# FinalProject

June 12, 2019

## 1 COGS 108 - Final Project

## 2 Overview

**1. Research Question**
Our research question
**2. Background and Prior Work**
The project's significance
**3. Hypothesis**
Our hypothesis
**4. Datasets**
The data sources
**5. Setup**
Imports packages
**6. Data Cleaning**
Datasets are loaded, merged, cleaned for further analysis.
**7. Data Analysis & Results**
Visualization and linear regression analysis about the relationships between different attributes and movie revenue. Prediction of movie revenue using multiple linear regression and non-linear regression. Models are tested by cross-validation.
**8. Ethics & Privacy**
Consideration regarding to privacy and ethics as well as possible biases of our analysis and prediction.
**9. Conclusion & Discussion**

## 3 Names


- Roger Cheng
- Deyin Chen
- Haoyin Xu
- Suzhou Yang
- Sowmya Parthiban

## 4 Group Members IDs

- A14743993
- A13790839
- A92099144
- A15305349
- A92090309

## 5 Research Question

Ever since the early 1900s, movies (or so called "moving pictures") have became a popular leisure activity. Due to the increased financial prosperity, people had more disposable money to allow themselves attending the cinemas far more regularly than the previous decade. As time progresses, styles and theme of movies evolve drastically due to both technological development and cultural evolution. Audience's favorite genres of movies have evolved accordingly as well. Thus, in this project, we are interested in seeing **what components have attributed to a movie's revenue in the recent decade. Examining a movie's components such as the genre, production country, title length, runtime, release time, vote average, budget, etc., we want to see how each component would affect the final revenue**. We believe that revenue could represent the popularity of a movie and thus it helps us to visualize the evolution of audience's interest in the recent decade.

## 6 Background and Prior Work

Films have been recognized to have artistic, educational and commercial values for a society. They are also a way to express and popularize current thoughts, ideas and concerns. More importantly, they indicate the development of technology and dominating ideology of the society. Big film companies in the world usually put billions of dollars into filmmaking as well as advertisement. However, over 70% of the films made negative profit, while the total average revenue of the film industry was around $10 billion each year [1]. Therefore, the high-risk but high-profit property of the films makes the study about factors contributing to a successful film and an accurate prediction method for the film revenue extremely desired for the filmmakers and investigators to make better investment decisions on film production and advertisement.

It would also be a huge benefit if there could be a prediction method of a successful film with a high accuracy. Indeed, there have been many studies about films trying to get a good prediction method. Studies found that multiple factors could be related to a movie's revenue. Although specific weights were not reported for most studies, they found that factors such as cast, budget, film review, actors, directors and genre contributed to the revenue [2, 3]. In particular, a study with modest prediction ability found that horror movies were the most popular movies and the Motion Picture Association of America film rating system had the largest contribution to domestic gross in the US [4].

Among the recent studies, Nithin et al. generated one of the most accurate models to predict the film revenue with around 51 percent accuracy using IMDB data and linear regression. However, they admitted the accuracy was not high enough for industrial use and suggested to use a larger training set [2]. Apte et al. also found that generally low revenue movies had a much lower prediction accuracy compared to high revenue movies due the incompleteness of data from global box office, and some genres might not have enough samples for them to train their model and resulted in a low accuracy [3]. Moreover, the data the groups used to train their models was out of date. Due to the inevitable changes in audience's tastes, using data only from 2000 to 2012 would make the model less accurate to predict film revenue after 2019.

Therefore, in this project, we will combine and organize two datasets that contain information about movies extracted from The Movie Database (TMDb) and MovieLens. These datasets have more than 50,000 entries in total and information up to July 2017. Our goal is to use these up-to-date datasets and a better algorithm to analyze the weights of factors that determine the revenue and to generate a model that will have a higher accuracy in revenue prediction.

#### 6.0.1 References:


- [1] "The Numbers - Movie Market Summary 1995 to 2011." The Numbers - Movie Box Office Data, Film Stars, Idle Speculation. Web. http://www.the-numbers.com/market/.
- [2] NithinV, R., & Babu, S. (2017). Predicting Movie Success Based On Imdb Data.
- [3] Apte, N., Forssell, M., & Sidhwa, A. (2011). Predicting Movie Revenue. CS229, Stanford University.
- [4] Hu, X. (n.d.). Predicting Domestic Gross of Movies. Retrieved from https://www.stat.berkeley.edu/~aldous/Research/Ugrad/ugrad_res_old.html.


## 7 Hypothesis

Our hypothesis is that quantitative predictors such as movie's runtime, budget, popularity score and viewer rating of the quality of the movie on a scale of 10 have a statistically significant effect on predicting the movie's revenue.

## 8 Dataset(s)

#### 8.0.1 Dataset 1

- Dataset Name: The Movies Dataset
- Link to the dataset: https://www.kaggle.com/rounakbanik/the-movies-dataset/downloads/the-movies-dataset.zip/7
- Number of observations: 45467

All movies released before and in July 2017 were collected from the Full MovieLens Dataset. 45467 movies were included in this dataset with each surveyed for its cast, crew, plot keywords, budget, revenue, posters, release dates, languages, production companies, countries, TMDb vote counts and vote averages.

#### 8.0.2 Dataset 2

- Dataset Name: TMDB 5000 Movie Dataset
- Link to the dataset: https://www.kaggle.com/tmdb/tmdb-movie-metadata#tmdb_5000_movies.csv
- Number of observations: 5000

5000 movies released between 1916 and 2017 were randomly extracted from The Movie Database (TMDb). Each movie was surveyed for its keywords, overview, production company, crew, cast, runtime, average rating, number of ratings, and revenue.

# 9 Setup

```
In [60]: # importing necessary packages for data editing
         import numpy as np
         import pandas as pd
         import matplotlib.pyplot as plt

         # important packages for data analysis
         import patsy
         import statsmodels.api as sm
         import scipy.stats as stats
         from scipy.stats import ttest_ind, chisquare, normaltest
```

# 10 Data Cleaning

```
In [61]: # import both dataset into a pandas dataframe
         tmdb_df = pd.read_csv('tmdb_5000_movies.csv')
         tmdb_df2 = pd.read_csv('movies_metadata.csv')

         # visualize one of the data table
         tmdb_df
```

```
/Users/winniexu/anaconda3/lib/python3.6/site-packages/IPython/core/interactiveshell.py:2728: D
  interactivity=interactivity, compiler=compiler, result=result)
```

```
Out[61]:          budget                                             genres  \
         0     237000000  [{"id": 28, "name": "Action"}, {"id": 12, "nam...
         1     300000000  [{"id": 12, "name": "Adventure"}, {"id": 14, "...
         2     245000000  [{"id": 28, "name": "Action"}, {"id": 12, "nam...
         3     250000000  [{"id": 28, "name": "Action"}, {"id": 80, "nam...
         4     260000000  [{"id": 28, "name": "Action"}, {"id": 12, "nam...
         5     258000000  [{"id": 14, "name": "Fantasy"}, {"id": 28, "na...
         6     260000000  [{"id": 16, "name": "Animation"}, {"id": 10751...
         7     280000000  [{"id": 28, "name": "Action"}, {"id": 12, "nam...
         8     250000000  [{"id": 12, "name": "Adventure"}, {"id": 14, "...
         9     250000000  [{"id": 28, "name": "Action"}, {"id": 12, "nam...
         10    270000000  [{"id": 12, "name": "Adventure"}, {"id": 14, "...
         11    200000000  [{"id": 12, "name": "Adventure"}, {"id": 28, "...
         12    200000000  [{"id": 12, "name": "Adventure"}, {"id": 14, "...
         13    255000000  [{"id": 28, "name": "Action"}, {"id": 12, "nam...
         14    225000000  [{"id": 28, "name": "Action"}, {"id": 12, "nam...
         15    225000000  [{"id": 12, "name": "Adventure"}, {"id": 10751...
         16    220000000  [{"id": 878, "name": "Science Fiction"}, {"id"...
         17    380000000  [{"id": 12, "name": "Adventure"}, {"id": 28, "...
         18    225000000  [{"id": 28, "name": "Action"}, {"id": 35, "nam...
         19    250000000  [{"id": 28, "name": "Action"}, {"id": 12, "nam...
         20    215000000  [{"id": 28, "name": "Action"}, {"id": 12, "nam...
```

```
21    200000000  [{"id": 28, "name": "Action"}, {"id": 12, "nam...
22    250000000  [{"id": 12, "name": "Adventure"}, {"id": 14, "...
23    180000000  [{"id": 12, "name": "Adventure"}, {"id": 14, "...
24    207000000  [{"id": 12, "name": "Adventure"}, {"id": 18, "...
25    200000000  [{"id": 18, "name": "Drama"}, {"id": 10749, "n...
26    250000000  [{"id": 12, "name": "Adventure"}, {"id": 28, "...
27    209000000  [{"id": 53, "name": "Thriller"}, {"id": 28, "n...
28    150000000  [{"id": 28, "name": "Action"}, {"id": 12, "nam...
29    200000000  [{"id": 28, "name": "Action"}, {"id": 12, "nam...
...         ...                                                ...
4773      27000                     [{"id": 35, "name": "Comedy"}]
4774      27000  [{"id": 18, "name": "Drama"}, {"id": 10749, "n...
4775          0  [{"id": 18, "name": "Drama"}, {"id": 35, "name...
4776          0  [{"id": 35, "name": "Comedy"}, {"id": 18, "nam...
4777          0                      [{"id": 18, "name": "Drama"}]
4778          0  [{"id": 28, "name": "Action"}, {"id": 18, "nam...
4779          0                     [{"id": 35, "name": "Comedy"}]
4780          0  [{"id": 53, "name": "Thriller"}, {"id": 80, "n...
4781      22000  [{"id": 35, "name": "Comedy"}, {"id": 10749, "...
4782          0  [{"id": 18, "name": "Drama"}, {"id": 10751, "n...
4783          0  [{"id": 53, "name": "Thriller"}, {"id": 27, "n...
4784          0  [{"id": 18, "name": "Drama"}, {"id": 35, "name...
4785          0                      [{"id": 18, "name": "Drama"}]
4786          0  [{"id": 35, "name": "Comedy"}, {"id": 10749, "...
4787          0  [{"id": 878, "name": "Science Fiction"}, {"id"...
4788      12000  [{"id": 27, "name": "Horror"}, {"id": 35, "nam...
4789          0                      [{"id": 18, "name": "Drama"}]
4790          0  [{"id": 18, "name": "Drama"}, {"id": 10769, "n...
4791         13                     [{"id": 27, "name": "Horror"}]
4792      20000  [{"id": 80, "name": "Crime"}, {"id": 27, "name...
4793          0                      [{"id": 18, "name": "Drama"}]
4794          0  [{"id": 53, "name": "Thriller"}, {"id": 27, "n...
4795          0                      [{"id": 18, "name": "Drama"}]
4796       7000  [{"id": 878, "name": "Science Fiction"}, {"id"...
4797          0  [{"id": 10769, "name": "Foreign"}, {"id": 53, ...
4798     220000  [{"id": 28, "name": "Action"}, {"id": 80, "nam...
4799       9000  [{"id": 35, "name": "Comedy"}, {"id": 10749, "...
4800          0  [{"id": 35, "name": "Comedy"}, {"id": 18, "nam...
4801          0                                                 []
4802          0                [{"id": 99, "name": "Documentary"}]

                                             homepage      id  \
0                         http://www.avatarmovie.com/   19995
1        http://disney.go.com/disneypictures/pirates/     285
2         http://www.sonypictures.com/movies/spectre/  206647
3                  http://www.thedarkknightrises.com/   49026
4                http://movies.disney.com/john-carter   49529
5      http://www.sonypictures.com/movies/spider-man3/     559
```

| | | |
|---|---|---|
| 6 | http://disney.go.com/disneypictures/tangled/ | 38757 |
| 7 | http://marvel.com/movies/movie/193/avengers_ag... | 99861 |
| 8 | http://harrypotter.warnerbros.com/harrypottera... | 767 |
| 9 | http://www.batmanvsupermandawnofjustice.com/ | 209112 |
| 10 | http://www.superman.com | 1452 |
| 11 | http://www.mgm.com/view/movie/234/Quantum-of-S... | 10764 |
| 12 | http://disney.go.com/disneypictures/pirates/ | 58 |
| 13 | http://disney.go.com/the-lone-ranger/ | 57201 |
| 14 | http://www.manofsteel.com/ | 49521 |
| 15 | NaN | 2454 |
| 16 | http://marvel.com/avengers_movie/ | 24428 |
| 17 | http://disney.go.com/pirates/index-on-stranger... | 1865 |
| 18 | http://www.sonypictures.com/movies/meninblack3/ | 41154 |
| 19 | http://www.thehobbit.com/ | 122917 |
| 20 | http://www.theamazingspiderman.com | 1930 |
| 21 | http://www.robinhoodthemovie.com/ | 20662 |
| 22 | http://www.thehobbit.com/ | 57158 |
| 23 | http://www.goldencompassmovie.com/index_german... | 2268 |
| 24 | NaN | 254 |
| 25 | http://www.titanicmovie.com | 597 |
| 26 | http://marvel.com/captainamericapremiere | 271110 |
| 27 | NaN | 44833 |
| 28 | http://www.jurassicworld.com/ | 135397 |
| 29 | http://www.skyfall-movie.com | 37724 |
| ... | ... | ... |
| 4773 | http://www.miramax.com/movie/clerks/ | 2292 |
| 4774 | NaN | 42497 |
| 4775 | NaN | 33693 |
| 4776 | NaN | 14585 |
| 4777 | NaN | 185465 |
| 4778 | NaN | 38780 |
| 4779 | NaN | 14022 |
| 4780 | NaN | 366967 |
| 4781 | https://www.facebook.com/DrySpellMovie | 255266 |
| 4782 | NaN | 17345 |
| 4783 | NaN | 226458 |
| 4784 | http://www.thepuffychairmovie.com | 24055 |
| 4785 | NaN | 287625 |
| 4786 | NaN | 44990 |
| 4787 | NaN | 86304 |
| 4788 | NaN | 692 |
| 4789 | NaN | 39851 |
| 4790 | NaN | 13898 |
| 4791 | http://tincanmanthemovie.com/ | 157185 |
| 4792 | NaN | 36095 |
| 4793 | NaN | 182291 |
| 4794 | NaN | 286939 |
| 4795 | NaN | 124606 |

```
4796                   http://www.primermovie.com   14337
4797                                          NaN   67238
4798                                          NaN    9367
4799                                          NaN   72766
4800  http://www.hallmarkchannel.com/signedsealeddel...  231617
4801                   http://shanghaicalling.com/  126186
4802                                          NaN   25975

                                               keywords original_language  \
0     [{"id": 1463, "name": "culture clash"}, {"id":...                en
1     [{"id": 270, "name": "ocean"}, {"id": 726, "na...                en
2     [{"id": 470, "name": "spy"}, {"id": 818, "name...                en
3     [{"id": 849, "name": "dc comics"}, {"id": 853,...                en
4     [{"id": 818, "name": "based on novel"}, {"id":...                en
5     [{"id": 851, "name": "dual identity"}, {"id": ...                en
6     [{"id": 1562, "name": "hostage"}, {"id": 2343,...                en
7     [{"id": 8828, "name": "marvel comic"}, {"id": ...                en
8     [{"id": 616, "name": "witch"}, {"id": 2343, "n...                en
9     [{"id": 849, "name": "dc comics"}, {"id": 7002...                en
10    [{"id": 83, "name": "saving the world"}, {"id"...                en
11    [{"id": 627, "name": "killing"}, {"id": 1568, ...                en
12    [{"id": 616, "name": "witch"}, {"id": 663, "na...                en
13    [{"id": 1556, "name": "texas"}, {"id": 2673, "...                en
14    [{"id": 83, "name": "saving the world"}, {"id"...                en
15    [{"id": 818, "name": "based on novel"}, {"id":...                en
16    [{"id": 242, "name": "new york"}, {"id": 5539,...                en
17    [{"id": 658, "name": "sea"}, {"id": 1316, "nam...                en
18    [{"id": 4379, "name": "time travel"}, {"id": 5...                en
19    [{"id": 417, "name": "corruption"}, {"id": 603...                en
20    [{"id": 1872, "name": "loss of father"}, {"id"...                en
21    [{"id": 4147, "name": "robin hood"}, {"id": 43...                en
22    [{"id": 603, "name": "elves"}, {"id": 604, "na...                en
23    [{"id": 392, "name": "england"}, {"id": 1461, ...                en
24    [{"id": 774, "name": "film business"}, {"id": ...                en
25    [{"id": 2580, "name": "shipwreck"}, {"id": 298...                en
26    [{"id": 393, "name": "civil war"}, {"id": 6091...                en
27    [{"id": 1721, "name": "fight"}, {"id": 4410, "...                en
28    [{"id": 1299, "name": "monster"}, {"id": 1718,...                en
29    [{"id": 470, "name": "spy"}, {"id": 4289, "nam...                en
...                                                 ...               ...
4773  [{"id": 1361, "name": "salesclerk"}, {"id": 30...                en
4774  [{"id": 1566, "name": "dream"}, {"id": 13059, ...                en
4775             [{"id": 171993, "name": "mumblecore"}]                en
4776  [{"id": 1438, "name": "office"}, {"id": 9673, ...                en
4777                                                 []                en
4778  [{"id": 10022, "name": "rampage"}, {"id": 1454...                en
4779  [{"id": 305, "name": "moon"}, {"id": 490, "nam...                en
4780                                                 []                en
```

```
4781  [{"id": 13043, "name": "dating"}, {"id": 15160...              en
4782  [{"id": 186, "name": "christianity"}, {"id": 4...              en
4783               [{"id": 9712, "name": "possession"}]              en
4784             [{"id": 171993, "name": "mumblecore"}]              en
4785                                                 []              en
4786        [{"id": 10183, "name": "independent film"}]              en
4787                [{"id": 9715, "name": "superhero"}]              en
4788  [{"id": 237, "name": "gay"}, {"id": 900, "name...              en
4789  [{"id": 6782, "name": "addiction"}, {"id": 155...              en
4790                                                 []              fa
4791          [{"id": 14903, "name": "home invasion"}]              en
4792  [{"id": 233, "name": "japan"}, {"id": 549, "na...              ja
4793  [{"id": 718, "name": "confession"}, {"id": 100...              en
4794                                                 []              en
4795  [{"id": 10726, "name": "gang"}, {"id": 33928, ...              en
4796  [{"id": 1448, "name": "distrust"}, {"id": 2101...              en
4797                                                 []              en
4798  [{"id": 5616, "name": "united states\u2013mexi...              es
4799                                                 []              en
4800  [{"id": 248, "name": "date"}, {"id": 699, "nam...              en
4801                                                 []              en
4802  [{"id": 1523, "name": "obsession"}, {"id": 224...              en

                                    original_title  \
0                                           Avatar
1         Pirates of the Caribbean: At World's End
2                                          Spectre
3                            The Dark Knight Rises
4                                      John Carter
5                                     Spider-Man 3
6                                          Tangled
7                          Avengers: Age of Ultron
8           Harry Potter and the Half-Blood Prince
9               Batman v Superman: Dawn of Justice
10                                Superman Returns
11                               Quantum of Solace
12     Pirates of the Caribbean: Dead Man's Chest
13                                 The Lone Ranger
14                                    Man of Steel
15        The Chronicles of Narnia: Prince Caspian
16                                    The Avengers
17     Pirates of the Caribbean: On Stranger Tides
18                                  Men in Black 3
19       The Hobbit: The Battle of the Five Armies
20                          The Amazing Spider-Man
21                                      Robin Hood
22             The Hobbit: The Desolation of Smaug
23                              The Golden Compass
```

```
24                                   King Kong
25                                     Titanic
26                  Captain America: Civil War
27                                  Battleship
28                              Jurassic World
29                                     Skyfall
...                                        ...
4773                                    Clerks
4774                            Pink Narcissus
4775                              Funny Ha Ha
4776                     In the Company of Men
4777                                    Manito
4778                                   Rampage
4779                                   Slacker
4780                               Dutch Kills
4781                                 Dry Spell
4782                                  Flywheel
4783                                  Backmask
4784                            The Puffy Chair
4785                      Stories of Our Lives
4786                          Breaking Upwards
4787                  All Superheroes Must Die
4788                            Pink Flamingos
4789                                     Clean
4790
4791                               Tin Can Man
4792
4793                            On The Downlow
4794              Sanctuary: Quite a Conundrum
4795                                      Bang
4796                                    Primer
4797                                    Cavite
4798                               El Mariachi
4799                                 Newlyweds
4800                  Signed, Sealed, Delivered
4801                           Shanghai Calling
4802                          My Date with Drew

                                               overview  popularity  \
0     In the 22nd century, a paraplegic Marine is di...  150.437577
1     Captain Barbossa, long believed to be dead, ha...  139.082615
2     A cryptic message from Bonds past sends him o...  107.376788
3     Following the death of District Attorney Harve...  112.312950
4     John Carter is a war-weary, former military ca...   43.926995
5     The seemingly invincible Spider-Man goes up ag...  115.699814
6     When the kingdom's most wanted-and most charmi...   48.681969
7     When Tony Stark tries to jumpstart a dormant p...  134.279229
8     As Harry begins his sixth year at Hogwarts, he...   98.885637
```

```
9       Fearing the actions of a god-like Super Hero l...  155.790452
10      Superman returns to discover his 5-year absenc...   57.925623
11      Quantum of Solace continues the adventures of ...  107.928811
12      Captain Jack Sparrow works his way out of a bl...  145.847379
13      The Texas Rangers chase down a gang of outlaws...   49.046956
14      A young boy learns that he has extraordinary p...   99.398009
15      One year after their incredible adventures in ...   53.978602
16      When an unexpected enemy emerges and threatens...  144.448633
17      Captain Jack Sparrow crosses paths with a woma...  135.413856
18      Agents J (Will Smith) and K (Tommy Lee Jones) ...   52.035179
19      Immediately after the events of The Desolation...  120.965743
20      Peter Parker is an outcast high schooler aband...   89.866276
21      When soldier Robin happens upon the dying Robe...   37.668301
22      The Dwarves, Bilbo and Gandalf have successful...   94.370564
23      After overhearing a shocking secret, precociou...   42.990906
24      In 1933 New York, an overly ambitious movie pr...   61.226010
25      84 years later, a 101-year-old woman named Ros...  100.025899
26      Following the events of Age of Ultron, the col...  198.372395
27      When mankind beams a radio signal into space, ...   64.928382
28      Twenty-two years after the events of Jurassic ...  418.708552
29      When Bond's latest assignment goes gravely wro...   93.004993
...                                                   ...         ...
4773    Convenience and video store clerks Dante and R...   19.748658
4774    An erotic poem set in the fantasies of a young...    0.027811
4775    Unsure of what to do next, 23-year-old Marnie ...    0.362633
4776    Two business executives--one an avowed misogyn...    2.634007
4777    Fifteen years ago, their Washington Heights ne...    0.039264
4778    The boredom of small town life is eating Bill ...    7.101197
4779    Presents a day in the life in Austin, Texas am...    3.320622
4780    A desperate ex-con is forced to gather his old...    0.038143
4781    Sasha tries to get her soon-to-be ex husband K...    0.048948
4782    Jay Austin wants to sell you a used car, but w...    1.048524
4783    During an all-night, drug-fueled party at an a...    3.619167
4784    Josh's life is pretty much in the toilet. He's...    1.243955
4785    Created by the members of a Nairobi-based arts...    0.327794
4786    'Breaking Upwards' explores a young, real-life...    0.674570
4787    Masked vigilantes Charge (Jason Trost), Cutthr...    3.545991
4788    Notorious Baltimore criminal and underground f...    4.553644
4789    After losing her husband to a heroin overdose,...    1.464566
4790    Various women struggle to function in the oppr...    1.193779
4791    Recently dumped by his girlfirend for another ...    0.332679
4792    A wave of gruesome murders is sweeping Tokyo. ...    0.212443
4793    Isaac and Angel are two young Latinos involved...    0.029757
4794    It should have been just a normal day of sex, ...    0.166513
4795    A young woman in L.A. is having a bad day: she...    0.918116
4796    Friends/fledgling entrepreneurs invent a devic...   23.307949
4797    Adam, a security guard, travels from Californi...    0.022173
4798    El Mariachi just wants to play his guitar and ...   14.269792
```

```
4799  A newlywed couple's honeymoon is upended by th...    0.642552
4800  "Signed, Sealed, Delivered" introduces a dedic...    1.444476
4801  When ambitious New York attorney Sam is sent t...    0.857008
4802  Ever since the second grade when he first saw ...    1.929883

                                    production_companies  \
0      [{"name": "Ingenious Film Partners", "id": 289...
1      [{"name": "Walt Disney Pictures", "id": 2}, {"...
2      [{"name": "Columbia Pictures", "id": 5}, {"nam...
3      [{"name": "Legendary Pictures", "id": 923}, {"...
4           [{"name": "Walt Disney Pictures", "id": 2}]
5      [{"name": "Columbia Pictures", "id": 5}, {"nam...
6      [{"name": "Walt Disney Pictures", "id": 2}, {"...
7      [{"name": "Marvel Studios", "id": 420}, {"name...
8      [{"name": "Warner Bros.", "id": 6194}, {"name"...
9      [{"name": "DC Comics", "id": 429}, {"name": "A...
10     [{"name": "DC Comics", "id": 429}, {"name": "L...
11             [{"name": "Eon Productions", "id": 7576}]
12     [{"name": "Walt Disney Pictures", "id": 2}, {"...
13     [{"name": "Walt Disney Pictures", "id": 2}, {"...
14     [{"name": "Legendary Pictures", "id": 923}, {"...
15     [{"name": "Walt Disney", "id": 5888}, {"name":...
16     [{"name": "Paramount Pictures", "id": 4}, {"na...
17     [{"name": "Walt Disney Pictures", "id": 2}, {"...
18     [{"name": "Amblin Entertainment", "id": 56}, {...
19     [{"name": "WingNut Films", "id": 11}, {"name":...
20     [{"name": "Columbia Pictures", "id": 5}, {"nam...
21     [{"name": "Imagine Entertainment", "id": 23}, ...
22     [{"name": "WingNut Films", "id": 11}, {"name":...
23     [{"name": "New Line Cinema", "id": 12}, {"name...
24     [{"name": "WingNut Films", "id": 11}, {"name":...
25     [{"name": "Paramount Pictures", "id": 4}, {"na...
26     [{"name": "Studio Babelsberg", "id": 264}, {"n...
27     [{"name": "Universal Pictures", "id": 33}, {"n...
28     [{"name": "Universal Studios", "id": 13}, {"na...
29              [{"name": "Columbia Pictures", "id": 5}]
...                                                  ...
4773   [{"name": "Miramax Films", "id": 14}, {"name":...
4774          [{"name": "Strand Releasing", "id": 3923}]
4775                                                  []
4776   [{"name": "Alliance Atlantis Communications", ...
4777                                                  []
4778   [{"name": "Boll Kino Beteiligungs GmbH & Co. K...
4779                                                  []
4780                                                  []
4781                                                  []
4782                                                  []
4783   [{"name": "GO Productions", "id": 2943}, {"nam...
```

```
4784                                                    []
4785                                                    []
4786                                                    []
4787                 [{"name": "Grindfest", "id": 18818}]
4788     [{"name": "Dreamland Productions", "id": 407}]
4789                                                    []
4790  [{"name": "Jafar Panahi Film Productions", "id...
4791  [{"name": "Park Films", "id": 21871}, {"name":...
4792               [{"name": "Daiei Studios", "id": 881}]
4793         [{"name": "Iconoclast Films", "id": 26677}]
4794  [{"name": "Gold Lion Films", "id": 37870}, {"n...
4795  [{"name": "Asylum Films", "id": 10571}, {"name...
4796                   [{"name": "Thinkfilm", "id": 446}]
4797                                                    []
4798            [{"name": "Columbia Pictures", "id": 5}]
4799                                                    []
4800  [{"name": "Front Street Pictures", "id": 3958}...
4801                                                    []
4802  [{"name": "rusty bear entertainment", "id": 87...

                                   production_countries release_date  \
0     [{"iso_3166_1": "US", "name": "United States o...   2009-12-10
1     [{"iso_3166_1": "US", "name": "United States o...   2007-05-19
2     [{"iso_3166_1": "GB", "name": "United Kingdom"...   2015-10-26
3     [{"iso_3166_1": "US", "name": "United States o...   2012-07-16
4     [{"iso_3166_1": "US", "name": "United States o...   2012-03-07
5     [{"iso_3166_1": "US", "name": "United States o...   2007-05-01
6     [{"iso_3166_1": "US", "name": "United States o...   2010-11-24
7     [{"iso_3166_1": "US", "name": "United States o...   2015-04-22
8     [{"iso_3166_1": "GB", "name": "United Kingdom"...   2009-07-07
9     [{"iso_3166_1": "US", "name": "United States o...   2016-03-23
10    [{"iso_3166_1": "US", "name": "United States o...   2006-06-28
11    [{"iso_3166_1": "GB", "name": "United Kingdom"...   2008-10-30
12    [{"iso_3166_1": "JM", "name": "Jamaica"}, {"is...   2006-06-20
13    [{"iso_3166_1": "US", "name": "United States o...   2013-07-03
14    [{"iso_3166_1": "GB", "name": "United Kingdom"...   2013-06-12
15    [{"iso_3166_1": "CZ", "name": "Czech Republic"...   2008-05-15
16    [{"iso_3166_1": "US", "name": "United States o...   2012-04-25
17    [{"iso_3166_1": "US", "name": "United States o...   2011-05-14
18    [{"iso_3166_1": "US", "name": "United States o...   2012-05-23
19    [{"iso_3166_1": "NZ", "name": "New Zealand"}, ...   2014-12-10
20    [{"iso_3166_1": "US", "name": "United States o...   2012-06-27
21    [{"iso_3166_1": "GB", "name": "United Kingdom"...   2010-05-12
22    [{"iso_3166_1": "NZ", "name": "New Zealand"}, ...   2013-12-11
23    [{"iso_3166_1": "GB", "name": "United Kingdom"...   2007-12-04
24    [{"iso_3166_1": "NZ", "name": "New Zealand"}, ...   2005-12-14
25    [{"iso_3166_1": "US", "name": "United States o...   1997-11-18
26    [{"iso_3166_1": "US", "name": "United States o...   2016-04-27
```

```
27    [{"iso_3166_1": "US", "name": "United States o...   2012-04-11
28    [{"iso_3166_1": "US", "name": "United States o...   2015-06-09
29    [{"iso_3166_1": "GB", "name": "United Kingdom"...   2012-10-25
...                                                 ...          ...
4773  [{"iso_3166_1": "US", "name": "United States o...   1994-09-13
4774  [{"iso_3166_1": "US", "name": "United States o...   1971-01-01
4775  [{"iso_3166_1": "US", "name": "United States o...   2002-09-20
4776  [{"iso_3166_1": "CA", "name": "Canada"}, {"iso...   1997-01-19
4777  [{"iso_3166_1": "US", "name": "United States o...   2002-01-15
4778  [{"iso_3166_1": "CA", "name": "Canada"}, {"iso...   2009-08-14
4779  [{"iso_3166_1": "US", "name": "United States o...   1990-07-27
4780                                                 []   2015-10-02
4781  [{"iso_3166_1": "US", "name": "United States o...   2013-02-14
4782  [{"iso_3166_1": "US", "name": "United States o...   2003-01-01
4783  [{"iso_3166_1": "US", "name": "United States o...   2015-01-16
4784                                                 []   2005-01-17
4785            [{"iso_3166_1": "KE", "name": "Kenya"}]   2014-09-05
4786  [{"iso_3166_1": "US", "name": "United States o...   2009-03-14
4787                                                 []   2011-10-26
4788  [{"iso_3166_1": "US", "name": "United States o...   1972-03-12
4789  [{"iso_3166_1": "GB", "name": "United Kingdom"...   2004-09-01
4790             [{"iso_3166_1": "IR", "name": "Iran"}]   2000-09-08
4791          [{"iso_3166_1": "IE", "name": "Ireland"}]   2007-01-01
4792            [{"iso_3166_1": "JP", "name": "Japan"}]   1997-11-06
4793  [{"iso_3166_1": "US", "name": "United States o...   2004-04-11
4794  [{"iso_3166_1": "US", "name": "United States o...   2012-01-20
4795  [{"iso_3166_1": "US", "name": "United States o...   1995-09-09
4796  [{"iso_3166_1": "US", "name": "United States o...   2004-10-08
4797                                                 []   2005-03-12
4798  [{"iso_3166_1": "MX", "name": "Mexico"}, {"iso...   1992-09-04
4799                                                 []   2011-12-26
4800  [{"iso_3166_1": "US", "name": "United States o...   2013-10-13
4801  [{"iso_3166_1": "US", "name": "United States o...   2012-05-03
4802  [{"iso_3166_1": "US", "name": "United States o...   2005-08-05

         revenue  runtime                                    spoken_languages  \
0     2787965087    162.0  [{"iso_639_1": "en", "name": "English"}, {"iso...
1      961000000    169.0           [{"iso_639_1": "en", "name": "English"}]
2      880674609    148.0  [{"iso_639_1": "fr", "name": "Fran\u00e7ais"},...
3     1084939099    165.0           [{"iso_639_1": "en", "name": "English"}]
4      284139100    132.0           [{"iso_639_1": "en", "name": "English"}]
5      890871626    139.0  [{"iso_639_1": "en", "name": "English"}, {"iso...
6      591794936    100.0           [{"iso_639_1": "en", "name": "English"}]
7     1405403694    141.0           [{"iso_639_1": "en", "name": "English"}]
8      933959197    153.0           [{"iso_639_1": "en", "name": "English"}]
9      873260194    151.0           [{"iso_639_1": "en", "name": "English"}]
10     391081192    154.0  [{"iso_639_1": "en", "name": "English"}, {"iso...
11     586090727    106.0  [{"iso_639_1": "en", "name": "English"}, {"iso...
```

| | | | |
|---|---|---|---|
| 12 | 1065659812 | 151.0 | [{"iso_639_1": "en", "name": "English"}, {"iso... |
| 13 | 89289910 | 149.0 | [{"iso_639_1": "en", "name": "English"}] |
| 14 | 662845518 | 143.0 | [{"iso_639_1": "en", "name": "English"}] |
| 15 | 419651413 | 150.0 | [{"iso_639_1": "en", "name": "English"}] |
| 16 | 1519557910 | 143.0 | [{"iso_639_1": "en", "name": "English"}] |
| 17 | 1045713802 | 136.0 | [{"iso_639_1": "en", "name": "English"}, {"iso... |
| 18 | 624026776 | 106.0 | [{"iso_639_1": "en", "name": "English"}] |
| 19 | 956019788 | 144.0 | [{"iso_639_1": "en", "name": "English"}] |
| 20 | 752215857 | 136.0 | [{"iso_639_1": "en", "name": "English"}] |
| 21 | 310669540 | 140.0 | [{"iso_639_1": "en", "name": "English"}, {"iso... |
| 22 | 958400000 | 161.0 | [{"iso_639_1": "en", "name": "English"}] |
| 23 | 372234864 | 113.0 | [{"iso_639_1": "is", "name": "\u00cdslenska"},... |
| 24 | 550000000 | 187.0 | [{"iso_639_1": "en", "name": "English"}] |
| 25 | 1845034188 | 194.0 | [{"iso_639_1": "en", "name": "English"}, {"iso... |
| 26 | 1153304495 | 147.0 | [{"iso_639_1": "ro", "name": "Rom\u00e2n\u0103... |
| 27 | 303025485 | 131.0 | [{"iso_639_1": "en", "name": "English"}, {"iso... |
| 28 | 1513528810 | 124.0 | [{"iso_639_1": "en", "name": "English"}] |
| 29 | 1108561013 | 143.0 | [{"iso_639_1": "en", "name": "English"}] |
| ... | ... | ... | ... |
| 4773 | 3151130 | 92.0 | [{"iso_639_1": "en", "name": "English"}] |
| 4774 | 0 | 64.0 | [{"iso_639_1": "en", "name": "English"}, {"iso... |
| 4775 | 76901 | 85.0 | [{"iso_639_1": "en", "name": "English"}] |
| 4776 | 0 | 97.0 | [{"iso_639_1": "en", "name": "English"}] |
| 4777 | 0 | 78.0 | [{"iso_639_1": "es", "name": "Espa\u00f1ol"}, ... |
| 4778 | 0 | 85.0 | [{"iso_639_1": "en", "name": "English"}] |
| 4779 | 0 | 97.0 | [{"iso_639_1": "en", "name": "English"}] |
| 4780 | 0 | 90.0 | [] |
| 4781 | 0 | 90.0 | [{"iso_639_1": "en", "name": "English"}] |
| 4782 | 0 | 120.0 | [{"iso_639_1": "en", "name": "English"}] |
| 4783 | 0 | 91.0 | [{"iso_639_1": "en", "name": "English"}] |
| 4784 | 0 | 85.0 | [] |
| 4785 | 0 | 60.0 | [{"iso_639_1": "sw", "name": "Kiswahili"}, {"i... |
| 4786 | 0 | 88.0 | [{"iso_639_1": "en", "name": "English"}] |
| 4787 | 0 | 78.0 | [{"iso_639_1": "en", "name": "English"}] |
| 4788 | 6000000 | 93.0 | [{"iso_639_1": "en", "name": "English"}] |
| 4789 | 0 | 111.0 | [{"iso_639_1": "cn", "name": "\u5e7f\u5dde\u8b... |
| 4790 | 0 | 90.0 | [{"iso_639_1": "fa", "name": "\u0641\u0627\u06... |
| 4791 | 0 | 84.0 | [{"iso_639_1": "en", "name": "English"}] |
| 4792 | 99000 | 111.0 | [{"iso_639_1": "ja", "name": "\u65e5\u672c\u8a... |
| 4793 | 0 | 90.0 | [] |
| 4794 | 0 | 82.0 | [{"iso_639_1": "en", "name": "English"}] |
| 4795 | 0 | 98.0 | [{"iso_639_1": "en", "name": "English"}] |
| 4796 | 424760 | 77.0 | [{"iso_639_1": "en", "name": "English"}] |
| 4797 | 0 | 80.0 | [] |
| 4798 | 2040920 | 81.0 | [{"iso_639_1": "es", "name": "Espa\u00f1ol"}] |
| 4799 | 0 | 85.0 | [] |
| 4800 | 0 | 120.0 | [{"iso_639_1": "en", "name": "English"}] |
| 4801 | 0 | 98.0 | [{"iso_639_1": "en", "name": "English"}] |

```
4802           0     90.0           [{"iso_639_1": "en", "name": "English"}]

        status                                          tagline  \
0     Released                      Enter the World of Pandora.
1     Released     At the end of the world, the adventure begins.
2     Released                             A Plan No One Escapes
3     Released                                   The Legend Ends
4     Released             Lost in our world, found in another.
5     Released                                The battle within.
6     Released           They're taking adventure to new lengths.
7     Released                                A New Age Has Come.
8     Released                             Dark Secrets Revealed
9     Released                                Justice or revenge
10    Released                                               NaN
11    Released     For love, for hate, for justice, for revenge.
12    Released                                      Jack is back!
13    Released                           Never Take Off the Mask
14    Released             You will believe that a man can fly.
15    Released                               Hope has a new face.
16    Released                           Some assembly required.
17    Released                      Live Forever Or Die Trying.
18    Released                        They are back... in time.
19    Released  Witness the defining chapter of the Middle-Ear...
20    Released                         The untold story begins.
21    Released    Rise and rise again, until lambs become lions.
22    Released  Beyond darkness... beyond desolation... lies t...
23    Released  There are worlds beyond our own - the compass ...
24    Released                  The eighth wonder of the world.
25    Released          Nothing on Earth could come between them.
26    Released                                   Divided We Fall
27    Released               The Battle for Earth Begins at Sea
28    Released                               The park is open.
29    Released                             Think on your sins.
...        ...                                               ...
4773  Released  Just because they serve you doesn't mean they ...
4774  Released            A unique experience in visual fantasy!
4775  Released                                               NaN
4776  Released     Are all men bastards...or just misunderstood?
4777  Released                                               NaN
4778  Released                           Vengeance is ruthless.
4779  Released                                               NaN
4780  Released                                               NaN
4781  Released      Getting divorced does funny things to a girl
4782  Released                                               NaN
4783  Released                                        nederlands
4784  Released                                               NaN
4785  Released                                               NaN
4786  Released                                               NaN
```

```
4787  Released                           May The Best Man Win
4788  Released                     An exercise in poor taste.
4789  Released        When you don't have a choice, you change.
4790  Released                                              NaN
4791  Released                 Everything You've Heard is True
4792  Released                        Madness. Terror. Murder.
4793  Released            Two gangs. One secret. One crossroad.
4794  Released                                              NaN
4795  Released             Sometimes you've got to break the rules
4796  Released                What happens if it actually works?
4797  Released                                              NaN
4798  Released  He didn't come looking for trouble, but troubl...
4799  Released  A newlywed couple's honeymoon is upended by th...
4800  Released                                              NaN
4801  Released                        A New Yorker in Shanghai
4802  Released                                              NaN

                                           title  vote_average  vote_count
0                                         Avatar           7.2       11800
1       Pirates of the Caribbean: At World's End           6.9        4500
2                                        Spectre           6.3        4466
3                          The Dark Knight Rises           7.6        9106
4                                    John Carter           6.1        2124
5                                   Spider-Man 3           5.9        3576
6                                        Tangled           7.4        3330
7                        Avengers: Age of Ultron           7.3        6767
8         Harry Potter and the Half-Blood Prince           7.4        5293
9             Batman v Superman: Dawn of Justice           5.7        7004
10                              Superman Returns           5.4        1400
11                             Quantum of Solace           6.1        2965
12    Pirates of the Caribbean: Dead Man's Chest           7.0        5246
13                               The Lone Ranger           5.9        2311
14                                  Man of Steel           6.5        6359
15      The Chronicles of Narnia: Prince Caspian           6.3        1630
16                                  The Avengers           7.4       11776
17   Pirates of the Caribbean: On Stranger Tides           6.4        4948
18                                Men in Black 3           6.2        4160
19     The Hobbit: The Battle of the Five Armies           7.1        4760
20                        The Amazing Spider-Man           6.5        6586
21                                    Robin Hood           6.2        1398
22           The Hobbit: The Desolation of Smaug           7.6        4524
23                            The Golden Compass           5.8        1303
24                                     King Kong           6.6        2337
25                                       Titanic           7.5        7562
26                    Captain America: Civil War           7.1        7241
27                                    Battleship           5.5        2114
28                                Jurassic World           6.5        8662
29                                       Skyfall           6.9        7604
```

```
...                                    ...       ...        ...
4773                                Clerks       7.4        755
4774                        Pink Narcissus       6.0          9
4775                          Funny Ha Ha        6.3          8
4776                In the Company of Men        6.8         44
4777                                Manito       5.5          2
4778                               Rampage       6.0        131
4779                               Slacker       6.4         77
4780                           Dutch Kills       0.0          0
4781                             Dry Spell       6.0          1
4782                              Flywheel       6.8         19
4783                              Backmask       4.7         79
4784                       The Puffy Chair       6.2         15
4785                  Stories of Our Lives       0.0          0
4786                      Breaking Upwards       5.6         12
4787               All Superheroes Must Die      4.2         13
4788                        Pink Flamingos       6.2        110
4789                                 Clean       6.7         17
4790                            The Circle       6.6         17
4791                           Tin Can Man       2.0          1
4792                                  Cure       7.4         63
4793                        On The Downlow       6.0          2
4794          Sanctuary: Quite a Conundrum       0.0          0
4795                                  Bang       6.0          1
4796                                Primer       6.9        658
4797                                Cavite       7.5          2
4798                            El Mariachi      6.6        238
4799                             Newlyweds       5.9          5
4800             Signed, Sealed, Delivered       7.0          6
4801                       Shanghai Calling      5.7          7
4802                      My Date with Drew      6.3         16

[4803 rows x 20 columns]
```

As shown from the previous section, the format of most of the cells was very messy. For instance, there were many random symbols and random information like "id" under the *"production_companies"* column. The same scenerio occured with *"production_countries"* and many other columns as well. Thus, we needed to clean up the format and extract the needed information from these cells.

Also, much of the information were redundant such as the "spoken language" column. Thus, we needed to drop all the unnecessary columns in order to facilitate our analysis process. During this process, we also dropped all the rows that doesn't contain a value for "revenue" column.

Finally, we merged both data set into one by their title, genres, original language, production countries, release date, runtime, popularity, vote count, vote average, budget and revenue.

```
In [62]: # Method to help extracting the genres information from cells
         def trim_genres(genres):
             g_list = []
```

```
    b = 0
    a = 0
    while (a != -1 and b != -1):
        a = genres.find("name")
        b = genres.find("}")
        g_list.append(genres[a+8:b-1])
        genres = genres [b+1:]
        if (len(genres) == 1):
            break
    g_list.sort()
    return g_list
```

In [63]:
```
# dropping unneeded columns and the rows with missing data
tmdb_df = tmdb_df.drop(columns = ['homepage','id','production_companies','keywords','c
tmdb_df = tmdb_df.dropna()

# Transform the genres and production_countries column to only contains the informatio
tmdb_df['genres'] = tmdb_df['genres'].apply(trim_genres)
tmdb_df['production_countries'] = tmdb_df['production_countries'].apply(trim_genres)
tmdb_df['genres'] = tmdb_df['genres'].apply(tuple)
tmdb_df['production_countries'] = tmdb_df['production_countries'].apply(tuple)
```

In [64]:
```
# same cleanning procedure for the second data set
tmdb_df2 = tmdb_df2.drop(columns = ['adult','belongs_to_collection','homepage', 'id',
tmdb_df2 = tmdb_df2.dropna()
tmdb_df2['genres'] = tmdb_df2['genres'].apply(trim_genres)
tmdb_df2['production_countries'] = tmdb_df2['production_countries'].apply(trim_genres)
tmdb_df2['genres'] = tmdb_df2['genres'].apply(tuple)
tmdb_df2['production_countries'] = tmdb_df2['production_countries'].apply(tuple)
tmdb_df2 = tmdb_df2.fillna(0)

# change the type of these columns
tmdb_df2.budget = tmdb_df2.budget.astype(np.int64)
tmdb_df2.popularity = tmdb_df2.popularity.astype(np.float64)
tmdb_df2.revenue = tmdb_df2.revenue.astype(np.int64)
tmdb_df2.vote_count = tmdb_df2.vote_count.astype(np.int64)
```

In [65]:
```
# merge two data set into one and drop the duplicated ones, save it as a new csv file
merge_df = pd.merge(tmdb_df,tmdb_df2,on = ['budget','genres','original_language','popu
merge_df = merge_df[merge_df['revenue'] != 0]
merge_df = merge_df.drop_duplicates(subset = 'title', keep = 'first')
merge_df=merge_df[['title','genres','original_language','production_countries','releas
merge_df.to_csv('trimmed_data.csv')
# visulize the new dataset
merge_df
```

Out[65]:
```
                                              title  \
0                                            Avatar
1          Pirates of the Caribbean: At World's End
```

```
2                                            Spectre
3                              The Dark Knight Rises
4                                        John Carter
5                                       Spider-Man 3
6                                            Tangled
7                             Avengers: Age of Ultron
8             Harry Potter and the Half-Blood Prince
9                 Batman v Superman: Dawn of Justice
10                                  Superman Returns
11                                 Quantum of Solace
12        Pirates of the Caribbean: Dead Man's Chest
13                                   The Lone Ranger
14                                      Man of Steel
15          The Chronicles of Narnia: Prince Caspian
16                                      The Avengers
17       Pirates of the Caribbean: On Stranger Tides
18                                    Men in Black 3
19         The Hobbit: The Battle of the Five Armies
20                            The Amazing Spider-Man
21                                        Robin Hood
22               The Hobbit: The Desolation of Smaug
23                                The Golden Compass
24                                         King Kong
25                                           Titanic
26                        Captain America: Civil War
27                                        Battleship
28                                    Jurassic World
29                                           Skyfall
...                                              ...
49057                                          Fanaa
49108                                  Atomic Blonde
49153                                        Dunkirk
49171                                       Bairavaa
49186                                        Gymkata
49265                       Confidential Assignment
49285          Yu-Gi-Oh!: The Dark Side of Dimensions
49291                                  Chasing Trane
49315                  Transformers: The Last Knight
49317                               Porn in the Hood
49321                        Mommies, Happy New Year!
49323                                       Pregnant
49329                                    On the Hook!
49360                                           Moka
49376                                      Good Time
49418                              One Hundred Steps
49427                                           2:22
49440                                       FC Venus
49482                                 The Dark Tower
```

```
49528                                       My Old Classmate
49557                    And Here's What's Happening to Me
49603                                         The Emoji Movie
49630                                              Wind River
49708                                                  Baasha
49710                                       Sivaji: The Boss
49836                                            Apartment 18
49854                                             All at Once
49856                                             The Miracle
49866                                            Pro Lyuboff
49876                                                 Antidur

                                                   genres original_language  \
0          (Action, Adventure, Fantasy, Science Fiction)                en
1                           (Action, Adventure, Fantasy)                en
2                             (Action, Adventure, Crime)                en
3                       (Action, Crime, Drama, Thriller)                en
4                   (Action, Adventure, Science Fiction)                en
5                           (Action, Adventure, Fantasy)                en
6                                    (Animation, Family)                en
7                   (Action, Adventure, Science Fiction)                en
8                           (Adventure, Family, Fantasy)                en
9                           (Action, Adventure, Fantasy)                en
10         (Action, Adventure, Fantasy, Science Fiction)                en
11                  (Action, Adventure, Crime, Thriller)                en
12                          (Action, Adventure, Fantasy)                en
13                          (Action, Adventure, Western)                en
14         (Action, Adventure, Fantasy, Science Fiction)                en
15                          (Adventure, Family, Fantasy)                en
16                  (Action, Adventure, Science Fiction)                en
17                          (Action, Adventure, Fantasy)                en
18                     (Action, Comedy, Science Fiction)                en
19                          (Action, Adventure, Fantasy)                en
20                          (Action, Adventure, Fantasy)                en
21                                   (Action, Adventure)                en
22                                  (Adventure, Fantasy)                en
23                                  (Adventure, Fantasy)                en
24                            (Action, Adventure, Drama)                en
25                            (Drama, Romance, Thriller)                en
26                  (Action, Adventure, Science Fiction)                en
27        (Action, Adventure, Science Fiction, Thriller)                en
28        (Action, Adventure, Science Fiction, Thriller)                en
29                         (Action, Adventure, Thriller)                en
...                                                  ...               ...
49057                 (Action, Drama, Romance, Thriller)                hi
49108                                 (Action, Thriller)                en
49153          (Action, Drama, History, Thriller, War)                en
49171                                          (Action,)                ta
```

```
49186                                   (Action, Drama)              en
49265                           (Action, Comedy, Drama)              ko
49285                            (Adventure, Animation)              ja
49291                                    (Documentary,)              en
49315    (Action, Adventure, Science Fiction, Thriller)              en
49317                                         (Comedy,)              fr
49321                                   (Comedy, Drama)              ru
49323                                         (Comedy,)              ru
49329                                 (Comedy, Romance)              ru
49360                                          (Drama,)              fr
49376                          (Crime, Drama, Thriller)              en
49418                                    (Crime, Drama)              it
49427                                 (Drama, Thriller)              en
49440                                 (Comedy, Romance)              fi
49482  (Action, Fantasy, Horror, Science Fiction, Wes...              en
49528                                        (Romance,)              en
49557                                          (Drama,)              ru
49603                       (Animation, Comedy, Family)              en
49630              (Action, Crime, Mystery, Thriller)              en
49708                                         (Action,)              ta
49710                           (Action, Comedy, Drama)              ta
49836                       (Horror, Mystery, Thriller)              ru
49854                                   (Comedy, Crime)              ru
49856                        (Drama, History, Mystery)              ru
49866                                  (Drama, Romance)              en
49876                 (Action, Comedy, Crime, Foreign)              ru

                                 production_countries release_date  \
0             (United Kingdom, United States of America)   2009-12-10
1                             (United States of America,)   2007-05-19
2             (United Kingdom, United States of America)   2015-10-26
3                             (United States of America,)   2012-07-16
4                             (United States of America,)   2012-03-07
5                             (United States of America,)   2007-05-01
6                             (United States of America,)   2010-11-24
7                             (United States of America,)   2015-04-22
8             (United Kingdom, United States of America)   2009-07-07
9                             (United States of America,)   2016-03-23
10                            (United States of America,)   2006-06-28
11            (United Kingdom, United States of America)   2008-10-30
12     (Bahamas, Dominica, Jamaica, United States of ...   2006-06-20
13                            (United States of America,)   2013-07-03
14            (United Kingdom, United States of America)   2013-06-12
15     (Czech Republic, Poland, Slovenia, United Stat...   2008-05-15
16                            (United States of America,)   2012-04-25
17                            (United States of America,)   2011-05-14
18                            (United States of America,)   2012-05-23
19               (New Zealand, United States of America)   2014-12-10
```

```
20                             (United States of America,)   2012-06-27
21             (United Kingdom, United States of America)   2010-05-12
22                (New Zealand, United States of America)   2013-12-11
23             (United Kingdom, United States of America)   2007-12-04
24       (Germany, New Zealand, United States of America)   2005-12-14
25                             (United States of America,)   1997-11-18
26                             (United States of America,)   2016-04-27
27                             (United States of America,)   2012-04-11
28                             (United States of America,)   2015-06-09
29             (United Kingdom, United States of America)   2012-10-25
...                                                    ...          ...
49057                                            (India,)   2006-05-26
49108        (Germany, Sweden, United States of America)   2017-07-26
49153  (France, Netherlands, United Kingdom, United S...   2017-07-19
49171                                            (India,)   2017-01-12
49186                (Japan, United States of America)   1985-05-03
49265                                     (South Korea,)   2017-01-18
49285                                            (Japan,)   2016-04-23
49291                          (United States of America,)   2017-04-14
49315                          (United States of America,)   2017-06-21
49317                                           (France,)   2012-07-11
49321                                           (Russia,)   2012-12-27
49323                                           (Russia,)   2011-07-21
49329                                           (Russia,)   2011-02-03
49360                              (France, Switzerland)   2016-08-17
49376                          (United States of America,)   2017-08-11
49418                                            (Italy,)   2000-08-31
49427            (Australia, United States of America)   2017-06-29
49440                                          (Finland,)   2005-12-30
49482          (South Africa, United States of America)   2017-08-03
49528                                            (China,)   2014-04-25
49557                                           (Russia,)   2012-12-09
49603                          (United States of America,)   2017-07-28
49630  (Canada, United Kingdom, United States of Amer...   2017-08-03
49708                                            (India,)   1995-01-15
49710                                            (India,)   2007-06-14
49836                                           (Russia,)   2014-03-13
49854                                           (Russia,)   2014-06-05
49856                                           (Russia,)   2009-10-09
49866                                           (Russia,)   2010-09-30
49876                                           (Russia,)   2007-09-06

       runtime  popularity  vote_count  vote_average     budget     revenue
0        162.0  150.437577       11800           7.2  237000000  2787965087
1        169.0  139.082615        4500           6.9  300000000   961000000
2        148.0  107.376788        4466           6.3  245000000   880674609
3        165.0  112.312950        9106           7.6  250000000  1084939099
4        132.0   43.926995        2124           6.1  260000000   284139100
```

| | | | | | | |
|---|---|---|---|---|---|---|
| 5 | 139.0 | 115.699814 | 3576 | 5.9 | 258000000 | 890871626 |
| 6 | 100.0 | 48.681969 | 3330 | 7.4 | 260000000 | 591794936 |
| 7 | 141.0 | 134.279229 | 6767 | 7.3 | 280000000 | 1405403694 |
| 8 | 153.0 | 98.885637 | 5293 | 7.4 | 250000000 | 933959197 |
| 9 | 151.0 | 155.790452 | 7004 | 5.7 | 250000000 | 873260194 |
| 10 | 154.0 | 57.925623 | 1400 | 5.4 | 270000000 | 391081192 |
| 11 | 106.0 | 107.928811 | 2965 | 6.1 | 200000000 | 586090727 |
| 12 | 151.0 | 145.847379 | 5246 | 7.0 | 200000000 | 1065659812 |
| 13 | 149.0 | 49.046956 | 2311 | 5.9 | 255000000 | 89289910 |
| 14 | 143.0 | 99.398009 | 6359 | 6.5 | 225000000 | 662845518 |
| 15 | 150.0 | 53.978602 | 1630 | 6.3 | 225000000 | 419651413 |
| 16 | 143.0 | 144.448633 | 11776 | 7.4 | 220000000 | 1519557910 |
| 17 | 136.0 | 135.413856 | 4948 | 6.4 | 380000000 | 1045713802 |
| 18 | 106.0 | 52.035179 | 4160 | 6.2 | 225000000 | 624026776 |
| 19 | 144.0 | 120.965743 | 4760 | 7.1 | 250000000 | 956019788 |
| 20 | 136.0 | 89.866276 | 6586 | 6.5 | 215000000 | 752215857 |
| 21 | 140.0 | 37.668301 | 1398 | 6.2 | 200000000 | 310669540 |
| 22 | 161.0 | 94.370564 | 4524 | 7.6 | 250000000 | 958400000 |
| 23 | 113.0 | 42.990906 | 1303 | 5.8 | 180000000 | 372234864 |
| 24 | 187.0 | 61.226010 | 2337 | 6.6 | 207000000 | 550000000 |
| 25 | 194.0 | 100.025899 | 7562 | 7.5 | 200000000 | 1845034188 |
| 26 | 147.0 | 198.372395 | 7241 | 7.1 | 250000000 | 1153304495 |
| 27 | 131.0 | 64.928382 | 2114 | 5.5 | 209000000 | 303025485 |
| 28 | 124.0 | 418.708552 | 8662 | 6.5 | 150000000 | 1513528810 |
| 29 | 143.0 | 93.004993 | 7604 | 6.9 | 200000000 | 1108561013 |
| ... | ... | ... | ... | ... | ... | ... |
| 49057 | 168.0 | 3.003526 | 53 | 6.7 | 5300000 | 22175908 |
| 49108 | 115.0 | 14.455104 | 748 | 6.1 | 30000000 | 90007945 |
| 49153 | 107.0 | 30.938854 | 2712 | 7.5 | 100000000 | 519876949 |
| 49171 | 168.0 | 1.459459 | 12 | 6.5 | 0 | 17000000 |
| 49186 | 90.0 | 1.542843 | 14 | 4.7 | 8500000 | 5730596 |
| 49265 | 125.0 | 1.758590 | 5 | 6.2 | 8520000 | 56100000 |
| 49285 | 120.0 | 3.235740 | 29 | 6.8 | 0 | 1015339 |
| 49291 | 99.0 | 0.519968 | 2 | 8.0 | 0 | 393970 |
| 49315 | 149.0 | 39.186819 | 1440 | 6.2 | 260000000 | 604942143 |
| 49317 | 98.0 | 9.754955 | 92 | 5.4 | 0 | 103504 |
| 49321 | 90.0 | 1.456046 | 9 | 5.3 | 2000000 | 11666088 |
| 49323 | 81.0 | 0.397106 | 7 | 3.1 | 2000000 | 8000000 |
| 49329 | 90.0 | 0.445269 | 3 | 4.7 | 3000000 | 1957000 |
| 49360 | 89.0 | 2.404466 | 24 | 6.1 | 0 | 126463 |
| 49376 | 99.0 | 5.798555 | 46 | 7.3 | 0 | 10893246 |
| 49418 | 114.0 | 4.675250 | 116 | 7.8 | 0 | 1805884 |
| 49427 | 99.0 | 37.484577 | 277 | 5.5 | 0 | 422 |
| 49440 | 107.0 | 0.947509 | 10 | 5.6 | 2196531 | 2411594 |
| 49482 | 95.0 | 50.903593 | 688 | 5.7 | 60000000 | 71000000 |
| 49528 | 0.0 | 0.504000 | 4 | 6.0 | 0 | 76000000 |
| 49557 | 72.0 | 0.181963 | 3 | 5.7 | 0 | 14353 |
| 49603 | 86.0 | 33.694599 | 327 | 5.8 | 50000000 | 66913939 |

```
49630    111.0   40.796775          181              7.4   11000000   184770205
49708    145.0    0.704162           14              7.8          0    15000000
49710    185.0    1.323587           25              6.9   12000000    19000000
49836     90.0    0.217441            4              4.4          0      320395
49854      0.0    0.201582            4              6.0     750000           3
49856    110.0    0.436028            3              6.3          0       50656
49866    107.0    0.121844            3              4.0    2000000     1268793
49876     91.0    0.039793            1              1.0    5000000     1413000

[7254 rows x 11 columns]
```

## 11 Data Analysis & Results

Among all the categories contained in our cleaned dataset, we selected genre, production country, title length, movie runtime, release month,vote average and movie budget as the potential factors influencing the final revenue of a newly released movie. First, we inspected the direct linear correlation between the movie revenue and these potential factors individually. The linear correlation is determined by the p-value and goodness of fit generated from the OLS Regression function. Next, we filter out most correlated categories for further multiple linear regression. This step would tell us about the possible interrelations among the categpries themselves. The goodness of fit was checked again and a non-linear regression was performed to test if there is a better fitting method.

### 11.1 Data set sorted according to the revenue

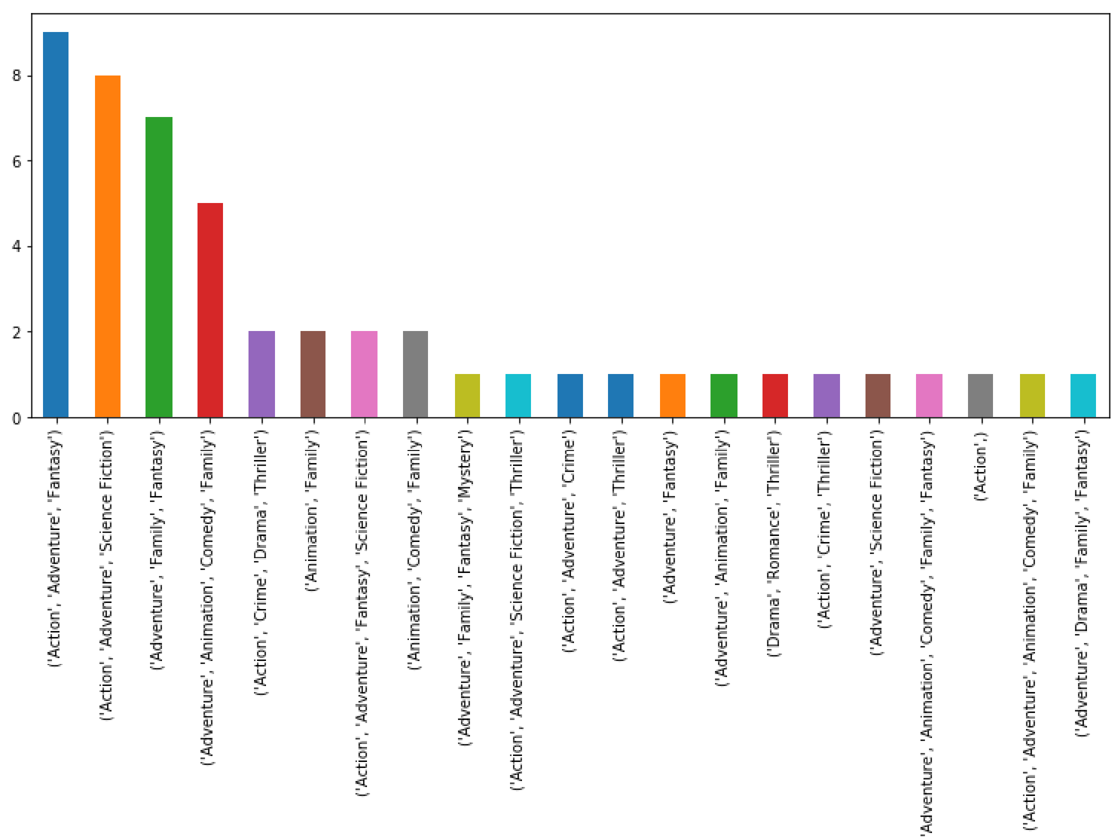
In [66]:
```
# open the trimmed data, sort the order by 'revenue' column to facilize the visualiza
# on the following cell, a sample table after sorting is shown
df = pd.read_csv('trimmed_data.csv',index_col = 0)
df = df.sort_values(by = ['revenue'],ascending=False)
df.head(5)
```

Out[66]:
```
                             title  \
0                           Avatar
31274  Star Wars: The Force Awakens
25                         Titanic
16                    The Avengers
28                  Jurassic World

                                                  genres original_language  \
0      ('Action', 'Adventure', 'Fantasy', 'Science Fi...                en
31274  ('Action', 'Adventure', 'Fantasy', 'Science Fi...                en
25                    ('Drama', 'Romance', 'Thriller')                  en
16              ('Action', 'Adventure', 'Science Fiction')              en
28     ('Action', 'Adventure', 'Science Fiction', 'Th...                en

                                 production_countries release_date  runtime  \
0      ('United Kingdom', 'United States of America')   2009-12-10    162.0
31274                    ('United States of America',)   2015-12-15    136.0
25                       ('United States of America',)   1997-11-18    194.0
```

```
16                        ('United States of America',)   2012-04-25    143.0
28                        ('United States of America',)   2015-06-09    124.0

       popularity  vote_count  vote_average     budget     revenue
0      150.437577       11800           7.2  237000000  2787965087
31274   31.626013        7993           7.5  245000000  2068223624
25     100.025899        7562           7.5  200000000  1845034188
16     144.448633       11776           7.4  220000000  1519557910
28     418.708552        8662           6.5  150000000  1513528810
```

Here, we plotted a bar graph in order to visualize the genres of the 50 most profitable movies. As the bar graph suggested, 9 movies has a combined genre of ('Action', 'Adventure', 'Fantasy'). 8 movies have a combined genre of ('Action', 'Adventure', 'Science Fiction'). 7 movies have a combined genre of ('Adventure', 'Family', 'Fantacy'). 5 movies have a combined genre of ('Adventure', 'Animation', 'Comedy', 'Family'). It appears that these combined genres have a good correlation with the movie revenue.

```
In [67]: pd.value_counts(df.head(50)['genres']).plot.bar(figsize = (13, 5))
         f1 = plt.gcf()
```



### 11.2 Linear Model Of Movie Genres and Revenue

We start the analysis by first fitting the movie genres column and the revenue column. In order to check for the potential correlation with the movie revenue, we performed an OLS regression on all the combined genres included in our dataset, using an alpha value of 0.05. We found, from the OLS regression output, that the combined genres of ('Action', 'Adventure', 'Fantasy') has a p value smaller than 0.001. The combined genre of ('Action', 'Adventure', 'Science Fiction') also has a p value smaller than 0.001. The same p value applies to the combined genres of ('Adventure', 'Family', 'Fantacy'), and ('Adventure', 'Animation', 'Comedy', 'Family'). It is then confirmed that there is a correlation bvetween these genres and the movie revenue.

```
In [88]: # First setting the revenue as the explanatory variable and genres as the dependent v
         # Then perform the fitting action, the result is printed.

         # The result first shows the overall data such as the R-squared value. Then on the lo
         # specific genre that is fitted to the revenue, and then the value of coefficient, st
         # shown on the following line. Notice that due to the formating issue, the coefficien
         # of the line above

         # Due to the length of the whole summary, we only depict the first could p-value. We 
         # p-value following the summary
         outcome_1, predictors_1 = patsy.dmatrices("revenue~genres",df)
         mod_1 = sm.OLS(outcome_1, predictors_1)
         res_1 = mod_1.fit()
         print(str(res_1.summary())[0:2992])
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                revenue   R-squared:                       0.359
Model:                            OLS   Adj. R-squared:                  0.280
Method:                 Least Squares   F-statistic:                     4.506
Date:                Wed, 12 Jun 2019   Prob (F-statistic):         6.93e-257
Time:                        23:19:32   Log-Likelihood:            -1.4508e+05
No. Observations:                7254   AIC:                         2.918e+05
Df Residuals:                    6450   BIC:                         2.973e+05
Df Model:                         803
Covariance Type:            nonrobust
==========================================================================================

------------------------------------------------------------------------------------------
Intercept
genres[T.('Action', 'Adventure')]
genres[T.('Action', 'Adventure', 'Animation')]
genres[T.('Action', 'Adventure', 'Animation', 'Comedy')]
genres[T.('Action', 'Adventure', 'Animation', 'Comedy', 'Drama', 'Family')]
genres[T.('Action', 'Adventure', 'Animation', 'Comedy', 'Family')]
genres[T.('Action', 'Adventure', 'Animation', 'Comedy', 'Family', 'Fantasy')]
genres[T.('Action', 'Adventure', 'Animation', 'Comedy', 'Family', 'Fantasy', 'Science Fiction')
```

```
genres[T.('Action', 'Adventure', 'Animation', 'Comedy', 'Family', 'Fantasy', 'Science Fiction'
```

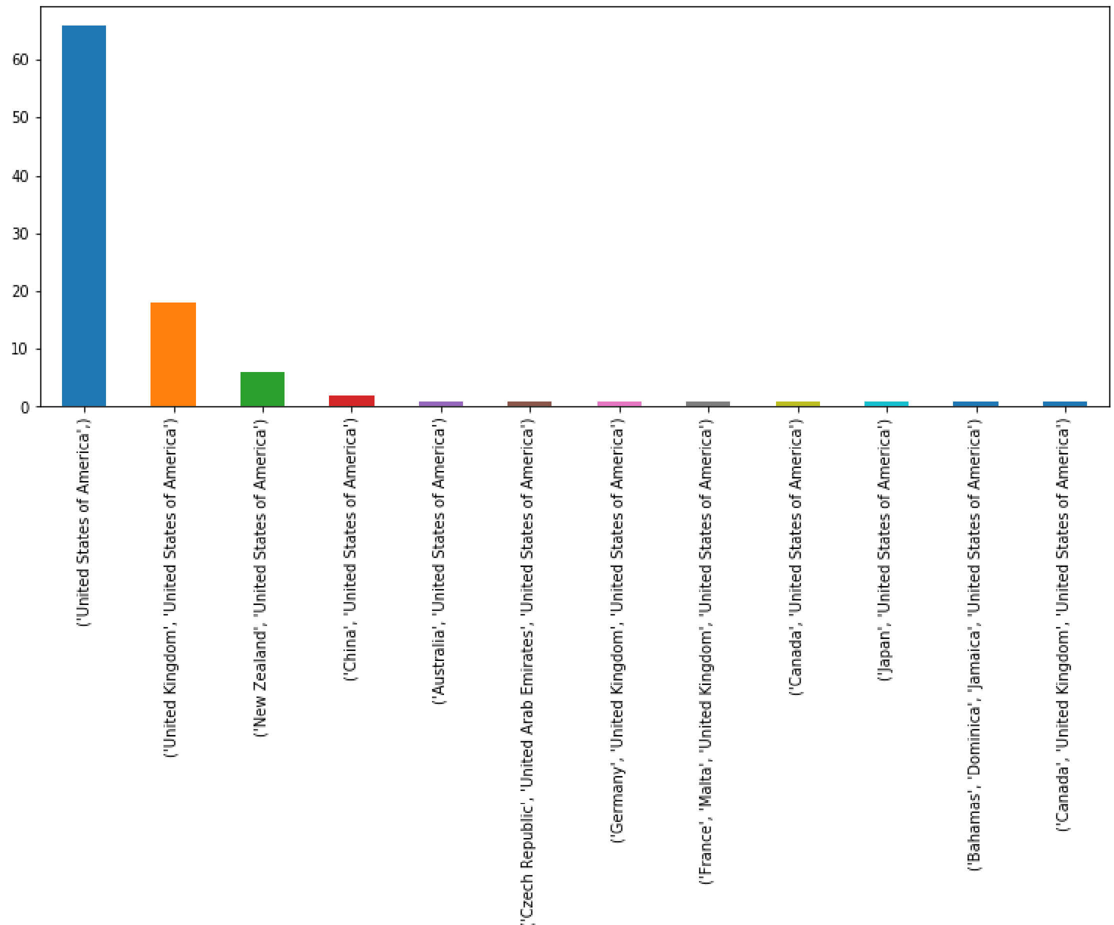
```
In [90]: # depicting the top 10 p-value (which is the smallest 10 p-value, since it implies th
         Genres_Pvalues = pd.DataFrame(res_1.model.exog_names, columns = ['Genres'])
         Genres_Pvalues['Pvalues'] =  res_1.pvalues
         pd.set_option('display.float_format','{:.3f}'.format)

         Genres_Pvalues = Genres_Pvalues.sort_values(by = ['Pvalues'],ascending=True)
         Genres_Pvalues.head(10)
```

```
Out[90]:                                           Genres  Pvalues
         96         genres[T.('Action', 'Adventure', 'Fantasy')]    0.000
         101  genres[T.('Action', 'Adventure', 'Fantasy', 'S...    0.000
         122  genres[T.('Action', 'Adventure', 'Science Fict...    0.000
         398        genres[T.('Adventure', 'Family', 'Fantasy')]    0.000
         300  genres[T.('Adventure', 'Animation', 'Comedy', ...    0.000
         299  genres[T.('Adventure', 'Animation', 'Comedy', ...    0.000
         373  genres[T.('Adventure', 'Drama', 'Fantasy', 'Ro...    0.000
         123  genres[T.('Action', 'Adventure', 'Science Fict...    0.000
         400  genres[T.('Adventure', 'Family', 'Fantasy', 'M...    0.000
         439         genres[T.('Animation', 'Comedy', 'Family')]    0.000
```

### 11.3 Movie production countries from top 100 movie revenues

Then, we proceed to the production countries. We plotted a bar graph of the 100 most profitable movies according to their country of production. As indicated by the bar graph, more than 60 movies were produced by ('United States of America'), suggesting that the ('United States of America') is potentially correlated with the movie revenue. We used an alpha value of 0.05.

```
In [14]: pd.value_counts(df.head(100)['production_countries']).plot.bar(figsize = (13, 5))
         f1 = plt.gcf()
```




An OLS regression test was performed to analyze the correlation between each country of production and the movie revenue. As expected, the ('United States of America') has a p value smaller than 0.001.

```
In [86]: outcome_1, predictors_1 = patsy.dmatrices("revenue~production_countries",df)
         mod_1 = sm.OLS(outcome_1, predictors_1)
         res_1 = mod_1.fit()
         print(str(res_1.summary())[0:3369])
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                revenue   R-squared:                       0.109
Model:                            OLS   Adj. R-squared:                  0.038
Method:                 Least Squares   F-statistic:                     1.526
Date:                Wed, 12 Jun 2019   Prob (F-statistic):           6.27e-13
Time:                        23:16:37   Log-Likelihood:            -1.4627e+05
No. Observations:                7254   AIC:                         2.936e+05
Df Residuals:                    6715   BIC:                         2.973e+05
Df Model:                         538
```

```
Covariance Type:            nonrobust
===========================================================================================

-------------------------------------------------------------------------------------------
Intercept
production_countries[T.('Afghanistan', 'France', 'Germany', 'United Kingdom')]
production_countries[T.('Algeria', 'Belgium', 'France', 'Morocco')]
production_countries[T.('Algeria', 'France')]
production_countries[T.('Algeria', 'Italy')]
production_countries[T.('Angola', 'France')]
production_countries[T.('Argentina', 'Brazil', 'Chile', 'France', 'Germany', 'Peru', 'United Ki
```

### 11.4 Relation between the title length and revenue

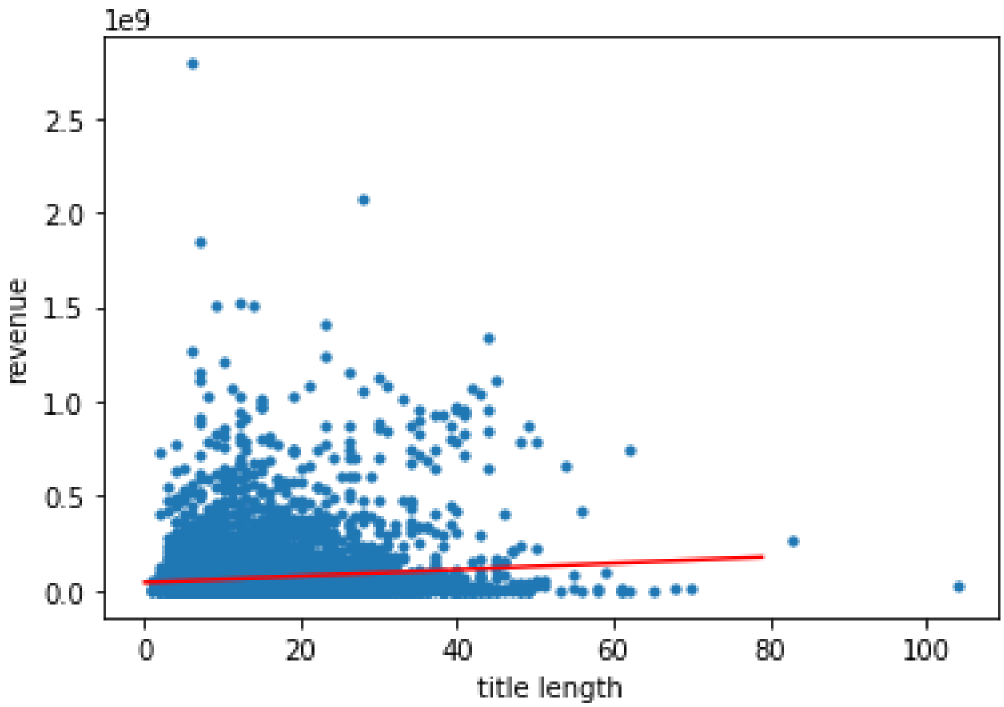
For our interest, we also analyze the correlation between title length and movie revenue. We counted the number of characters contained in each movie title, which is later designated as the title length.

```
In [16]: # Extract the needed data, title length and revenue, for this analysis. Transform the
         # each movie title
         lenth_rev = df[['title','revenue']]
         lenth_rev['title'] = lenth_rev['title'].apply(len)
```

```
/Users/winniexu/anaconda3/lib/python3.6/site-packages/ipykernel_launcher.py:4: SettingWithCopyW
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
  after removing the cwd from sys.path.
```

A scatter plot was made for title length vs. revenue. A linear regression was thencomputed and drawn on top of the scatter plot. The final plot indicats that there is some kind of correlation between the title length and the movie revenue.

```
In [18]: a1, b1 = np.polyfit(lenth_rev['title'], lenth_rev['revenue'], 1)
         title_len = np.arange(0,80,dtype = 'float')
         pred_rev = title_len * a1 + b1

         plt.scatter(x = lenth_rev['title'], y = lenth_rev['revenue'], s=10)
         plt.plot(title_len, pred_rev, linestyle='-',color = "red")
         plt.xlabel('title length')
         plt.ylabel('revenue')

Out[18]: Text(0,0.5,'revenue')
```




To confirm this, we used an OLS regression test to analyze the correlation between the title length and the movie revenue, with an alpha value of 0.05. The returned p value is smaller than 0.001, suggesting that there is a correlation between the title length and movie revenue. Yet,the R-squared value (0.009) indicates that a linear regression is a poor fitting for the correlation between the title length and movie revenue.

```
In [19]: lenth_rev.rename(columns={'title':'title_len'}, inplace=True)
         outcome_1, predictors_1 = patsy.dmatrices("revenue~title_len",lenth_rev)
         mod_1 = sm.OLS(outcome_1, predictors_1)
         res_1 = mod_1.fit()
         print(res_1.summary())
```

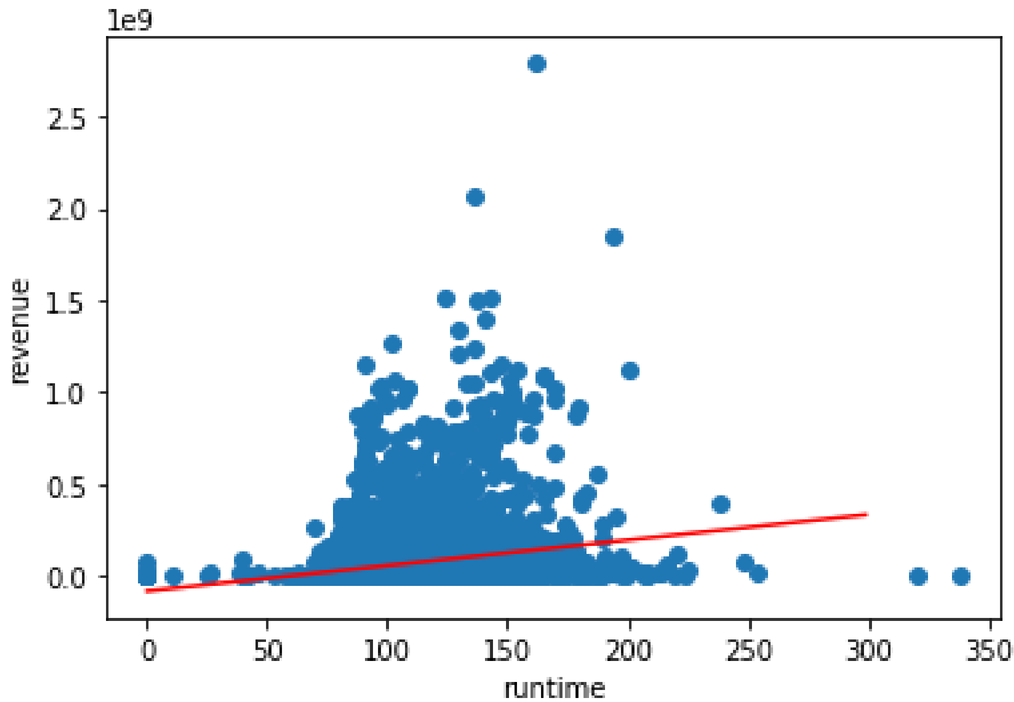
```
                            OLS Regression Results
==============================================================================
Dep. Variable:                revenue   R-squared:                       0.009
Model:                            OLS   Adj. R-squared:                  0.009
Method:                 Least Squares   F-statistic:                     69.34
Date:                Wed, 12 Jun 2019   Prob (F-statistic):           9.83e-17
Time:                        20:58:30   Log-Likelihood:            -1.4666e+05
No. Observations:                7254   AIC:                         2.933e+05
Df Residuals:                    7252   BIC:                         2.933e+05
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
```

```
------------------------------------------------------------------------------
Intercept    4.306e+07   3.55e+06     12.138      0.000    3.61e+07       5e+07
title_len    1.699e+06   2.04e+05      8.327      0.000     1.3e+06     2.1e+06
==============================================================================
Omnibus:                     7499.325   Durbin-Watson:                   0.023
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           595287.165
Skew:                           5.088   Prob(JB):                         0.00
Kurtosis:                      46.197   Cond. No.                         36.1
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.


/Users/winniexu/anaconda3/lib/python3.6/site-packages/pandas/core/frame.py:3027: SettingWithCop
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
  return super(DataFrame, self).rename(**kwargs)
```

### 11.5 Relation between the movie runtime and revenue

We performed a linear regression on the scatter plot of movie runtime vs. movie revenue. As the plot has suggested, it seems like there is also some kind of correlation between movie runtime and revenue.

```
In [20]: timelenth_rev = df[['runtime','revenue']]
         a, b = np.polyfit(timelenth_rev['runtime'], timelenth_rev['revenue'], 1)
         time_len = np.arange(0,300,dtype = 'float')
         pred_rev = a * time_len + b
         plt.scatter(x = timelenth_rev['runtime'], y = timelenth_rev['revenue'])
         plt.plot(time_len, pred_rev, linestyle='-',color = "red")
         plt.xlabel('runtime')
         plt.ylabel('revenue')

Out[20]: Text(0,0.5,'revenue')
```




Again, we used an OLS regression to confirm our conjecture. The test returned a p value was smaller than 0.001, indicating a linear correlation between the movie runtime and the movie revenue.However,the R-squared value (0.041) indicates that a linear regression is a poor fitting for the correlation between the movie runtime and movie revenue.

```
In [21]: outcome_1, predictors_1 = patsy.dmatrices("revenue~runtime",timelenth_rev)
         mod_1 = sm.OLS(outcome_1, predictors_1)
         res_1 = mod_1.fit()
         print(res_1.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                revenue   R-squared:                       0.041
Model:                            OLS   Adj. R-squared:                  0.041
Method:                 Least Squares   F-statistic:                     313.8
Date:                Wed, 12 Jun 2019   Prob (F-statistic):           9.01e-69
Time:                        20:59:07   Log-Likelihood:            -1.4654e+05
No. Observations:                7254   AIC:                         2.931e+05
Df Residuals:                    7252   BIC:                         2.931e+05
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept  -8.039e+07    8.6e+06     -9.352      0.000   -9.72e+07   -6.35e+07
```

```
runtime     1.386e+06   7.83e+04     17.714      0.000    1.23e+06    1.54e+06
==============================================================================
Omnibus:                     7340.372   Durbin-Watson:                   0.085
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           542103.202
Skew:                           4.930   Prob(JB):                         0.00
Kurtosis:                      44.187   Cond. No.                         560.
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

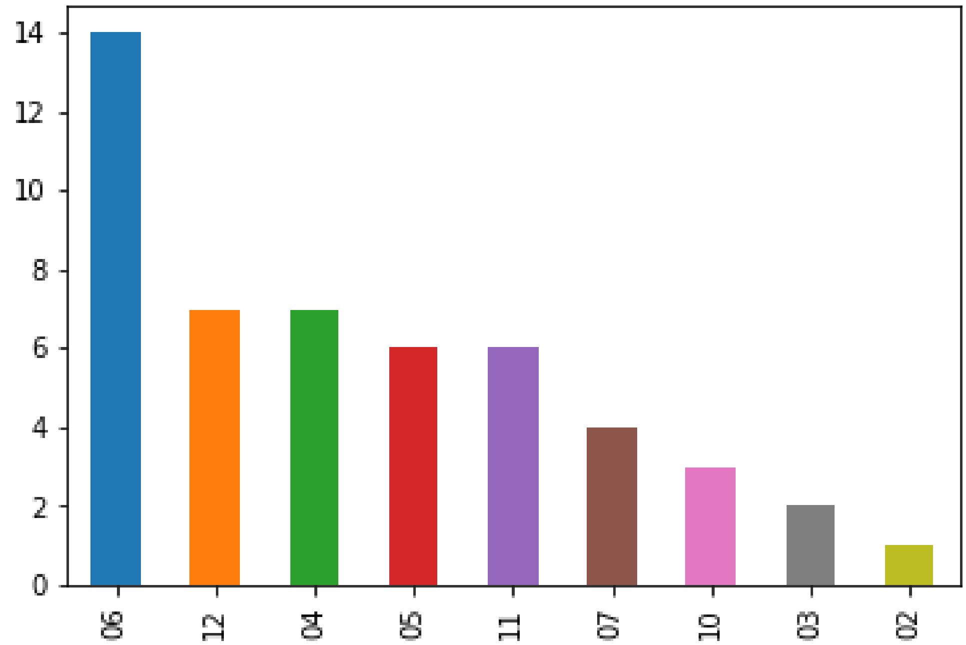
### 11.6 Movie release month of top 50 movie revenues

We plotted a bar graph of the release month for the 50 most profitable movies. According to the bar graph, June has the most movie release (14 movies).

```
In [30]: # Select the needed column and only extract the month information
         date_rev = df[['release_date','revenue']]
         date_rev['release_date']=date_rev['release_date'].str[5:7]
```

```
/Users/winniexu/anaconda3/lib/python3.6/site-packages/ipykernel_launcher.py:3: SettingWithCopyW
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
  This is separate from the ipykernel package so we can avoid doing imports until
```

```
In [32]: # Note that in this plot, the x axis represents the month and the y axis represents th
         # The order of month was sorted by the movie count.
         pd.value_counts(date_rev.head(50)['release_date']).plot.bar()
         f1 = plt.gcf()
```




An OLS regression test was performed to provide further insight into the correlation between the release month and the movie revenue, using an alpha value of 0.05. June has a p value smaller than 0.001, indicating that June is indeed correlated with higher movie revenues.

```
In [33]: date_rev.rename(columns={'release_date':'release_month'}, inplace=True)
         outcome_1, predictors_1 = patsy.dmatrices("revenue~release_month",date_rev)
         mod_1 = sm.OLS(outcome_1, predictors_1)
         res_1 = mod_1.fit()
         print(res_1.summary())
```

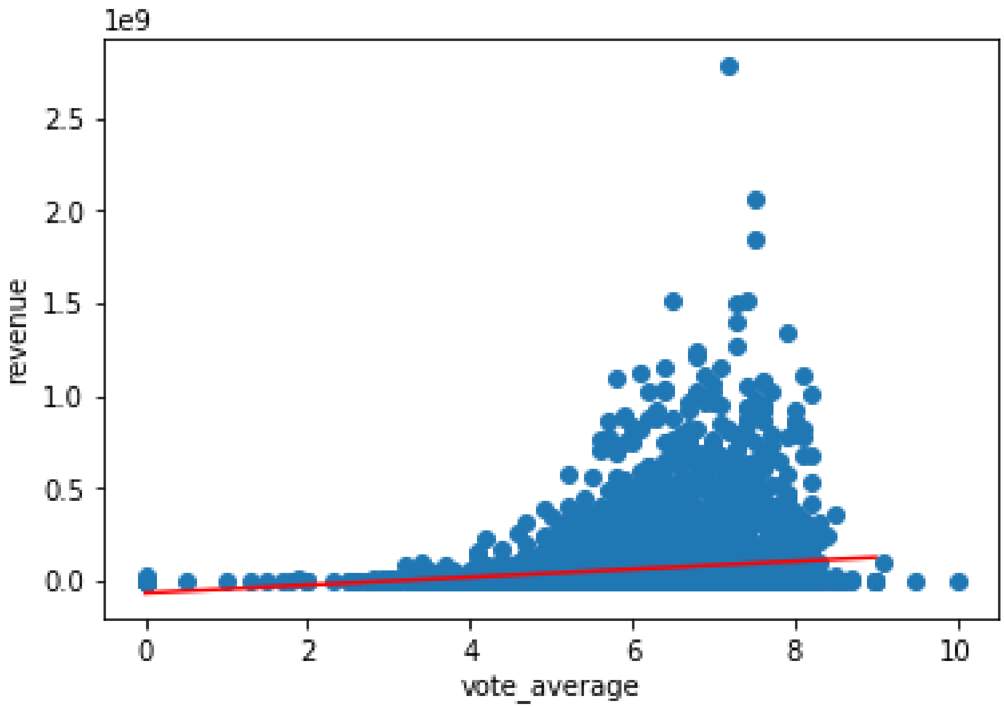
```
                            OLS Regression Results
==============================================================================
Dep. Variable:                revenue   R-squared:                       0.038
Model:                            OLS   Adj. R-squared:                  0.037
Method:                 Least Squares   F-statistic:                     26.24
Date:                Wed, 12 Jun 2019   Prob (F-statistic):           2.99e-54
Time:                        21:04:07   Log-Likelihood:            -1.4655e+05
No. Observations:                7254   AIC:                         2.931e+05
Df Residuals:                    7242   BIC:                         2.932e+05
Df Model:                          11
Covariance Type:            nonrobust
=======================================================================================
                          coef    std err          t      P>|t|      [0.025      0.975]
---------------------------------------------------------------------------------------
Intercept            3.213e+07   6.41e+06      5.013      0.000    1.96e+07    4.47e+07
release_month[T.02]  1.958e+07   9.06e+06      2.160      0.031    1.81e+06    3.73e+07
```

```
release_month[T.03]  3.165e+07   8.88e+06      3.566      0.000     1.42e+07     4.9e+07
release_month[T.04]  3.088e+07    8.9e+06      3.470      0.001     1.34e+07    4.83e+07
release_month[T.05]  6.576e+07   8.74e+06      7.521      0.000     4.86e+07    8.29e+07
release_month[T.06]  8.789e+07   8.75e+06     10.039      0.000     7.07e+07    1.05e+08
release_month[T.07]  6.122e+07   8.86e+06      6.910      0.000     4.39e+07    7.86e+07
release_month[T.08]  1.332e+07   8.55e+06      1.558      0.119    -3.44e+06    3.01e+07
release_month[T.09]  4.258e+05   8.02e+06      0.053      0.958    -1.53e+07    1.62e+07
release_month[T.10]  1.696e+07   8.44e+06      2.009      0.045     4.09e+05    3.35e+07
release_month[T.11]  6.561e+07   8.96e+06      7.323      0.000      4.8e+07    8.32e+07
release_month[T.12]   6.46e+07   8.45e+06      7.648      0.000      4.8e+07    8.12e+07
==============================================================================
Omnibus:                     7482.742   Durbin-Watson:                   0.079
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           603672.653
Skew:                           5.061   Prob(JB):                         0.00
Kurtosis:                      46.530   Cond. No.                         14.1
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.


/Users/winniexu/anaconda3/lib/python3.6/site-packages/pandas/core/frame.py:3027: SettingWithCop
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
  return super(DataFrame, self).rename(**kwargs)
```

### 11.7 Relation between the vote average and revenue

We created a scatter plot to visualize the data and then computed a linear regression based on the scatter plot. As suggested by the plot, there is some kind of correlation between vote average and revenue.

```
In [34]: voteagv_rev = df[['vote_average','revenue']]
         a, b = np.polyfit(voteagv_rev['vote_average'], voteagv_rev['revenue'], 1)
         vot_average = np.arange(0,10,dtype = 'float')
         pred_vote = a * vot_average + b
         plt.scatter(x = voteagv_rev['vote_average'], y = voteagv_rev['revenue'])
         plt.plot(vot_average, pred_vote, linestyle='-',color = "red")
         plt.xlabel('vote_average')
         plt.ylabel('revenue')

Out[34]: Text(0,0.5,'revenue')
```




An OLS regression test has provided further information pertaining to the correlation between the average vote and the movie revenue. The test has returned a p values smaller than 0.001, which is smaller than the alpha value used (0.05). This suggests the existance of a correlation. Yet, the R-squared value is 0.022, which suggests that a linear regression is a poor fitting for the correlation between the average vote and the movie revenue.

```
In [35]: outcome_1, predictors_1 = patsy.dmatrices("revenue~vote_average",voteagv_rev)
         mod_1 = sm.OLS(outcome_1, predictors_1)
         res_1 = mod_1.fit()
         print(res_1.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                revenue   R-squared:                       0.022
Model:                            OLS   Adj. R-squared:                  0.022
Method:                 Least Squares   F-statistic:                     163.1
Date:                Wed, 12 Jun 2019   Prob (F-statistic):           5.79e-37
Time:                        21:04:59   Log-Likelihood:            -1.4661e+05
No. Observations:                7254   AIC:                         2.932e+05
Df Residuals:                    7252   BIC:                         2.932e+05
Df Model:                           1
Covariance Type:            nonrobust
================================================================================
                   coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------
```

```
Intercept    -6.423e+07   1.06e+07     -6.080      0.000   -8.49e+07   -4.35e+07
vote_average   2.14e+07   1.68e+06     12.773      0.000    1.81e+07    2.47e+07
==============================================================================
Omnibus:                     7453.695   Durbin-Watson:                   0.048
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           578376.552
Skew:                           5.043   Prob(JB):                         0.00
Kurtosis:                      45.566   Cond. No.                         40.1
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

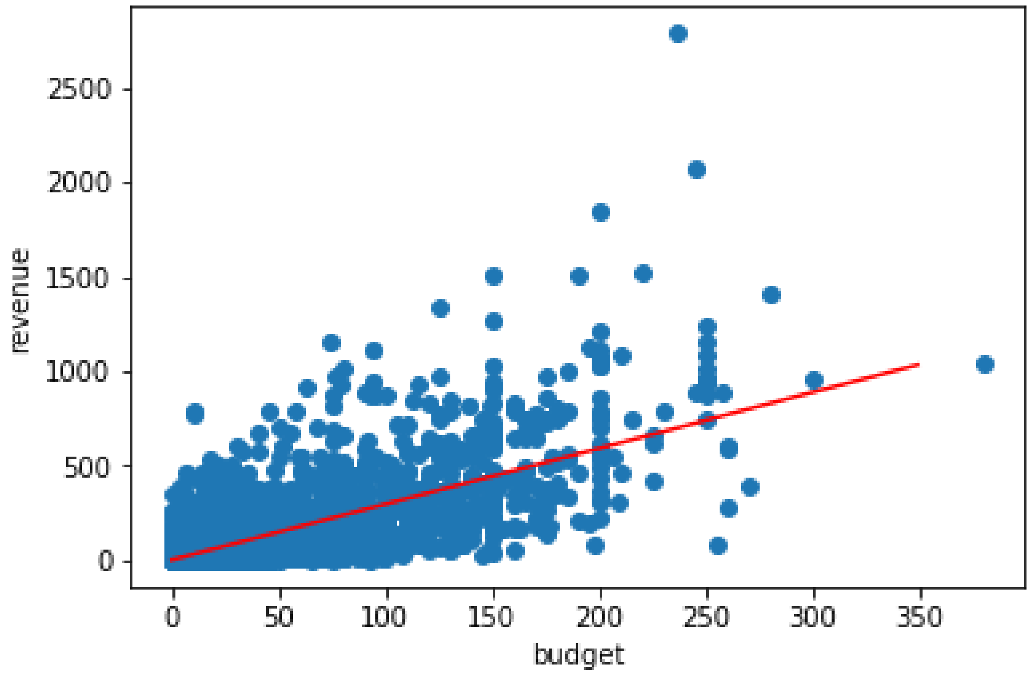
## 11.8 Relation between the budget and revenue

We plotted the scatter plot with the budget in million dollars against the movie revenue. A linear regression was computed on the same scatter plot. According to the final plot, the budget appears to have a correlation with the movie revenue.

```
In [36]: # Exact the necessary columns and factor each value of budgets by 1,000,000
         budget_rev = df[['budget','revenue']]
         budget_rev['budget'] = budget_rev['budget']/1000000
         budget_rev['revenue'] = budget_rev['revenue']/1000000
         a, b = np.polyfit(budget_rev['budget'], budget_rev['revenue'], 1)
         budget = np.arange(0,350,dtype = 'float')
         pred_budget = a * budget + b
         plt.scatter(x = budget_rev['budget'], y = budget_rev['revenue'])
         plt.plot(budget, pred_budget, linestyle='-',color = "red")
         plt.xlabel('budget')
         plt.ylabel('revenue')
```

```
/Users/winniexu/anaconda3/lib/python3.6/site-packages/ipykernel_launcher.py:2: SettingWithCopy
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.htm

/Users/winniexu/anaconda3/lib/python3.6/site-packages/ipykernel_launcher.py:3: SettingWithCopy
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.htm
  This is separate from the ipykernel package so we can avoid doing imports until
```

```
Out[36]: Text(0,0.5,'revenue')
```




The OLS regression test has returned a p value smaller than 0.001, which is smaller than the alpha value we used (0.05). Meanwhile, the R-squared value is 0.557. These two results indicate that there is a linear collrelation between the movie budget and the movie revenue.

```
In [37]: outcome_1, predictors_1 = patsy.dmatrices("revenue~budget",budget_rev)
         mod_1 = sm.OLS(outcome_1, predictors_1)
         res_1 = mod_1.fit()
         print(res_1.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                revenue   R-squared:                       0.557
Model:                            OLS   Adj. R-squared:                  0.557
Method:                 Least Squares   F-statistic:                     9108.
Date:                Wed, 12 Jun 2019   Prob (F-statistic):               0.00
Time:                        21:05:02   Log-Likelihood:                -43523.
No. Observations:                7254   AIC:                         8.705e+04
Df Residuals:                    7252   BIC:                         8.706e+04
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      1.9546      1.344      1.454      0.146      -0.680       4.589
budget         2.9529      0.031     95.436      0.000       2.892       3.014
==============================================================================
```

```
Omnibus:                     6998.804   Durbin-Watson:                   1.004
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           919260.536
Skew:                           4.302   Prob(JB):                         0.00
Kurtosis:                      57.473   Cond. No.                         50.9
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

## 12 Multiple Linear Regression

With the results of the previous linear regression analysis, we filtered out budget, popularity, runtime and vote average as the possible contributing factor of final movie revenue. We decided to ignore genre as a factor as each movie fell into multiple genres hence making it a much complex task to assign each category a dummy variable for prediction purposes. We proceeded with multiple linear regression to indicate if there are any interrelations among these 4 potential contributors. Multiple linear regression could also provide better goodness of fit and futher rule out the non-significant factors.

In [44]:
```python
from sklearn.utils import shuffle
from sklearn.model_selection import KFold
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split

X = df[['budget', 'popularity', 'runtime','vote_average']]
y = df['revenue']
X = sm.add_constant(X) # adding a constant

model = sm.OLS(y, X).fit() #multiple regression model
predictions = model.predict(X)
print_model = model.summary()
print(print_model)

reg= LinearRegression().fit(X, y)
reg.score(X,y)
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                revenue   R-squared:                       0.637
Model:                            OLS   Adj. R-squared:                  0.637
Method:                 Least Squares   F-statistic:                     3176.
Date:                Wed, 12 Jun 2019   Prob (F-statistic):               0.00
Time:                        21:17:43   Log-Likelihood:            -1.4302e+05
No. Observations:                7254   AIC:                         2.860e+05
Df Residuals:                    7249   BIC:                         2.861e+05
Df Model:                           4
Covariance Type:            nonrobust
```

```
==============================================================================
                   coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         -7.598e+07    7.4e+06    -10.263      0.000   -9.05e+07   -6.15e+07
budget            2.3166      0.034     69.001      0.000       2.251       2.382
popularity     1.613e+06   4.53e+04     35.569      0.000    1.52e+06     1.7e+06
runtime        3.814e+04   5.17e+04      0.738      0.461   -6.32e+04    1.39e+05
vote_average   9.879e+06    1.1e+06      9.011      0.000    7.73e+06     1.2e+07
==============================================================================
Omnibus:                     7241.603   Durbin-Watson:                   1.147
Prob(Omnibus):                  0.000   Jarque-Bera (JB):          1394320.661
Skew:                           4.425   Prob(JB):                         0.00
Kurtosis:                      70.341   Cond. No.                     3.12e+08
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 3.12e+08. This might indicate that there are
strong multicollinearity or other numerical problems.
```

Out[44]: 0.6367253570884439

In [45]:
```
# Since the analysis above indicates that the runtime of a movie may not be an influe
# we eliminated the run_time variable and perform another regression.
X = df[['budget', 'popularity','vote_average']]
y = df['revenue']
X = sm.add_constant(X) # adding a constant

model = sm.OLS(y, X).fit() #multiple regression model
predictions = model.predict(X)
print_model = model.summary()
print(print_model)

reg= LinearRegression().fit(X, y)
reg.score(X,y)

print(np.average(cross_val_score(reg, X, y, cv=10)))
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                revenue   R-squared:                       0.637
Model:                            OLS   Adj. R-squared:                  0.637
Method:                 Least Squares   F-statistic:                     4235.
Date:                Wed, 12 Jun 2019   Prob (F-statistic):               0.00
Time:                        21:17:44   Log-Likelihood:            -1.4302e+05
No. Observations:                7254   AIC:                         2.860e+05
Df Residuals:                    7250   BIC:                         2.861e+05
```

```
Df Model:                           3
Covariance Type:            nonrobust
================================================================================
                  coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------
const       -7.344e+07   6.55e+06    -11.208      0.000   -8.63e+07   -6.06e+07
budget          2.3213      0.033     70.392      0.000       2.257       2.386
popularity   1.613e+06   4.53e+04     35.569      0.000    1.52e+06     1.7e+06
vote_average 1.011e+07   1.05e+06      9.641      0.000    8.06e+06    1.22e+07
==============================================================================
Omnibus:                     7246.207   Durbin-Watson:                   1.147
Prob(Omnibus):                  0.000   Jarque-Bera (JB):          1397409.911
Skew:                           4.430   Prob(JB):                         0.00
Kurtosis:                      70.416   Cond. No.                     2.78e+08
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.78e+08. This might indicate that there are
strong multicollinearity or other numerical problems.
-26262.561446148833
```

The p-value for runtime from the previous model turned out to be 0.461, which is much larger than the critical alpha value. This shows that the total length of the movie may actually not have a significant effect in predicting the revenue of a movie.

Hence we eliminated this variable as a part of backward stepwise variable selection and only kept budget, popularity, and vote average as predictors. The accuracy score we got for this new model was the same as the last one,which further proves that run-time was not an important factor in predicting revenue for movies.

The accuracy score of this model is 0.637 ( R^2), which is much higher than any of the simple linear regression models designed in the previous sections. Hoever, We wanted to explore if this score could get better with a non-linear model. We decided to perform a RandomForest Regression and the result is on the following section.

## 13 Non Linear Regression

In [41]:
```
# Non linear Regression
from sklearn.ensemble import RandomForestRegressor
from sklearn.datasets import make_regression
X, y = make_regression(n_features=4, n_informative=2,
                       random_state=0, shuffle=False)
regr = RandomForestRegressor(max_depth=2, random_state=0,
                             n_estimators=100)
regr.fit(X, y)

print(regr.score(X,y))  #R^2
```

```python
from sklearn.model_selection import cross_val_score

print(np.average(cross_val_score(regr, X, y, cv=10))) #average of 10 fold cross valid
```

```
0.8435830500065526
0.6182275266155581
```

The prediction accuracy score for this model turned out to be the highest we have ever had which is a 84% accuracy at predicting the revenue of movies.

To show that this model was not overfitting, we carried out 10-fold cross validation, which still returned a fairly positive accuracy score : 0.618.

We compared this to the cross-val score of the multiple linear regression model which returned a high negative number. This negative number implies that the regression model extremely underfits our test data and hence is not generally usable.

## 14 Ethics & Privacy

Our question is about what factors contribute to a movie's revenue. The datasets we used were extracted from two well-known movie databases. The user names or any other user information is not presented in the datasets and the datasets are themselves public, so there is no issue about the privacy or informed consent problem. The datasets included any language, any genre and from any country, so there would be no territory bias on the movies themselves. However, we were concerned about the user distribution of the two movie databases, in other words, who would make the ratings. We found that data from these two databases were frequently used to determine the contribution of factors to the revenue, while they were used to predict American movie revenues most of the time. We thought that the reason could be that they were used mainly by users from English-speaking countries and were popular in the US. As a result, our prediction may not be indicative for a non-English-speaking country, while it has a good accuracy to predict the world's revenue.

There is also a concern about the balance between movie revenue and diversity protection. By figuring out how the movie's budget, popularity and vote average, etc affect the revenue, movie investors would have an easier time to decide which movie would be more profitable. However, after finding out that there is a specific type of movie that will make a higher revenue, the society will tend to have more of that type of movie being produced since less people will be willing to take a risk. This kind of studies will potentially lead to a less diverse movie market. We want to emphasize the importance of having diverse movie market here since it could be a conservation of culture and maintain the movie's artistic value.

The movie trend at a certain time era may also affect the revenue. For example, people now may be more into action movies, and romantic movies may become more popular after a few years. Thus, there may be bias over the time frame of the data set collected. Although we are trying to use up-to-date data, the prediction model needs to be updated in the future to predict the most accurate results that reflect the trend of that particular time.

## 15 Conclusion & Discussion

After all, we concluded that the budget, popularity, and vote_average are the most influential factors to a movie's revenue. Throughout the process of this project, we first investigate each variable individually in order to do a brief filter for the possible components. Then we selected those with higher p-values to perform a multi-variable regression to further investigate the correlation between these variables and movie revenue. We selected average vote, budget, and popularity in our multiple linear regression model. The result shows that they had a positive correlation with the revenue and the prediction accuracy score on the training data was 63.7%. However, the model drastically failed with our test data showing a highly negative R^2 value. From this result, we thought a linear regression may not be the best regression to describe the relationship. Thus, we proceed to use a non-linear regression method and the result is exciting. Our results suggest that a non-linear model, in this case, a RandomForest Regression model does a much better job of predicting the movie revenue than linear model even with "test-data" as seen through our 10-fold cross validation method.

If we continue on this project, our next step would be try to introduce a way to take genre into account for regressions based on multiple variables. In reality, we believe that a movie's genre would take a significant role on determine a movie's final revenue (as shown from the high revenue from Avenger seires). We didn't analyze the genre portion this time because we couldn't find a way to build a classification on genre since there were so many combination in there. It was hard to find an appropriate dataset that contains enough information to allow us compare different combinations. Another future direction would taking inflation into account in the model. This would cause the result a lot more reliable for any year in history. Of course, we could look into more robust non-linear model to produce a better model as well.

Overall, it was interesting to see that without genre, a budget, popularity, and vote_average are the most influential components for a movie's revenue. This indicates for future film investiment, one should first find a reputatble film industry to be reponsible for production (which should bring popularity at the time of probaganda). Budget wise, it may be explained by the fact that generally the more budget a movie has, the better quality it would be. A good voting score would help to gain the popularity as well. Surprisingly, the running time isn't an influential factor for a movie. This may be due to the fact that most movie are between 1.5~2 hours. But also it could be an indication that a movie's quality could overcome the uncomfotableness of sitting on the same seat for a long time (e.g. lord of the ring)