Project: Investigate a Dataset (TMDB movies)

Table of Contents

Introduction

1-In order to complete my Investigate a dataset project i am using a dataset called TMDB movies file downloaded from the udacity.

2-the shape of the dataset is 10866 rows with 21 columns.

Questions that can be used to analyse the dataset are:

1-Movies with profit and loss.

2-Movies with lower and higher budget.

3-Movies with lower and higher revenue.

4-Most of the genres.

5-Most of the cast.

6-Movie with shortest and longest runtime.

In [1]:

```
# Use this cell to set up import statements for all of the packages that you
#   plan to use.

# Remember to include a 'magic word' so that your visualizations are plotted
#   inline with the notebook. See this page for more:
#   http://ipython.readthedocs.io/en/stable/interactive/magics.html
```

In [2]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
```

Data Wrangling

--Data Wrangling is used to clean the dataset for better results.As per my observation from the dataset there are some of the cleanliness need to be done.

General Properties

In [3]:

```
# Load your data and print out a few lines. Perform operations to inspect data
#   types and look for instances of missing or possibly errant data.
df=pd.read_csv('tmdb-movies.csv')
df.head(10)
```

Out[3]:

	id	imdb_id	popularity	budget	revenue	original_title	cast	
0	135397	tt0369610	32.985763	150000000	1513528810	Jurassic World	Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi...	
1	76341	tt1392190	28.419936	150000000	378436354	Mad Max: Fury Road	Tom Hardy|Charlize Theron|Hugh Keays-Byrne|Nic...	
2	262500	tt2908446	13.112507	110000000	295238201	Insurgent	Shailene Woodley|Theo James|Kate Winslet|Ansel...	ht
3	140607	tt2488496	11.173104	200000000	2068178225	Star Wars: The Force Awakens	Harrison Ford|Mark Hamill|Carrie Fisher|Adam D...	
4	168259	tt2820852	9.335014	190000000	1506249360	Furious 7	Vin Diesel|Paul Walker|Jason Statham|Michelle ...	
5	281957	tt1663202	9.110700	135000000	532950503	The Revenant	Leonardo DiCaprio|Tom Hardy|Will Poulter|Domhn...	ht
6	87101	tt1340138	8.654359	155000000	440603537	Terminator Genisys	Arnold Schwarzenegger|Jason Clarke|Emilia Clar...	
7	286217	tt3659388	7.667400	108000000	595380321	The Martian	Matt Damon|Jessica Chastain|Kristen Wiig|Jeff ...	
8	211672	tt2293640	7.404165	74000000	1156730962	Minions	Sandra Bullock|Jon Hamm|Michael Keaton|Allison...	
9	150540	tt2096673	6.326804	175000000	853708609	Inside Out	Amy Poehler|Phyllis Smith|Richard Kind|Bill Ha...	

10 rows × 21 columns

In [4]:

```
df.shape
```

Out[4]:

```
(10866, 21)
```

In [5]:

```
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 [6]:

```
df_clean=df.copy()
df_clean.head(10)
```

Out[6]:

	id	imdb_id	popularity	budget	revenue	original_title	cast	
0	135397	tt0369610	32.985763	150000000	1513528810	Jurassic World	Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi...	
1	76341	tt1392190	28.419936	150000000	378436354	Mad Max: Fury Road	Tom Hardy|Charlize Theron|Hugh Keays-Byrne|Nic...	
2	262500	tt2908446	13.112507	110000000	295238201	Insurgent	Shailene Woodley|Theo James|Kate Winslet|Ansel...	ht
3	140607	tt2488496	11.173104	200000000	2068178225	Star Wars: The Force Awakens	Harrison Ford|Mark Hamill|Carrie Fisher|Adam D...	
4	168259	tt2820852	9.335014	190000000	1506249360	Furious 7	Vin Diesel|Paul Walker|Jason Statham|Michelle ...	
5	281957	tt1663202	9.110700	135000000	532950503	The Revenant	Leonardo DiCaprio|Tom Hardy|Will Poulter|Domhn...	ht
6	87101	tt1340138	8.654359	155000000	440603537	Terminator Genisys	Arnold Schwarzenegger|Jason Clarke|Emilia Clar...	
7	286217	tt3659388	7.667400	108000000	595380321	The Martian	Matt Damon|Jessica Chastain|Kristen Wiig|Jeff ...	
8	211672	tt2293640	7.404165	74000000	1156730962	Minions	Sandra Bullock|Jon Hamm|Michael Keaton|Allison...	
9	150540	tt2096673	6.326804	175000000	853708609	Inside Out	Amy Poehler|Phyllis Smith|Richard Kind|Bill Ha...	

10 rows × 21 columns

In [7]:

```
df_clean.shape
```

Out[7]:

```
(10866, 21)
```

In [8]:

```
df_clean.describe()
```

Out[8]:

	id	popularity	budget	revenue	runtime	vote_count	v
count	10866.000000	10866.000000	1.086600e+04	1.086600e+04	10866.000000	10866.000000	1
mean	66064.177434	0.646441	1.462570e+07	3.982332e+07	102.070863	217.389748	
std	92130.136561	1.000185	3.091321e+07	1.170035e+08	31.381405	575.619058	
min	5.000000	0.000065	0.000000e+00	0.000000e+00	0.000000	10.000000	
25%	10596.250000	0.207583	0.000000e+00	0.000000e+00	90.000000	17.000000	
50%	20669.000000	0.383856	0.000000e+00	0.000000e+00	99.000000	38.000000	
75%	75610.000000	0.713817	1.500000e+07	2.400000e+07	111.000000	145.750000	
max	417859.000000	32.985763	4.250000e+08	2.781506e+09	900.000000	9767.000000	

In [9]:

```
df_clean.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
```

Data Cleaning (Replace this with more specific notes!)

As per my observation made from the dataset, There are some cleaning,calculation and change of datatype of some columns are need to be done.

1-There are 4 columns that are duplicated values in the original_title column from the dataset.

2-There are some unwanted columns that need to be droped.

3-caluclation for profit and profit_adj(from columns budget,revenue,budget_adj,revenue_adj)

4-changinf the datatypes of some columns.

Note:-for the currency related columns i am going to assume that the currency will be in Dollors, as dollors is used as the international currency.

In [10]:

```
# After discussing the structure of the data and any problems that need to be
#   cleaned, perform those cleaning steps in the second part of this section.
```

Finding sum of null values in the Dataset:

In [11]:

```
df_clean.isnull().sum()
```

Out[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
```

Finding the number of Duplicated values in the original_titles column:

In [15]:

```
df_clean.original_title.duplicated(False).value_counts()
```

Out[15]:

```
False    10294
True       572
Name: original_title, dtype: int64
```

In [20]:

```
df_clean[df_clean.original_title.duplicated(False)]
```

Out[20]:

	id	imdb_id	popularity	budget	revenue	original_title	cast	
5	281957	tt1663202	9.110700	135000000	532950503	The Revenant	Leonardo DiCaprio|Tom Hardy|Will Poulter|Domhn...	http://w
18	150689	tt1661199	5.556818	95000000	542351353	Cinderella	Lily James|Cate Blanchett|Richard Madden|Helen...	
46	228161	tt2224026	2.976436	135000000	368871007	Home	Jim Parsons|Rihanna|Steve Martin|Jennifer Lope...	

Comparing two columns to find the duplicate values columns are original_title and cast:

In [21]:

```
df_clean.duplicated(['original_title','cast'],False).value_counts()
```

Out[21]:

```
False    10862
True         4
dtype: int64
```

In [23]:

```
df_clean[df_clean.duplicated(['original_title','cast'],False)]
```

Out[23]:

	id	imdb_id	popularity	budget	revenue	original_title	cast	homepag
2089	42194	tt0411951	0.596430	30000000	967000	TEKKEN	Jon Foo|Kelly Overton|Cary-Hiroyuki Tagawa|Ian...	Na
2090	42194	tt0411951	0.596430	30000000	967000	TEKKEN	Jon Foo|Kelly Overton|Cary-Hiroyuki Tagawa|Ian...	Na
4063	28004	tt0359639	0.371510	6000000	0	Madea's Family Reunion	Tyler Perry|Blair Underwood|Lynn Whitfield|Bor...	Na
6701	16781	tt0455612	0.552267	6000000	57231524	Madea's Family Reunion	Tyler Perry|Blair Underwood|Lynn Whitfield|Bor...	Na

4 rows × 21 columns

Droping the duplicate values except the first occurrence and checking the duplicate values:

In [31]:

```
df_clean.drop_duplicates(['original_title','cast'],keep='first',inplace=True)
```

In [32]:

```
df_clean[df_clean.duplicated(['original_title','cast'])]
```

Out[32]:

id	imdb_id	popularity	budget	revenue	original_title	cast	homepage	director	tagline	...

0 rows × 21 columns

Dropping unwanted columns such as id,imdb_id,keywords,overview,production_companies:

In [33]:

```
df_clean.drop(['id','imdb_id','homepage','keywords','overview','production_companies','home
```

In [34]:

```
df_clean.head(10)
```

Out[34]:

	popularity	budget	revenue	original_title	cast	director	runtime
0	32.985763	150000000	1513528810	Jurassic World	Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi...	Colin Trevorrow	124
1	28.419936	150000000	378436354	Mad Max: Fury Road	Tom Hardy|Charlize Theron|Hugh Keays-Byrne|Nic...	George Miller	120
2	13.112507	110000000	295238201	Insurgent	Shailene Woodley|Theo James|Kate Winslet|Ansel...	Robert Schwentke	119
3	11.173104	200000000	2068178225	Star Wars: The Force Awakens	Harrison Ford|Mark Hamill|Carrie Fisher|Adam D...	J.J. Abrams	136
4	9.335014	190000000	1506249360	Furious 7	Vin Diesel|Paul Walker|Jason Statham|Michelle ...	James Wan	137
5	9.110700	135000000	532950503	The Revenant	Leonardo DiCaprio|Tom Hardy|Will Poulter|Domhn...	Alejandro GonzÃ¡lez IÃ±Ã¡rritu	156
6	8.654359	155000000	440603537	Terminator Genisys	Arnold Schwarzenegger|Jason Clarke|Emilia Clar...	Alan Taylor	125
7	7.667400	108000000	595380321	The Martian	Matt Damon|Jessica Chastain|Kristen Wiig|Jeff ...	Ridley Scott	141
8	7.404165	74000000	1156730962	Minions	Sandra Bullock|Jon Hamm|Michael Keaton|Allison...	Kyle Balda|Pierre Coffin	91
9	6.326804	175000000	853708609	Inside Out	Amy Poehler|Phyllis Smith|Richard Kind|Bill Ha...	Pete Docter	94

Change of datat types to some of the columns:

1-budget_adj column from float64 to int64.

2-revenue_adj column from float6 to int 64.

3-release_date from str to datetime.

In [35]:

```
df_clean['budget_adj']=df_clean['budget_adj'].astype('int64')
```

In [36]:

```
df_clean['revenue_adj']=df_clean['revenue_adj'].astype('int64')
```

In [37]:

```
df_clean['release_date']=pd.to_datetime(df_clean['release_date'],format='%m/%d/%y')
```

In [38]:

```
df_clean.dtypes
```

Out[38]:

```
popularity               float64
budget                     int64
revenue                    int64
original_title            object
cast                      object
director                  object
runtime                    int64
genres                    object
release_date      datetime64[ns]
vote_count                 int64
vote_average             float64
release_year               int64
budget_adj                 int64
revenue_adj                int64
dtype: object
```

Caluclation for profit and profit_adj:

In [39]:

```
df_clean['profit']=df_clean['revenue']-df_clean['budget']
```

In [40]:

```
df_clean['profit_adj']=(df_clean['revenue_adj']-df_clean['budget_adj']).astype('int64')
```

In [41]:

```
df_clean.head(10)
```

Out[41]:

	popularity	budget	revenue	original_title	cast	director	runtime
0	32.985763	150000000	1513528810	Jurassic World	Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi...	Colin Trevorrow	124
1	28.419936	150000000	378436354	Mad Max: Fury Road	Tom Hardy|Charlize Theron|Hugh Keays-Byrne|Nic...	George Miller	120
2	13.112507	110000000	295238201	Insurgent	Shailene Woodley|Theo James|Kate Winslet|Ansel...	Robert Schwentke	119
3	11.173104	200000000	2068178225	Star Wars: The Force Awakens	Harrison Ford|Mark Hamill|Carrie Fisher|Adam D...	J.J. Abrams	136
4	9.335014	190000000	1506249360	Furious 7	Vin Diesel|Paul Walker|Jason Statham|Michelle ...	James Wan	137
5	9.110700	135000000	532950503	The Revenant	Leonardo DiCaprio|Tom Hardy|Will Poulter|Domhn...	Alejandro GonzÃ¡lez IÃ±Ã¡rritu	156
6	8.654359	155000000	440603537	Terminator Genisys	Arnold Schwarzenegger|Jason Clarke|Emilia Clar...	Alan Taylor	125
7	7.667400	108000000	595380321	The Martian	Matt Damon|Jessica Chastain|Kristen Wiig|Jeff ...	Ridley Scott	141
8	7.404165	74000000	1156730962	Minions	Sandra Bullock|Jon Hamm|Michael Keaton|Allison...	Kyle Balda|Pierre Coffin	91
9	6.326804	175000000	853708609	Inside Out	Amy Poehler|Phyllis Smith|Richard Kind|Bill Ha...	Pete Docter	94

In [42]:

```
df_clean.budget.replace((0),np.NaN,inplace=True)
```

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.

Question 1-Movies with profit and loss.

In [63]:

```
movie_profit=pd.DataFrame(df_clean.loc[df_clean.profit.idxmax()])
movie_loss=pd.DataFrame(df_clean.loc[df_clean.profit.idxmin()])
pd.concat([movie_profit,movie_loss],axis=1)
```

Out[63]:

	1386	2244
popularity	9.43277	0.25054
budget	2.37e+08	4.25e+08
revenue	2781505847	11087569
original_title	Avatar	The Warrior's Way
cast	Sam Worthington|Zoe Saldana|Sigourney Weaver|S...	Kate Bosworth|Jang Dong-gun|Geoffrey Rush|Dann...
director	James Cameron	Sngmoo Lee
runtime	162	100
genres	Action|Adventure|Fantasy|Science Fiction	Adventure|Fantasy|Action|Western|Thriller
release_date	2009-12-10 00:00:00	2010-12-02 00:00:00
vote_count	8458	74
vote_average	7.1	6.4
release_year	2009	2010
budget_adj	240886902	425000000
revenue_adj	2827123750	11087569
profit	2544505847	-413912431
profit_adj	2586236848	-413912431

Question 2-Movies with lower and higher budget.

In [67]:

```
movie_budget_high=pd.DataFrame(df_clean.loc[df_clean.budget.idxmax()])
movie_budget_low=pd.DataFrame(df_clean.loc[df_clean.budget.idxmin()])
pd.concat([movie_budget_high,movie_budget_low],axis=1)
```

Out[67]:

	2244	1151
popularity	0.25054	0.177102
budget	4.25e+08	1
revenue	11087569	0
original_title	The Warrior's Way	Fear Clinic
cast	Kate Bosworth|Jang Dong-gun|Geoffrey Rush|Dann...	Thomas Dekker|Robert Englund|Cleopatra Coleman...
director	Sngmoo Lee	Robert Hall
runtime	100	95
genres	Adventure|Fantasy|Action|Western|Thriller	Horror
release_date	2010-12-02 00:00:00	2014-10-31 00:00:00
vote_count	74	15
vote_average	6.4	4.1
release_year	2010	2014
budget_adj	425000000	0
revenue_adj	11087569	0
profit	-413912431	-1
profit_adj	-413912431	0

Question 3-Movies with lower and higher revenue.

In [72]:

```
movie_revenue_high=pd.DataFrame(df_clean.loc[df_clean.revenue.idxmax()])
movie_revenue_low=pd.DataFrame(df_clean.loc[df_clean.revenue.idxmin()])
pd.concat([movie_revenue_high,movie_revenue_low],axis=1)
```

Out[72]:

	1386	48
popularity	9.43277	2.93234
budget	2.37e+08	3e+07
revenue	2781505847	0
original_title	Avatar	Wild Card
cast	Sam Worthington|Zoe Saldana|Sigourney Weaver|S...	Jason Statham|Michael Angarano|Milo Ventimigli...
director	James Cameron	Simon West
runtime	162	92
genres	Action|Adventure|Fantasy|Science Fiction	Thriller|Crime|Drama
release_date	2009-12-10 00:00:00	2015-01-14 00:00:00
vote_count	8458	481
vote_average	7.1	5.3
release_year	2009	2015
budget_adj	240886902	27599987
revenue_adj	2827123750	0
profit	2544505847	-30000000
profit_adj	2586236848	-27599987

Question 4-Movie with shortest and longest runtime.

In [74]:

```
movie_runtime_high=pd.DataFrame(df_clean.loc[df_clean.runtime.idxmax()])
movie_runtime_low=pd.DataFrame(df_clean.loc[df_clean.runtime.idxmin()])
pd.concat([movie_runtime_high,movie_runtime_low],axis=1)
```

Out[74]:

	3894	92
popularity	0.006925	1.87604
budget	NaN	NaN
revenue	0	0
original_title	The Story of Film: An Odyssey	Mythica: The Necromancer
cast	Mark Cousins|Jean-Michel Frodon|Cari Beauchamp...	Melanie Stone|Adam Johnson|Kevin Sorbo|Nicola ...
director	Mark Cousins	A. Todd Smith
runtime	900	0
genres	Documentary	Fantasy|Action|Adventure
release_date	2011-09-03 00:00:00	2015-12-19 00:00:00
vote_count	14	11
vote_average	9.2	5.4
release_year	2011	2015
budget_adj	0	0
revenue_adj	0	0
profit	0	0
profit_adj	0	0

Count of genres in the dataset:

In [125]:

```
def genres(column):
    genres=df_clean[column].str.cat(sep='|')
    genres=pd.Series(genres.split("|"))
    count=genres.value_counts()
    return count
genres_count=genres('genres')
genres_count
```

Out[125]:

```
Drama              4759
Comedy             3792
Thriller           2907
Action             2384
Romance            1711
Horror             1637
Adventure          1471
Crime              1354
Family             1231
Science Fiction    1229
Fantasy             916
Mystery             810
Animation           699
Documentary         520
Music               408
History             334
War                 270
Foreign             188
TV Movie            167
Western             165
dtype: int64
```

Count of casts in the dataset:

In [174]:

```
def cast(column):
    cast=df_clean[column].str.cat(sep='|')
    cast=pd.Series(cast.split("|"))
    count=cast.value_counts()
    return count
cast_count=cast('cast')
cast_count
```

Out[174]:

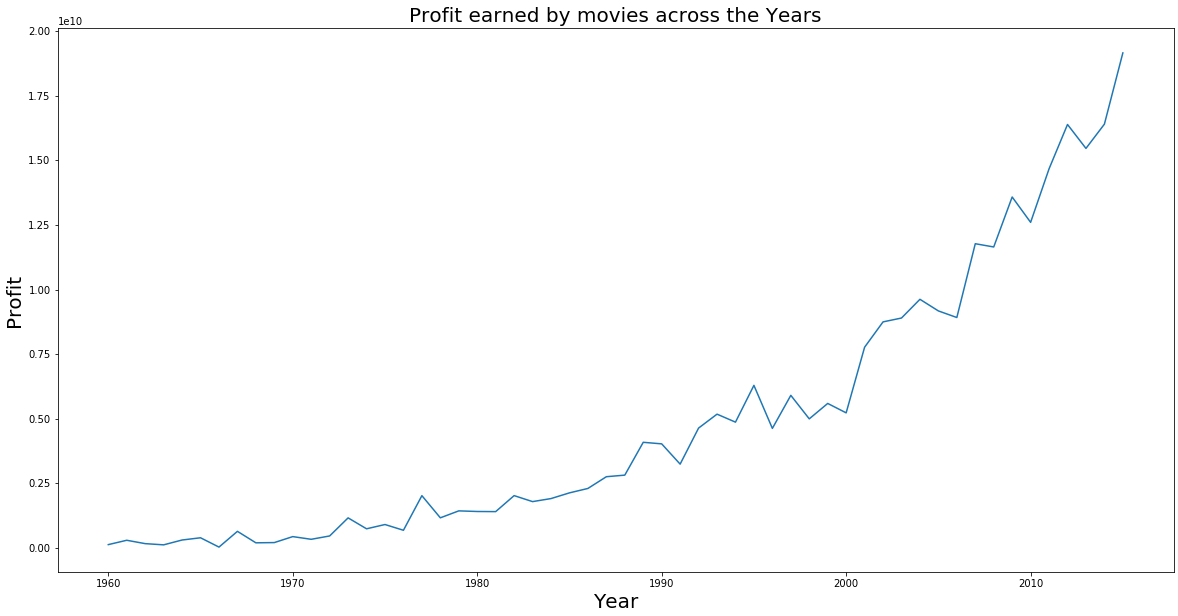
```
Robert De Niro        72
Samuel L. Jackson     71
Bruce Willis          62
Nicolas Cage          61
Michael Caine         53
Robin Williams        51
John Cusack           50
Morgan Freeman        49
John Goodman          49
Susan Sarandon        48
Liam Neeson           48
Alec Baldwin          47
Julianne Moore        47
Tom Hanks             46
Christopher Walken    46
Gene Hackman          46
Johnny Depp           46
Willem Dafoe          45
Sylvester Stallone    45
Dennis Quaid          45
Clint Eastwood        44
Meryl Streep          44
Ed Harris             44
Donald Sutherland     44
Robert Downey Jr.     43
Woody Harrelson       43
Keanu Reeves          43
Antonio Banderas      43
Ewan McGregor         43
Eddie Murphy          42
                      ..
Matt Corboy            1
Monica Swinn           1
Bernie McInerney       1
Gisele Fraga           1
GÃ©rard Darmon         1
George Williams        1
Barry Nelson           1
Yuta Hiraoka           1
ShÅ Hayami            1
Yukihide Benny         1
David Bagby            1
Rita Tushingham        1
Lymari Nadal           1
Daniela Ruah           1
Kevin Clash            1
Monique Alexander      1
Andrew Fullerton       1
Joyce Jameson          1
```

```
Pete Smith              1
Jesse Garcia            1
Thorsten Kaye           1
Jake Siegel             1
Michel Bouquet          1
Michael Polley          1
Kris Black              1
Will Tranfo             1
Bill Johnson            1
Dean Marshall           1
Alexa Nikolas           1
Ayrton Senna            1
Length: 19026, dtype: int64
```

Profit earned by movies across the year:

In [180]:

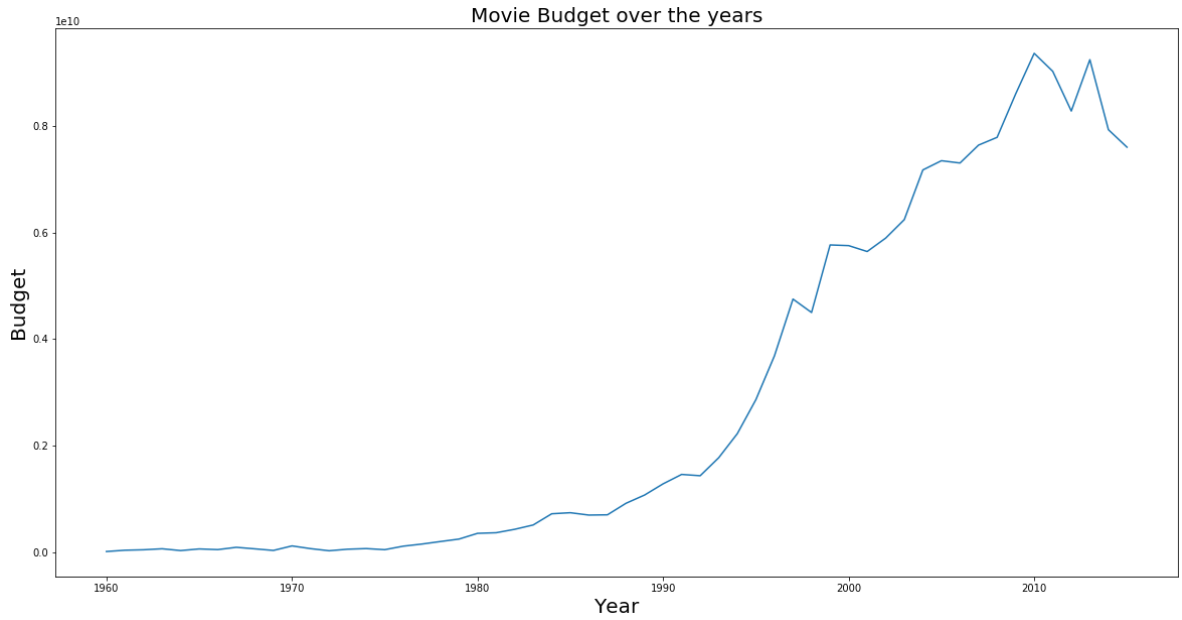
```
profit_year=df_clean.groupby('release_year')['profit'].sum()
plt.figure(figsize=(20,10));
plt.xlabel('Year',fontsize=20)
plt.ylabel('Profit',fontsize=20)
plt.title('Profit earned by movies across the Years',fontsize=20)
plt.plot(profit_year);
```



Movie budget Over the years:

In [181]:

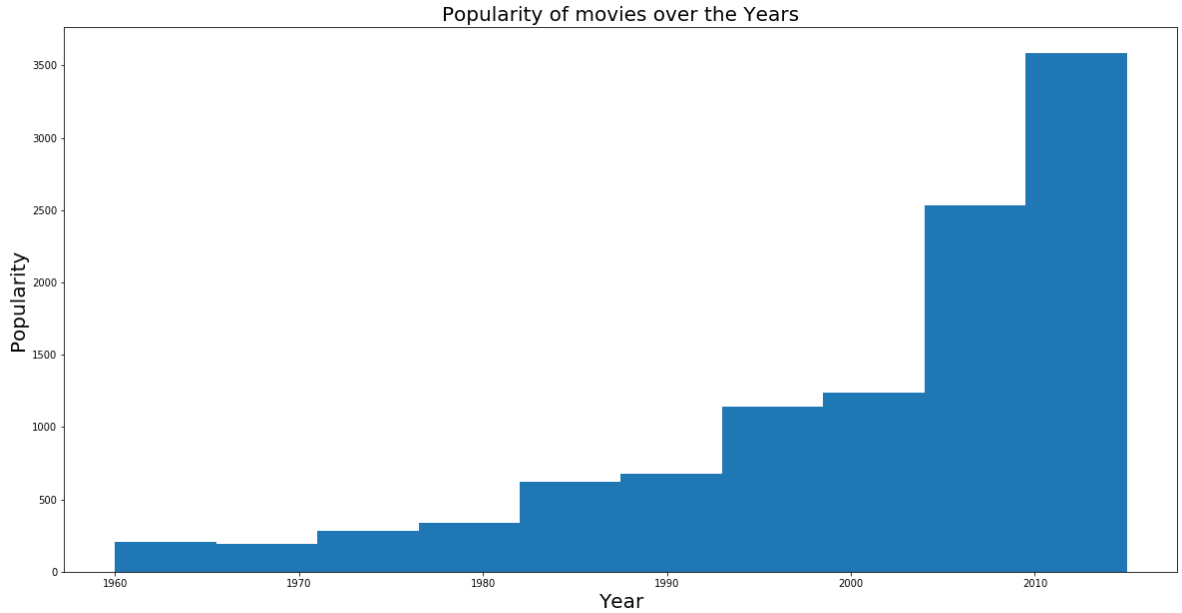
```
budget_year=df_clean.groupby('release_year')['budget'].sum()
plt.figure(figsize=(20,10));
plt.xlabel('Year',fontsize=20);
plt.ylabel('Budget',fontsize=20);
plt.title('Movie Budget over the years',fontsize=20);
plt.plot(budget_year);
```



Popularity of the movies over the years:

In [179]:

```
popularity=df_clean.groupby('popularity')['release_year'].mean()
plt.figure(figsize=(20,10));
plt.xlabel('Year',fontsize=20);
plt.ylabel('Popularity',fontsize=20);
plt.title('Popularity of movies over the Years',fontsize=20)
plt.hist(popularity);
```



Conclusions

> **Tip**: Finally, summarize your findings and the results that have been performed. Make sure that you are clear with regards to the limitations of your exploration. If you haven't done any statistical tests, do not imply any statistical conclusions. And make sure you avoid implying causation from correlation!

> **Tip**: Once you are satisfied with your work here, check over your report to make sure that it is satisfies all the areas of the rubric (found on the project submission page at the end of the lesson). You should also probably remove all of the "Tips" like this one so that the presentation is as polished as possible.

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'])
```