Investigate_a_Dataset

February 17, 2021

1 TMDB 5000 Movie Dataset- Genres- The higher the ranking the more revenue

Does the ranking of a movie have a direct correlation to how much budget there was for the movie? Which Genre has made the most revenue in 2015?

Introduction: The data being used in this project is TMDB 5000 Movie Data set provided by Kaggle. Each of the columns are fairly straight forward to understand apart from budget adj and revenue adj. These columns are the adjusted values of budget and revenue after inflation however we will not be using these values since we are just looking for the overall revenue and budget for our two unanswered questions.

```
In [12]: import pandas as pd
         import numpy as np
         import datetime as dt
         import matplotlib.pyplot as plt
         import seaborn as sns
         % matplotlib inline
         df_movies = pd.read_csv('tmdb-movies.csv')
```

Data Wrangling

Now that we have uploaded the data as well as all the packages necessary it is time to take a look at the actual data and see what information we need to answer our questions.

```
In [13]: df_movies.head()
```

```
Out[13]:        id    imdb_id  popularity     budget     revenue  \
         0  135397  tt0369610   32.985763  150000000  1513528810
         1   76341  tt1392190   28.419936  150000000   378436354
         2  262500  tt2908446   13.112507  110000000   295238201
         3  140607  tt2488496   11.173104  200000000  2068178225
         4  168259  tt2820852    9.335014  190000000  1506249360

                          original_title  \
         0                Jurassic World
         1            Mad Max: Fury Road
         2                     Insurgent
         3  Star Wars: The Force Awakens
```

```
4                    Furious 7

                                                cast  \
0  Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi...
1  Tom Hardy|Charlize Theron|Hugh Keays-Byrne|Nic...
2  Shailene Woodley|Theo James|Kate Winslet|Ansel...
3  Harrison Ford|Mark Hamill|Carrie Fisher|Adam D...
4  Vin Diesel|Paul Walker|Jason Statham|Michelle ...

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

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

                                            overview runtime  \
0  Twenty-two years after the events of Jurassic ...     124
1  An apocalyptic story set in the furthest reach...     120
2  Beatrice Prior must confront her inner demons ...     119
3  Thirty years after defeating the Galactic Empi...     136
4  Deckard Shaw seeks revenge against Dominic Tor...     137

                                      genres  \
0  Action|Adventure|Science Fiction|Thriller
1  Action|Adventure|Science Fiction|Thriller
2         Adventure|Science Fiction|Thriller
3   Action|Adventure|Science Fiction|Fantasy
4                      Action|Crime|Thriller

                                production_companies release_date vote_count  \
0  Universal Studios|Amblin Entertainment|Legenda...       6/9/15       5562
1  Village Roadshow Pictures|Kennedy Miller Produ...      5/13/15       6185
2  Summit Entertainment|Mandeville Films|Red Wago...      3/18/15       2480
3          Lucasfilm|Truenorth Productions|Bad Robot     12/15/15       5292
4  Universal Pictures|Original Film|Media Rights ...       4/1/15       2947

   vote_average  release_year    budget_adj   revenue_adj
0           6.5          2015  1.379999e+08  1.392446e+09
1           7.1          2015  1.379999e+08  3.481613e+08
2           6.3          2015  1.012000e+08  2.716190e+08
```

```
3           7.5          2015  1.839999e+08  1.902723e+09
4           7.3          2015  1.747999e+08  1.385749e+09

[5 rows x 21 columns]
```

In [14]:
```
df_movies.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10866 entries, 0 to 10865
Data columns (total 21 columns):
id                      10866 non-null int64
imdb_id                 10856 non-null object
popularity              10866 non-null float64
budget                  10866 non-null int64
revenue                 10866 non-null int64
original_title          10866 non-null object
cast                    10790 non-null object
homepage                2936 non-null object
director                10822 non-null object
tagline                 8042 non-null object
keywords                9373 non-null object
overview                10862 non-null object
runtime                 10866 non-null int64
genres                  10843 non-null object
production_companies    9836 non-null object
release_date            10866 non-null object
vote_count              10866 non-null int64
vote_average            10866 non-null float64
release_year            10866 non-null int64
budget_adj              10866 non-null float64
revenue_adj             10866 non-null float64
dtypes: float64(4), int64(6), object(11)
memory usage: 1.7+ MB
```

Looking at the data we can see that there are 21 columns and 10866 rows. Since the columns that are necessary for our two burning questions are budget, revenue, genres, release_year lets take a look to see which values are missing from the genres column

In [15]:
```
df_movies[df_movies.genres.isnull()]
```

Out[15]:
```
          id    imdb_id  popularity  budget  revenue  \
424   363869  tt4835298    0.244648       0        0
620   361043  tt5022680    0.129696       0        0
997   287663        NaN    0.330431       0        0
1712   21634  tt1073510    0.302095       0        0
1897   40534  tt1229827    0.020701       0        0
2370  127717  tt1525359    0.081892       0        0
2376  315620  tt1672218    0.068411       0        0
```

```
2853    57892  tt0270053    0.130018       0        0
3279    54330  tt1720044    0.145331       0        0
4547   123024  tt2305700    0.520520       0        0
4732   139463  tt2084977    0.235911       0        0
4797   369145        NaN    0.167501       0        0
4890   126909  tt2219564    0.083202       0        0
5830   282848  tt2986512    0.248944       0        0
5934   200204  tt2808968    0.067433       0        0
6043   190940  tt2797242    0.039080       0        0
6530   168891  tt0818519    0.092724       0        0
8234    56804  tt0114844    0.028874       0        0
8614    65595  tt0117880    0.273934       0        0
8878    92208  tt0250593    0.038045       0        0
9307   141859  tt0097446    0.094652       0        0
9799    48847  tt0193716    0.175008       0        0
10659    4255  tt0065904    0.344172    5000        0

                                          original_title  \
424                                        Belli di papÃă
620                                     All Hallows' Eve 2
997                   Star Wars Rebels: Spark of Rebellion
1712                                     Prayers for Bobby
1897                 Jonas Brothers: The Concert Experience
2370                                        Freshman Father
2376                            Doctor Who: A Christmas Carol
2853                                              Vizontele
3279                                          ìêÿřì ë
4547    London 2012 Olympic Opening Ceremony: Isles of...
4732                                          The Scapegoat
4797                                Doctor Who: The Snowmen
4890                             Cousin Ben Troop Screening
5830                      Doctor Who: The Time of the Doctor
5934                                           Prada: Candy
6043                                         Bombay Talkies
6530                                            Saw Rebirth
8234                                        Viaggi di nozze
8614                            T2 3-D: Battle Across Time
8878                         Mom's Got a Date With a Vampire
9307                                              Goldeneye
9799                                            The Amputee
10659                           The Party at Kitty and Stud's

                                                    cast  \
424     Diego Abatantuono|Matilde Gioli|Andrea Pisani|...
620                                                   NaN
997     Freddie Prinze Jr.|Vanessa Marshall|Steve Blum...
1712    Ryan Kelley|Sigourney Weaver|Henry Czerny|Dan ...
1897    Nick Jonas|Joe Jonas|Kevin Jonas|John Lloyd Ta...
```

```
2370     Britt Irvin|Merrilyn Gann|Barbara Tyson|Anthon...
2376     Matt Smith|Karen Gillan|Arthur Darvill|Michael...
2853     YÄślmaz ErdoÄan|Demet Akbag|Altan Erkekli|Cem...
3279              Jang Keun-suk|Song Ha-yoon|Kim Jeong-Nan
4547     Queen Elizabeth II|Mike Oldfield|Kenneth Brana...
4732     Andrew Scott|Jodhi May|Eileen Atkins|Matthew R...
4797     Matt Smith|Jenna Coleman|Richard E. Grant|Ian ...
4890                                   Jason Schwartzman
5830                                Matt Smith|Jenna Coleman
5934              Peter Gadiot|Rodolphe Pauly|LÃĽa Seydoux
6043     Aamir Khan|Rani Mukerji|Randeep Hooda|Saqib Sa...
6530     Whit Anderson|Stan Kirsch|Jeff Shuter|George W...
8234     Carlo Verdone|Claudia Gerini|Veronica Pivetti|...
8614     Arnold Schwarzenegger|Linda Hamilton|Edward Fu...
8878     Matt O'Leary|Laura Vandervoort|Myles Jeffrey|C...
9307     Charles Dance|Phyllis Logan|Patrick Ryecart|La...
9799                    Catherine E. Coulson|David Lynch
10659    Sylvester Stallone|Henrietta Holm|Nicholas War...

                                                homepage  \
424                                                  NaN
620                                                  NaN
997                                                  NaN
1712                      http://www.prayersforbobby.com/
1897                                                 NaN
2370                                                 NaN
2376                                                 NaN
2853                                                 NaN
3279                                                 NaN
4547                          http://www.london2012.com/
4732     http://www.island-pictures.co.uk/extras/the-sc...
4797                                                 NaN
4890     http://www.funnyordie.com/videos/fc132ce8b2/co...
5830                                                 NaN
5934                                                 NaN
6043     http://en.wikipedia.org/wiki/Bombay_Talkies_%2...
6530                                                 NaN
8234                                                 NaN
8614                                                 NaN
8878                                                 NaN
9307                                                 NaN
9799                                                 NaN
10659                                                NaN

                                                director  \
424                                          Guido Chiesa
620      Antonio Padovan|Bryan Norton|Marc Roussel|Ryan...
997                                 Steward Lee|Steven G. Lee
```

```
1712                                     Russell Mulcahy
1897                                     Bruce Hendricks
2370                                       Michael Scott
2376                                                 NaN
2853                                    YÄślmaz ErdoÄan
3279                                       Kim Jin-Yeong
4547                                         Danny Boyle
4732                                   Charles Sturridge
4797                                                 NaN
4890                                        Wes Anderson
5830                                         James Payne
5934                          Wes Anderson|Roman Coppola
6043   Anurag Kashyap|Dibakar Banerjee|Zoya Akhtar|Ka...
6530                           Jeff Shuter|Daniel Viney
8234                                       Carlo Verdone
8614                                       James Cameron
8878                                         Steve Boyum
9307                                            Don Boyd
9799                                         David Lynch
10659                                        Morton Lewis

                                             tagline     ...        \
424                                                  NaN     ...
620                                                  NaN     ...
997                                                  NaN     ...
1712   Before you echo "amen" in your home and place ...     ...
1897                                                 NaN     ...
2370                                                 NaN     ...
2376                                                 NaN     ...
2853                                                 NaN     ...
3279                                                 NaN     ...
4547                               Inspire a generation.     ...
4732                                                 NaN     ...
4797                                                 NaN     ...
4890                                                 NaN     ...
5830                        A change is going to come...     ...
5934                                               short     ...
6043                                                 NaN     ...
6530     Somewhere... Somehow... Something went wrong...     ...
8234                                                 NaN     ...
8614                                                 NaN     ...
8878                                                 NaN     ...
9307                                                 NaN     ...
9799                                                 NaN     ...
10659                                                NaN     ...

                                            overview runtime  genres  \
424    Italian remake of the Mexican 2013 hit, "We th...     100     NaN
```

620	A woman finds a VHS tape on her doorstep that ...	90	NaN
997	A Long Time Ago In A Galaxy Far, Far Awayâę A...	44	NaN
1712	True story of Mary Griffith, gay rights crusad...	88	NaN
1897	Secure your VIP pass to a once-in-a-lifetime e...	76	NaN
2370	NaN	0	NaN
2376	Amy Pond and Rory Williams are trapped on a cr...	62	NaN
2853	The story takes place in a small town (called ...	110	NaN
3279	Joon-soo (Jang Geun -Seok) is a rebellious hig...	96	NaN
4547	The London 2012 Olympic Games Opening Ceremony...	220	NaN
4732	Set in 1952, as England prepares for the coron...	100	NaN
4797	Christmas Eve, 1892, and the falling snow is t...	60	NaN
4890	Cousin Ben hosts a screening of Wes Anderson's...	2	NaN
5830	Orbiting a quiet backwater planet, the massed ...	60	NaN
5934	Candy is a modern chic french woman. She meets...	3	NaN
6043	One hundred years of Hindi cinema is celebrate...	127	NaN
6530	This comic, set in the world of SAW goes back ...	6	NaN
8234	Le vicessitudini di tre coppie di novelli spos...	103	NaN
8614	Three freedom fighters attack a large corporat...	12	NaN
8878	The Hansen kids are in a jam. Adam and his bes...	85	NaN
9307	Fact-based biography of James Bond author, Ian...	105	NaN
9799	A double leg amputated woman sits and writes a...	5	NaN
10659	Kitty and Stud are lovers. They enjoy a robust...	71	NaN

	production_companies	release_date	vote_count \
424	NaN	10/29/15	21
620	Ruthless Pictures|Hollywood Shorts	10/6/15	13
997	NaN	10/3/14	13
1712	Daniel Sladek Entertainment	2/27/09	57
1897	NaN	2/27/09	11
2370	NaN	6/5/10	12
2376	NaN	12/25/10	11
2853	NaN	2/2/01	12
3279	NaN	8/13/08	11
4547	BBC	7/27/12	12
4732	Island Pictures	9/9/12	12
4797	BBC Television UK	12/25/12	10
4890	NaN	1/1/12	14
5830	NaN	12/25/13	26
5934	NaN	3/25/13	27
6043	Viacom 18 Motion Pictures	5/3/13	12
6530	NaN	10/24/05	24
8234	NaN	12/15/95	44
8614	NaN	1/1/96	14
8878	Walt Disney Pictures	10/13/00	16
9307	Anglia Television	8/26/89	10
9799	NaN	1/1/74	11
10659	Stallion Releasing Inc.	2/10/70	10

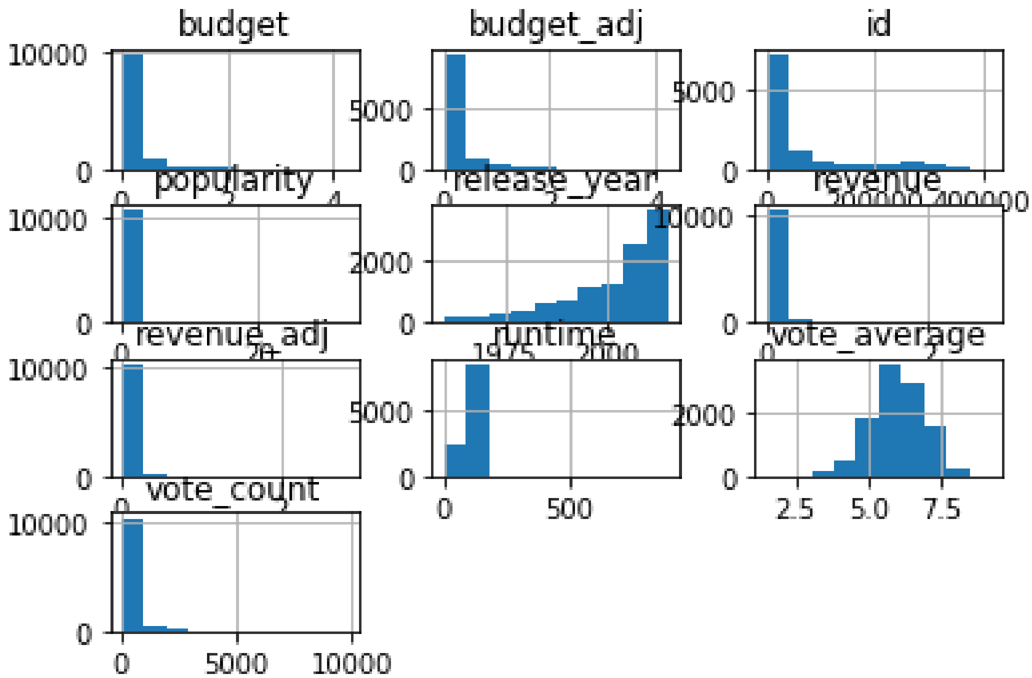
```
       vote_average  release_year   budget_adj  revenue_adj
424             6.1          2015      0.00000          0.0
620             5.0          2015      0.00000          0.0
997             6.8          2014      0.00000          0.0
1712            7.4          2009      0.00000          0.0
1897            7.0          2009      0.00000          0.0
2370            5.8          2010      0.00000          0.0
2376            7.7          2010      0.00000          0.0
2853            7.2          2001      0.00000          0.0
3279            6.1          2008      0.00000          0.0
4547            8.3          2012      0.00000          0.0
4732            6.2          2012      0.00000          0.0
4797            7.8          2012      0.00000          0.0
4890            7.0          2012      0.00000          0.0
5830            8.5          2013      0.00000          0.0
5934            6.9          2013      0.00000          0.0
6043            5.9          2013      0.00000          0.0
6530            5.9          2005      0.00000          0.0
8234            6.7          1995      0.00000          0.0
8614            6.7          1996      0.00000          0.0
8878            5.4          2000      0.00000          0.0
9307            5.3          1989      0.00000          0.0
9799            5.0          1974      0.00000          0.0
10659           3.0          1970  28081.84172          0.0

[23 rows x 21 columns]
```

Since there are only 23 lines in the missing dataset of 10866 (<1% of the whole dataset) we can just remove these values but this will be in the data cleaning portion.

```
In [16]: df_movies.hist()
```

```
Out[16]: array([[<matplotlib.axes._subplots.AxesSubplot object at 0x7fc71c2442e8>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x7fc71da88208>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x7fc71da41278>],
                [<matplotlib.axes._subplots.AxesSubplot object at 0x7fc71d9fd198>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x7fc71d9b2588>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x7fc71d9b25c0>],
                [<matplotlib.axes._subplots.AxesSubplot object at 0x7fc71d993c88>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x7fc71d94cc88>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x7fc71d908c88>],
                [<matplotlib.axes._subplots.AxesSubplot object at 0x7fc71d965c50>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x7fc71d87a208>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x7fc71d830278>]], dtype=objec
```




```
In [17]: df_movies.describe()
```

Out[17]:

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

	vote_count	vote_average	release_year	budget_adj	revenue_adj
count	10866.000000	10866.000000	10866.000000	1.086600e+04	1.086600e+04
mean	217.389748	5.974922	2001.322658	1.755104e+07	5.136436e+07
std	575.619058	0.935142	12.812941	3.430616e+07	1.446325e+08
min	10.000000	1.500000	1960.000000	0.000000e+00	0.000000e+00
25%	17.000000	5.400000	1995.000000	0.000000e+00	0.000000e+00
50%	38.000000	6.000000	2006.000000	0.000000e+00	0.000000e+00
75%	145.750000	6.600000	2011.000000	2.085325e+07	3.369710e+07
max	9767.000000	9.200000	2015.000000	4.250000e+08	2.827124e+09

Looking at the histograms and the statistics summary we can see that budget and revenue are heavily skewed to the right and the voting average is skewed slightly to the left. The years we have in this data set are between 1960-2015, and the voting average which is key to answering one of our questions is roughly 6 but ranges from 1.5-9.2. The assumption is the voting average is out of 10

1.0.1 Data Cleaning - Removing nulls, and changing format taking only data that is needed

```
In [18]: df_movies = df_movies.dropna(axis=0, subset=['genres'])
```

The first thing we wanted to do is remove the null values.

```
In [19]: df_movies.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 10843 entries, 0 to 10865
Data columns (total 21 columns):
id                      10843 non-null int64
imdb_id                 10835 non-null object
popularity              10843 non-null float64
budget                  10843 non-null int64
revenue                 10843 non-null int64
original_title          10843 non-null object
cast                    10768 non-null object
homepage                2931 non-null object
director                10801 non-null object
tagline                 8037 non-null object
keywords                9368 non-null object
overview                10840 non-null object
runtime                 10843 non-null int64
genres                  10843 non-null object
production_companies    9827 non-null object
release_date            10843 non-null object
vote_count              10843 non-null int64
vote_average            10843 non-null float64
release_year            10843 non-null int64
budget_adj              10843 non-null float64
revenue_adj             10843 non-null float64
dtypes: float64(4), int64(6), object(11)
memory usage: 1.8+ MB
```

We have now removed the 23 missing values from the genres column and can move forward with making the data more readable.

```
In [21]: df_movies['genres'] = df_movies['genres'].str.replace('|',',')
         df_movies.head()
```

```
Out[21]:        id    imdb_id  popularity     budget     revenue  \
         0  135397  tt0369610   32.985763  150000000  1513528810
         1   76341  tt1392190   28.419936  150000000   378436354
         2  262500  tt2908446   13.112507  110000000   295238201
```

```
3  140607  tt2488496   11.173104  200000000  2068178225
4  168259  tt2820852    9.335014  190000000  1506249360

               original_title  \
0              Jurassic World
1          Mad Max: Fury Road
2                   Insurgent
3  Star Wars: The Force Awakens
4                   Furious 7

                                                cast  \
0  Chris Pratt,Bryce Dallas Howard,Irrfan Khan,Vi...
1  Tom Hardy,Charlize Theron,Hugh Keays-Byrne,Nic...
2  Shailene Woodley,Theo James,Kate Winslet,Ansel...
3  Harrison Ford,Mark Hamill,Carrie Fisher,Adam D...
4  Vin Diesel,Paul Walker,Jason Statham,Michelle ...

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

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

                                            overview runtime  \
0  Twenty-two years after the events of Jurassic ...     124
1  An apocalyptic story set in the furthest reach...     120
2  Beatrice Prior must confront her inner demons ...     119
3  Thirty years after defeating the Galactic Empi...     136
4  Deckard Shaw seeks revenge against Dominic Tor...     137

                                      genres  \
0  Action,Adventure,Science Fiction,Thriller
1  Action,Adventure,Science Fiction,Thriller
2         Adventure,Science Fiction,Thriller
3   Action,Adventure,Science Fiction,Fantasy
4                      Action,Crime,Thriller

                                production_companies release_date vote_count  \
0  Universal Studios|Amblin Entertainment|Legenda...       6/9/15       5562
1  Village Roadshow Pictures|Kennedy Miller Produ...      5/13/15       6185
```

```
2  Summit Entertainment|Mandeville Films|Red Wago...      3/18/15       2480
3           Lucasfilm|Truenorth Productions|Bad Robot     12/15/15       5292
4  Universal Pictures|Original Film|Media Rights ...       4/1/15       2947

   vote_average  release_year    budget_adj   revenue_adj
0           6.5          2015  1.379999e+08  1.392446e+09
1           7.1          2015  1.379999e+08  3.481613e+08
2           6.3          2015  1.012000e+08  2.716190e+08
3           7.5          2015  1.839999e+08  1.902723e+09
4           7.3          2015  1.747999e+08  1.385749e+09

[5 rows x 21 columns]
```

Due to the readability in the genres column it is important to remove the | and replace it with commas. Lets now create two seperate dataframes to answer our two burning questions: is the rating of a movie directly correlated to how much budget is put towards a movie? and which genres have produced the most revenue in 2015?

```
In [22]: df_movies1 = df_movies[['budget', 'original_title', 'vote_average']]
```

```
In [23]: df_movies2 = df_movies[['genres', 'revenue', 'release_year' ]]
```

1.0.2 Is the rating of a movie directly correlated to how much budget is put towards the movie? (Hypothesis: the greater the budget the better the rating)

```
In [24]: df_movies1.head()
```

```
Out[24]:       budget                original_title  vote_average
         0  150000000                Jurassic World           6.5
         1  150000000            Mad Max: Fury Road           7.1
         2  110000000                     Insurgent           6.3
         3  200000000  Star Wars: The Force Awakens           7.5
         4  190000000                     Furious 7           7.3
```

```
In [25]: df_movies1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 10843 entries, 0 to 10865
Data columns (total 3 columns):
budget            10843 non-null int64
original_title    10843 non-null object
vote_average      10843 non-null float64
dtypes: float64(1), int64(1), object(1)
memory usage: 338.8+ KB
```

it is always good to double check and see that all columns have values, since we are exploring whether the voting average and budget are correlated, we need to sort the voting average descending.

```
In [26]: df_movies1= df_movies1.sort_values('vote_average',
                                            ascending=False)
```

```
In [27]: df_movies1.head(2)
```

```
Out[27]:       budget                  original_title  vote_average
         3894       0  The Story of Film: An Odyssey           9.2
         538        0              The Mask You Live In           8.9
```

we want to get a better view so lets take all movies ranked over 8 with their budget to see if the budget and voting are correlated

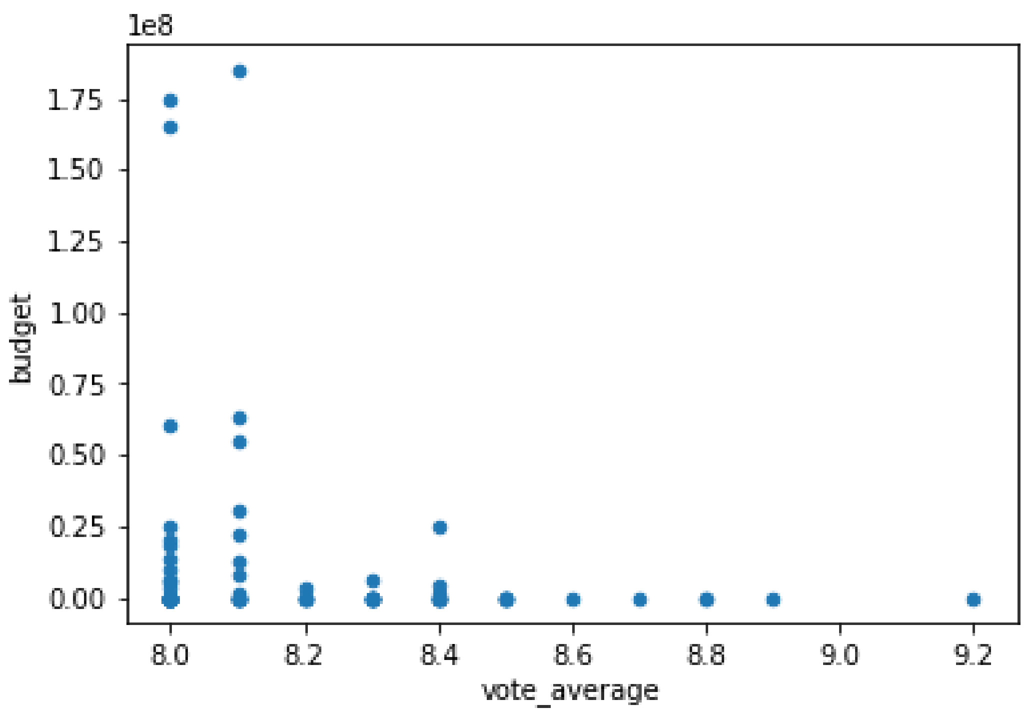
```
In [28]: df_movies1 = df_movies1[df_movies1.vote_average >= 8.0]
         df_movies1.head(5)
```

```
Out[28]:       budget                  original_title  vote_average
         3894       0  The Story of Film: An Odyssey           9.2
         538        0              The Mask You Live In           8.9
         1200       0  Black Mirror: White Christmas           8.8
         2269       0                       Life Cycles           8.8
         6911       0                Pink Floyd: Pulse           8.7
```

We know that the budget is our independant variable and the voting average is the dependant variable, it is good to see a visual of how our data is distributed.

```
In [32]: df_movies1.plot(x='vote_average', y='budget', kind='scatter')
```

```
Out[32]: <matplotlib.axes._subplots.AxesSubplot at 0x7fc71d6ae0b8>
```



Since the data does not show the distribution, let's remove all budgets that are zero.

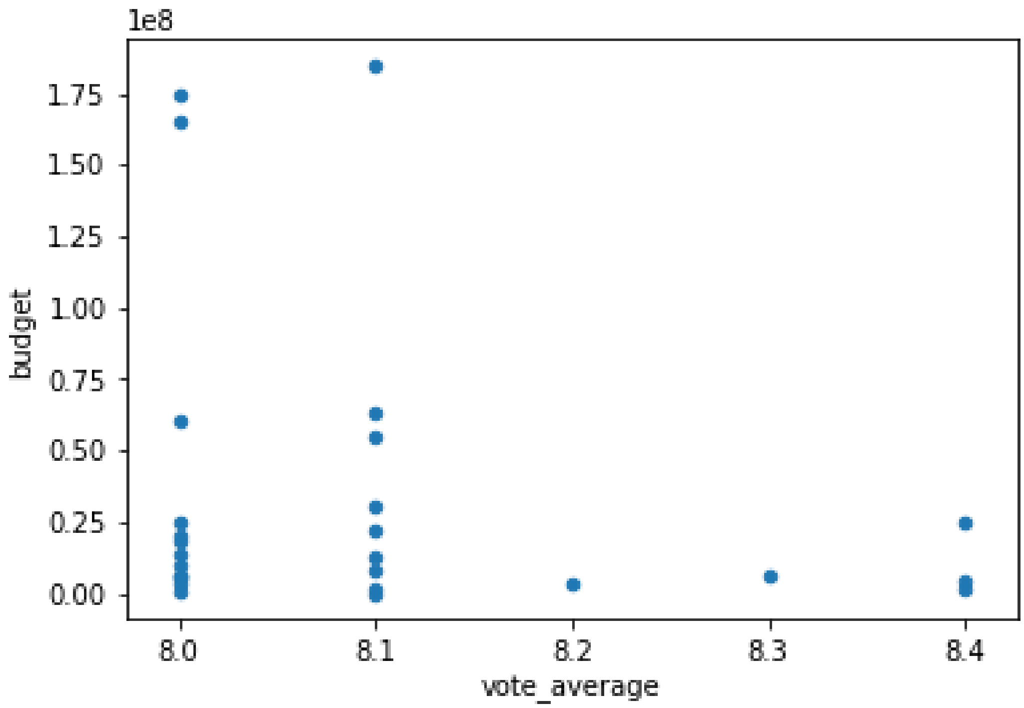
```
In [36]: df_movies1 = df_movies1[df_movies1.budget != 0]
```

```
In [39]: df_movies1.head(20)
```

```
Out[39]:          budget                   original_title  vote_average
         4178   25000000         The Shawshank Redemption           8.4
         7948    1200000                Stop Making Sense           8.4
         5986    4000000                 Guten Tag, RamÃşn           8.4
         7269    6000000                    The Godfather           8.3
         650     3300000                         Whiplash           8.2
         2409   63000000                       Fight Club           8.1
         3826   30000000  Kill Bill: The Whole Bloody Affair        8.1
         4946        110      Bones Brigade: An Autobiography       8.1
         8043    1100000        Michael Jackson's Thriller          8.1
         9758   13000000           The Godfather: Part II           8.1
         2875  185000000                  The Dark Knight           8.1
         4179   55000000                     Forrest Gump           8.1
         10222  22000000                 Schindler's List           8.1
         4177    8000000                     Pulp Fiction           8.1
         9979   25000000                       Goodfellas           8.0
         5914   10000000        One Direction: This Is Us           8.0
         7309   18000000          The Empire Strikes Back           8.0
         9805    3000000  One Flew Over the Cuckoo's Nest           8.0
         9     175000000                       Inside Out           8.0
         718     4900000                            Mommy           8.0
```

```
In [38]: df_movies1.plot(x='vote_average', y='budget', kind='scatter')
```

```
Out[38]: <matplotlib.axes._subplots.AxesSubplot at 0x7fc71c74a3c8>
```



As we can see above in our table, it is clear that budget is not a direct reflection on the vote average. As we can see the max budget paid for a movie at $185000000 for the Dark Knight had a lower voting average than other movies that were voted higher with a lower budget.

1.0.3 Research Question 2 Which top genres of movies that were released had the most revenue?

In [40]:
```
df_movies2.head()
```

Out[40]:

	genres	revenue	release_year
0	Action,Adventure,Science Fiction,Thriller	1513528810	2015
1	Action,Adventure,Science Fiction,Thriller	378436354	2015
2	Adventure,Science Fiction,Thriller	295238201	2015
3	Action,Adventure,Science Fiction,Fantasy	2068178225	2015
4	Action,Crime,Thriller	1506249360	2015

Lets have revenue sorted from largest to smallest

In [41]:
```
df_movies2= df_movies2.sort_values('revenue', ascending=False)
df_movies2.head()
```

Out[41]:

	genres	revenue	release_year
1386	Action,Adventure,Fantasy,Science Fiction	2781505847	2009
3	Action,Adventure,Science Fiction,Fantasy	2068178225	2015
5231	Drama,Romance,Thriller	1845034188	1997
4361	Science Fiction,Action,Adventure	1519557910	2012
0	Action,Adventure,Science Fiction,Thriller	1513528810	2015

Now that we have the columns we need as well as the sorted revenue largest to smallest, it is time to filter the year to 2015 to see which genre had earned the most money.

```
In [42]: df_movies2=df_movies2[df_movies2.release_year ==2015]
```

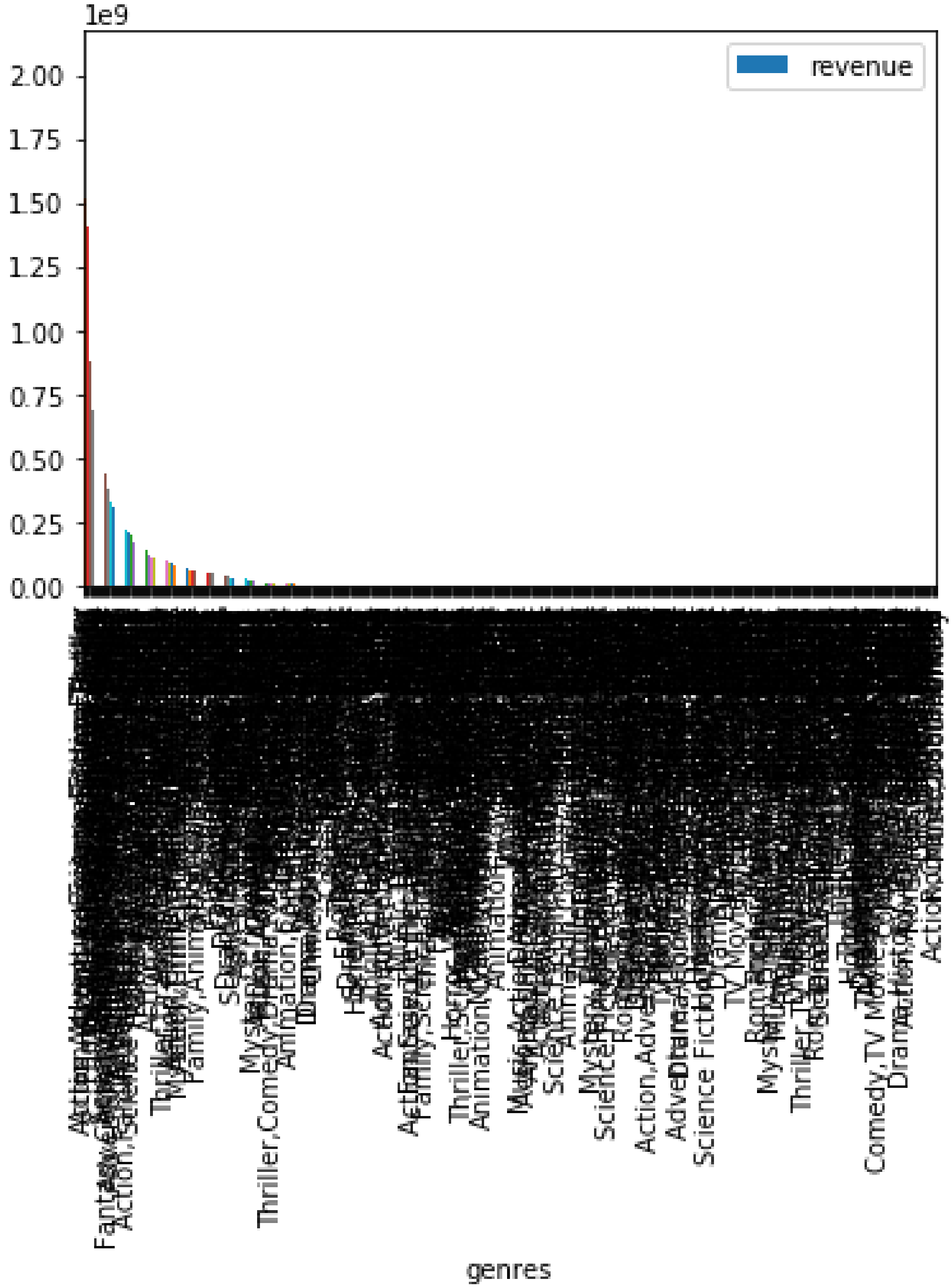
```
In [52]: df_movies2.head()
```

```
Out[52]:                                    genres     revenue  release_year
         3    Action,Adventure,Science Fiction,Fantasy  2068178225          2015
         0   Action,Adventure,Science Fiction,Thriller  1513528810          2015
         4                       Action,Crime,Thriller  1506249360          2015
         14           Action,Adventure,Science Fiction  1405035767          2015
         8            Family,Animation,Adventure,Comedy  1156730962          2015
```

Now that we have all of our necessary data we can do a visualization to see which genres have pulled in the most revenue in 2015

```
In [53]: df_movies2.plot(x='genres', y='revenue', kind='bar')
```

```
Out[53]: <matplotlib.axes._subplots.AxesSubplot at 0x7fc71250d1d0>
```



The top genres in 2015 that produced the most revenue are: Action, Adventure, Fantasy, Science Fiction pulling in roughly 2 billion dollars. The visualization is difficult to read due to the genres having multiple different genres within the title, however the table answers which are the top genres of 2015. Just for fun the worst genres of 2015 are:

```
In [54]: df_movies2.tail()
```

Out[54]:

	genres	revenue	release_year
501	Drama	0	2015
502	Romance,Drama,Music	0	2015
503	Drama	0	2015
504	Music,Documentary	0	2015
505	Action,Crime,Drama,Thriller	0	2015

Conclusions The conclusions to my presentation are that budget and vote average are not in correlation even though the assumption was that if there is more budget towards a film the better the voting average will be. This was not a true hypothesis. It is clear that even though the Dark Knight had the largest budget it was still not the highest in votes.

The second observation was that in 2015 the top genres were Action Adventure and the least favourite was Drama.

1.1 Submitting your Project

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

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

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

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

Out[ ]: 0

In [ ]: