Project 2: Investigating the dataset on The Movie Database (TMDb)

from Bastian Boegel Submitted in June 2020

Introduction

In this project, I am going to analyse the dataset on The Movie Database (TMDb) provided from Kaggle. Source: https://www.kaggle.com/tmdb/tmdb-movie-metadata (https://www.kaggle.com/tmdb/tmdb-movie-metadata) The dataset includes Metadata on ~5,000 movies from TMDb. The database contains data on the plot, cast, crew, budget, etc. of several thousand films. Moreover, this dataset also provides data on revenues and popularity of the individual movies.

I chose to analyse the movie database as the mvie industry is gaining more and more importance and revenue. The increasing success of movie streaming platforms like netflix and Disney plus confirm the importance and demand of the movie production. Besides, I am personally a big fan of movies.

With my data analysis I am aiming to identify the relationship between variables such as revenues, budget, genres and popularity. In particular, I want to analyse potential predictors for the movies success in terms of a) revenue, b) profit or c) popularity. Potential predictors include the budget, the genre and the main actor. Question 1: Does the budget, the main actor or the genre of a movie predict a movie's success?

As a second part of the analysis, I want to analyse which genres are most popular and how did this change in the years that are covered in this dataset. Question 2: Which genres have been the most popular during the years?

One assumption I have to make is that the given revenue and budget for each movie is in the same currency (USD). There was no explanation provided on kaggle.com.

Data wrangling

Loading and importing data:

In [1]:

```
# import and load data
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
df = pd.read_csv('tmdb-movies.csv')
```

Gathering, assessing and cleaning data:

I will assess the structure of the dataset, check for errors, or missing values. Further, I will remove information and data that will not be used as well as duplicate information. I am aiming to get a clean datset for the data exploration step.

In [2]:

```
#Assessing data
#Getting a first overview
df.head()
```

Out[2]:

	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...	http:/
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 rows × 21 columns

Revenue and budget are given for the movies, however, no profit. I will calculate and add the profit as variable.

In [4]:

```
#Inspecting the dataset including datatypes and check for missing values
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10866 entries, 0 to 10865
Data columns (total 21 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   id                    10866 non-null  int64  
 1   imdb_id               10856 non-null  object 
 2   popularity            10866 non-null  float64
 3   budget                10866 non-null  int64  
 4   revenue               10866 non-null  int64  
 5   original_title        10866 non-null  object 
 6   cast                  10790 non-null  object 
 7   homepage              2936 non-null   object 
 8   director              10822 non-null  object 
 9   tagline               8042 non-null   object 
 10  keywords              9373 non-null   object 
 11  overview              10862 non-null  object 
 12  runtime               10866 non-null  int64  
 13  genres                10843 non-null  object 
 14  production_companies  9836 non-null   object 
 15  release_date          10866 non-null  object 
 16  vote_count            10866 non-null  int64  
 17  vote_average          10866 non-null  float64
 18  release_year          10866 non-null  int64  
 19  budget_adj            10866 non-null  float64
 20  revenue_adj           10866 non-null  float64
dtypes: float64(4), int64(6), object(11)
memory usage: 1.7+ MB
```

In [5]:

```
# Assessing basic statistics for each column:
df.describe()
```

Out[5]:

	id	popularity	budget	revenue	runtime	vote_count
count	10866.000000	10866.000000	1.086600e+04	1.086600e+04	10866.000000	10866.000000
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

Cleaning data:

- Removing data that is not needed like irrelevant columns (e.g. relaese date, homepage, etc.)
- Removing data that is double
- Adding the variable profit calculated as revenue - budget
- Preparing the variable genres as this will be further analyzed as independent variable later (values need to be splitted around the separator "|")
- Same holds for the variable main actor

In [15]:

```
#Unnecessary columns can be dropped
df.drop(['homepage', 'tagline', 'overview', 'imdb_id', 'keywords', 'release_date'], axi
s=1, inplace=True)
df.head()
```

Out[15]:

	id	popularity	budget	revenue	original_title	cast	director	runt
0	135397	32.985763	150000000	1513528810	Jurassic World	Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi...	Colin Trevorrow	
1	76341	28.419936	150000000	378436354	Mad Max: Fury Road	Tom Hardy|Charlize Theron|Hugh Keays-Byrne|Nic...	George Miller	
2	262500	13.112507	110000000	295238201	Insurgent	Shailene Woodley|Theo James|Kate Winslet|Ansel...	Robert Schwentke	
3	140607	11.173104	200000000	2068178225	Star Wars: The Force Awakens	Harrison Ford|Mark Hamill|Carrie Fisher|Adam D...	J.J. Abrams	
4	168259	9.335014	190000000	1506249360	Furious 7	Vin Diesel|Paul Walker|Jason Statham|Michelle ...	James Wan	

In [16]:

```
#Removing data that is double
#1. Duplicate rows
sum(df.duplicated())
```

Out[16]:

```
0
```

In [18]:

```
#df.drop_duplicates(inplace=True)
#print(sum(df.duplicated()))
```

In [19]:

```
#2. Missing values
df.isnull().sum()
```

Out[19]:

```
id                         0
popularity                 0
budget                     0
revenue                    0
original_title             0
cast                      76
director                  44
runtime                    0
genres                    23
production_companies    1030
vote_count                 0
vote_average               0
release_year               0
budget_adj                 0
revenue_adj                0
dtype: int64
```

In [20]:

```
#Critical variables do not have (many) missing avlues. I will remove rows with missing
 cast, production companies, etc...
df.dropna(inplace=True)
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 9772 entries, 0 to 10865
Data columns (total 15 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   id                    9772 non-null   int64  
 1   popularity            9772 non-null   float64
 2   budget                9772 non-null   int64  
 3   revenue               9772 non-null   int64  
 4   original_title        9772 non-null   object 
 5   cast                  9772 non-null   object 
 6   director              9772 non-null   object 
 7   runtime               9772 non-null   int64  
 8   genres                9772 non-null   object 
 9   production_companies  9772 non-null   object 
 10  vote_count            9772 non-null   int64  
 11  vote_average          9772 non-null   float64
 12  release_year          9772 non-null   int64  
 13  budget_adj            9772 non-null   float64
 14  revenue_adj           9772 non-null   float64
dtypes: float64(4), int64(6), object(5)
memory usage: 1.2+ MB
```

Additional cleaning checks:

- No renaming of columns needed (lower cases are used if applicable)
- The main dependent variables 'revenue' and 'budget' are integers, whicha allows for calculating the profit

In [21]:

```
#Create new variable "profit"
df['profit'] = df['revenue'] - df['budget']
df.head()
```

Out[21]:

	id	popularity	budget	revenue	original_title	cast	director	runt
0	135397	32.985763	150000000	1513528810	Jurassic World	Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi...	Colin Trevorrow	
1	76341	28.419936	150000000	378436354	Mad Max: Fury Road	Tom Hardy|Charlize Theron|Hugh Keays-Byrne|Nic...	George Miller	
2	262500	13.112507	110000000	295238201	Insurgent	Shailene Woodley|Theo James|Kate Winslet|Ansel...	Robert Schwentke	
3	140607	11.173104	200000000	2068178225	Star Wars: The Force Awakens	Harrison Ford|Mark Hamill|Carrie Fisher|Adam D...	J.J. Abrams	
4	168259	9.335014	190000000	1506249360	Furious 7	Vin Diesel|Paul Walker|Jason Statham|Michelle ...	James Wan	

In [25]:

```
#Check: There is no negative number for profit:
df.profit.describe()
```

Out[25]:

```
count    9.772000e+03
mean     2.805154e+07
std      1.014445e+08
min     -4.139124e+08
25%      0.000000e+00
50%      0.000000e+00
75%      1.420195e+07
max      2.544506e+09
Name: profit, dtype: float64
```

In [26]:

```
df.loc[df['profit'] < 0, 'profit'] = 0
```

In [27]:

```
df.profit.describe()
```

Out[27]:

```
count    9.772000e+03
mean     3.112680e+07
std      9.995743e+07
min      0.000000e+00
25%      0.000000e+00
50%      0.000000e+00
75%      1.420195e+07
max      2.544506e+09
Name: profit, dtype: float64
```

In [28]:

```
df.release_year.describe()
```

Out[28]:

```
count    9772.000000
mean     2000.878428
std        13.036794
min      1960.000000
25%      1994.000000
50%      2005.000000
75%      2011.000000
max      2015.000000
Name: release_year, dtype: float64
```

In [29]:

```
#I'll allocate the release years into groups for decades for a better analysis of the c
hange over year:
bin_edges = [1960, 1970, 1980, 1990, 2000, 2010, 2015]
bin_names = ['60s', '70s', '80s', '90s', '2000s', '2010s']
df['decades'] = pd.cut(df['release_year'], bin_edges, labels=bin_names)
df.head()
```

Out[29]:

	id	popularity	budget	revenue	original_title	cast	director	runt
0	135397	32.985763	150000000	1513528810	Jurassic World	Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi...	Colin Trevorrow	
1	76341	28.419936	150000000	378436354	Mad Max: Fury Road	Tom Hardy|Charlize Theron|Hugh Keays-Byrne|Nic...	George Miller	
2	262500	13.112507	110000000	295238201	Insurgent	Shailene Woodley|Theo James|Kate Winslet|Ansel...	Robert Schwentke	
3	140607	11.173104	200000000	2068178225	Star Wars: The Force Awakens	Harrison Ford|Mark Hamill|Carrie Fisher|Adam D...	J.J. Abrams	
4	168259	9.335014	190000000	1506249360	Furious 7	Vin Diesel|Paul Walker|Jason Statham|Michelle ...	James Wan	

I want to split the genres values in order to identify the main genre and consider that one as the "main genre". The old genre column will be replaced by 5 new columns with the individual genre. I will do the same afterwards for the cast column: I will consider the first mentioned actor as "main actor" later.

Please note, I found this code in the udacity chatroom posted by one of the students.
https://hub.udacity.com/rooms/community:nd002:843718-project-107-smg-2?contextType=room (https://hub.udacity.com/rooms/community:nd002:843718-project-107-smg-2?contextType=room)

In [30]:

```
#Preparing the variable genres (values need to be splitted around the separator "I")
df_genres = df['genres'].str.split('|', expand=True)
df_genres.head()
```

Out[30]:

	0	1	2	3	4
0	Action	Adventure	Science Fiction	Thriller	None
1	Action	Adventure	Science Fiction	Thriller	None
2	Adventure	Science Fiction	Thriller	None	None
3	Action	Adventure	Science Fiction	Fantasy	None
4	Action	Crime	Thriller	None	None

In [35]:

```
#Extending the dataset by splitted genre values to show genres separately
df[['genre1','genre2','genre3','genre4','genre5']] = df['genres'].str.split('|', expand
=True)
df.head()
```

Out[35]:

	id	popularity	budget	revenue	original_title	cast	director	runt
0	135397	32.985763	150000000	1513528810	Jurassic World	Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi...	Colin Trevorrow	
1	76341	28.419936	150000000	378436354	Mad Max: Fury Road	Tom Hardy|Charlize Theron|Hugh Keays-Byrne|Nic...	George Miller	
2	262500	13.112507	110000000	295238201	Insurgent	Shailene Woodley|Theo James|Kate Winslet|Ansel...	Robert Schwentke	
3	140607	11.173104	200000000	2068178225	Star Wars: The Force Awakens	Harrison Ford|Mark Hamill|Carrie Fisher|Adam D...	J.J. Abrams	
4	168259	9.335014	190000000	1506249360	Furious 7	Vin Diesel|Paul Walker|Jason Statham|Michelle ...	James Wan	

5 rows × 27 columns

In [42]:

```
#Check:
df['genre1'].unique()
```

Out[42]:

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

In [36]:

```
#Preparing the variable main actors (values for cast need to be splitted around the sep
arator "I")
df_actors = df['cast'].str.split('|', expand=True)
df_actors.head()
```

Out[36]:

	0	1	2	3	4
0	Chris Pratt	Bryce Dallas Howard	Irrfan Khan	Vincent D'Onofrio	Nick Robinson
1	Tom Hardy	Charlize Theron	Hugh Keays-Byrne	Nicholas Hoult	Josh Helman
2	Shailene Woodley	Theo James	Kate Winslet	Ansel Elgort	Miles Teller
3	Harrison Ford	Mark Hamill	Carrie Fisher	Adam Driver	Daisy Ridley
4	Vin Diesel	Paul Walker	Jason Statham	Michelle Rodriguez	Dwayne Johnson

In [37]:

```
#Extending the dataset by splitted cast values to show actors separately
df[['actor1','actor2','actor3','actor4','actor5']] = df['cast'].str.split('|', expand=True)
df.head()
```

Out[37]:

	id	popularity	budget	revenue	original_title	cast	director	runt
0	135397	32.985763	150000000	1513528810	Jurassic World	Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi...	Colin Trevorrow	
1	76341	28.419936	150000000	378436354	Mad Max: Fury Road	Tom Hardy|Charlize Theron|Hugh Keays-Byrne|Nic...	George Miller	
2	262500	13.112507	110000000	295238201	Insurgent	Shailene Woodley|Theo James|Kate Winslet|Ansel...	Robert Schwentke	
3	140607	11.173104	200000000	2068178225	Star Wars: The Force Awakens	Harrison Ford|Mark Hamill|Carrie Fisher|Adam D...	J.J. Abrams	
4	168259	9.335014	190000000	1506249360	Furious 7	Vin Diesel|Paul Walker|Jason Statham|Michelle ...	James Wan	

5 rows × 27 columns

In [43]:

```
#Check:
df['actor1'].unique()
```

Out[43]:

```
array(['Chris Pratt', 'Tom Hardy', 'Shailene Woodley', ...,
       'Innokentiy Smoktunovskiy', 'Tatsuya Mihashi', 'Harold P. Warren'],
      dtype=object)
```

In [44]:

```
#Inspecting the dataset again
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 9772 entries, 0 to 10865
Data columns (total 27 columns):
 #   Column                Non-Null Count  Dtype   
---  ------                --------------  -----   
 0   id                    9772 non-null   int64   
 1   popularity            9772 non-null   float64 
 2   budget                9772 non-null   int64   
 3   revenue               9772 non-null   int64   
 4   original_title        9772 non-null   object  
 5   cast                  9772 non-null   object  
 6   director              9772 non-null   object  
 7   runtime               9772 non-null   int64   
 8   genres                9772 non-null   object  
 9   production_companies  9772 non-null   object  
 10  vote_count            9772 non-null   int64   
 11  vote_average          9772 non-null   float64 
 12  release_year          9772 non-null   int64   
 13  budget_adj            9772 non-null   float64 
 14  revenue_adj           9772 non-null   float64 
 15  profit                9772 non-null   int64   
 16  decades               9740 non-null   category
 17  genre1                9772 non-null   object  
 18  genre2                7813 non-null   object  
 19  genre3                4710 non-null   object  
 20  genre4                1848 non-null   object  
 21  genre5                507 non-null    object  
 22  actor1                9772 non-null   object  
 23  actor2                9688 non-null   object  
 24  actor3                9635 non-null   object  
 25  actor4                9566 non-null   object  
 26  actor5                9359 non-null   object  
dtypes: category(1), float64(4), int64(7), object(15)
memory usage: 2.0+ MB
```

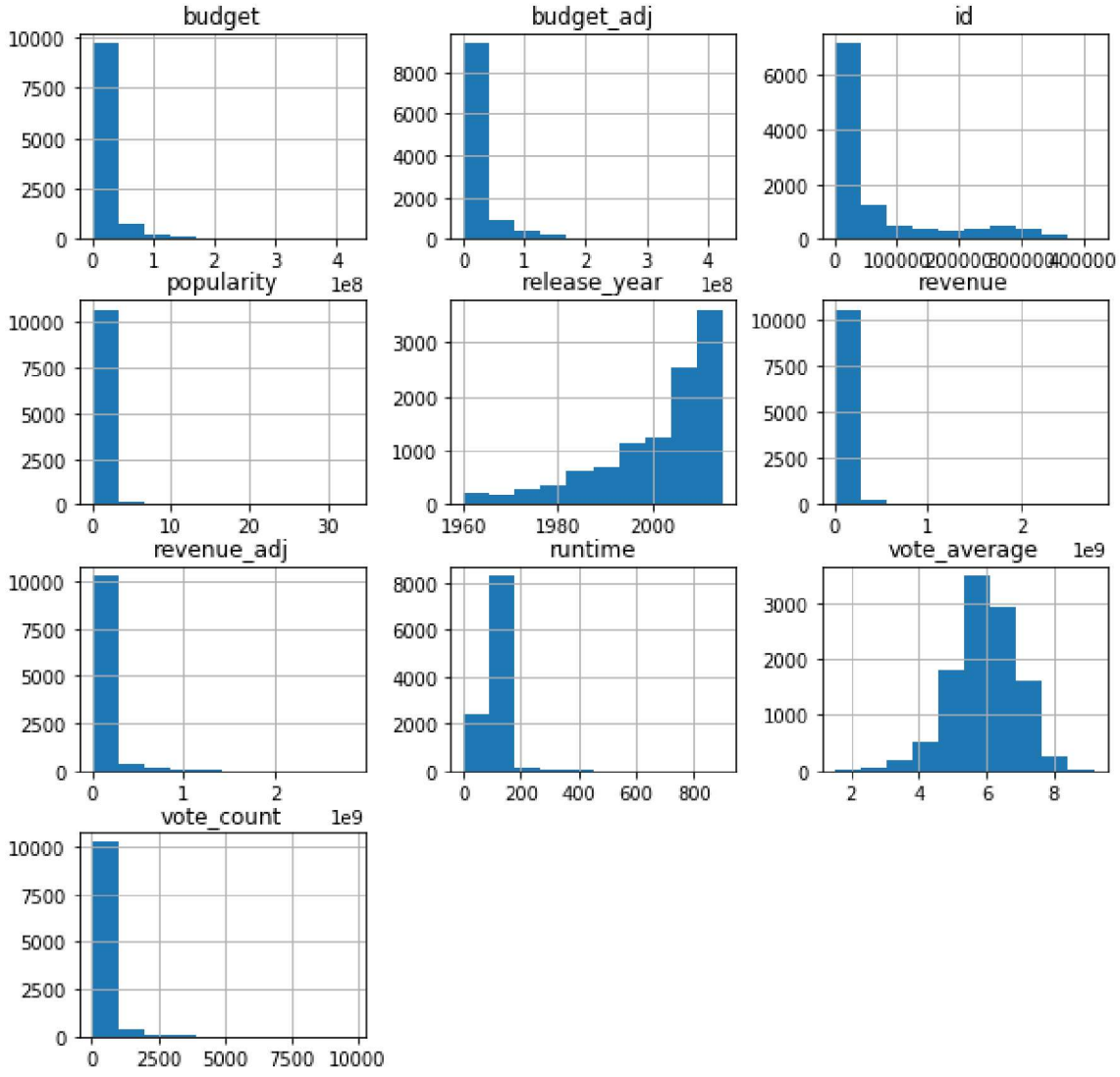
The data looks clean and ready to be further analysed.

Exploratory Data Analysis

General exploratory analysis

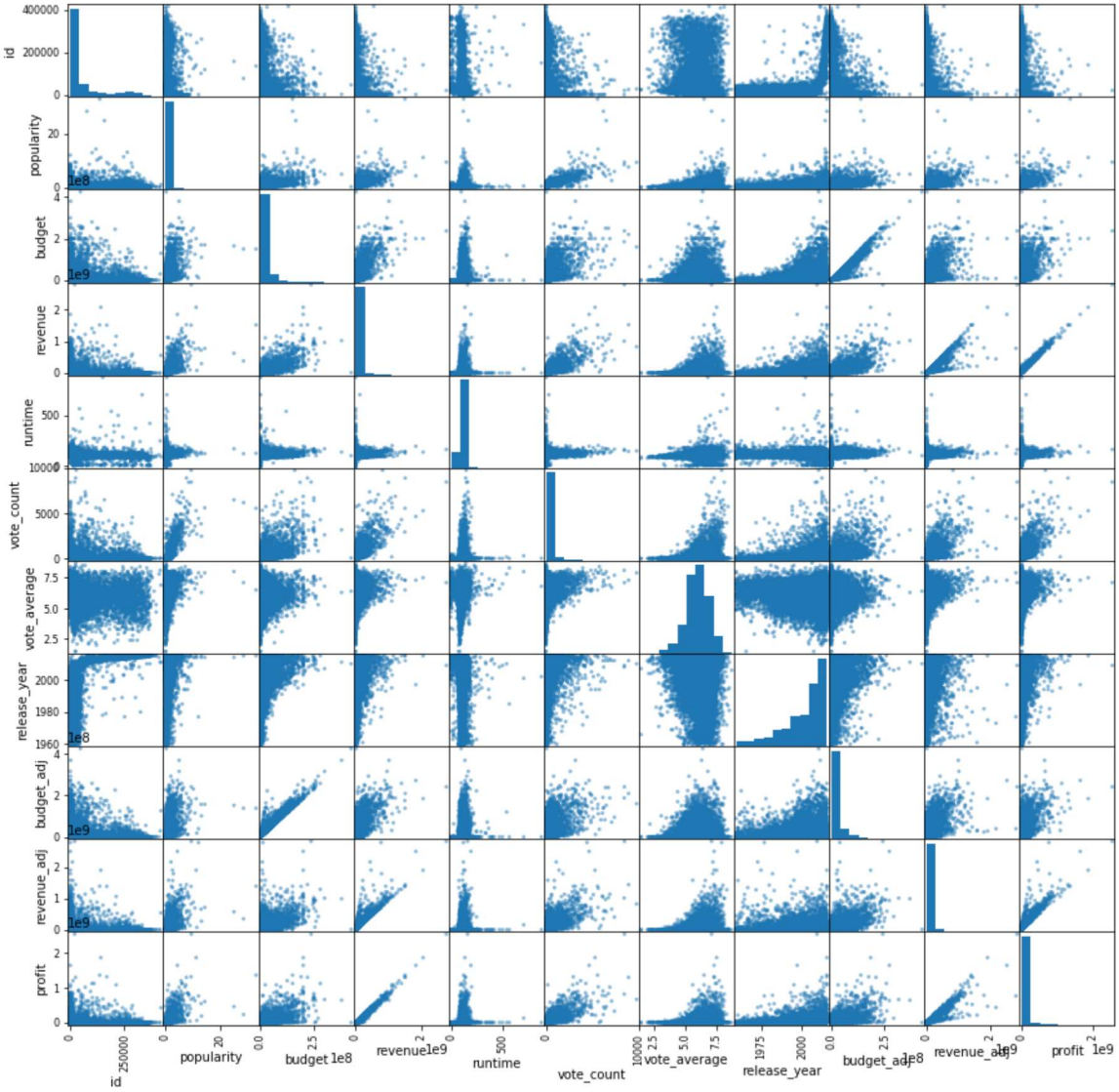
In [6]:

```
#Assessing numerical data:
df.hist(figsize=(10, 10));
```



In [45]:

```
pd.plotting.scatter_matrix(df, figsize=(15, 15));
```



Question 1: Does the budget, the main actor or the genre of a movie predict a movie's success?

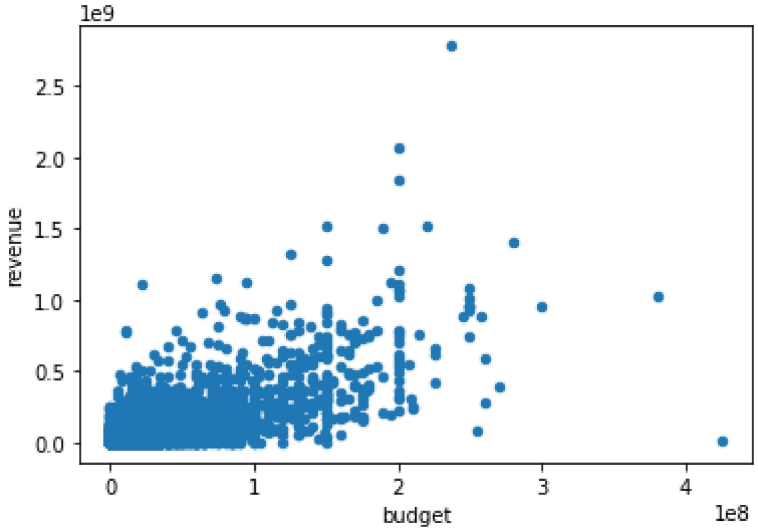
The first variable to be looked at is the budget. A scatterplot can be created in order to detect a potential correlation between a movie's budget and its revenue, its profit or its popularity.

In [46]:

```
# plot relationship between budget and revenue
df.plot(x = 'budget', y = 'revenue', kind ='scatter')
```

Out[46]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x22766e8ecc8>
```

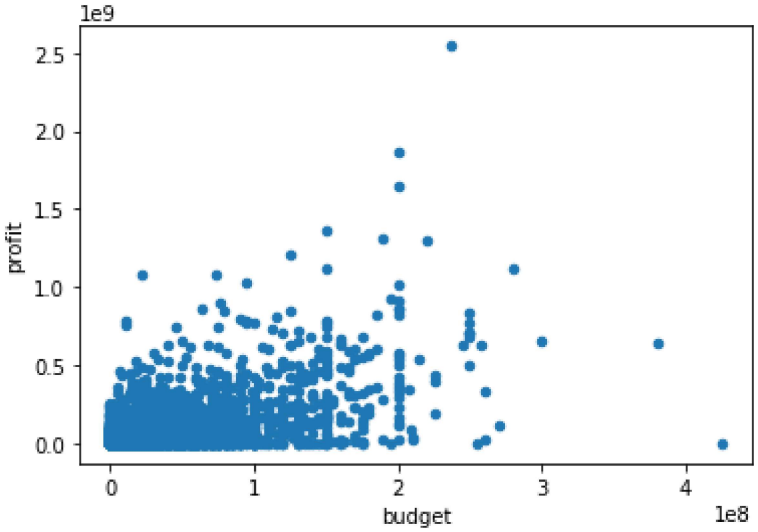


In [47]:

```
# plot relationship between budget and revenue
df.plot(x = 'budget', y = 'profit', kind ='scatter')
```

Out[47]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2276748f648>
```

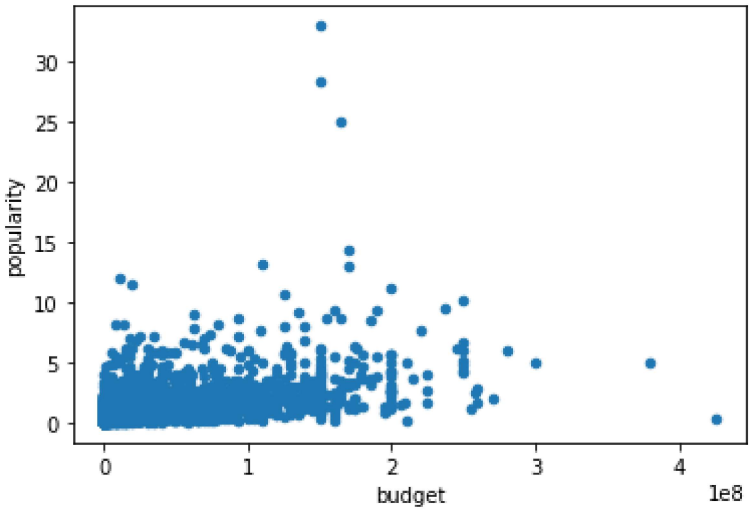


In [48]:

```
# plot relationship between budget and revenue
df.plot(x = 'budget', y = 'popularity', kind ='scatter')
```

Out[48]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x22767013cc8>
```



-> First findings:

- Revenue and budget have are positively correlated
- Profit and budget are not really correlated
- Popularity and budget are not really correlated

The second variable to be looked at is the main actor. Which (main) actors are associated with movies with the highest revenues?

In [101]:

```
#Dataset is to be analysed within the top 100 movies based on revenue:
df_top_100_movies = df.sort_values(by=['revenue'], ascending=False).head(100)
df_top_100_movies['original_title'].nunique()
```

Out[101]:

```
100
```

In [102]:

```
df_top_100_movies.describe()
```

Out[102]:

	id	popularity	budget	revenue	runtime	vote_count	vote
count	100.000000	100.000000	1.000000e+02	1.000000e+02	100.000000	100.000000	1
mean	43314.780000	5.191795	1.542350e+08	8.979584e+08	130.070000	3475.060000	
std	55833.708222	4.467371	6.694002e+07	3.189943e+08	25.168625	1839.603382	
min	11.000000	0.436803	1.050000e+07	6.118994e+08	88.000000	201.000000	
25%	673.750000	2.755044	1.122500e+08	7.097980e+08	110.250000	2245.750000	
50%	12299.500000	4.264860	1.500000e+08	8.079637e+08	132.000000	3169.500000	
75%	69080.750000	6.067551	2.000000e+08	9.590500e+08	144.250000	4266.000000	
max	211672.000000	32.985763	3.800000e+08	2.781506e+09	201.000000	9767.000000	

The top movies have a revenue of >= 9.590500e+08 (upper quartile of top 100 movies).

In [98]:

```
df_top_100_movies.head(5)
```

Out[98]:

	id	popularity	budget	revenue	original_title	cast	director
1386	19995	9.432768	237000000	2781505847	Avatar	Sam Worthington|Zoe Saldana|Sigourney Weaver|S...	James Cameron
3	140607	11.173104	200000000	2068178225	Star Wars: The Force Awakens	Harrison Ford|Mark Hamill|Carrie Fisher|Adam D...	J.J. Abrams
5231	597	4.355219	200000000	1845034188	Titanic	Kate Winslet|Leonardo DiCaprio|Frances Fisher|...	James Cameron
4361	24428	7.637767	220000000	1519557910	The Avengers	Robert Downey Jr.|Chris Evans|Mark Ruffalo|Chr...	Joss Whedon
0	135397	32.985763	150000000	1513528810	Jurassic World	Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi...	Colin Trevorrow

5 rows × 27 columns

-> Sam Worthington, Harrison Ford, Kate Winslet, Robert Downey Jr. and Chris Pratt are the main actors in the top 5 movies accroding to revenue.

In [105]:

```
#Which actors are the main actors within the top movies of revenue?
df_top_100_movies['actor1'].value_counts()
```

Out[105]:

```
Daniel Radcliffe          8
Kristen Stewart           4
Jennifer Lawrence         4
Johnny Depp               4
Robert Downey Jr.         4
Will Smith                3
Elijah Wood               3
Mike Myers                3
Tobey Maguire             3
Shia LaBeouf              3
Tom Hanks                 3
Sandra Bullock            3
Harrison Ford             2
Daniel Craig              2
Chris Pratt               2
Martin Freeman            2
Vin Diesel                2
Christian Bale            2
Ewan McGregor             2
Andrew Garfield           2
Ray Romano                2
Jack Black                2
Tom Cruise                2
Angelina Jolie            1
Andy Serkis               1
John Cusack               1
Mark Hamill               1
William Moseley           1
Sam Worthington           1
John Leguizamo            1
Keanu Reeves              1
Jonathan Taylor Thomas    1
Hugh Jackman              1
Jim Caviezel              1
Scott Adsit               1
Matthew McConaughey       1
Ian McKellen              1
Henry Thomas              1
Billy Crystal             1
Steve Carell              1
Craig T. Nelson           1
Amy Poehler               1
Leonardo DiCaprio         1
Liam Neeson               1
Henry Cavill              1
Kate Winslet              1
Bruce Willis              1
Ed Asner                  1
Paul Walker               1
Sam Neill                 1
Albert Brooks             1
Patton Oswalt             1
Ben Stiller               1
Mia Wasikowska            1
Chris Evans               1
Kristen Bell              1
Name: actor1, dtype: int64
```

-> Among the top 100 movies according to revenue, the following 5 actors appear the most: Daniel Radcliffe, Kristen Stewart, Jennifer Lawrence, Johnny Depp and Robert Downey Jr..

The third variable to be looked at is the genre. Which (main) genres are associated with movies with the highest revenues?

In [106]:

```
#Which genres are the main genres within the top movies of revenue?
df_top_100_movies['genre1'].value_counts()
```

Out[106]:

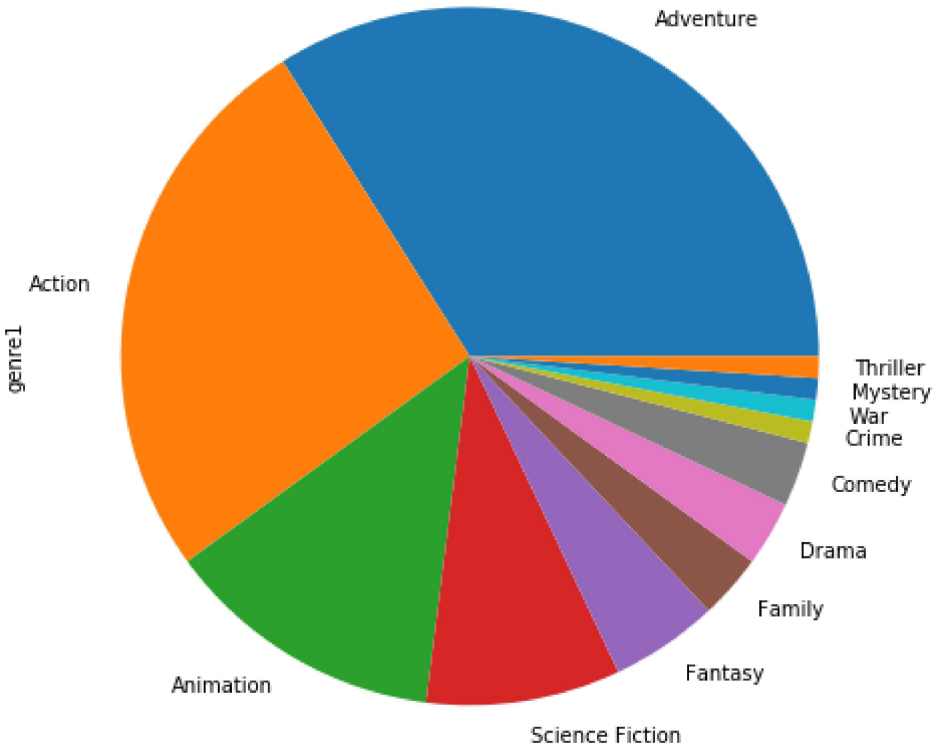
```
Adventure          34
Action             26
Animation          13
Science Fiction     9
Fantasy             5
Family              3
Drama               3
Comedy              3
Crime               1
War                 1
Mystery             1
Thriller            1
Name: genre1, dtype: int64
```

In [107]:

```
# also view with a pie chart
df_top_100_movies['genre1'].value_counts().plot(kind='pie', figsize=(8,8))
```

Out[107]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x227627d3e88>
```

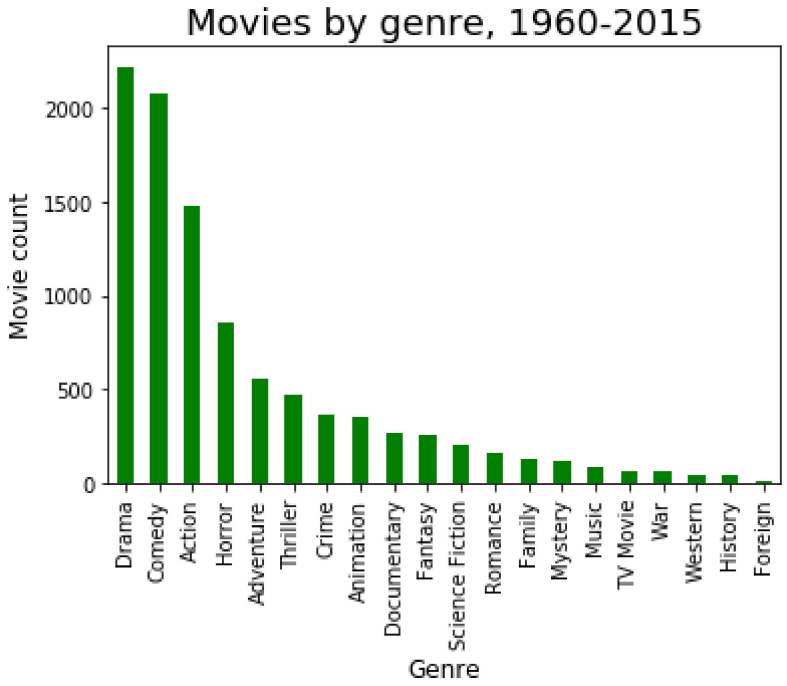


-> Among the top 100 movies according to revenue, the following 5 genres appear the most: Adventure, Action, Animation, Science Fiction and Fantasy. Looking at the pie chart it looks like the top 3 genres Adventure, Action and Animation make almost 75% of the topp genres among the top 100 movies.

Question 2: Which genres have been the most popular (highest number of movies) during the years?

In [119]:

```
#Overview of genres over time:
df['genre1'].value_counts().plot(kind='bar', color='g');
plt.title('Movies by genre, 1960-2015', size=18)
plt.xlabel('Genre', size=12)
plt.ylabel('Movie count', size=12);
```



In [120]:

```
genres_per_decades = df.groupby(['decades'])['genre1'].value_counts()
```

In [118]:

```
#The list above is too long, I will look at top 3 genres per decade:
genres_per_decades.groupby(level=0).nlargest(3).reset_index(level=0, drop=True)
```

Out[118]:

```
decades  genre1
60s      Drama      75
         Comedy     68
         Action     56
70s      Drama     115
         Action     98
         Comedy     77
80s      Comedy    260
         Drama     194
         Action    178
90s      Comedy    437
         Drama     417
         Action    317
2000s    Comedy    805
         Drama     791
         Action    502
2010s    Drama     626
         Comedy    430
         Action    320
Name: genre1, dtype: int64
```

-> Drama and comedy have been the most popular genres: From the 60s to the 80s, Drama has been the most poular genre and the agaain in 2010s. During the 90s and the 2000s, Comedy has been the most popular one.

Conclusions

In the first part of my data analysis I was aiming to identify potential predictors for a movie's success in terms of a) revenue, b) profit or c) popularity. Potential predictors include the budget, the genre and the main actor. The analyses above show: -> Revenue and budget have are positively correlated, whereas I could not show a clear correlation between profit and budget and popularity and budget;

Regarding the actors and genres a dataset of the top 100 movies according to revenue has been analysed. -> Some actors show up more frequently in successful movies than others: Sam Worthington, Harrison Ford, Kate Winslet, Robert Downey Jr. and Chris Pratt are the main actors in the top 5 movies accroding to revenue; -> Among the top 100 movies according to revenue, the following 5 actors appear the most: Daniel Radcliffe, Kristen Stewart, Jennifer Lawrence, Johnny Depp and Robert Downey Jr.; Among the top 100 movies according to revenue, the following 5 genres appear the most: -> Adventure, Action, Animation, Science Fiction and Fantasy.

As a second part of the analysis, I analysed which genres are most popular and how did this change in the years that are covered in this dataset: -> Drama and comedy have been the most popular genres: From the 60s to the 80s, Drama has been the most poular genre and the agaain in 2010s. During the 90s and the 2000s, Comedy has been the most popular one.