Investigate_a_Dataset

December 20, 2021

1 Project: TMDB Movies Data Analysis

1.1 Table of Contents

## Introduction

1.1.1 Dataset Description

Here we are investigating The Movies DataBase (TMDB) which contains information about 10,000 movies and their revenues, the release date...etc in order to figure out some questions abot the elements of a successful movie and other questions

1.1.2 Question(s) for Analysis

1) The relation between the number of movies made and the year of release?

2) What is the amount of profit over the years?

3)What is the most popular genres in movies?
4)What is the average runtime of movies over the years?

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

```
In [ ]: # Upgrade pandas to use dataframe.explode() function.
        !pip install --upgrade pandas==0.25.0
```

## Data Wrangling

```
In [2]: df = pd.read_csv('tmdb-movies.csv')
        df.head(3)
```

```
Out[2]:        id    imdb_id  popularity     budget     revenue      original_title  \
        0  135397  tt0369610   32.985763  150000000  1513528810      Jurassic World
        1   76341  tt1392190   28.419936  150000000   378436354  Mad Max: Fury Road
        2  262500  tt2908446   13.112507  110000000   295238201           Insurgent

                                                        cast  \
        0  Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi...
        1  Tom Hardy|Charlize Theron|Hugh Keays-Byrne|Nic...
        2  Shailene Woodley|Theo James|Kate Winslet|Ansel...

                                                homepage          director  \
        0                    http://www.jurassicworld.com/   Colin Trevorrow
        1                      http://www.madmaxmovie.com/     George Miller
        2  http://www.thedivergentseries.movie/#insurgent  Robert Schwentke

                              tagline      ...       \
        0          The park is open.      ...
        1         What a Lovely Day.      ...
        2  One Choice Can Destroy You      ...

                                                    overview runtime  \
        0  Twenty-two years after the events of Jurassic ...     124
        1  An apocalyptic story set in the furthest reach...     120
        2  Beatrice Prior must confront her inner demons ...     119

                                              genres  \
        0  Action|Adventure|Science Fiction|Thriller
        1  Action|Adventure|Science Fiction|Thriller
        2         Adventure|Science Fiction|Thriller

                                        production_companies release_date vote_count  \
        0  Universal Studios|Amblin Entertainment|Legenda...       6/9/15       5562
        1  Village Roadshow Pictures|Kennedy Miller Produ...      5/13/15       6185
        2  Summit Entertainment|Mandeville Films|Red Wago...      3/18/15       2480

           vote_average  release_year    budget_adj   revenue_adj
        0           6.5          2015  1.379999e+08  1.392446e+09
        1           7.1          2015  1.379999e+08  3.481613e+08
        2           6.3          2015  1.012000e+08  2.716190e+08

        [3 rows x 21 columns]
```

```
In [3]: df.describe()
```

```
Out[3]:                   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
```

```
In [4]: 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
```

1.1.3 Data Cleaning

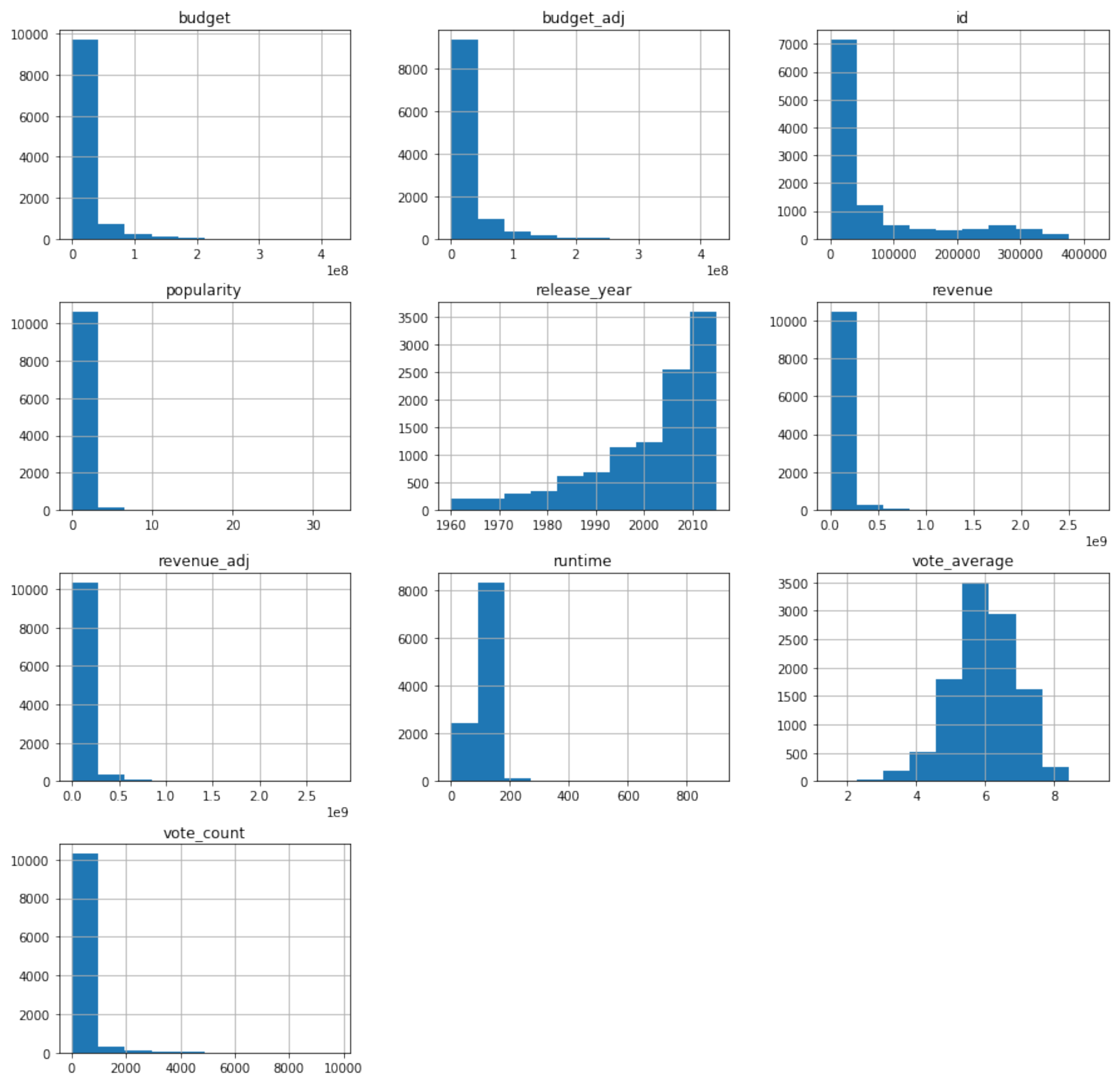
After looking at the data set i found some columns that won't be useful for my analysis so it's better to drop them to get a much cleaner data set.

```
In [5]: df.drop(['cast', 'homepage', 'tagline', 'keywords', 'overview', 'production_companies',
```

```
In [6]: df.info()
```

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

```
In [7]: df.hist(figsize=(15, 15));
```



So here i checked the NaN values and couldn't fill them because they were strings, so i dropped them to have better data.

```
In [8]: df.dropna(inplace=True)
```

```
In [9]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 10835 entries, 0 to 10865
Data columns (total 14 columns):
id                  10835 non-null int64
imdb_id             10835 non-null object
popularity          10835 non-null float64
budget              10835 non-null int64
revenue             10835 non-null int64
```

```
original_title    10835 non-null object
runtime           10835 non-null int64
genres            10835 non-null object
release_date      10835 non-null object
vote_count        10835 non-null int64
vote_average      10835 non-null float64
release_year      10835 non-null int64
budget_adj        10835 non-null float64
revenue_adj       10835 non-null float64
dtypes: float64(4), int64(6), object(4)
memory usage: 1.2+ MB
```

Here i checked for duplicates and only found one so i dropped it

```
In [10]: sum(df.duplicated())
```

```
Out[10]: 1
```

```
In [11]: df.drop_duplicates(inplace=True)
```

So when i tried to calculate profit i found that there was a lot of zeros in budget and revenue column so i replaced them with NAN so i can use dropna function on them and remove them in order to have a much better data set

```
In [12]: df_list = ['budget', 'revenue']
         df[df_list]= df[df_list].replace(0, np.NAN)
         df.dropna(subset = df_list, inplace=True)
```

```
In [13]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 3854 entries, 0 to 10848
Data columns (total 14 columns):
id                3854 non-null int64
imdb_id           3854 non-null object
popularity        3854 non-null float64
budget            3854 non-null float64
revenue           3854 non-null float64
original_title    3854 non-null object
runtime           3854 non-null int64
genres            3854 non-null object
release_date      3854 non-null object
vote_count        3854 non-null int64
vote_average      3854 non-null float64
release_year      3854 non-null int64
budget_adj        3854 non-null float64
revenue_adj       3854 non-null float64
dtypes: float64(6), int64(4), object(4)
memory usage: 451.6+ KB
```

1.1.4 Data Cleaning Summary:

the steps i took to clean the DataBase are not complex but useful in a way

first: i looked at the Data and saw some columns that were not going to be useful for the analysis so i removed them

second: i checked for null values in columns and i found some so i also dropped them for better results in my graphs

third: i checked for duplicates in my data and only found one row and dropping it won't affect the data so i dropped it

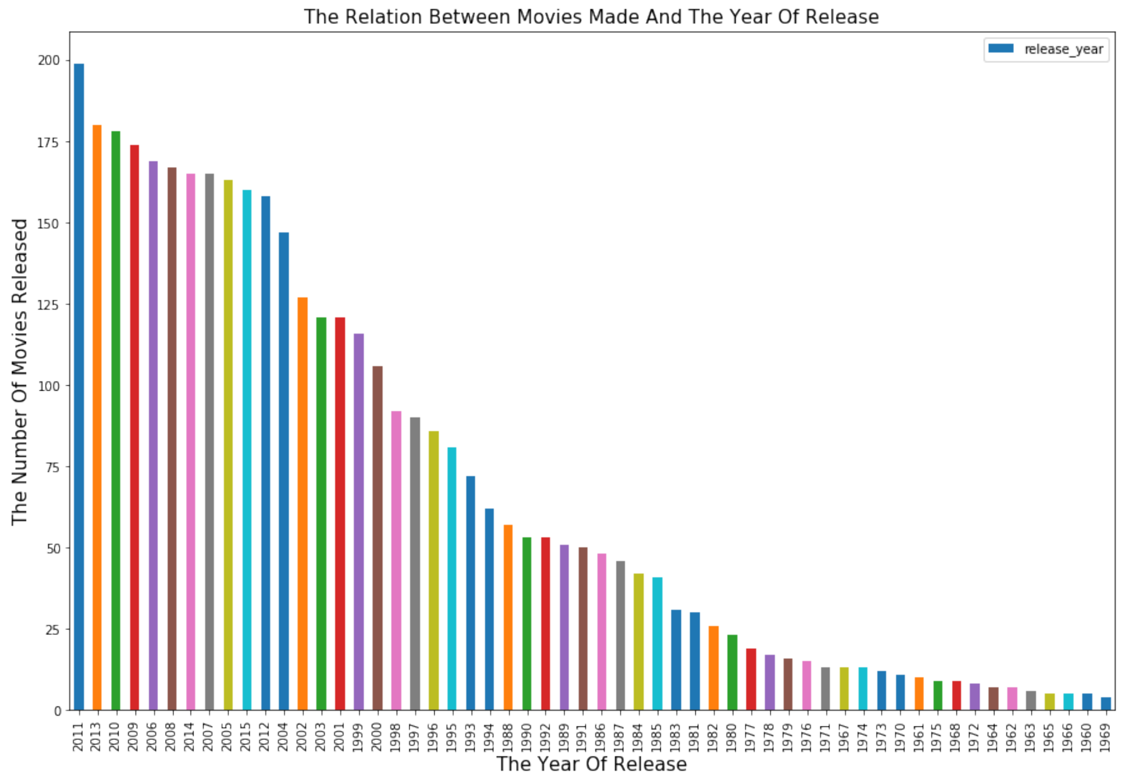
fourth: when i tried to calculate the profit of the movies over the years i found alot of zeros so i returned to the cleaning phase and replaced all the zeros in budget and revenue columns to null values (NAN) so i can drop them

## Exploratory Data Analysis

1.1.5 Research Question 1 : The relation between the number of movies made and the year of release?

```
In [21]: df.release_year.value_counts().plot(kind='bar', figsize=(15,10));
         plt.title('The Relation Between Movies Made And The Year Of Release', fontsize= '15')
         plt.xlabel('The Year Of Release', fontsize= '15')
         plt.ylabel('The Number Of Movies Released', fontsize= '15')
         plt.legend()
```

```
Out[21]: <matplotlib.legend.Legend at 0x7ff4ea18cf60>
```

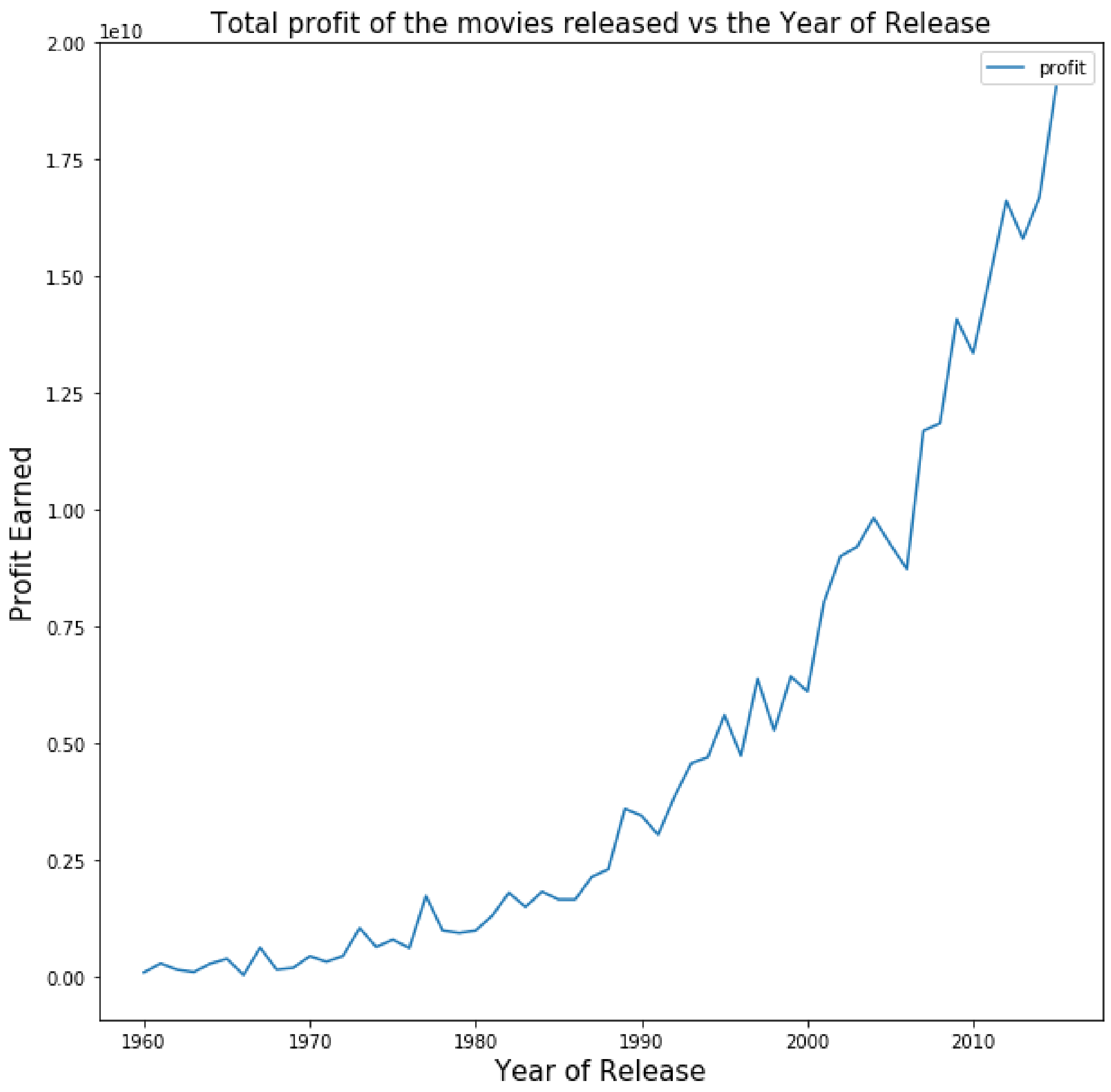


We can clearly see that the number of movies increase over the years

1.1.6 Research Question 2: What is the amount of profit over the years?

```
In [22]: df['profit'] = df['revenue'] - df['budget']
         df.groupby('release_year')['profit'].sum().plot(kind='line', figsize=(10,10));
         plt.title('Total profit of the movies released vs the Year of Release', fontsize= '15')
         plt.xlabel('Year of Release', fontsize= '15')
         plt.ylabel('Profit Earned', fontsize= '15')
         plt.legend()
```

```
Out[22]: <matplotlib.legend.Legend at 0x7ff4e97dd0b8>
```

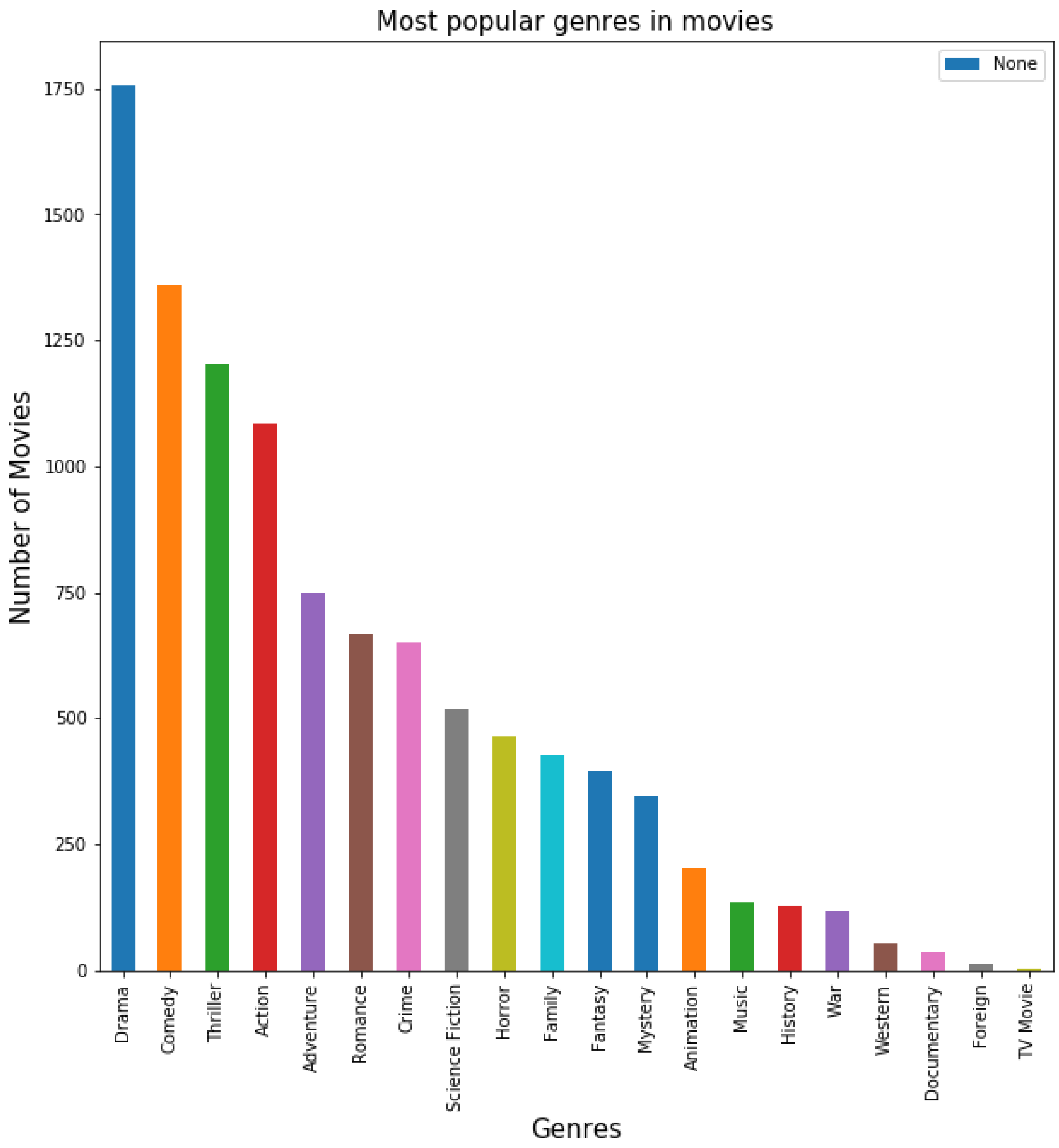


After looking at this graph we can see that profit increased over the years which means the movie industry got much bigger than in the 60s

1.1.7 Research Question 3: What is the most popular genres in movies?

```
In [20]: data = df['genres'].str.cat(sep='|')
         data = pd.Series(data.split('|'))
         data.value_counts().plot(kind='bar', figsize=(10, 10));
         plt.title('Most popular genres in movies', fontsize = '15')
         plt.xlabel('Genres', fontsize = '15')
         plt.ylabel('Number of Movies', fontsize = '15')
         plt.legend()
```

```
Out[20]: <matplotlib.legend.Legend at 0x7ff4ea9a5780>
```



looks like Drama genre is the most popular genre in movies and comes after comedy, thriller and action

1.1.8 Research Question 4: What is the average runtime of movies over the years?

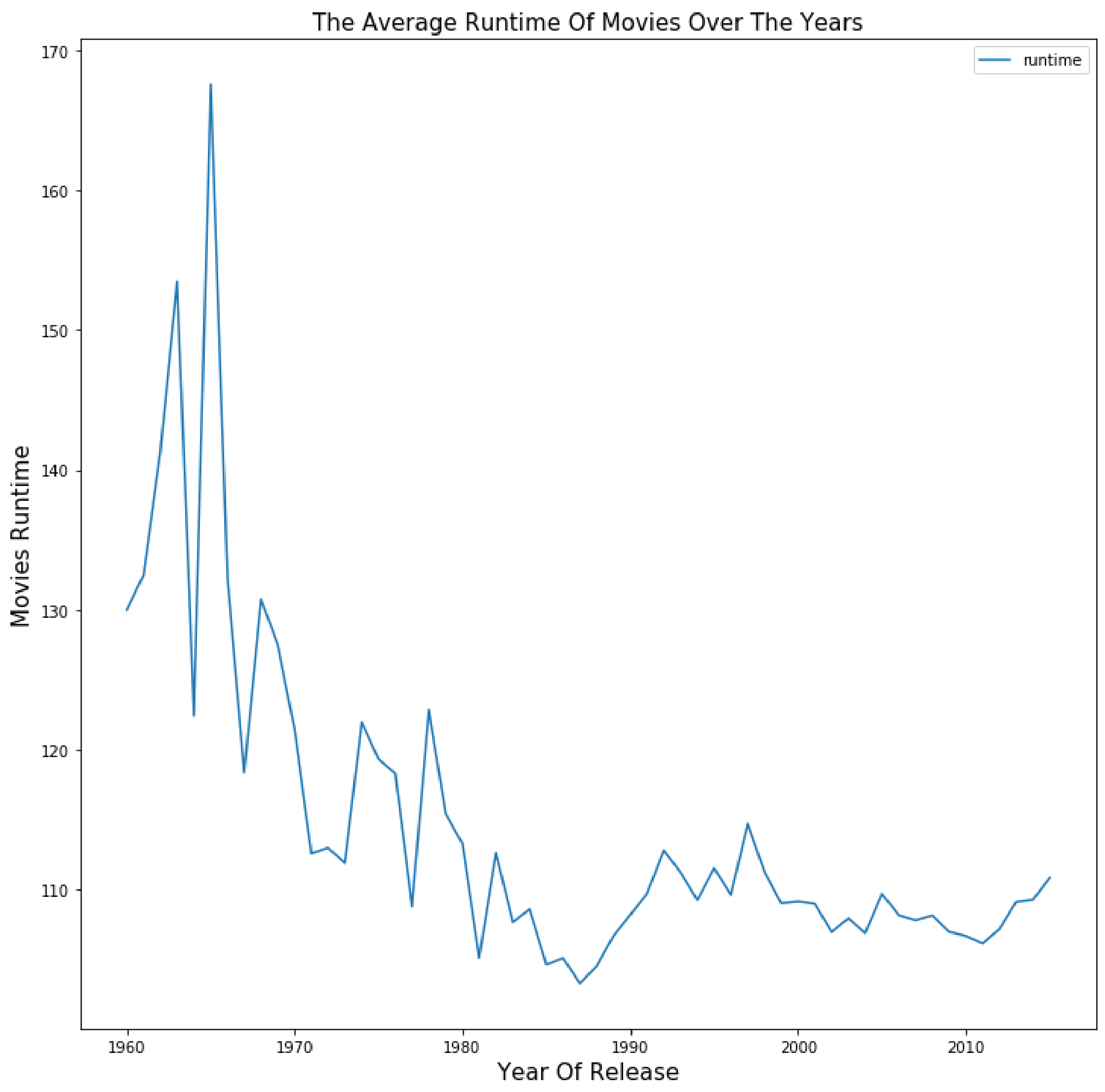
```
In [30]: def avg_amount(column):
             return df[column].mean()
         avg_amount('runtime')

Out[30]: 109.22029060716139
```

Here we used the function to get the average amount of any column in an easy way, and we got the average runtime of all movies in the data set

```
In [54]: df.groupby('release_year')['runtime'].mean().plot(kind='line', figsize=(12, 12))
         plt.legend()
         plt.title('The Average Runtime Of Movies Over The Years', fontsize = '15')
         plt.xlabel('Year Of Release', fontsize = '15')
         plt.ylabel('Movies Runtime', fontsize = '15')

Out[54]: Text(0,0.5,'Movies Runtime')
```



1.1.9 in the graph above we can see that the average runtime of movies decreased over the years, which implies that people are more likely to watch movies between 100 and 125 minutes.

## Conclusions

the first thing to catch your eye is that the filming industry became huge and the amount of profit in it is massive and grew over the years also the amount of movies increased throughout the years. also the most popular genre in movies is the drama genre which means people love watching dramatic movies

1.1.10 Limitations

my only limitation is that these statistics are on a limited amount of movies and can't really represent all the movies produced and they represent a small amount of movies that had such huge profits

```
In [56]: from subprocess import call
         call(['python', '-m', 'nbconvert', 'Investigate_a_Dataset.ipynb'])

Out[56]: 0
```