Investigating the TMDb Movies Dataset

August 9, 2022

1 Project: Investigate a Dataset (TMDb Movies Data)

1.1 Table of Contents

Introduction

Since the invention of motion pictures, many movies have been produced across different genres.

The Movie Database (TMDB) is a community built movie and TV database. It is a reputable source of movies data avaiblable on the internet. Since 2008, TMDb has consistently kept an up to date record of movies records.

In order to help better understand the investigation, below is a description of the content of each column

id: THis is the unique identifier for each movie in the dataset.

Popularity: This is the popularity rating for each movie.

Budget: This is the budgeted amount for the production of the movie.

Revenue: This is revenue generated by the movie after its release.

Original_title: The original title of the movie.

Cast: This is the cast of the movie.

Director: The directors of the movie.

Runtime: The time in minutes the movie runs for.

Production_companies: This columns carries the companies that produced the movie.

Release_date: This the year the movies are intended to be released.

Vote_count: A count of votes on a movie.

Vote_average: Average vote per movie.

Budget_adj: This column represents the adjusted budget accounting for inflation.

Revenue_adj: This the adjusted revenue per movies accounting for inflation.

We will investigate the data we have pulled from TMDb with the goal of getting insights and answering some pertinent questions. These questions are as follows:

What are the 20 most popular movies and what are their features?

What are the 20 highest grossing movies in terms of revenue?

What features are associated with the 20 highest grossing movies?

Does the budget of a movie affect its revenue or popularity?

What is the correlation revenue and popularity?

What is the correlation between the higest grossing movies and their popularity?

[1]:
```
# Use this cell to set up import statements for all of the packages that you
#   plan to use.
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

# 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
```

Data Wrangling

> We will attempt to explore the dataset with the goal of identifying the noise within it. These noise will cleaned and made ready to answer the questions we have outlined above.
> ### General Properties

[2]:
```
# 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_movies = pd.read_csv('tmdb-movies.csv')

# a summary look of our data
df_movies.head()
```

[2]:
```
       id    imdb_id  popularity     budget     revenue  \
0  135397  tt0369610   32.985763  150000000  1513528810
1   76341  tt1392190   28.419936  150000000   378436354
2  262500  tt2908446   13.112507  110000000   295238201
3  140607  tt2488496   11.173104  200000000  2068178225
4  168259  tt2820852    9.335014  190000000  1506249360

                 original_title  \
0                Jurassic World
```

```
1           Mad Max: Fury Road
2                    Insurgent
3  Star Wars: The Force Awakens
4                    Furious 7

                                                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…
3  Harrison Ford|Mark Hamill|Carrie Fisher|Adam D…
4  Vin Diesel|Paul Walker|Jason Statham|Michelle …

                                            homepage          director  \
0                     http://www.jurassicworld.com/   Colin Trevorrow
1                       http://www.madmaxmovie.com/     George Miller
2     http://www.thedivergentseries.movie/#insurgent  Robert Schwentke
3  http://www.starwars.com/films/star-wars-episod…      J.J. Abrams
4                          http://www.furious7.com/          James Wan

                        tagline  …  \
0            The park is open.  …
1           What a Lovely Day.  …
2       One Choice Can Destroy You  …
3  Every generation has a story.  …
4           Vengeance Hits Home  …

                                            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
3  Thirty years after defeating the Galactic Empi…     136
4  Deckard Shaw seeks revenge against Dominic Tor…     137

                                      genres  \
0  Action|Adventure|Science Fiction|Thriller
1  Action|Adventure|Science Fiction|Thriller
2         Adventure|Science Fiction|Thriller
3   Action|Adventure|Science Fiction|Fantasy
4                      Action|Crime|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
3          Lucasfilm|Truenorth Productions|Bad Robot     12/15/15       5292
4  Universal Pictures|Original Film|Media Rights …       4/1/15       2947
```

```
   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           7.5          2015  1.839999e+08  1.902723e+09
4           7.3          2015  1.747999e+08  1.385749e+09

[5 rows x 21 columns]
```

[3]:
```python
# exploring the shape of our data set
df_movies.shape
```

[3]:
```
(10866, 21)
```

Above, we can see that there are **10866** rows in our data set with **21** columns or features

[4]:
```python
# a general info of our data set
df_movies.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
```

From the out put above, we can that there are **4** columns in our data set with float datatypes, **6** integer datatype columns and **11** object datatype columns. Additionally, when we look closely, we can see that some of the columns have less than **10866** records. This indicates missing values. We will investigate this further in the cell that follows.

[5]:
```
# checking for null values
df_movies.isnull().sum()
```

[5]:
```
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
```

Above, we can see the columns with missing values and count of missing values they have. We will deal with the missing values depending on how they impact our analysis.

[6]:
```
# drop columns with missing which do not impact our analysis
df_movies = df_movies.drop(['homepage', 'tagline', 'keywords', 'imdb_id',␣
↪'overview'], axis=1)
```

[7]:
```
# confirming that the columns above have been dropped
df_movies.isnull().sum()
```

[7]:
```
id                       0
popularity               0
budget                   0
revenue                  0
original_title           0
cast                    76
director                44
```

```
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
```

From the out put above, we have successfully dropped the columns with missing values which we consider irrelevant to the question we want address in this investigation. For the columns which still missing data as you can see above, we will use the mode of the to replace the missing data. This is suitable since the data is categorical in nature.

1.1.1 Replace missing values

In this section, we will replace missing values ##### Production_companies column

[8]:
```
# determine the mode of the production companies column
df_movies['production_companies'].mode()
```

[8]:
```
0    Paramount Pictures
Name: production_companies, dtype: object
```

[9]:
```
#use fillna() function to fill in missing values using the column's mode
df_movies['production_companies'].fillna('Paramount Pictures', inplace=True)
```

[10]:
```
#confirm missing values have been replaced
df_movies['production_companies'].isnull().sum()
```

[10]:
```
0
```

Director column

[11]:
```
# determine the mode of the genres column
df_movies['director'].mode()
```

[11]:
```
0    Woody Allen
Name: director, dtype: object
```

Above we can see that the mode in our column is **Woody Allen**. In the cell below, we use the *fillna()* function to replace the null values with the mode.

[12]:
```
#use fillna() function to fill in missing values using the column's mode
df_movies['director'].fillna('Woody Allen', inplace=True)
```

```
[13]: #confirm missing values have been replaced
df_movies['director'].isnull().sum()
```

```
[13]: 0
```

Dropping rows with null values

We will now drop rows that still have null values within them. Becuase it is a small part of our data set, our data loss is not significant.

```
[14]: # dropping rows with null values
df_movies.dropna(inplace=True)
```

```
[15]: # confirming that our data set has no null values
df_movies.isnull().sum()
```

```
[15]: id                      0
popularity              0
budget                  0
revenue                 0
original_title          0
cast                    0
director                0
runtime                 0
genres                  0
production_companies    0
release_date            0
vote_count              0
vote_average            0
release_year            0
budget_adj              0
revenue_adj             0
dtype: int64
```

We can see from the output above that our dataset no longer has any null values.

1.1.2 Checking for duplicates

We will now check our dataset to make there are no duplicate entries.

```
[16]: # checking for duplicates records
df_movies.duplicated().sum()
```

```
[16]: 1
```

We can see that we have one duplicate entry in our dataset. To deal with this, we will drop it.

[17]:
```
# dropping duplicates
df_movies.drop_duplicates(inplace=True)
```

[18]:
```
# confirm that duplicate entry has been dropped
df_movies.duplicated().sum()
```

[18]:
```
0
```

Now that we have cleaned our data, we will take a look at what the shape of our data looks like and also do some statistical analysis.

Shape of the dataset after cleaning

[19]:
```
# the shape of the data set
df_movies.shape
```

[19]:
```
(10767, 16)
```

Our data set now has **10767** rows and **16** columns after cleaning.

1.1.3 Summary Statistical Analysis

Our goal is to get a summary statistical analysis of our dataset following the cleaning we carried out. This will give an insight into the spread of our data set and also the correlation between our colums. We will also employ the use of some visuals to give us a better perspective.

[20]:
```
# using describe() function to a summary statistics of our dataset.
df_movies.describe()
```

[20]:
```
                  id    popularity        budget       revenue       runtime  \
count   10767.000000  10767.000000  1.076700e+04  1.076700e+04  10767.000000
mean    65477.144144      0.650924  1.475532e+07  4.018610e+07    102.413393
std     91703.303390      1.003565  3.102387e+07  1.174783e+08     30.906009
min         5.000000      0.000065  0.000000e+00  0.000000e+00      0.000000
25%     10559.500000      0.209957  0.000000e+00  0.000000e+00     90.000000
50%     20423.000000      0.386062  0.000000e+00  0.000000e+00     99.000000
75%     74507.500000      0.719253  1.600000e+07  2.476490e+07    112.000000
max    417859.000000     32.985763  4.250000e+08  2.781506e+09    900.000000

         vote_count  vote_average  release_year    budget_adj   revenue_adj
count  10767.000000  10767.000000  10767.000000  1.076700e+04  1.076700e+04
mean     219.137364      5.967549   2001.283459  1.770705e+07  5.183338e+07
std      577.964702      0.931426     12.815909  3.442339e+07  1.452125e+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%       39.000000      6.000000   2006.000000  0.000000e+00  0.000000e+00
75%      147.000000      6.600000   2011.000000  2.103337e+07  3.432264e+07
max     9767.000000      9.200000   2015.000000  4.250000e+08  2.827124e+09
```

It should be noted that the *describe()* function runs on only numeric data. Hence, the summary statistics it generates is only on dataset that are numeric in nature or continuous variables. **Count** from the output represents a count of the records in each column whereas *mean* is the average value, *std* is standard deviation. The orders are self explanatory.

Exploratory Data Analysis

The dataset will be explored with the aim to answer questions which were outlined in the beginning of the report. ### Research Question 1: What are the 20 most popular movies and what are their features?

[21]:
```
# we will extract the 20 most popular movies
Top_20_movies = df_movies.sort_values(by='popularity', ascending=False)
Top_20_movies.head(20)
```

[21]:
```
          id  popularity     budget     revenue  \
0     135397   32.985763  150000000  1513528810
1      76341   28.419936  150000000   378436354
629   157336   24.949134  165000000   621752480
630   118340   14.311205  170000000   773312399
2     262500   13.112507  110000000   295238201
631   100402   12.971027  170000000   714766572
1329      11   12.037933   11000000   775398007
632   245891   11.422751   20000000    78739897
3     140607   11.173104  200000000  2068178225
633   131631   10.739009  125000000   752100229
634   122917   10.174599  250000000   955119788
1386   19995    9.432768  237000000  2781505847
1919   27205    9.363643  160000000   825500000
4     168259    9.335014  190000000  1506249360
5     281957    9.110700  135000000   532950503
2409     550    8.947905   63000000   100853753
635   177572    8.691294  165000000   652105443
6      87101    8.654359  155000000   440603537
2633     120    8.575419   93000000   871368364
2875     155    8.466668  185000000  1001921825

                                  original_title  \
0                                 Jurassic World
1                             Mad Max: Fury Road
629                                 Interstellar
630                      Guardians of the Galaxy
2                                      Insurgent
631          Captain America: The Winter Soldier
1329                                   Star Wars
632                                    John Wick
3                   Star Wars: The Force Awakens
633            The Hunger Games: Mockingjay - Part 1
```

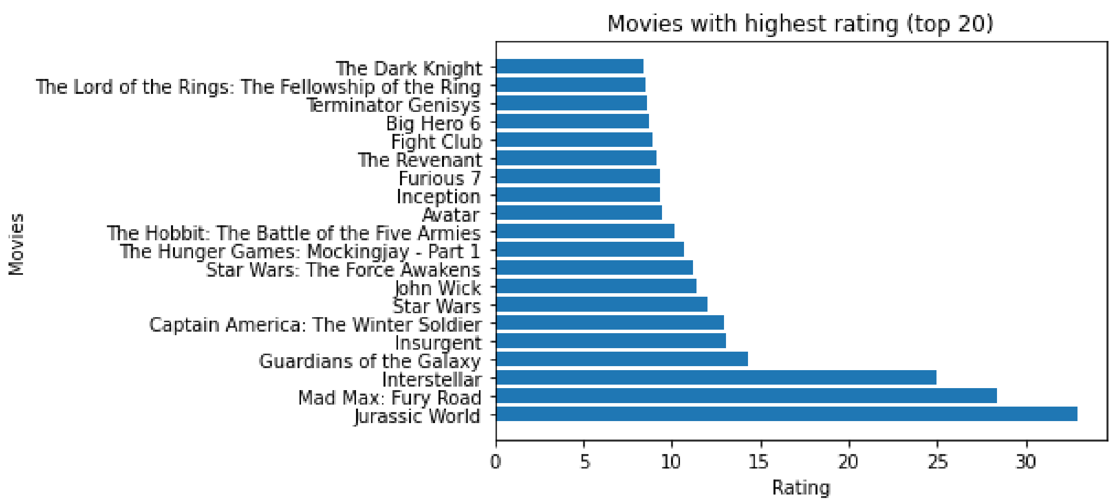
```
634           The Hobbit: The Battle of the Five Armies
1386                                             Avatar
1919                                          Inception
4                                             Furious 7
5                                          The Revenant
2409                                         Fight Club
635                                          Big Hero 6
6                                   Terminator Genisys
2633  The Lord of the Rings: The Fellowship of the Ring
2875                                    The Dark Knight

                                                  cast  \
0     Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi…
1     Tom Hardy|Charlize Theron|Hugh Keays-Byrne|Nic…
629   Matthew McConaughey|Jessica Chastain|Anne Hath…
630   Chris Pratt|Zoe Saldana|Dave Bautista|Vin Dies…
2     Shailene Woodley|Theo James|Kate Winslet|Ansel…
631   Chris Evans|Scarlett Johansson|Sebastian Stan|…
1329  Mark Hamill|Harrison Ford|Carrie Fisher|Peter …
632   Keanu Reeves|Michael Nyqvist|Alfie Allen|Wille…
3     Harrison Ford|Mark Hamill|Carrie Fisher|Adam D…
633   Jennifer Lawrence|Josh Hutcherson|Liam Hemswor…
634   Martin Freeman|Ian McKellen|Richard Armitage|K…
1386  Sam Worthington|Zoe Saldana|Sigourney Weaver|S…
1919  Leonardo DiCaprio|Joseph Gordon-Levitt|Ellen P…
4     Vin Diesel|Paul Walker|Jason Statham|Michelle …
5     Leonardo DiCaprio|Tom Hardy|Will Poulter|Domhn…
2409  Edward Norton|Brad Pitt|Meat Loaf|Jared Leto|H…
635   Scott Adsit|Ryan Potter|Daniel Henney|T.J. Mil…
6     Arnold Schwarzenegger|Jason Clarke|Emilia Clar…
2633  Elijah Wood|Ian McKellen|Viggo Mortensen|Liv T…
2875  Christian Bale|Michael Caine|Heath Ledger|Aaro…

                       director  runtime  \
0               Colin Trevorrow      124
1                 George Miller      120
629           Christopher Nolan      169
630                  James Gunn      121
2             Robert Schwentke      119
631        Joe Russo|Anthony Russo      136
1329                George Lucas      121
632   Chad Stahelski|David Leitch      101
3                    J.J. Abrams      136
633            Francis Lawrence      123
634                Peter Jackson      144
1386               James Cameron      162
1919           Christopher Nolan      148
```

```
4                         James Wan      137
5     Alejandro GonzÃ¡lez IÃ±Ã¡rritu      156
2409                  David Fincher      139
635         Don Hall|Chris Williams      102
6                       Alan Taylor      125
2633                  Peter Jackson      178
2875              Christopher Nolan      152

                                              genres  \
0            Action|Adventure|Science Fiction|Thriller
1            Action|Adventure|Science Fiction|Thriller
629                    Adventure|Drama|Science Fiction
630                   Action|Science Fiction|Adventure
2                  Adventure|Science Fiction|Thriller
631                   Action|Adventure|Science Fiction
1329                  Adventure|Action|Science Fiction
632                                    Action|Thriller
3             Action|Adventure|Science Fiction|Fantasy
633                Science Fiction|Adventure|Thriller
634                                  Adventure|Fantasy
1386          Action|Adventure|Fantasy|Science Fiction
1919  Action|Thriller|Science Fiction|Mystery|Adventure
4                              Action|Crime|Thriller
5                   Western|Drama|Adventure|Thriller
2409                                             Drama
635           Adventure|Family|Animation|Action|Comedy
6            Science Fiction|Action|Thriller|Adventure
2633                        Adventure|Fantasy|Action
2875                        Drama|Action|Crime|Thriller

                                production_companies release_date  \
0     Universal Studios|Amblin Entertainment|Legenda…       6/9/15
1     Village Roadshow Pictures|Kennedy Miller Produ…      5/13/15
629   Paramount Pictures|Legendary Pictures|Warner B…      11/5/14
630   Marvel Studios|Moving Picture Company (MPC)|Bu…      7/30/14
2     Summit Entertainment|Mandeville Films|Red Wago…      3/18/15
631                                      Marvel Studios      3/20/14
1329   Lucasfilm|Twentieth Century Fox Film Corporation      3/20/77
632   Thunder Road Pictures|Warner Bros.|87Eleven|De…     10/22/14
3             Lucasfilm|Truenorth Productions|Bad Robot     12/15/15
633                                 Lionsgate|Color Force     11/18/14
634   WingNut Films|New Line Cinema|3Foot7|Metro-Gol…     12/10/14
1386  Ingenious Film Partners|Twentieth Century Fox …     12/10/09
1919            Legendary Pictures|Warner Bros.|Syncopy      7/14/10
4     Universal Pictures|Original Film|Media Rights …       4/1/15
5     Regency Enterprises|Appian Way|CatchPlay|Anony…     12/25/15
2409  Regency Enterprises|Fox 2000 Pictures|Taurus F…     10/14/99
```

```
635   Walt Disney Pictures|Walt Disney Animation Stu…     10/24/14
6               Paramount Pictures|Skydance Productions      6/23/15
2633  WingNut Films|New Line Cinema|The Saul Zaentz …     12/18/01
2875  DC Comics|Legendary Pictures|Warner Bros.|Syncopy      7/16/08

      vote_count  vote_average  release_year    budget_adj   revenue_adj
0           5562           6.5          2015  1.379999e+08  1.392446e+09
1           6185           7.1          2015  1.379999e+08  3.481613e+08
629         6498           8.0          2014  1.519800e+08  5.726906e+08
630         5612           7.9          2014  1.565855e+08  7.122911e+08
2           2480           6.3          2015  1.012000e+08  2.716190e+08
631         3848           7.6          2014  1.565855e+08  6.583651e+08
1329        4428           7.9          1977  3.957559e+07  2.789712e+09
632         2712           7.0          2014  1.842182e+07  7.252661e+07
3           5292           7.5          2015  1.839999e+08  1.902723e+09
633         3590           6.6          2014  1.151364e+08  6.927528e+08
634         3110           7.1          2014  2.302728e+08  8.797523e+08
1386        8458           7.1          2009  2.408869e+08  2.827124e+09
1919        9767           7.9          2010  1.600000e+08  8.255000e+08
4           2947           7.3          2015  1.747999e+08  1.385749e+09
5           3929           7.2          2015  1.241999e+08  4.903142e+08
2409        5923           8.1          1999  8.247033e+07  1.320229e+08
635         4185           7.8          2014  1.519800e+08  6.006485e+08
6           2598           5.8          2015  1.425999e+08  4.053551e+08
2633        6079           7.8          2001  1.145284e+08  1.073080e+09
2875        8432           8.1          2008  1.873655e+08  1.014733e+09
```

Above are the 20 most popular movies in our dataset. Interestingly, Jurassic world is considered the most popular movie with 32.9, followed by Mad Max: Fuy Road with 28.4 and Interstellar with 24.9.

[29]:
```python
# plotting a graph of movies against their popularity rating
x = Top_20_movies['original_title'].head(20)
y = Top_20_movies['popularity'].head(20)
plt.barh(x, y)
plt.title('Movies with highest rating (top 20)')
plt.xlabel('Rating')
plt.ylabel('Movies')
plt.show()
```



The graph above helps us visualize movies against their popularity rating.

1.1.4 Research Question 2: What is the spread of movies according to release?

[23]:
```
# count of movies according to release year
movies_by_year = df_movies.groupby(['release_year'])['original_title'].count().
↪reset_index (name="count")
```

Above, we grouped and counted our movies by the year they were released and then saved it in a variable.

[24]:
```
# we sort the movies in decending order
sorted_by_count_movies = movies_by_year.sort_values(by='count', ascending=False)
sorted_by_count_movies.head(20)
```

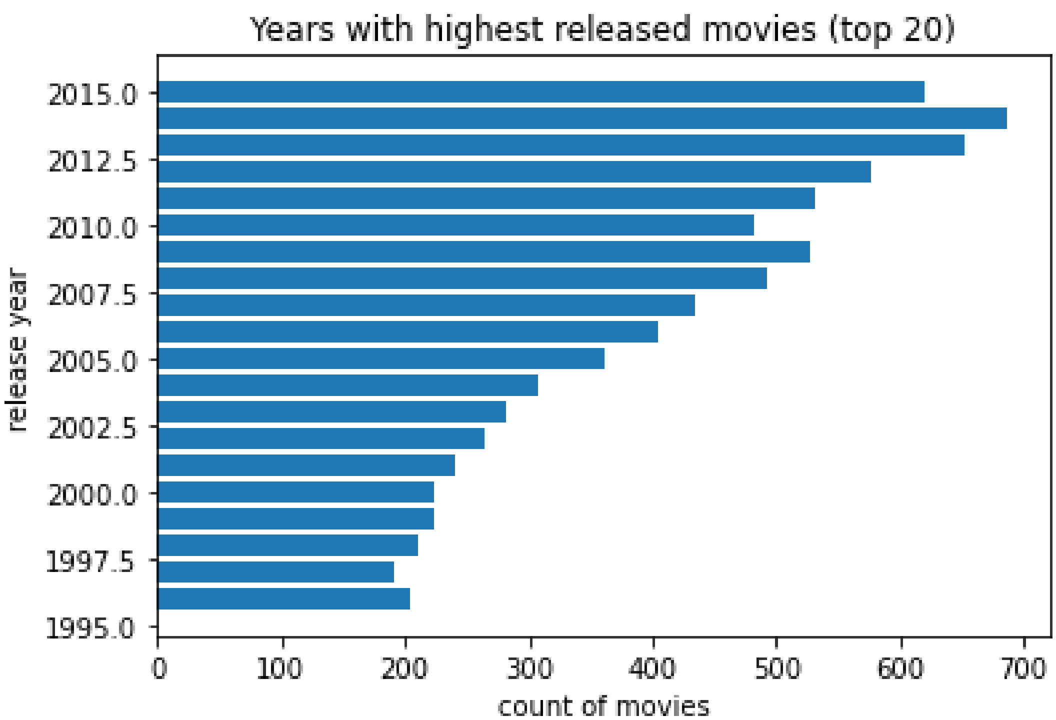
[24]:

	release_year	count
54	2014	687
53	2013	653
55	2015	620
52	2012	576
51	2011	532
49	2009	528
48	2008	492
50	2010	482
47	2007	435
46	2006	404
45	2005	361
44	2004	307
43	2003	281
42	2002	264

41	2001	241
40	2000	224
39	1999	224
38	1998	210
36	1996	203
37	1997	191

Since our goal is to see the top 20 years with heighest producing movies, we sort the count of movies we did earlier above in descending order and the filter the top 20 using the *head(20)* function.

[27]:
```
x = sorted_by_count_movies['release_year'].head(20)
y = sorted_by_count_movies['count'].head(20)
plt.barh(x,y)
plt.title('Years with highest released movies (top 20)')
plt.xlabel('count of movies')
plt.ylabel('release year')
plt.show()
```



Above is a graphical representation of the number of movies produced yearly.

1.1.5 Research Question 3: What are the 20 highest grossing movies in terms of revenue?

[30]:
```
# top five movies by the revenue generated
movies_by_revenue = df_movies.sort_values(by='revenue', ascending=False)
movie_viz = movies_by_revenue.head(20)
movie_viz
```

[30]:
```
          id  popularity     budget     revenue  \
1386   19995    9.432768  237000000  2781505847
3     140607   11.173104  200000000  2068178225
5231     597    4.355219  200000000  1845034188
4361   24428    7.637767  220000000  1519557910
0     135397   32.985763  150000000  1513528810
4     168259    9.335014  190000000  1506249360
14     99861    5.944927  280000000  1405035767
3374   12445    5.711315  125000000  1327817822
5422  109445    6.112766  150000000  1274219009
5425   68721    4.946136  200000000  1215439994
8     211672    7.404165   74000000  1156730962
3522   38356    0.760503  195000000  1123746996
4949     122    7.122455   94000000  1118888979
4365   37724    5.603587  200000000  1108561013
8094    1642    1.136610   22000000  1106279658
4363   49026    6.591277  250000000  1081041287
6555      58    4.205992  200000000  1065659812
1930   10193    2.711136  200000000  1063171911
1921   12155    5.572950  200000000  1025467110
3375    1865    4.955130  380000000  1021683000

                                     original_title  \
1386                                         Avatar
3                      Star Wars: The Force Awakens
5231                                        Titanic
4361                                   The Avengers
0                                    Jurassic World
4                                         Furious 7
14                          Avengers: Age of Ultron
3374   Harry Potter and the Deathly Hallows: Part 2
5422                                         Frozen
5425                                     Iron Man 3
8                                           Minions
3522                Transformers: Dark of the Moon
4949  The Lord of the Rings: The Return of the King
4365                                        Skyfall
8094                                        The Net
4363                          The Dark Knight Rises
```

```
6555    Pirates of the Caribbean: Dead Man's Chest
1930                                   Toy Story 3
1921                           Alice in Wonderland
3375   Pirates of the Caribbean: On Stranger Tides

                                                   cast  \
1386  Sam Worthington|Zoe Saldana|Sigourney Weaver|S…
3     Harrison Ford|Mark Hamill|Carrie Fisher|Adam D…
5231  Kate Winslet|Leonardo DiCaprio|Frances Fisher|…
4361  Robert Downey Jr.|Chris Evans|Mark Ruffalo|Chr…
0     Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi…
4     Vin Diesel|Paul Walker|Jason Statham|Michelle …
14    Robert Downey Jr.|Chris Hemsworth|Mark Ruffalo…
3374  Daniel Radcliffe|Rupert Grint|Emma Watson|Alan…
5422  Kristen Bell|Idina Menzel|Jonathan Groff|Josh …
5425  Robert Downey Jr.|Gwyneth Paltrow|Guy Pearce|D…
8     Sandra Bullock|Jon Hamm|Michael Keaton|Allison…
3522  Shia LaBeouf|John Malkovich|Ken Jeong|Frances …
4949  Elijah Wood|Ian McKellen|Viggo Mortensen|Liv T…
4365  Daniel Craig|Judi Dench|Javier Bardem|Ralph Fi…
8094  Sandra Bullock|Jeremy Northam|Dennis Miller|We…
4363  Christian Bale|Michael Caine|Gary Oldman|Anne …
6555  Johnny Depp|Orlando Bloom|Keira Knightley|Bill…
1930  Tom Hanks|Tim Allen|Ned Beatty|Joan Cusack|Mic…
1921  Mia Wasikowska|Johnny Depp|Anne Hathaway|Helen…
3375  Johnny Depp|PenÃ©lope Cruz|Geoffrey Rush|Ian M…

                     director  runtime  \
1386            James Cameron      162
3                 J.J. Abrams      136
5231            James Cameron      194
4361              Joss Whedon      143
0             Colin Trevorrow      124
4                   James Wan      137
14                Joss Whedon      141
3374              David Yates      130
5422   Chris Buck|Jennifer Lee      102
5425              Shane Black      130
8     Kyle Balda|Pierre Coffin       91
3522              Michael Bay      154
4949            Peter Jackson      201
4365               Sam Mendes      143
8094            Irwin Winkler      114
4363        Christopher Nolan      165
6555           Gore Verbinski      151
1930              Lee Unkrich      103
1921               Tim Burton      108
```

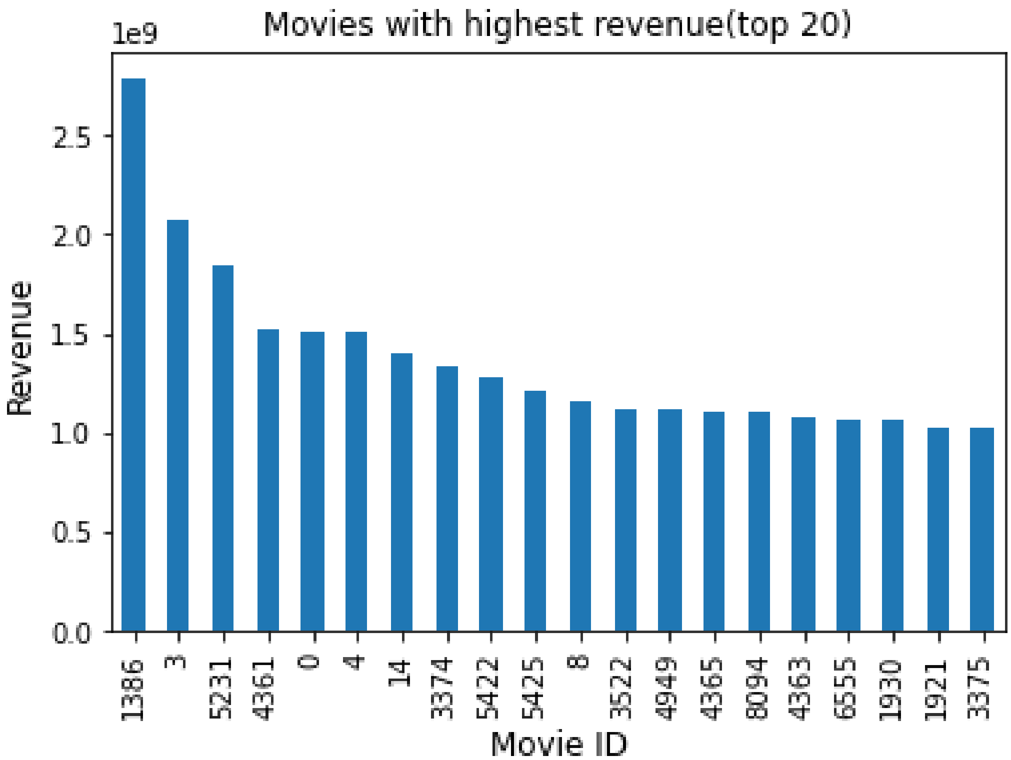
```
3375            Rob Marshall      136

                                       genres  \
1386   Action|Adventure|Fantasy|Science Fiction
3      Action|Adventure|Science Fiction|Fantasy
5231                    Drama|Romance|Thriller
4361          Science Fiction|Action|Adventure
0     Action|Adventure|Science Fiction|Thriller
4                        Action|Crime|Thriller
14            Action|Adventure|Science Fiction
3374                Adventure|Family|Fantasy
5422                  Animation|Adventure|Family
5425          Action|Adventure|Science Fiction
8            Family|Animation|Adventure|Comedy
3522          Action|Science Fiction|Adventure
4949                  Adventure|Fantasy|Action
4365                 Action|Adventure|Thriller
8094        Crime|Drama|Mystery|Thriller|Action
4363                  Action|Crime|Drama|Thriller
6555                  Adventure|Fantasy|Action
1930                   Animation|Family|Comedy
1921                  Family|Fantasy|Adventure
3375                  Adventure|Action|Fantasy

                                 production_companies release_date  \
1386  Ingenious Film Partners|Twentieth Century Fox …     12/10/09
3             Lucasfilm|Truenorth Productions|Bad Robot     12/15/15
5231  Paramount Pictures|Twentieth Century Fox Film …     11/18/97
4361                                     Marvel Studios      4/25/12
0     Universal Studios|Amblin Entertainment|Legenda…       6/9/15
4     Universal Pictures|Original Film|Media Rights …       4/1/15
14    Marvel Studios|Prime Focus|Revolution Sun Studios      4/22/15
3374  Warner Bros.|Heyday Films|Moving Picture Compa…       7/7/11
5422  Walt Disney Pictures|Walt Disney Animation Stu…     11/27/13
5425                                     Marvel Studios      4/18/13
8         Universal Pictures|Illumination Entertainment      6/17/15
3522  Paramount Pictures|Di Bonaventura Pictures|Has…      6/28/11
4949                     WingNut Films|New Line Cinema      12/1/03
4365                                  Columbia Pictures     10/25/12
8094                                  Columbia Pictures      7/28/95
4363  Legendary Pictures|Warner Bros.|DC Entertainme…      7/16/12
6555  Walt Disney Pictures|Jerry Bruckheimer Films|S…      6/20/06
1930       Walt Disney Pictures|Pixar Animation Studios      6/16/10
1921  Walt Disney Pictures|Team Todd|Tim Burton Prod…       3/3/10
3375  Walt Disney Pictures|Jerry Bruckheimer Films|M…      5/11/11

      vote_count  vote_average  release_year    budget_adj   revenue_adj
```

1386	8458	7.1	2009	2.408869e+08	2.827124e+09
3	5292	7.5	2015	1.839999e+08	1.902723e+09
5231	4654	7.3	1997	2.716921e+08	2.506406e+09
4361	8903	7.3	2012	2.089437e+08	1.443191e+09
0	5562	6.5	2015	1.379999e+08	1.392446e+09
4	2947	7.3	2015	1.747999e+08	1.385749e+09
14	4304	7.4	2015	2.575999e+08	1.292632e+09
3374	3750	7.7	2011	1.211748e+08	1.287184e+09
5422	3369	7.5	2013	1.404050e+08	1.192711e+09
5425	6882	6.9	2013	1.872067e+08	1.137692e+09
8	2893	6.5	2015	6.807997e+07	1.064192e+09
3522	2456	6.1	2011	1.890326e+08	1.089358e+09
4949	5636	7.9	2003	1.114231e+08	1.326278e+09
4365	6137	6.8	2012	1.899489e+08	1.052849e+09
8094	201	5.6	1995	3.148127e+07	1.583050e+09
4363	6723	7.5	2012	2.374361e+08	1.026713e+09
6555	3181	6.8	2006	2.163338e+08	1.152691e+09
1930	2924	7.5	2010	2.000000e+08	1.063172e+09
1921	2853	6.3	2010	2.000000e+08	1.025467e+09
3375	3180	6.3	2011	3.683713e+08	9.904175e+08

In the output above, we are able to see the top five movies with the highest producing revenue. What is also interesting to note is that *Jurassic World* which we earlier saw to be the most popular movies did not turn out to be the highest grossing movie.

[44]:
```
# top 20 movies with the highest revenue
movie_viz['revenue'].plot.bar();
plt.title('Movies with highest revenue(top 20)')
plt.xlabel('Movie ID', fontsize=12)
plt.ylabel('Revenue', fontsize=12)
```

[44]:
```
Text(0, 0.5, 'Revenue')
```



We represent the data we extracted on the highest grossing movies in a bar graph to aid easy visualization.

1.1.6 Research Question 4: What is the correlation between budget and revenue?

[81]:

```
# a function to calculate correlation between budget and revenue
def finding_corr(df, col_1, col_2):
    for col in df:
        corr_result = df[[col_1, col_2]].corr()
        return corr_result

finding_corr(df_movies, 'revenue', 'budget')
```

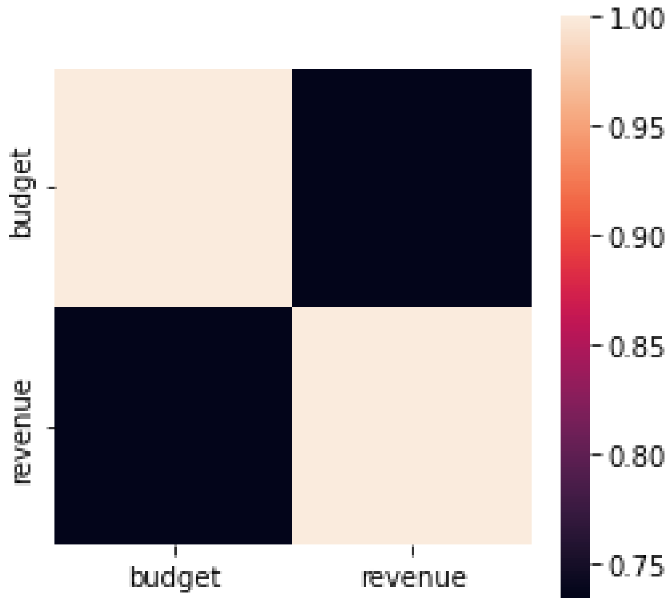
[81]:

	revenue	budget
revenue	1.000000	0.734608
budget	0.734608	1.000000

From the above output of our query, there exist a strongly positive correlation *0.73* between the budget of a movie and the revenue generated by that movie. It is therefore safe to say that the more money spent in the production of a movie, the more likely the movie will generate high revenue.

[86]:
```
# function to plot correlation heatmaps
def corr_heatmap(df, col_1, col_2):
    for col in df:
        plt.figure(figsize=(4,4))
        corr_hmap = sns.heatmap(df[[col_1, col_2]].corr(), square=True)
        return corr_hmap

finding_corr(df_movies, 'budget', 'revenue')
```

[86]:
```
<AxesSubplot:>
```



Above is a heatmap of the correlation between budget and revenue.

1.1.7 Research Question 5: What is the correlation between revenue and popularity?

[82]:
```
# calling finding_corr function correlation between revenue and popularity
finding_corr(df_movies, 'revenue', 'popularity')
```

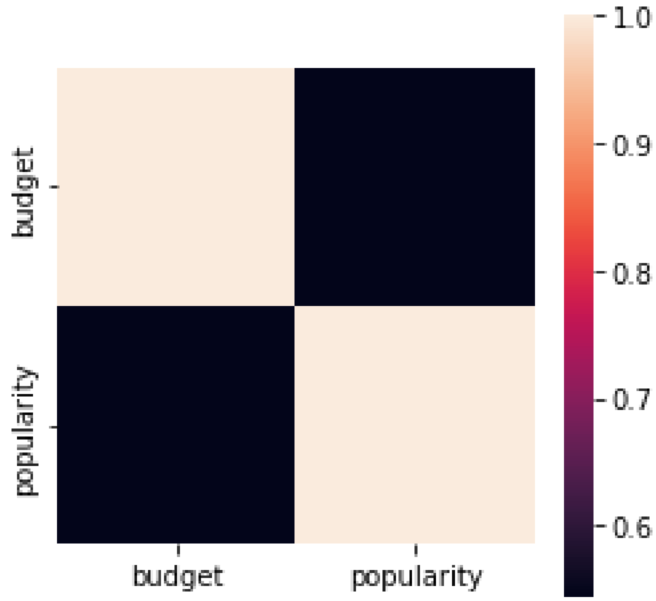
[82]:

	revenue	popularity
revenue	1.000000	0.662994
popularity	0.662994	1.000000

We can see from the code block above that there is a positive correlation between revenue and popularity. This is equally represented in the heatmap below. The dark squares shows the correlation between the popularity of a movie and the revenue generated from that movie.

[87]:
```
# plotting a correlation heatmap of revenue and popularity
finding_corr(df_movies, 'budget', 'popularity')
```

[87]: <AxesSubplot:>



Conclusions

The TMDb movies dataset had 10866 rows and 21 columns. After cleaning which involved removing duplicates, null values and deleting irrelevant columns, we were left with 10767 rows and 16 columns. The dataset was investigated upon some questions which were posed at the begining and the following have been reached: #### Observations

The most popular movies did not make the highest revenues.

The year 2014 saw the production of the highest number of movies at 687.

The movie Avatar generated a revenue of 2,781,505,847 and had a popularity score 9.4 whereas Jurassic world generated 1,513,528,810 and had a popularity score of 32.9.

The top 20 movies that generated the highest revenue are mostly action, science fiction and adventure.

There is a positive correlation of 0.734608 between the budget made for a movie and the revenue generated. This indicates that the amount spent in the production of a movie positively impacts the amount generated by the movie.

The popularity of a movies does not translate to higher revenue from the movies.

Limitations Some limitations to this investigation include:

Some of the columns had incomplete records which resulted in deleting those columns. Indeed, had the records been complete, it would have added depth to the investigation.

Time has also been another factor, dedicating more time to the research will result in more findings.

[ ]:
```

```