Investigate_a_Dataset

May 25, 2020

Data Analytics: Welcome to the Investigate a Dataset project! Most people can understand the visualizations, as 40% of the people can answer basic questions about the information provided on the record visualizations. Therefore, when providing information in the form of linear charts, people show a good understanding of the plotsand provide accurate forecasts in this project..

1 Project: Analysis of the reasons for success and failure in the film industry

1.1 Table of Contents

Introduction
Data Wrangling
Exploratory Data Analysis
Conclusions
Introduction

Success in any film production business requires great potential, especially in light of competition from major companies with long experience.Choosing the content for the audience's desire remains the first of the basics, as diversity in films, whether Funny, social, historical, etc. has its own audience.

Perhaps the success of the drama is due to many factors, including excitement, photography and the content of the story, in addition to employing the talents required. Therefore, we see films at the top that generate revenues and films at the bottom that do not achieve anything. In this study, we analyze data on the revenues of films most interested and compare production success in the film industry.

According to industry statistics, six or seven out of ten films are unprofitable, which makes business risky at best? Given this inherent danger, how do movie studios decide which films to place their bets on? Are there common factors, such as the duration of the show, gender, staffing, social style of the audience, or production budget, that explain the financial success of a movie in relation to another? And based on this is determined the desire of the public? Are there common factors, such as revenue (views), voting, gender, or year? This question forms the basis of this research project. This question forms the basis of this research project. To answer that

Other questions : A comparison of budget for modern and traditional films? Which companies have big capital and generate revenues? And companies that do not generate significant revenues? How do you rate the most interesting and popular films? Determine the best, why did he achieve revenue in all years? And the best films that have achieved revenues for each year? Defining audiences for each genre?

In [1]:
```
#importing library
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
sns.set_style("darkgrid")
%matplotlib inline
```

In [2]:
```
# Load dataset
df = pd.read_csv('tmdb-movies.csv')
df.head(1)
```

Out[2]:
```
       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]
```

Data Wrangling

Tip: In this section of the report, you will load in the data, check for cleanliness, and then trim and clean your dataset for analysis. Make sure that you document your steps carefully and justify your cleaning decisions.

1.1.1 General Properties

Revenue project receipts were presented as a dependent variable, with popularity, budget, revenue, director, runtime, genres, production_companies, vote_count, release_year ratings as standalone variables in the final project. The results showed that budget, runtime, vote_count, release year and some genres were statistically significant and positively contributed to the film's domestic revenue.

1.1.2 Data Cleaning (drop many variables!)

Multiple columns have already become a problem in the original data set, this has led to the deletion of many variables and the inclusion of variables of interest to the project.

```
In [3]: df.drop(['id','imdb_id','original_title','cast','homepage','tagline','keywords','overvie
        # iterating the columns
        for col in df.columns:
            print(col)
```

```
popularity
budget
revenue
director
runtime
genres
production_companies
vote_count
release_year
```

```
In [4]: df.shape
```

```
Out[4]: (10866, 9)
```

```
In [5]: df.query('budget == 0').budget
```

```
Out[5]: 30       0
        36       0
        72       0
        74       0
        75       0
        88       0
        92       0
```

95	0
100	0
101	0
103	0
116	0
119	0
122	0
125	0
128	0
130	0
132	0
134	0
139	0
140	0
143	0
146	0
147	0
148	0
151	0
152	0
153	0
158	0
161	0
	..
10830	0
10831	0
10833	0
10834	0
10836	0
10837	0
10838	0
10839	0
10840	0
10842	0
10843	0
10844	0
10845	0
10846	0
10847	0
10849	0
10850	0
10851	0
10852	0
10853	0
10854	0
10856	0
10857	0
10858	0

```
10859    0
10860    0
10861    0
10862    0
10863    0
10864    0
Name: budget, Length: 5696, dtype: int64
```

In [6]:
```
# replace the zero values to nan in revenue,runtime and budget column
col_list = ['budget', 'revenue', 'runtime']
df[col_list] = df[col_list].replace(0, np.NAN) # replacing '0' value to NAN
#dropping NaN value in temp_list
df.dropna(subset = col_list, inplace = True)
```

In [7]:
```
df.release_year
```

Out[7]:
```
0        2015
1        2015
2        2015
3        2015
4        2015
5        2015
6        2015
7        2015
8        2015
9        2015
10       2015
11       2015
12       2015
13       2015
14       2015
15       2015
16       2015
17       2015
18       2015
19       2015
20       2015
21       2015
22       2015
23       2015
```

```
24       2015
25       2015
26       2015
27       2015
28       2015
29       2015
         ...
10690    1965
10691    1965
10692    1965
10716    1965
10724    1969
10725    1969
10727    1969
10728    1969
10755    1978
10756    1978
10757    1978
10758    1978
10759    1978
10760    1978
10762    1978
10770    1978
10771    1978
10775    1978
10777    1978
10778    1978
10779    1978
10780    1978
10788    1978
10791    1978
10793    1978
10822    1966
10828    1966
10829    1966
10835    1966
10848    1966
Name: release_year, Length: 3855, dtype: int64
```

Our study group that contains 10,866 films that were released worldwide in the years (1966-2015) Given the large number of data samples from movie releases and in order to determine the variables that determine the success of the most popular films, we chose the dataset for years instead of Films every year where five years of forty-nine years (2011-2015) were chosen to represent modern films and five years of forty-nine years (2005-2010) were chosen to represent traditional films and were combined into one large unorganized dataset. This method proved an effective way to answer the research question as it focused on the most profitable films and tried to explain their success, rather than finding similarities between random films that are too small and too big, something that might happen if films were chosen each year randomly and variable data was obtained

In [8]:
```
df.dtypes
```

Out[8]:
```
popularity              float64
budget                  float64
revenue                 float64
director                 object
runtime                 float64
genres                   object
production_companies     object
vote_count                int64
release_year              int64
dtype: object
```

In [9]:
```
data = df[(df['release_year'] >= 2005) & (df['release_year'] <= 2015)]
data.release_year
```

Out[9]:
```
0       2015
1       2015
2       2015
3       2015
4       2015
5       2015
6       2015
7       2015
8       2015
9       2015
10      2015
11      2015
12      2015
13      2015
14      2015
15      2015
16      2015
17      2015
18      2015
19      2015
20      2015
21      2015
22      2015
```

```
23      2015
24      2015
25      2015
26      2015
27      2015
28      2015
29      2015
        ...
7620    2007
7630    2007
7637    2007
7638    2007
7643    2007
7653    2007
7654    2007
7665    2007
7667    2007
7668    2007
7670    2007
7675    2007
7685    2007
7697    2007
7706    2007
7707    2007
7708    2007
7714    2007
7717    2007
7718    2007
7733    2007
7739    2007
7758    2007
7761    2007
7776    2007
7785    2007
7797    2007
7808    2007
7813    2007
7819    2007
Name: release_year, Length: 1879, dtype: int64
```

```
In [10]: data.isnull().sum()
```

```
Out[10]: popularity              0
         budget                  0
         revenue                 0
         director                1
         runtime                 0
         genres                  0
         production_companies    25
         vote_count              0
         release_year            0
         dtype: int64
```

```
In [11]: df[df.director.isnull()]
```

```
Out[11]:       popularity     budget     revenue director  runtime  \
         3276    0.147657  4180000.0  11000000.0      NaN    153.0

                                  genres production_companies  vote_count  \
         3276  Drama|Comedy|Romance|Foreign      Tips Industries          11

               release_year
         3276          2008
```

Dropping duplicates Eliminating duplicates will not be identical to two movie at all

```
In [12]: sum(data.duplicated())
```

```
Out[12]: 1
```

```
In [14]: data.drop_duplicates(inplace=True)
```

```
/opt/conda/lib/python3.6/site-packages/ipykernel_launcher.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html#
  """Entry point for launching an IPython kernel.
```

```
In [15]: data.duplicated().sum()
```

```
Out[15]: 0
```

```
In [17]: list_datatype=['budget', 'revenue']
         data[list_datatype]=data[list_datatype].applymap(np.int64)
```

```
/opt/conda/lib/python3.6/site-packages/pandas/core/frame.py:3140: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html#
  self[k1] = value[k2]
```

```
In [18]: data.dtypes
```

```
Out[18]: popularity              float64
         budget                    int64
         revenue                   int64
         director                 object
         runtime                 float64
         genres                   object
         production_companies     object
         vote_count                int64
         release_year              int64
         dtype: object
```

Exploratory Data Analysis

> **Tip**: Now that you've trimmed and cleaned your data, you're ready to move on to exploration. Compute statistics and create visualizations with the goal of addressing the research questions that you posed in the Introduction section. It is recommended that you be systematic with your approach. Look at one variable at a time, and then follow it up by looking at relationships between variables.

1.1.3 Research Question 1 (Comparing modern and traditional movies in the last ten years !)

```
In [19]: df_15 = data[(data['release_year'] >= 2011) & (data['release_year'] <= 2015)]
         df_15.release_year
```

```
Out[19]: 0       2015
         1       2015
         2       2015
         3       2015
         4       2015
         5       2015
         6       2015
         7       2015
         8       2015
         9       2015
         10      2015
         11      2015
         12      2015
         13      2015
         14      2015
         15      2015
         16      2015
         17      2015
         18      2015
         19      2015
         20      2015
         21      2015
         22      2015
         23      2015
         24      2015
         25      2015
         26      2015
         27      2015
         28      2015
         29      2015
                 ...
         5673    2013
         5674    2013
         5679    2013
         5697    2013
         5704    2013
         5713    2013
         5722    2013
         5727    2013
         5732    2013
         5741    2013
         5746    2013
         5750    2013
         5772    2013
         5775    2013
         5785    2013
         5787    2013
         5812    2013
```

```
5833    2013
5837    2013
5840    2013
5846    2013
5852    2013
5860    2013
5875    2013
5903    2013
5908    2013
5932    2013
6010    2013
6041    2013
6065    2013
Name: release_year, Length: 862, dtype: int64
```

In [20]:
```
df_05 = data[(data['release_year'] >= 2005) & (data['release_year'] <= 2010)]
df_05.release_year
```

Out[20]:
```
1386    2009
1387    2009
1388    2009
1389    2009
1390    2009
1391    2009
1392    2009
1393    2009
1394    2009
1395    2009
1396    2009
1397    2009
1398    2009
1399    2009
1400    2009
1401    2009
1402    2009
1403    2009
1404    2009
1405    2009
1406    2009
1407    2009
1408    2009
1410    2009
1411    2009
1412    2009
1413    2009
```

```
1414    2009
1415    2009
1416    2009
        ...
7620    2007
7630    2007
7637    2007
7638    2007
7643    2007
7653    2007
7654    2007
7665    2007
7667    2007
7668    2007
7670    2007
7675    2007
7685    2007
7697    2007
7706    2007
7707    2007
7708    2007
7714    2007
7717    2007
7718    2007
7733    2007
7739    2007
7758    2007
7761    2007
7776    2007
7785    2007
7797    2007
7808    2007
7813    2007
7819    2007
Name: release_year, Length: 1016, dtype: int64
```

```
In [21]: df_05.shape
```

```
Out[21]: (1016, 9)
```

```
In [22]: df_15.shape
```

```
Out[22]: (862, 9)
```

In [23]:
```
# ensure these queries included each sample exactly once
num_samples = data.shape[0]
num_samples == df_05['revenue'].count() + df_15['revenue'].count() # should be True
```

Out[23]:
```
True
```

In [24]:
```
df_05.describe().revenue
```

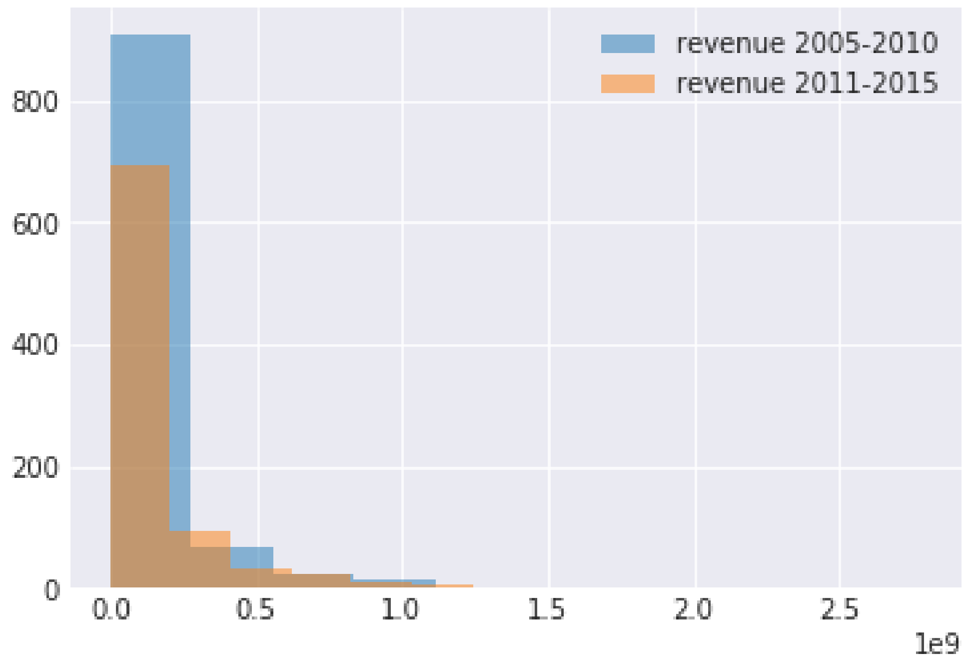
Out[24]:
```
count    1.016000e+03
mean     1.109685e+08
std      1.868657e+08
min      3.000000e+00
25%      1.231831e+07
50%      4.791468e+07
75%      1.228296e+08
max      2.781506e+09
Name: revenue, dtype: float64
```

In [25]:
```
df_15.describe().revenue
```

Out[25]:
```
count    8.620000e+02
mean     1.417822e+08
std      2.322957e+08
min      1.100000e+01
25%      1.113568e+07
50%      5.474676e+07
75%      1.597275e+08
max      2.068178e+09
Name: revenue, dtype: float64
```

Through the results it was found that traditional films achieved higher revenues than modern films as a result of the following: Through an average study where the best movies achieved the highest revenue is 2.781506e+09 while the lowest real revenue is 3.000000e+00 and on that the data set of the traditional films was chosen to know the factors That contributed to success

In [26]:
```
df_05.revenue.hist(alpha=0.5,label='revenue 2005-2010')
df_15.revenue.hist(alpha=0.5,label='revenue 2011-2015');
plt.legend();
```



1.1.4 Research Question 2 (How do movie studios decide which films to place their bets on !)

Are there common factors, such as the duration of the show, gender, staffing, social style of the audience, or production budget, that explain the financial success of a movie in relation to another?

In [27]:
```
data.describe().revenue
```

Out[27]:
```
count    1.878000e+03
mean     1.251120e+08
std      2.094543e+08
min      3.000000e+00
25%      1.173206e+07
50%      5.076060e+07
75%      1.407145e+08
max      2.781506e+09
Name: revenue, dtype: float64
```

In [28]:
```
high_revenue = data.revenue>=1.406333e+08
mid_revenue = (data.revenue >= 5.065008e+07) & (data.revenue <= 1.406333e+08)
lower_revenue = (data.revenue >= 1.166930e+07) & (data.revenue <= 5.065008e+07)
```

factor budget relating revenue

In [29]:
```
data.budget[high_revenue].mean()
```

Out[29]: 101218968.09361702

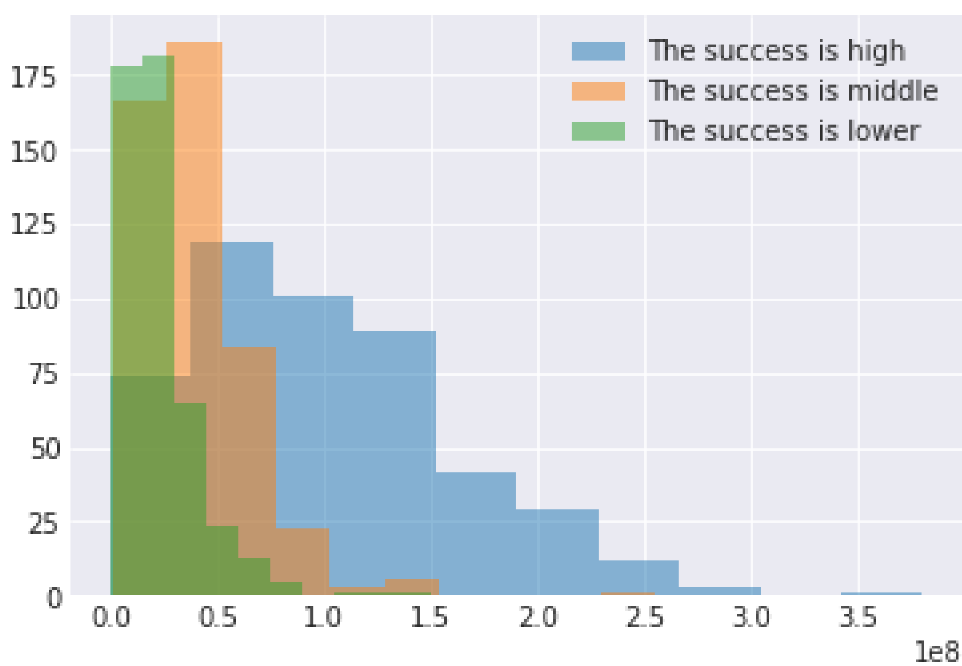
In [30]:
```
data.budget[mid_revenue].mean()
```

Out[30]: 39480899.044776119

In [31]:
```
data.budget[lower_revenue].mean()
```

Out[31]: 23855186.989361703

In [32]:
```
data.budget[high_revenue].hist(label='The success is high',alpha=0.5)
data.budget[mid_revenue].hist(label='The success is middle',alpha=0.5)
data.budget[lower_revenue].hist(label='The success is lower',alpha=0.5);
plt.legend();
```



The big companies that have big capital for making films make big revenues, while the companies that don't have a big budget make small revenues

factor runtime relating revenue

In [33]:
```
data.runtime[high_revenue].mean()
```

Out[33]: 114.64680851063829

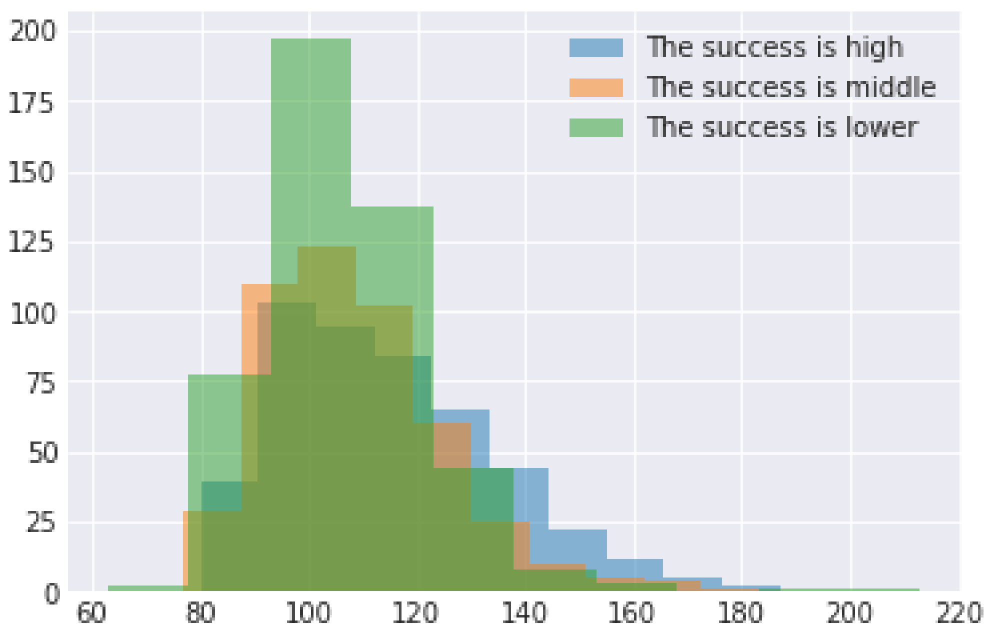
In [34]:
```
data.runtime[mid_revenue].mean()
```

Out[34]: 108.37100213219617

In [35]:
```
data.runtime[lower_revenue].mean()
```

Out[35]: 106.19787234042553

In [40]:
```
data.runtime[high_revenue].hist(label='The success is high',alpha=0.5)
data.runtime[mid_revenue].hist(label='The success is middle',alpha=0.5)
data.runtime[lower_revenue].hist(label='The success is lower',alpha=0.5);
plt.legend();
```



Among the success factors for the industry, films are the show duration, that is, the longer the show, the more revenue, and the less the offer, the less revenue

factor vote count relating revenue

In [37]:
```
data.vote_count[high_revenue].mean()
```

Out[37]: 1784.4212765957448

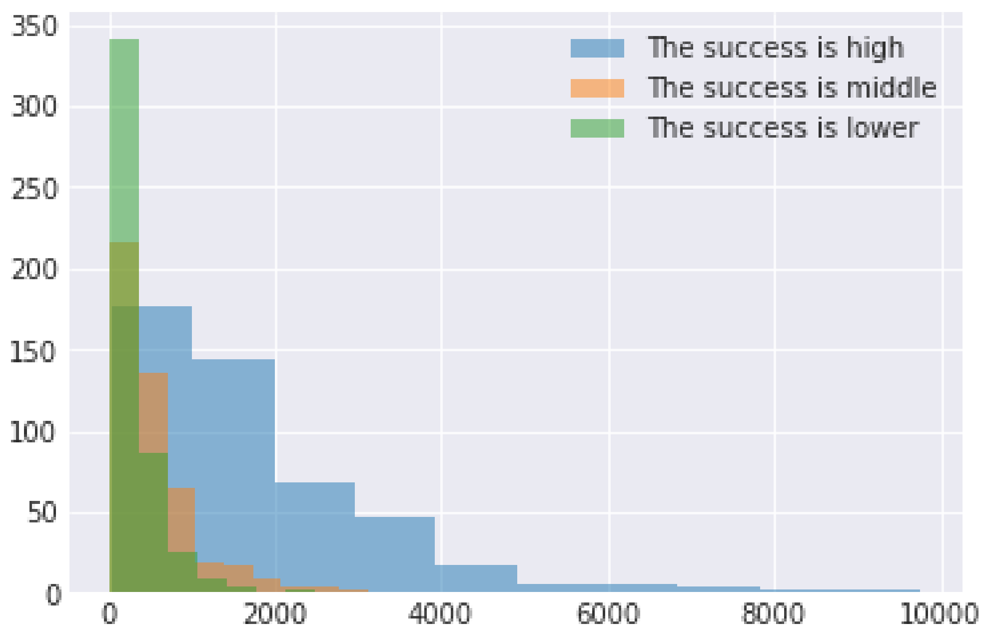
In [38]:
```
data.vote_count[mid_revenue].mean()
```

Out[38]: 552.04477611940297

```
In [39]: data.vote_count[lower_revenue].mean()
```

Out[39]: 309.67659574468087

```
In [41]: data.vote_count[high_revenue].hist(label='The success is high',alpha=0.5)
         data.vote_count[mid_revenue].hist(label='The success is middle',alpha=0.5)
         data.vote_count[lower_revenue].hist(label='The success is lower',alpha=0.5);
         plt.legend();
```



Likewise, the voting component increases the more votes, the more revenue, and the lower the percentage of voting, the less revenue

1.1.5 Research Question 3 (What is the best so I have not achieved revenue in all years !)

> In order to analyze the reasons for the success of these films, some questions are asked here: What is the best so I have not achieved revenue in all years? The best films that have earned revenue for each year?

By examining the number of films per year and by checking the best films that make money from ten years ago

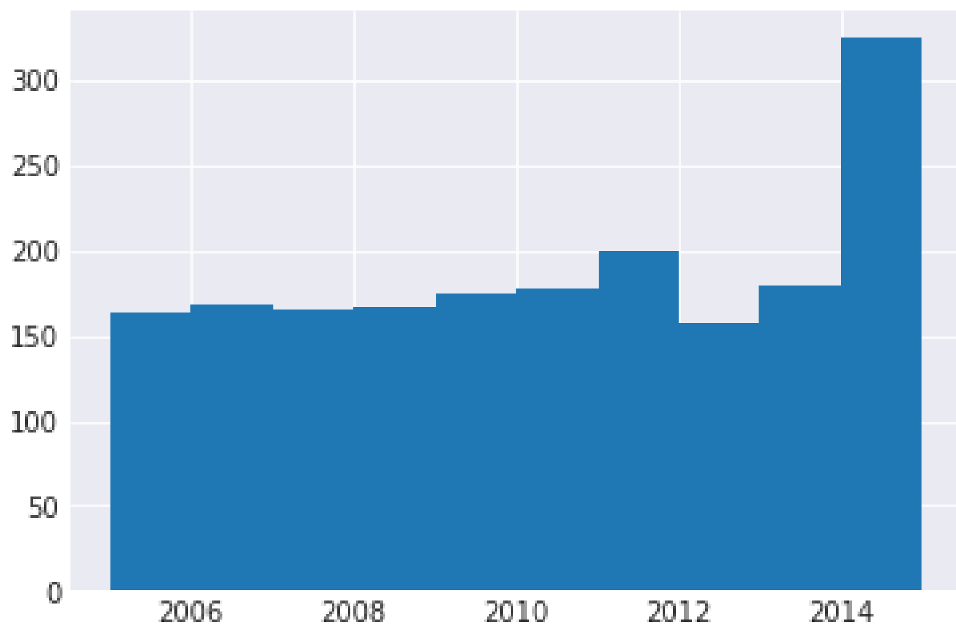
```
In [43]: data.revenue.max()
```

Out[43]: 2781505847

```
In [44]: data.groupby('release_year').max().revenue
```

Out[44]:
```
release_year
2005     895921036
2006    1065659812
2007     961000000
2008    1001921825
2009    2781505847
2010    1063171911
2011    1327817822
2012    1519557910
2013    1274219009
2014     955119788
2015    2068178225
Name: revenue, dtype: int64
```

In [47]:
```
data['release_year'].hist();
```



Conclusions

Modern filmmaking, which is worth nearly 10 billion dolar a year, is a noisy business and highly profitable There was an important theoretical relationship between the number of revenues and the amount of money the film studio spent in producing the film

The variable was recorded as analyzes indicate a large variation in movie revenue, with approximately 80% -85% of total movie revenue coming from the best 20% of movies. The film that is a supplement or belongs to a well-established property will have an impact on competition in the release year. As the release year affects films evaluation

By identifying the big companies that have modern equipment and have big capital for the film industry, they achieve great revenues, while the average companies achieve small revenues and accordingly we can find out the reasons for the success of these companies or the failure of other companies.

In the event that success is achieved for a previous movie in the series, the company will strive to produce successful and profitable films in the coming days, because success will be followed by other successes and whoever succeeds in one of the works does not accept failure in other works, and all of this will result in increasing the different audiences. There is no specific work for success, as we found in the analysis that it is difficult to divide success according to the type of film in the study: action (ACTION), science fiction (SCIFI), comedy (COM), documentary (DOC), foreign (FOREIGN), romance (ROM), adventure (ADVENT) and horror (HORROR). Therefore, it is difficult to evaluate the database according to gender, and there are other factors that affect the popularity of films, such as music, photography, award nominations, and the strength of stars, which were important positive determinants of success.

Voting clearly plays an important role in determining movie revenues, as some votes can say something about the nature of the movie and can restrict the film market.

Another variable whose importance was questioned in the analysis but worthy of inclusion was a measure of the strength of the director and actor associated with a film project. It indicates that the analysis believes that the strength of the director and the star is important, which supports the assumption of rent picking that the actor has a market value through large salaries and does little to influence the profitability of films. And successful films may make the stars. Due to the ambiguity of the effect of this variable and the inconsistency of our qualifications

Most of the time, we found that the strength of the directors, production budgets and sequences contributed positively to the film's revenue.

Special effects and computer technology have come a long way in the past ten years, and may have contributed to changing consumers' tastes and preferences for certain types of movies.

Better quality films will be more successful.

If a movie is released in the holiday season, it is expected to see an increase in revenue, while the summer release will bring an expected increase in views.

Comedies tend to experience positive success in the supply market, although the influence of other genres is inconclusive.

1.2 Submitting your Project

Before you submit your project, you need to create a .html or .pdf version of this notebook in the workspace here. To do that, run the code cell below. If it worked correctly, you should get a return code of 0, and you should see the generated .html file in the workspace directory (click on the orange Jupyter icon in the upper left).

Alternatively, you can download this report as .html via the **File** > **Download as** submenu, and then manually upload it into the workspace directory by clicking on the orange Jupyter icon in the upper left, then using the Upload button.

Once you've done this, you can submit your project by clicking on the "Submit Project" button in the lower right here. This will create and submit a zip file with this .ipynb doc and the .html or .pdf version you created. Congratulations!

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