years and get the mean of the popularity for every year. The result visualization shows that the popularity of movies has increased in 2010.

1.1.4 Research Question 2 - Which year has released the highest number of movies?

Number of released movies over years

```
In [13]: count = df.groupby('release_year')['id'].count().plot(kind='bar', figsize=(12,5))
         count
         plt.title('Movies Total Number Over Years')
         plt.xlabel('Years')
         plt.ylabel('Movies number');
```

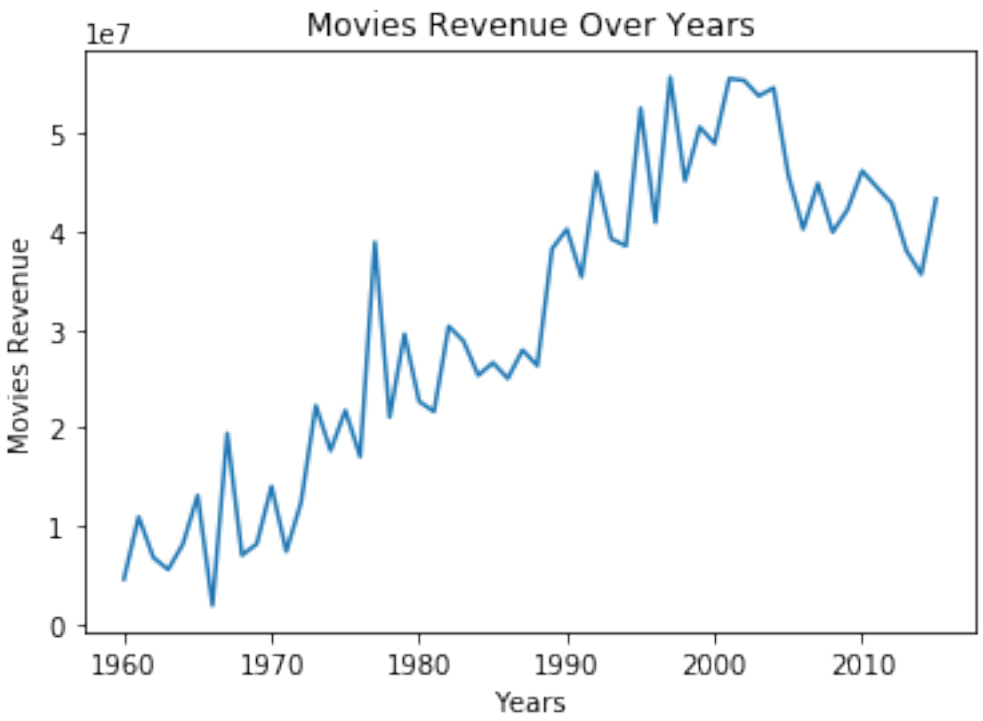


To get the answer to this question, I have grouped the movies into their released years and count the number of movies in each year. The result visualization shows that the number of released movies has been increasing over years and the most movies has released in 2014

1.1.5 Research Question 3 - Does the revenue of releasing movies has changed over years?

Movies' revenue over year

```
In [14]: revenue = df.groupby('release_year')['revenue'].mean()
         plt.plot(revenue)
         plt.title('Movies Revenue Over Years')
         plt.xlabel('Years')
         plt.ylabel('Movies Revenue');
```

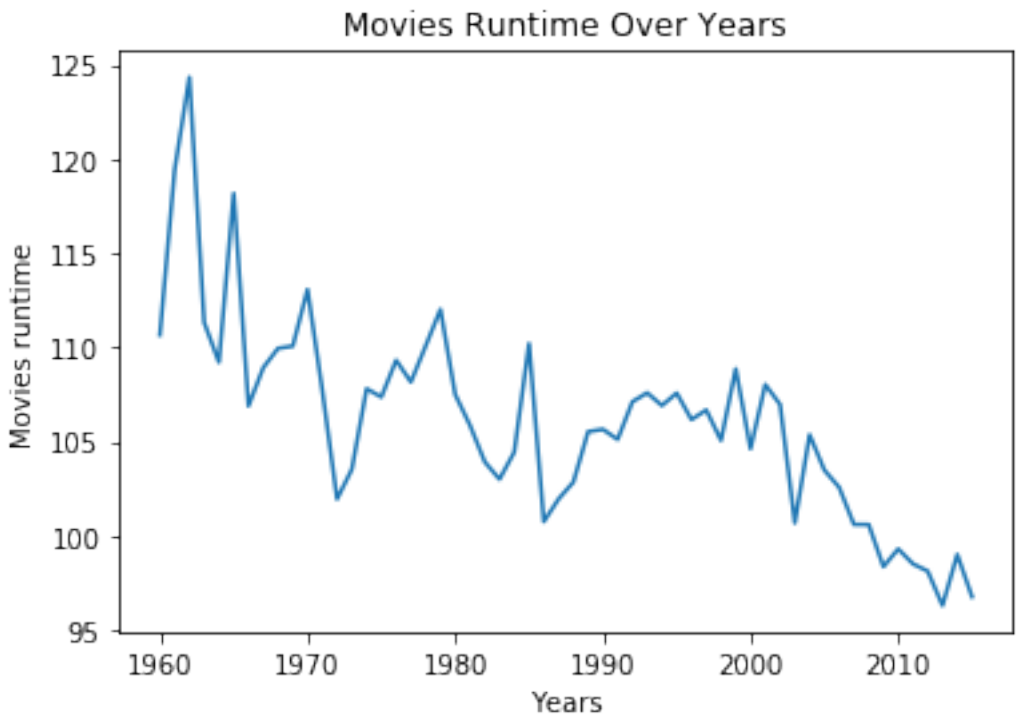


To get the answer to this question, I have grouped the movies into their released years and get the mean of the revenue. The result visualization shows that the revenue has obviously increased over years.

1.1.6 Research Question 4 - Does the runtime duration changed over years?

Movies' runtime duration over years

```
In [15]: runtime = df.groupby('release_year')['runtime'].mean()
         plt.plot(runtime)
         plt.title('Movies Runtime Over Years')
         plt.xlabel('Years')
         plt.ylabel('Movies runtime');
```

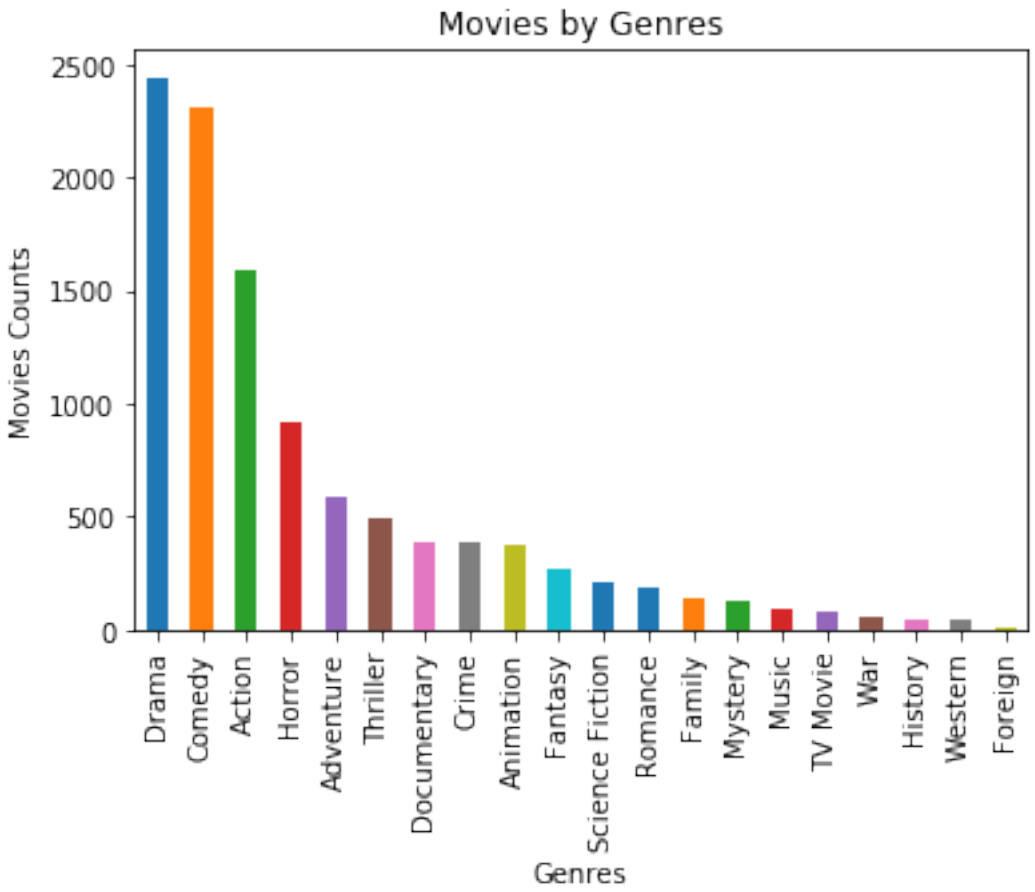


To get the answer to this question, I have grouped the movies into their released years and get the mean of the runtime. The result visualization shows that the runtime duration has decreased over years.

1.1.7 Research Question 5 - What are the most popular genres?

Total number of each genre over years

```
In [16]: df['genres'].value_counts().plot(kind='bar');
         plt.title('Movies by Genres')
         plt.xlabel('Genres')
         plt.ylabel('Movies Counts');
```



To get the answer to this question, I count the number of genres repetitions. The result visualization shows that drama and comedy are the most popular genres. ## Conclusions

`Finally, the data analyzation can show the data details based on the released date and number, M`

For movies popularity, in 60s, it was almost 0.5 and dropped down to 0.3, then it increasd to 0.5 in 70s. that was the lowes point of popularty of movies. after 70s, the popularity is lighting raise to 0.7 in 2000.the popularity trends to 1.0 in 2010 after it goes down to 0.6 becouse of the total number of movies was increased 700 at the same period of time.

On the other hand, the movies runtime it keep decreasing for the same period of time. It decreased from 125 in 60s to below 100 in 2010. When we compare it with the revenue, it keep increasing for whenever the movie time is decreased

At the end, the genres is important to be focus on depends on people intrest. the three trends genres is drama, comedy, and action which they are more than 1500 movie for each genres which most of people are intrest on and make more profit. Horror, advanture comes next between 1000 and 500. The rest of geners are below 500 movie 1960 to 2010.

1.1.8 Limitiation

We have used TMBD Movies dataset for our analysis and worked with popularity, revenue and runtime. Our analysis is limited to only the provided dataset. For example, the dataset does not confirm that every release of every director is listed.

There is no normalization or exchange rate or currency conversion is considered during this analysis and our analysis is limited to the numerical values of revenue.

Dropping missing or Null values from variables of our interest might skew our analysis and could show unintentional bias towards the relationship being analyzed.

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