raw_template_v3

June 30, 2019

1 Project 2: Investigating The Movies Industry Through Budgeting, Genres and Fiscal Quarters In Relation to Revenue.

1.1 Table of Contents

## Introduction

According to Wikipedia, and by 2018 estimates, the global movie industry is worth $136 billion dollars (USD). Consequently, the movie industry is continuously attempting to understand any and all factors which may influence movie viewing. The better industry officials understand these influences, the better chance they have to control and manipulate the most influential factors. Of course, this leads to securing, and potentially increasing, their initial capital investment through larger revenues.

This project investigates three common movie variables (budget, genre and release date) seeking to find whether any of these variables show a relationship to a movie's overall box office revenue. The data is culled from over 10,000 movies catorgized by the Internet Movie Database (IMDB). Please note, this analysis is for the purpose of showing what can be accomplished using Jupyter notebooks, Pandas, Numpy, Matplotlib and a general knowledge of statiscal analysis. Per Kaggle (https://www.kaggle.com/tmdb/tmdb-movie-metadata), the data is mostly self-reported and isn't verified. For our purposes here, we'll disregard that and move forward under the assumption of its veracity.

Using an initial broadstroke of descriptive statistics, we can comfortably describe the data we'll be extracting. We must, of course, be cautious not to over-extend our analysis. An indepth analysis of this data would be better conducted using inferential statistics be that through traditional methods or through the use of machine learning. Still, much can be gleaned from an inital foray into this subject via descriptive statiscal analysis alone.

The project begins by importing the necessary packages required for analysis.

[59]:
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

## Data Wrangling

Data wrangling begins by loading the CSV data file from IMDB into a dataframe that will subsequently be used for cleaning.

1.1.1 General Properties

A new dataframe (df) is created and the first few lines are pulled for observation.

[2]:

```
df_mov = pd.read_csv('tmdb-movies.csv')
df_mov.head(3)
```

[2]:

```
       id    imdb_id  popularity     budget     revenue      original_title  \
0  135397  tt0369610   32.985763  150000000  1513528810      Jurassic World
1   76341  tt1392190   28.419936  150000000   378436354  Mad Max: Fury Road
2  262500  tt2908446   13.112507  110000000   295238201           Insurgent

                                                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...

                                         homepage          director  \
0                   http://www.jurassicworld.com/   Colin Trevorrow
1                     http://www.madmaxmovie.com/     George Miller
2  http://www.thedivergentseries.movie/#insurgent  Robert Schwentke

                      tagline  ...  \
0           The park is open.  ...
1          What a Lovely Day.  ...
2  One Choice Can Destroy You  ...

                                            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

                                      genres  \
0  Action|Adventure|Science Fiction|Thriller
1  Action|Adventure|Science Fiction|Thriller
2         Adventure|Science Fiction|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

   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 rows x 21 columns]
```

The next few lines of code speak to the shape (in rows & columns) and data types of the df.

[3]:
```
df_mov.shape
```

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

[4]:
```
df_mov.dtypes
```

[4]:
```
id                        int64
imdb_id                  object
popularity              float64
budget                    int64
revenue                   int64
original_title           object
cast                     object
homepage                 object
director                 object
tagline                  object
keywords                 object
overview                 object
runtime                   int64
genres                   object
production_companies     object
release_date             object
vote_count                int64
vote_average            float64
release_year              int64
budget_adj              float64
revenue_adj             float64
dtype: object
```

The "budget" and "revenue" columns are already integers; "genres" and "release_date" require a deeper investigation.

[5]:
```
type(df_mov['genres'][0])
```

[5]:
```
str
```

[6]:
```
type(df_mov['release_date'][0])
#if str, will change to datetime while data cleaning
```

```
[6]: str
```

During data cleaning release_date will be changed from string to datetime.

1.1.2 Data Cleaning

This code changes release_date from str to datetime. This will be useful during the investigation of the 3rd research question that follows.

```
[8]: df_mov['release_date'] = pd.to_datetime(df_mov['release_date'])
     df_mov.dtypes
```

```
[8]: id                                int64
     imdb_id                          object
     popularity                      float64
     budget                            int64
     revenue                           int64
     original_title                   object
     cast                             object
     homepage                         object
     director                         object
     tagline                          object
     keywords                         object
     overview                         object
     runtime                           int64
     genres                           object
     production_companies             object
     release_date             datetime64[ns]
     vote_count                        int64
     vote_average                    float64
     release_year                      int64
     budget_adj                      float64
     revenue_adj                     float64
     dtype: object
```

The next set of code summarizes how many duplicate rows there were, then removes the duplicates in place and verifies that they were removed.

```
[9]: sum(df_mov.duplicated())
```

```
[9]: 1
```

```
[10]: df_mov.drop_duplicates(inplace=True)
```

```
[11]: sum(df_mov.duplicated())
```

```
[11]: 0
```

These lines of code identify null values in the df. They also will drop the null values within specific columns.

[12]:
```
df_mov.shape
```

[12]:
```
(10865, 21)
```

[13]:
```
df_mov.isnull().sum()
```

[13]:
```
id                         0
imdb_id                   10
popularity                 0
budget                     0
revenue                    0
original_title             0
cast                      76
homepage                7929
director                  44
tagline                 2824
keywords                1493
overview                   4
runtime                    0
genres                    23
production_companies    1030
release_date               0
vote_count                 0
vote_average               0
release_year               0
budget_adj                 0
revenue_adj                0
dtype: int64
```

[15]:
```
df_mov.dropna(subset=['genres'], inplace=True)
```

[16]:
```
df_mov.dropna(subset=['director'], inplace=True)
```

[17]:
```
df_mov.isnull().sum()
```

[17]:
```
id                         0
imdb_id                    5
popularity                 0
budget                     0
revenue                    0
original_title             0
cast                      69
homepage                7879
director                   0
tagline                 2774
keywords                1449
overview                   2
runtime                    0
genres                     0
production_companies     994
```

```
release_date               0
vote_count                 0
vote_average               0
release_year               0
budget_adj                 0
revenue_adj                0
dtype: int64
```

To make the df more viewable on screen, extraneous columns unnecessary to the research project will be dropped.

[18]:
```
for i, v in enumerate(df_mov.columns):
    print(i,v)
```

```
0 id
1 imdb_id
2 popularity
3 budget
4 revenue
5 original_title
6 cast
7 homepage
8 director
9 tagline
10 keywords
11 overview
12 runtime
13 genres
14 production_companies
15 release_date
16 vote_count
17 vote_average
18 release_year
19 budget_adj
20 revenue_adj
```

This drops the columns determined to be unnecessary from above

[19]:
```
df_mov.drop(['popularity', 'cast', 'homepage', 'tagline', 'keywords',␣
 ↪'overview', 'vote_count', 'vote_average'], axis=1, inplace=True)
```

[20]:
```
for i, v in enumerate(df_mov.columns):
    print(i,v)
```

```
0 id
1 imdb_id
2 budget
```

```
3 revenue
4 original_title
5 director
6 runtime
7 genres
8 production_companies
9 release_date
10 release_year
11 budget_adj
12 revenue_adj
```

This is the final shape of the df after data cleaning. Intially, it had 10866 rows and 21 columns. It has been brought down to 10800 rows and 13 columns.

[21]:
```
df_mov.shape
```

[21]:
```
(10800, 13)
```

## Exploratory Data Analysis

1.1.3 Question 1: Do the top 25% Budgeted Movies correlate to the top 25% Highest Revenued Movies?

The research begins by statistically describing the budget column, paying attention for the 75th percentile. From this point to the maximum is where the top 25% of the budgeted amounts lie.

[22]:
```
# Budget: describe the 75th percertile
df_mov.describe().budget
```

[22]:
```
count    1.080000e+04
mean     1.471114e+07
std      3.098677e+07
min      0.000000e+00
25%      0.000000e+00
50%      0.000000e+00
75%      1.564374e+07
max      4.250000e+08
Name: budget, dtype: float64
```

Creating a new "over/under" column of the 75th Budgeted percentile allows for the groupby function to be used further along. This code creates the edges of the bins of the new column. There will be three edges and two categories in the new column.

[23]:
```
#Budget: create over/under 75th percentile column
#Budget: Bin edges to cut the data into groups
#budget_bin_edges = [0.0, 1.564374e+07, 4.250000e+08]; 0 & 0.0 don't capture␣
 ↪info as bin edge, try -1.
budget_bin_edges = [-1, 15643740, 425000000]
```

These are the two categories ("under" & "over") within the new column

[24]:
```
#Budget: create over/under 75th percentile column
#Budget: options for new column
budget_bin_names = ['under', 'over']
```

This combines the above column elements together and creates the new column based on those elements. The column will be titled "75th_%tile_budget".

[25]:
```
#Budget: create over/under 75th percentile column
#Budget: create the column
df_mov['75th_%tile_budget'] = pd.cut(df_mov['budget'], budget_bin_edges,␣
 ↪labels=budget_bin_names)
```

The exact same steps above are taken to create another new column describing revenue rather than budget. Once created, this new column will be titled "75th_%tile_revenue".

[26]:
```
# Revenue: describe the 75th percertile
df_mov.describe().revenue
```

[26]:
```
count    1.080000e+04
mean     4.006558e+07
std      1.173193e+08
min      0.000000e+00
25%      0.000000e+00
50%      0.000000e+00
75%      2.455409e+07
max      2.781506e+09
Name: revenue, dtype: float64
```

[27]:
```
#Revenue: create over/under 75th percentile column
#Revenue: Bin edges to cut the data into groups
#revenue_bin_edges = [0.0, 2.455409e+07, 2.781506e+09]
revenue_bin_edges = [-1, 24554090, 2781506000]
```

[28]:
```
#Revenue: create over/under 75th percentile column
#Revenue: options for new column
revenue_bin_names = ['under', 'over']
```

[29]:
```
#Revenue: create over/under 75th percentile column
#Revenue: create the column
df_mov['75th_%tile_revenue'] = pd.cut(df_mov['revenue'], revenue_bin_edges,␣
 ↪labels=revenue_bin_names)
```

The two new columns have been created and are shown below.

[31]:
```
#Budget & Revenue: show two new 75th percentile columns
df_mov.head(3)
```

[31]:
```
       id    imdb_id     budget     revenue      original_title  \
0  135397  tt0369610  150000000  1513528810      Jurassic World
```

```
1   76341  tt1392190  150000000   378436354  Mad Max: Fury Road
2  262500  tt2908446  110000000   295238201           Insurgent

           director  runtime                                     genres  \
0   Colin Trevorrow      124  Action|Adventure|Science Fiction|Thriller
1     George Miller      120  Action|Adventure|Science Fiction|Thriller
2  Robert Schwentke      119         Adventure|Science Fiction|Thriller

                                production_companies release_date  \
0  Universal Studios|Amblin Entertainment|Legenda...   2015-06-09
1  Village Roadshow Pictures|Kennedy Miller Produ...   2015-05-13
2  Summit Entertainment|Mandeville Films|Red Wago...   2015-03-18

   release_year    budget_adj   revenue_adj 75th_%tile_budget  \
0          2015  1.379999e+08  1.392446e+09              over
1          2015  1.379999e+08  3.481613e+08              over
2          2015  1.012000e+08  2.716190e+08              over

  75th_%tile_revenue
0               over
1               over
2               over
```

[32]:

```
#Budget & Revenue: show two new 75th percentile columns
df_mov.tail(3)
```

[32]:

```
          id    imdb_id  budget  revenue            original_title  \
10863  39768  tt0060161        0        0       Beregis Avtomobilya
10864  21449  tt0061177        0        0    What's Up, Tiger Lily?
10865  22293  tt0060666    19000        0  Manos: The Hands of Fate

               director  runtime          genres     production_companies  \
10863    Eldar Ryazanov       94  Mystery|Comedy                  Mosfilm
10864       Woody Allen       80   Action|Comedy  Benedict Pictures Corp.
10865  Harold P. Warren       74          Horror                Norm-Iris

      release_date  release_year     budget_adj  revenue_adj  \
10863   2066-01-01          1966       0.000000          0.0
10864   2066-11-02          1966       0.000000          0.0
10865   2066-11-15          1966  127642.279154          0.0

      75th_%tile_budget 75th_%tile_revenue
10863             under              under
10864             under              under
10865             under              under
```

As a further check, there should not be any nulls in the two new columns (because they should all be filled with an "under" or an "over").

[34]:
```
#Budget & Revenue: how many nulls in new columns? s/b 0 for both
df_mov.isnull().sum()
```

[34]:
```
id                        0
imdb_id                   5
budget                    0
revenue                   0
original_title            0
director                  0
runtime                   0
genres                    0
production_companies    994
release_date              0
release_year              0
budget_adj                0
revenue_adj               0
75th_%tile_budget         0
75th_%tile_revenue        0
dtype: int64
```

two new df's are created (bud_over75_df & rev_over75_df) for analysis. They are both based on the newly created columns above.

[36]:
```
#Budget: create a new df that is only over 75% percentile of budget
bud_over75_df = df_mov[df_mov['75th_%tile_budget'].str.contains('over')]
bud_over75_df.shape
```

[36]:
```
(2700, 15)
```

[37]:
```
#Revenue: create a new df that is only over 75% percentile of revenue
rev_over75_df = df_mov[df_mov['75th_%tile_revenue'].str.contains('over')]
rev_over75_df.shape
```

[37]:
```
(2700, 15)
```

Now created, the new df's are combined into a single df.

[38]:
```
#create new df appending bud_over75 to rev_over75.
#new df will show either bud or rev over 75%
bud_or_rev_over75_df = bud_over75_df.append(rev_over75_df)
bud_or_rev_over75_df.head(3)
```

[38]:
```
       id    imdb_id     budget     revenue      original_title  \
0  135397  tt0369610  150000000  1513528810      Jurassic World
1   76341  tt1392190  150000000   378436354  Mad Max: Fury Road
2  262500  tt2908446  110000000   295238201           Insurgent

           director  runtime                                      genres  \
0  Colin Trevorrow      124  Action|Adventure|Science Fiction|Thriller
1    George Miller      120  Action|Adventure|Science Fiction|Thriller
```

```
2  Robert Schwentke      119         Adventure|Science Fiction|Thriller

                               production_companies release_date  \
0  Universal Studios|Amblin Entertainment|Legenda...   2015-06-09
1  Village Roadshow Pictures|Kennedy Miller Produ...   2015-05-13
2  Summit Entertainment|Mandeville Films|Red Wago...   2015-03-18

   release_year    budget_adj   revenue_adj 75th_%tile_budget  \
0          2015  1.379999e+08  1.392446e+09              over
1          2015  1.379999e+08  3.481613e+08              over
2          2015  1.012000e+08  2.716190e+08              over

  75th_%tile_revenue
0               over
1               over
2               over
```

[39]:

```
bud_or_rev_over75_df.shape
```

[39]:

```
(5400, 15)
```

However, upon investigation, the bud_or_rev_over75_df wasn't going to help in analysis of the research question.

What was required was a df that inlcuded the dataset of movies that were both over the budget AND over the revenue of the 75th percentile.

So, following a long Google investigation, a new df was created that was compiled from movies that were both over the budget AND the revenue of the 75th percentile. The new df of both combined was smaller (3836 rows) then the df that was budget OR revnue over the 75th percentile (5400 rows).

[40]:

```
#create new df to show both bud AND rev over 75%.
#this df should be a smaller data set than bud OR rev over 75%.
#this df will be necessary to compare budget to revenue
#a google search produced this code:
#df[(df['col_name'].str.contains('apple')) & (df['col_name'].str.
 ↪contains('banana'))]
bud_and_rev_over75_df =␣
 ↪bud_or_rev_over75_df[(bud_or_rev_over75_df['75th_%tile_revenue'].str.
 ↪contains('over')) & (bud_or_rev_over75_df['75th_%tile_budget'].str.
 ↪contains('over'))]
bud_and_rev_over75_df.shape
```

[40]:

```
(3836, 15)
```

The following statistically describes the two df's (the original all inclusive df_mov and the newly created bud_and_rev_over75_df) when viewing each df's for its budget and revenue.

[43]:
```
df_mov.describe().budget
```

[43]:
```
count    1.080000e+04
mean     1.471114e+07
std      3.098677e+07
min      0.000000e+00
25%      0.000000e+00
50%      0.000000e+00
75%      1.564374e+07
max      4.250000e+08
Name: budget, dtype: float64
```

[44]:
```
df_mov.describe().revenue
```

[44]:
```
count    1.080000e+04
mean     4.006558e+07
std      1.173193e+08
min      0.000000e+00
25%      0.000000e+00
50%      0.000000e+00
75%      2.455409e+07
max      2.781506e+09
Name: revenue, dtype: float64
```

[46]:
```
bud_and_rev_over75_df.describe().budget
```

[46]:
```
count    3.836000e+03
mean     6.153833e+07
std      4.615358e+07
min      1.600000e+07
25%      3.000000e+07
50%      4.700000e+07
75%      7.814665e+07
max      3.800000e+08
Name: budget, dtype: float64
```

[47]:
```
bud_and_rev_over75_df.describe().revenue
```

[47]:
```
count    3.836000e+03
mean     1.873410e+08
std      2.171156e+08
min      2.471922e+07
25%      5.879581e+07
50%      1.150394e+08
75%      2.228096e+08
max      2.781506e+09
Name: revenue, dtype: float64
```

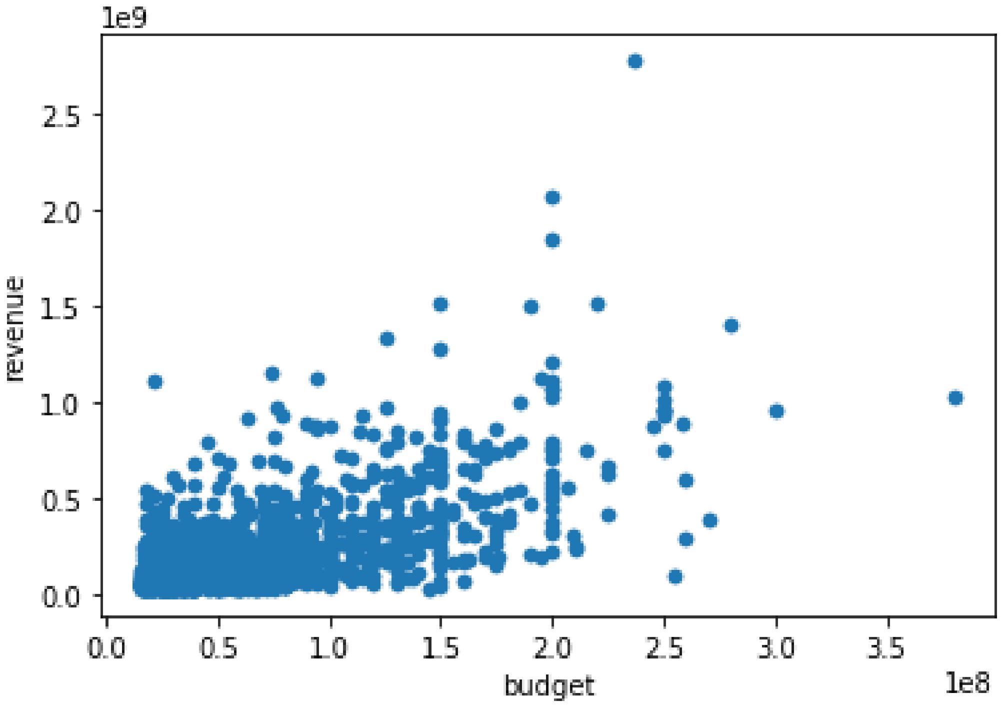
Some might use the above information and, comparing the mean of budget and revenue of each df to the other df, notice that the bud_and_rev_over75_df values are larger than the mov_df and conclude a positive response to the initial research question ("Do the top 25% Budgeted Movies correlate to the top 25% Highest Revenued Movies?").

However, to do that is faulty analysis. The mean of the budget and revenue are larger in the newly created df strictly because the new df is based on all those movies that are over the 75th percentile in budget and revenue. They have to be larger!

So, why go through the trouble of statistically describing these dataframes in the first place? It's an error checkpoint in the analysis. If the mean averages in budget & revenue for the bud_and_rev_over75_df had been less than the df_mov budget and revenue means, then the coding would have proven faulty somewhere previously and would have required further investigation before continuing.

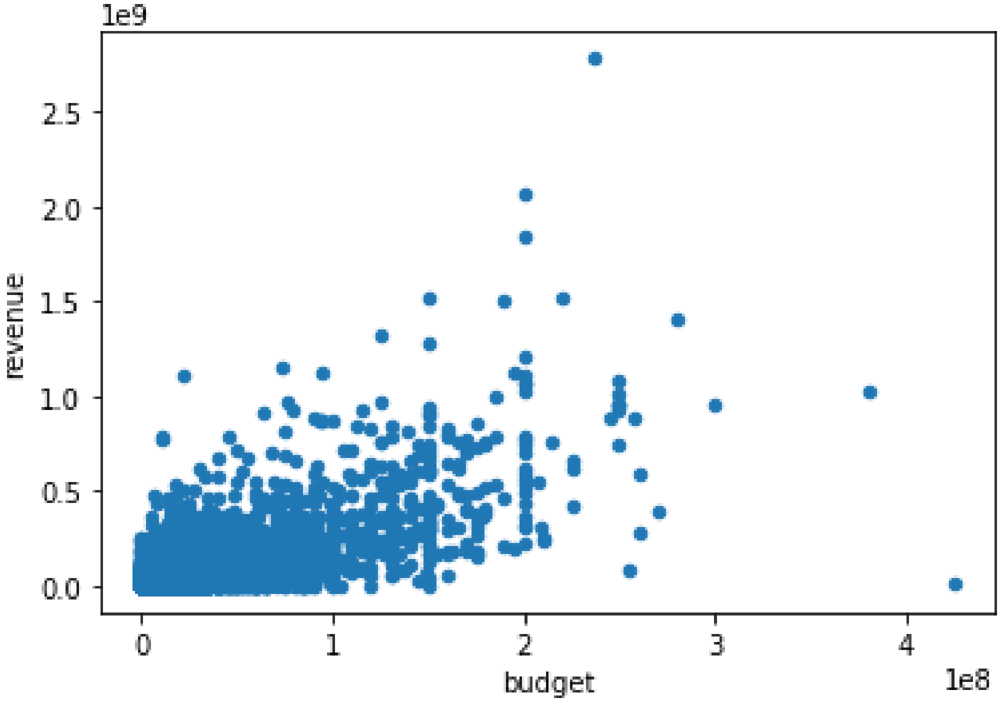
Instead of falling in the trap of faulty analysis, the focus of research for this question will be to compare budget to revenue in the over 75th percentile data set (bud_and_rev_over75_df) and then again in the overall movie data set (mov_df). Comparing a scatter plot of both data sets (focusing on determining a positive relationship and, if existent, its linearity) more accurately answers question one.

[48]:
```
#plot relationship between budget and revenue from the dataset that contains␣
 ↪movies that were
# both in the equal or over 75th percentile in budget and revenue
bud_and_rev_over75_df.plot(x='budget', y='revenue', kind='scatter');
```



[50]:
```
#plot relationship between budget and revenue from the dataset that contains␣
 ↪movies that were
# both in the equal or over 75th percentile in budget and revenue
```

```
df_mov.plot(x='budget', y='revenue', kind='scatter');
```



If the top 25% Budgeted Movies correlate to the top 25% Highest Revenued Movies then when comparing the scatter plots, there should be a marked distinction between the two plots. The plot of the top 25% should be more positive (as X increases, so does y) and should be more grouped linearly than the general movie plot. Again, bearing in mind that the top 25% data set is merely a microscoped section of the general movie data set, the former should still show a marked difference from the latter, if the research question is to be proved correct.

Upon analysis, both plots appear quite similarly. Of course, running the correlation coefficient on both data sets individually and comparing the values would conclusively show the strength the correlation of each data set. Unfortunately, that analysis is beyond the scrope of this project.

In conclusion, based upon the scatter plots, the top 25% budgeted movies don't appear to correlate to the top 25% revenued movies, though in general, there appears to be a slightly positive correlation between budget and revenue.

1.1.4 What 3 Genres are Most Represented in the Top 25% Highest Revenued Movies?

This question focuses on movie genres and their representation within the confines of high revenue movies. Research begins by calling the (previously established) revenue over 75 percentile data frame and then determining the count of the unique number of genres in the df.

[54]:
```
rev_over75_df.shape
```

[54]:
```
(2700, 15)
```

[55]:
```
#find number of genres
rev_over75_df.nunique()
```

[55]:
```
id                      2700
imdb_id                 2700
budget                   265
revenue                 2657
original_title          2664
director                1133
runtime                  123
genres                   840
production_companies    2056
release_date            2199
release_year              56
budget_adj              1470
revenue_adj             2695
75th_%tile_budget          2
75th_%tile_revenue         1
dtype: int64
```

Finding the unique count of genres was fruitful. The search result shows that there were 840 unique counts. This requires a more indepth analysis into what comprises the genre field.

[57]:
```
#840 genres! that's ridiculous.  how are those genres presented?
rev_over75_df['genres'].unique()
```

[57]:
```
array(['Action|Adventure|Science Fiction|Thriller',
       'Adventure|Science Fiction|Thriller',
       'Action|Adventure|Science Fiction|Fantasy',
       'Action|Crime|Thriller', 'Western|Drama|Adventure|Thriller',
       'Science Fiction|Action|Thriller|Adventure',
       'Drama|Adventure|Science Fiction',
       'Family|Animation|Adventure|Comedy', 'Comedy|Animation|Family',
       'Action|Adventure|Crime',
       'Science Fiction|Fantasy|Action|Adventure',
       'Drama|Science Fiction', 'Action|Comedy|Science Fiction',
       'Action|Adventure|Science Fiction', 'Crime|Drama|Mystery|Western',
       'Crime|Action|Thriller', 'Science Fiction|Action|Adventure',
       'Romance|Fantasy|Family|Drama', 'War|Adventure|Science Fiction',
       'Action|Family|Science Fiction|Adventure|Mystery', 'Action|Drama',
       'Action|Drama|Thriller', 'Drama|Romance', 'Comedy|Drama', 'Action',
       'Comedy', 'Crime|Comedy|Action|Adventure',
       'Drama|Thriller|History', 'Action|Science Fiction|Thriller',
       'Mystery|Drama', 'Crime|Action|Science Fiction', 'Comedy|Music',
       'Thriller|Drama', 'Adventure|Horror|Comedy', 'Drama|Thriller',
       'Adventure|Animation|Comedy|Family|Fantasy',
       'Action|Crime|Drama|Mystery|Thriller', 'Drama|Music',
       'Fantasy|Action|Adventure', 'History|Drama',
       'Comedy|Action|Adventure', 'Fantasy|Drama|Romance',
       'Fantasy|Comedy|Animation|Science Fiction|Family',
       'Drama|Mystery|Romance', 'Romance|Drama', 'Adventure|Drama',
       'Drama', 'Science Fiction|Mystery|Thriller', 'Comedy|Adventure',
       'Mystery|Horror', 'Action|Comedy|Crime',
       'Romance|Comedy|Crime|Drama',
       'Thriller|Drama|Adventure|Action|History', 'Drama|Thriller|War',
       'Crime|Thriller', 'Adventure|Family|Fantasy',
       'Adventure|Animation|Comedy|Family', 'Animation|Comedy|Family',
       'Adventure|Drama|Thriller', 'Adventure|Comedy', 'Horror|Thriller',
       'Thriller|Mystery', 'Adventure|Animation|Fantasy',
       'Romance|Comedy', 'Horror', 'Action|Adventure|Comedy|Family',
       'Thriller', 'Crime|Drama', 'Horror|Comedy|Fantasy',
       'Action|Thriller', 'Comedy|Action', 'Science Fiction|Thriller',
       'Thriller|Action|Crime', 'Mystery|Thriller|Fantasy|Horror|Drama',
       'Comedy|Drama|Music', 'Crime|Drama|Mystery', 'Family|Animation',
       'Adventure|Drama|Family', 'Family|Animation|Comedy|Adventure',
       'Action|Crime|Comedy', 'Drama|Horror|Thriller',
       'Drama|Science Fiction|Thriller', 'Drama|Comedy',
       'Adventure|Comedy|Romance', 'Drama|Adventure|Comedy',
       'Thriller|Horror', 'Adventure|Drama|Science Fiction',
       'Action|Science Fiction|Adventure',
       'Science Fiction|Adventure|Thriller', 'Adventure|Fantasy',
       'Adventure|Family|Animation|Action|Comedy',
       'History|Drama|Thriller|War',
       'Action|Mystery|Science Fiction|Thriller',
       'Mystery|Thriller|Drama', 'War|Drama|Action',
       'Adventure|Comedy|Fantasy|Family',
       'Action|Adventure|Fantasy|Science Fiction',
       'Action|Science Fiction',
       'Science Fiction|Action|Adventure|Fantasy|Comedy',
       'Crime|Drama|Thriller', 'Science Fiction|Action|Drama|Thriller',
       'Action|Adventure|Fantasy', 'War|Action', 'Adventure|Drama|Action',
       'Fantasy|Adventure|Action|Family|Romance',
       'Fantasy|Action|Adventure|Animation|Comedy',
       'Action|Adventure|Thriller',
       'Adventure|Fantasy|Animation|Comedy|Action',
       'Horror|Action|Drama|Fantasy|War', 'Fantasy|Comedy|Music',
       'Action|War', 'Crime|Comedy|Action',
       'Crime|Drama|Mystery|Thriller', 'Action|Adventure',
       'Drama|Adventure', 'Romance|Drama|Music', 'Action|Comedy',
       'Action|Thriller|Mystery', 'Science Fiction|Action',
       'Thriller|Science Fiction|Drama|Mystery', 'Comedy|Romance',
       'Animation|Adventure|Family', 'Animation|Adventure|Comedy|Family',
       'War|Drama|History|Action', 'Drama|War',
       'Action|Adventure|History|Romance|Drama',
       'Music|Comedy|Drama|Family',
       'Romance|Animation|Adventure|Comedy|Family',
       'Action|Crime|Drama|Thriller', 'Animation|Comedy|Family|Fantasy',
       'Thriller|Action', 'Animation|Comedy|Adventure|Family',
       'Action|Crime|Drama', 'Drama|Fantasy|Mystery|Romance',
       'Comedy|Drama|Romance', 'Family|Comedy', 'Music|Drama',
       'Family|Drama', 'Thriller|Horror|Drama',
       'Comedy|Adventure|Crime|Family',
       'Family|Adventure|Science Fiction', 'Thriller|Crime|Drama',
       'Adventure|Action|Science Fiction', 'Adventure|Action|Thriller',
       'Fantasy|Family|Animation|Adventure', 'Science Fiction|Drama',
       'Drama|History|War', 'Action|Adventure|Comedy',
       'Drama|Action|Crime|Thriller', 'Horror|Thriller|Fantasy',
       'Action|Adventure|Drama|Thriller', 'Fantasy|Comedy',
       'Drama|Action|Thriller|War',
       'Fantasy|Action|Adventure|Science Fiction|Thriller',
       'Adventure|Fantasy|Family', 'Animation|Comedy|Family|Adventure',
       'Drama|Action|Thriller|Science Fiction',
       'Action|Thriller|Science Fiction',
       'Adventure|Drama|Action|Comedy|Thriller',
       'Action|Mystery|Science Fiction',
       'Action|Adventure|Animation|Science Fiction|Thriller',
       'Romance|Family|Animation|Music', 'Comedy|Horror',
       'Adventure|Fantasy|Drama|Romance', 'Comedy|Romance|Drama',
       'Adventure|Action|Thriller|Science Fiction', 'Comedy|Mystery',
       'Animation|Family', 'Adventure|Fantasy|Action|Comedy|Family',
       'Animation|Drama', 'Fantasy|Drama', 'Drama|Family',
       'Drama|Crime|Thriller', 'Thriller|Science Fiction',
       'Action|Thriller|Crime',
       'Action|Adventure|Drama|Mystery|Science Fiction',
       'Fantasy|Action|Adventure|Family|Comedy', 'Thriller|Drama|Crime',
       'Comedy|Romance|Fantasy', 'Fantasy|Comedy|Romance',
       'Animation|Family|Adventure|Science Fiction',
       'History|Crime|Drama', 'Horror|Thriller|Mystery', 'Horror|Mystery',
       'Comedy|Family|Animation|Fantasy|Music',
       'Comedy|Crime|Drama|Thriller',
       'Adventure|Family|Fantasy|Science Fiction|Thriller',
       'Horror|Mystery|Thriller',
       'Fantasy|Horror|Action|Thriller|Science Fiction',
       'Drama|Romance|Fantasy', 'Drama|History|Romance',
       'Drama|Comedy|Romance', 'Comedy|War', 'Family|Fantasy',
       'Adventure|Comedy|Science Fiction', 'Drama|History',
       'Science Fiction|Animation|Family|Comedy|Adventure',
       'Comedy|Family', 'Adventure|Drama|History',
       'Adventure|Fantasy|Mystery', 'Adventure|Fantasy|Action|Thriller',
       'Animation|Action|Family|Science Fiction', 'Drama|Thriller|Crime',
       'Comedy|Drama|Family|Music|Romance',
       'Action|Adventure|Crime|Drama|Thriller',
       'Adventure|Comedy|Family|Fantasy|Science Fiction',
       'Drama|Music|Romance', 'Comedy|Crime|Drama',
       'Action|Thriller|Science Fiction|Mystery|Adventure',
       'Family|Fantasy|Adventure', 'Drama|Mystery|Thriller',
       'Action|Animation|Family', 'Fantasy|Adventure|Animation|Family',
       'Drama|Thriller|Mystery', 'Animation|Family|Comedy',
       'Action|Thriller|Crime|Mystery', 'Action|Crime',
       'Action|Thriller|Romance', 'Adventure|Fantasy|Action|Romance',
       'Drama|Adventure|Western', 'Adventure|Fantasy|Action',
       'Fantasy|Adventure|Action|Comedy|Drama',
       'Action|Science Fiction|Adventure|Thriller',
       'Comedy|Adventure|Fantasy|Animation|Family', 'Drama|Romance|War',
       'Thriller|Action|Adventure|Comedy|Crime',
       'Action|Adventure|Comedy|Crime|Thriller', 'Action|Comedy|Thriller',
       'Animation|Action|Comedy|Family|Science Fiction',
       'Action|Adventure|Family|Fantasy', 'Fantasy|Horror|Thriller',
       'Horror|Crime', 'Action|Adventure|Drama|Family',
       'Romance|Drama|Thriller|Crime', 'Horror|Comedy',
       'Action|Comedy|Thriller|Romance',
       'Science Fiction|Comedy|Adventure',
       'Animation|Adventure|Family|Fantasy', 'Horror|Drama|Thriller',
       'Comedy|Family|Fantasy', 'Science Fiction', 'Action|Comedy|Family',
       'Drama|Fantasy', 'Drama|Crime', 'Mystery|Horror|Action',
       'Comedy|Family|Animation|Adventure',
       'Action|Adventure|Horror|Science Fiction',
       'Adventure|Fantasy|Horror|Action|Thriller', 'Fantasy|Drama|Crime',
       'Drama|Romance|Thriller', 'Drama|Fantasy|Thriller|Mystery',
       'Romance|Drama|Mystery',
       'Crime|Adventure|Action|Comedy|Science Fiction',
       'Comedy|Family|Science Fiction', 'Comedy|Crime', 'Fantasy',
       'Drama|Mystery|Thriller|Crime',
       'Action|Adventure|Comedy|Drama|Science Fiction',
       'Action|Fantasy|Horror|Mystery',
       'Horror|Action|Thriller|Science Fiction',
       'Drama|Action|Thriller|Crime', 'Comedy|Science Fiction',
       'Thriller|Crime|Mystery', 'Fantasy|Comedy|Adventure',
       'Horror|Thriller|Fantasy|Mystery',
       'Horror|Comedy|Action|Science Fiction|Thriller',
       'Adventure|Fantasy|Action|Comedy|Science Fiction',
       'Music|Animation|Family|Fantasy', 'Crime|Comedy|Thriller',
       'Romance|Comedy|Drama', 'Drama|Comedy|Crime|Romance',
       'Thriller|Action|Comedy|Crime',
       'Comedy|Drama|Family|Science Fiction', 'Drama|Mystery',
       'Adventure|Drama|Action|History|War',
       'Fantasy|Animation|Family|Comedy',
       'Drama|Science Fiction|Adventure', 'Thriller|Crime',
       'History|Romance|War', 'Action|Adventure|Drama|Fantasy|Horror',
       'Comedy|Family|Romance',
       'Thriller|Science Fiction|Action|Adventure', 'War',
       'Action|History|War', 'Adventure|Drama|Romance|Action',
       'Action|Comedy|Crime|Thriller', 'Horror|Drama|Mystery|Thriller',
       'Adventure|Fantasy|Animation|Action|Comedy',
       'Drama|Mystery|Romance|Science Fiction|Thriller',
       'Adventure|Action|Animation|Fantasy|Science Fiction',
       'Drama|Thriller|Mystery|Romance', 'Comedy|Family|Romance|Fantasy',
       'Action|Drama|Thriller|War', 'Crime|Mystery|Thriller|Action',
       'Action|Comedy|Crime|Romance', 'Action|Drama|War',
       'Mystery|Adventure|Comedy|Romance|Drama',
       'Drama|Family|Romance|Music', 'Action|Comedy|Family|Adventure',
       'Crime|Drama|Thriller|Horror', 'Mystery|Thriller|Crime',
       'Action|Adventure|Drama', 'Drama|Comedy|Crime', 'Adventure|Action',
       'Adventure|Action|Thriller|Crime',
       'Fantasy|Drama|Thriller|Mystery|Romance',
       'Adventure|Fantasy|Science Fiction',
       'Action|Science Fiction|Adventure|Comedy|Family',
       'Action|Adventure|Thriller|Crime', 'Drama|Thriller|History|War',
       'Adventure|Action|Drama|Fantasy', 'Fantasy|Action',
       'Mystery|Thriller|Action', 'Music|Drama|Romance',
       'Animation|Family|Adventure|Comedy', 'Drama|Action',
       'Fantasy|Comedy|Family|Romance', 'Comedy|Drama|Crime',
       'Comedy|Drama|Music|Romance',
       'Thriller|Animation|Action|Science Fiction|Adventure',
       'Horror|Thriller|Crime', 'Family|Action|Science Fiction',
       'Animation|Comedy|Family|Adventure|Fantasy',
       'Action|Drama|History|War', 'Thriller|Horror|Mystery',
       'Drama|Mystery|Science Fiction|Thriller',
       'Action|Comedy|Thriller|Crime|Science Fiction',
       'Adventure|Comedy|Family',
       'Comedy|Science Fiction|Adventure|Family',
       'Horror|Science Fiction|Thriller', 'Adventure|Comedy|Drama|Family',
       'Adventure|Comedy|Family|Fantasy',
       'Action|Drama|Foreign|Mystery|Thriller',
       'Adventure|Action|Fantasy',
       'Thriller|Action|Drama|Science Fiction',
       'Fantasy|Thriller|Action|Crime',
       'Thriller|Science Fiction|Mystery', 'Action|Thriller|Adventure',
       'Thriller|Crime|Mystery|Drama', 'Thriller|Mystery|Science Fiction',
       'Action|Science Fiction|Drama', 'Adventure|Action|Crime|Mystery',
       'Adventure|Animation|Action|Family|Mystery',
       'Action|Fantasy|Thriller', 'Adventure|Fantasy|Romance',
       'Action|Thriller|Science Fiction|Adventure',
       'Comedy|Romance|Family', 'Action|Science Fiction|Thriller|Western',
       'Animation|Comedy|Family|Western|Adventure',
       'Drama|Thriller|Horror', 'Adventure|Action|Thriller|Comedy',
       'Action|Fantasy|Horror|Science Fiction|Thriller',
       'Thriller|Action|Mystery', 'Action|Mystery|Thriller',
       'Thriller|Romance|Fantasy', 'Fantasy|Action|Drama',
       'Action|Thriller|Crime|Drama', 'Drama|Comedy|Family',
       'Thriller|Drama|Horror',
       'Action|Adventure|Animation|Family|Fantasy',
       'Crime|Adventure|Action|Comedy|Thriller',
       'Drama|Thriller|Science Fiction', 'Drama|Fantasy|Romance',
       'Horror|Science Fiction|Mystery|Thriller',
       'Adventure|Fantasy|Action|Comedy',
       'Horror|Thriller|Science Fiction', 'Music|Family|Animation|Comedy',
       'Adventure|Animation|Family',
       'Comedy|Fantasy|Family|Music|Animation',
       'Animation|Family|Adventure|Comedy|Fantasy', 'Animation',
       'Comedy|Family|Music', 'Family|Comedy|Action',
       'Drama|Crime|History', 'Action|Comedy|Adventure',
       'Documentary|Music|Family', 'Action|Drama|Mystery|Thriller',
       'Horror|Action|Science Fiction',
       'Action|Adventure|Comedy|Science Fiction',
       'Action|Thriller|Science Fiction|Mystery',
       'Adventure|Animation|Family|Fantasy|Science Fiction',
       'Science Fiction|Horror|Action|Thriller', 'Drama|History|Crime',
       'Family|Adventure|Animation|Comedy',
       'Drama|Thriller|Science Fiction|Mystery', 'Drama|Family|Romance',
       'Action|Thriller|Drama',
       'Science Fiction|Action|Adventure|Thriller',
       'Science Fiction|Comedy|Crime|Action', 'Fantasy|Comedy|Family',
       'Crime|Mystery|Thriller', 'Mystery|Adventure|Comedy',
       'Western|Animation|Adventure|Comedy|Family',
       'Action|Fantasy|Adventure', 'Action|Adventure|Comedy|Thriller',
       'Documentary', 'Comedy|Fantasy',
       'Science Fiction|Adventure|Action', 'Drama|Action|History|War',
       'Action|Adventure|Crime|Thriller', 'Thriller|Drama|History',
       'Drama|Fantasy|Horror', 'Adventure|Fantasy|Animation|Family',
       'Action|Comedy|Crime|Drama|Music',
       'Action|Fantasy|Science Fiction|Thriller', 'Adventure',
       'Drama|Science Fiction|Mystery|Romance',
       'War|Adventure|Drama|Romance', 'Drama|Action|Thriller|Romance',
       'Drama|Mystery|Thriller|Horror',
       'Comedy|Crime|Drama|Romance|Thriller',
       'Adventure|Family|Science Fiction|Thriller',
       'Action|Adventure|Comedy|Drama|Family', 'Crime|Thriller|Horror',
       'Drama|Romance|Comedy|Music', 'Action|Adventure|Family|Foreign',
       'Family|Animation|Drama', 'Comedy|Action|Crime', 'Horror|Romance',
       'Romance|Comedy|Crime|Fantasy', 'Crime|Thriller|Drama',
       'Fantasy|Drama|Comedy|Family', 'Drama|Western|War|Romance',
       'Drama|Action|Thriller', 'Action|Comedy|Drama|Family',
       'Drama|Thriller|Crime|Mystery', 'Fantasy|Action|Thriller',
       'Fantasy|Drama|Family', 'Drama|Thriller|Crime|Mystery|Romance',
       'Comedy|Romance|Science Fiction',
       'Adventure|Fantasy|Action|Thriller|Science Fiction',
       'Thriller|Science Fiction|Action|Crime', 'Drama|Action|Western',
       'Action|Fantasy|Science Fiction',
       'Drama|Horror|Science Fiction|Romance', 'Romance|Comedy|Family',
       'Fantasy|Action|Horror', 'Drama|Western',
       'Science Fiction|Adventure|Mystery',
       'Animation|Adventure|Comedy|Family|Action',
       'Adventure|Fantasy|Drama', 'Comedy|Music|Romance',
       'Science Fiction|Adventure|Fantasy', 'Fantasy|Animation|Family',
       'Action|Horror|Fantasy',
       'Thriller|Action|Adventure|Science Fiction',
       'Adventure|Fantasy|Drama|Comedy|Science Fiction',
       'Science Fiction|Drama|Thriller', 'Action|Thriller|War',
       'Thriller|Action|Drama|Crime', 'Animation|Adventure|Family|Comedy',
       'Adventure|Drama|Romance', 'War|Action|Thriller|Science Fiction',
       'Action|Comedy|Thriller|Crime|Romance',
       'Drama|Action|Adventure|History|War', 'Fantasy|Family',
       'Action|Horror|Science Fiction', 'Documentary|Music',
       'Romance|Comedy|Family|Fantasy',
       'Fantasy|Action|Crime|Science Fiction|Thriller',
       'Fantasy|Action|Thriller|Science Fiction',
       'Family|Animation|Adventure', 'Adventure|Action|Comedy', 'Music',
       'Drama|Action|War|History',
       'Adventure|Action|Comedy|Thriller|Crime',
       'Action|Adventure|Fantasy|Thriller', 'Mystery|Thriller',
       'Thriller|Crime|Drama|Mystery',
       'Action|Thriller|Adventure|Science Fiction',
       'Action|Adventure|Comedy|Western',
       'Action|Adventure|Mystery|Science Fiction|Thriller',
       'Action|Adventure|Comedy|Crime',
       'Thriller|Fantasy|Comedy|Family|Mystery', 'Comedy|Drama|Family',
       'Action|Drama|Mystery|Thriller|Crime',
       'Adventure|Action|Comedy|Thriller', 'Western',
       'Comedy|Drama|Crime|Thriller', 'Comedy|Adventure|Family',
       'Drama|Horror|Science Fiction|Thriller',
       'Action|Adventure|Comedy|Family|Science Fiction',
       'Crime|Comedy|Romance',
       'Adventure|Action|Science Fiction|Thriller',
       'Animation|Family|Fantasy|Adventure|Comedy',
       'Action|Crime|Fantasy|Science Fiction',
       'Science Fiction|Horror|Action', 'Horror|Science Fiction|Mystery',
       'Science Fiction|Comedy|Crime',
       'Action|Crime|Science Fiction|Thriller',
       'Drama|Horror|Mystery|Thriller', 'Adventure|Horror|Thriller',
       'Drama|Science Fiction|Mystery', 'Action|Thriller|Adventure|Crime',
       'Thriller|Action|Romance|Science Fiction|Adventure',
       'History|Drama|Adventure|War', 'Adventure|Action|Drama|Thriller',
       'Action|Adventure|Fantasy|Horror|Science Fiction',
       'Documentary|Drama', 'Adventure|Fantasy|Drama|Comedy|Music',
       'Drama|History|Mystery', 'Science Fiction|Thriller|Drama',
       'Science Fiction|Thriller|Action|Adventure', 'Crime|Drama|Comedy',
       'Action|Science Fiction|Adventure|Mystery',
       'Romance|Science Fiction|Drama', 'Science Fiction|Action|Thriller',
       'Science Fiction|Horror|Thriller',
       'Action|Drama|Horror|Science Fiction|Thriller',
       'Adventure|Comedy|Drama|Fantasy',
       'Action|Crime|Drama|Comedy|Thriller',
       'Comedy|Drama|Science Fiction', 'Horror|Comedy|Romance',
       'Fantasy|Horror|Action', 'Action|Adventure|Drama|Mystery|Romance',
       'Fantasy|Action|Comedy|Crime', 'Drama|Horror',
       'Comedy|Music|Romance|Drama', 'Action|Family|Fantasy',
       'Comedy|Drama|History', 'Action|Adventure|Western',
       'Crime|Drama|Action|Thriller', 'Fantasy|Adventure|Family',
       'Thriller|Crime|Action',
       'Action|Adventure|Romance|Science Fiction|Thriller',
       'Animation|Comedy|Adventure|Family|Science Fiction', 'Romance',
       'Animation|Family|Adventure', 'Fantasy|Family|Comedy|Animation',
       'Action|Science Fiction|Adventure|Fantasy',
       'Science Fiction|Thriller|Horror', 'Documentary|Comedy',
       'Adventure|Comedy|Science Fiction|Family',
       'Action|Adventure|Drama|Science Fiction',
       'Action|Adventure|Drama|Thriller|War',
       'Crime|Drama|Romance|Thriller', 'Horror|Thriller|Comedy',
       'Horror|Science Fiction', 'Romance|Western',
       'Action|Adventure|Comedy|Romance', 'Drama|Action|Comedy|Romance',
       'Action|Adventure|Comedy|Drama', 'Action|Thriller|Fantasy',
       'Adventure|Comedy|Family|Science Fiction',
       'Action|Comedy|Drama|Family|Thriller',
       'Adventure|Thriller|Science Fiction',
       'Animation|Comedy|Family|Science Fiction',
       'Action|Crime|Adventure', 'Action|Comedy|Drama|Thriller',
       'Action|Adventure|Drama|Western',
       'Family|Fantasy|Science Fiction|Adventure',
       'Romance|Fantasy|Animation|Music', 'Comedy|Fantasy|Romance',
       'Comedy|Family|Adventure|Fantasy|Romance', 'Romance|Drama|History',
       'Drama|Thriller|Crime|Romance',
       'Action|Adventure|Science Fiction|Thriller|Horror',
       'Adventure|Fantasy|Action|Comedy|Thriller',
       'Mystery|Drama|Thriller|Crime', 'Drama|Action|History|Thriller',
       'Crime|Drama|Horror|Thriller', 'Crime|Drama|Family',
       'Horror|Thriller|Drama', 'Thriller|Drama|Mystery',
       'Comedy|Drama|Mystery|Romance', 'Comedy|Family|Adventure|Romance',
       'Action|Adventure|Comedy|Drama|Mystery',
       'Adventure|Fantasy|Action|Family', 'Thriller|Horror|Drama|Mystery',
       'Fantasy|Action|Science Fiction|Thriller',
       'Music|Drama|Romance|Crime', 'Animation|Family|Comedy|Adventure',
       'Action|Adventure|Comedy|Family|Fantasy',
       'Adventure|Fantasy|Action|Science Fiction',
       'Comedy|Drama|Fantasy|Romance', 'Crime|Fantasy|Drama',
       'Drama|Thriller|Action', 'Comedy|Drama|Family|Romance',
       'Music|Animation|Comedy|Family', 'Fantasy|Action|Adventure|Family',
       'Thriller|Action|Adventure|Drama',
       'Fantasy|Adventure|Animation|Comedy|Family',
       'Comedy|Drama|Family|Fantasy|Romance', 'War|Drama|History',
       'Mystery|Thriller|Horror', 'Comedy|Family|Adventure',
       'Fantasy|Drama|Thriller|Romance', 'Action|Adventure|Drama|War',
       'Drama|History|Thriller', 'Horror|Action|Crime|Thriller',
       'Action|Comedy|Documentary', 'Adventure|Comedy|Drama',
       'Drama|History|Crime|Thriller|Action',
       'Fantasy|Drama|Action|Comedy|Crime',
       'Comedy|Drama|Romance|Thriller', 'Drama|Thriller|Fantasy|Mystery',
       'Action|Adventure|Crime|Drama', 'Science Fiction|Drama|Romance',
       'Action|Adventure|Animation|Family|Science Fiction',
       'War|Adventure|Drama', 'Adventure|Action|Thriller|Mystery',
       'Horror|Adventure|Action', 'Fantasy|Action|Science Fiction',
       'Science Fiction|Action|Horror|Thriller|Adventure',
       'Adventure|Fantasy|Comedy|Family',
       'Animation|Action|Comedy|Family',
       'Mystery|Action|Thriller|Science Fiction|Adventure',
       'Mystery|Fantasy|Adventure|Comedy',
       'Adventure|Science Fiction|Action',
       'War|History|Action|Adventure|Drama', 'Horror|Mystery|Crime',
       'Action|Drama|Thriller|Crime', 'Adventure|War|History',
       'Action|Comedy|Crime|Drama', 'Western|History|War',
       'Comedy|Drama|Horror', 'Drama|Romance|Comedy',
       'Western|Animation|Family|Comedy|Music', 'Western|Adventure',
       'Adventure|Drama|Action|Thriller',
       'Music|Adventure|Animation|Action|Comedy',
       'Music|Comedy|Family|Romance', 'Comedy|Thriller|Crime',
       'Adventure|Action|Horror|Science Fiction|Thriller',
       'Thriller|Drama|Romance',
       'Crime|Drama|Horror|Mystery|Science Fiction',
       'Documentary|Comedy|Drama', 'Drama|Adventure|Thriller',
       'Music|Comedy|Action|Crime', 'Romance|Adventure|Drama',
       'Adventure|Fantasy|Family|Mystery',
       'Drama|Horror|Action|Thriller|Science Fiction',
       'Action|Adventure|Mystery|Thriller',
       'Adventure|Fantasy|Romance|Family', 'Action|Adventure|War',
       'Thriller|Action|Fantasy|Horror', 'Crime|Action|Comedy',
       'Drama|Horror|Music', 'Comedy|Music|Family|Fantasy|Animation',
       'Comedy|Family|Fantasy|Romance',
       'Comedy|Adventure|Fantasy|Action|Science Fiction',
       'Thriller|Action|Drama',
       'Fantasy|Action|Science Fiction|Thriller|Horror',
       'Drama|Crime|Mystery|Thriller', 'Action|Adventure|Drama|History',
       'Fantasy|Drama|Mystery', 'Drama|Thriller|Crime|History',
       'Action|Comedy|Drama', 'Action|Thriller|Comedy|Crime',
       'Animation|Fantasy', 'Thriller|Action|Horror',
       'Comedy|Drama|Adventure', 'Action|Horror|Crime',
       'History|Drama|Thriller|Crime|Mystery',
       'Action|Adventure|Crime|Drama|Fantasy', 'Drama|Mystery|Crime',
       'Adventure|Family', 'Action|Adventure|Fantasy|War',
       'Action|History', 'Action|Adventure|Thriller|Science Fiction',
       'Adventure|Family|Science Fiction|Action',
       'Drama|Comedy|Western|Romance', 'Romance|Comedy|Horror',
       'Fantasy|Action|Comedy|Science Fiction|Family',
       'Fantasy|Horror|Comedy', 'Drama|History|Music',
       'Drama|Family|Music|Romance',
       'Action|Adventure|Drama|Romance|Science Fiction',
       'Mystery|Thriller|Crime|Drama', 'Music|Drama|Crime|Romance',
       'Adventure|Drama|Action|Romance', 'Drama|Adventure|History',
       'Comedy|Horror|Fantasy',
       'Comedy|Action|Adventure|Fantasy|Science Fiction',
       'Comedy|Adventure|Romance',
       'Science Fiction|Action|Thriller|Crime|Drama',
       'Drama|Fantasy|Horror|Science Fiction|Thriller',
       'Drama|Action|Music|Romance', 'Science Fiction|Thriller|Mystery',
       'Adventure|Animation|Drama|Family',
       'Science Fiction|Fantasy|Action|Crime',
       'Romance|Animation|Family|Comedy|Adventure',
       'Crime|Drama|Mystery|Thriller|Action',
       'Action|Drama|Science Fiction|Thriller', 'Action|Fantasy',
       'Action|Adventure|Science Fiction|Family|Fantasy',
       'Action|Adventure|Drama|Romance', 'Romance|Drama|Crime',
       'Animation|Family|Comedy|Adventure|Fantasy',
       'Comedy|Family|Adventure|Crime',
       'Action|Crime|Fantasy|Science Fiction|Thriller',
       'Science Fiction|Action|Horror', 'Romance|Horror', 'Music|Comedy',
       'Thriller|Action|Drama|Music|Romance',
       'Comedy|Family|Fantasy|Drama',
       'Action|Adventure|Drama|History|Romance', 'Drama|Thriller|Romance',
       'Adventure|Fantasy|Drama|Science Fiction|Romance',
       'Romance|Crime|Drama|Thriller',
       'Fantasy|Animation|Science Fiction|Family',
       'Adventure|Drama|Romance|Western', 'Fantasy|Thriller|Horror',
       'Drama|Horror|Thriller|Crime',
       'Family|Fantasy|Science Fiction|Adventure|Comedy',
       'Comedy|Fantasy|Science Fiction', 'Family|Fantasy|Animation',
       'Animation|Comedy|Drama|Family|Fantasy',
       'Drama|Fantasy|Romance|Science Fiction',
       'Horror|Action|Thriller|Crime', 'Crime|Horror|Mystery',
       'Fantasy|Comedy|Romance|Science Fiction', 'Comedy|Drama|Thriller',
       'Drama|Thriller|Mystery|War', 'Drama|Action|Crime',
       'Fantasy|Drama|Horror|Thriller',
       'Fantasy|Drama|Comedy|Science Fiction|Romance',
       'History|Drama|Music', 'Action|Comedy|Music|Family|Adventure',
       'Comedy|Drama|Fantasy', 'TV Movie|Adventure|Drama|Science Fiction',
       'Fantasy|Horror|Science Fiction', 'Action|Drama|Adventure',
       'Family|Comedy|Fantasy', 'Drama|History|War|Action',
       'Drama|Comedy|Music', 'Thriller|Science Fiction|Action',
       'Comedy|Crime|Action', 'Crime|Drama|Science Fiction|Thriller',
       'Crime|Comedy', 'Drama|Crime|Music',
       'Comedy|Drama|Romance|Fantasy', 'Adventure|Action|Comedy|Western',
       'Comedy|Drama|Romance|Music', 'Thriller|Action|Science Fiction',
       'Science Fiction|Comedy|Family|Romance',
       'Drama|Adventure|Romance|Thriller',
       'Animation|Action|Science Fiction|Family|Adventure',
       'Action|Adventure|Drama|History|War',
       'Action|Comedy|Drama|Romance',
       'Adventure|Animation|Action|Comedy|Family',
       'Science Fiction|Adventure|Family|Fantasy',
       'Action|Adventure|Thriller|War',
       'Thriller|Action|Comedy|Crime|Drama',
       'Science Fiction|Action|Adventure|Drama|Thriller',
       'Comedy|Drama|Family|Music', 'Horror|Action',
       'Adventure|Animation|Drama|Family|Music',
       'Action|Thriller|Crime|Adventure', 'Mystery|Science Fiction',
       'Mystery|Science Fiction|Thriller',
       'Adventure|Family|Science Fiction',
       'Action|Adventure|Crime|Drama|Mystery',
       'Action|Adventure|Comedy|Crime|Drama', 'Drama|Fantasy|Comedy',
       'History', 'Adventure|Comedy|Western|Romance|Thriller',
       'Comedy|Adventure|Fantasy|Science Fiction|Action',
       'Romance|Comedy|Crime', 'Thriller|Fantasy|Crime|Drama',
       'Adventure|Comedy|Drama|Family|Fantasy', 'Crime|Mystery',
       'Horror|Mystery|Science Fiction', 'Adventure|Documentary',
       'Crime|Action|Drama',
       'Adventure|Comedy|Family|Science Fiction|Action',
       'Comedy|Horror|Thriller', 'Family|Adventure',
       'Action|Comedy|Thriller|Crime|Family',
       'Drama|Animation|Family|Comedy|Fantasy',
       'Action|Comedy|Crime|Drama|Thriller', 'Drama|Crime|Mystery',
       'Drama|Family|Fantasy', 'Comedy|Crime|Mystery',
       'Romance|Family|Animation|Fantasy|Music',
       'Action|Adventure|Science Fiction|Family',
       'Drama|Thriller|Crime|Adventure', 'Thriller|Action|Crime|Drama',
       'Action|Adventure|Comedy|Drama|Western', 'Action|Comedy|War',
       'Mystery', 'Fantasy|Drama|Comedy|Romance|Family',
       'Fantasy|Animation|Comedy|Crime|Family', 'Fantasy|Comedy|Drama',
       'Adventure|Comedy|Crime', 'Adventure|Thriller',
       'Adventure|Family|Fantasy|Comedy|Romance',
       'Crime|Drama|History|Thriller', 'Comedy|Drama|War',
       'Horror|Drama|Romance|Thriller',
       'Comedy|Family|Fantasy|Science Fiction',
       'Thriller|Comedy|Crime|Adventure',
       'Crime|Drama|Action|Thriller|Mystery', 'Family|Comedy|Drama',
       'Drama|Comedy|Romance|Family', 'Science Fiction|Mystery|Adventure',
       'Adventure|Science Fiction|Mystery', 'Horror|Drama|Mystery',
       'Comedy|Western', 'Crime|Comedy|Drama',
       'Horror|Thriller|Adventure', 'Comedy|Horror|Music|Science Fiction',
       'Adventure|Drama|History|War', 'Western|Action|Adventure|Drama',
       'Comedy|Music|Family|Fantasy',
       'Adventure|Action|Comedy|Science Fiction|Family',
       'Adventure|Drama|Western',
       'Science Fiction|Action|Adventure|Comedy|Family', 'Action|Horror',
       'Action|Adventure|Crime|Science Fiction|Thriller',
       'Fantasy|Comedy|Science Fiction|Romance', 'Drama|Mystery|Horror',
       'Action|Science Fiction|Crime',
       'Adventure|Animation|Crime|Family|Fantasy', 'Music|Crime|Drama',
       'Action|Drama|History', 'Adventure|Action|Romance|Science Fiction',
       'Action|Adventure|Drama|Horror|Science Fiction',
       'Drama|History|Mystery|Thriller', 'Comedy|Crime|Romance|Thriller',
       'Romance|Fantasy|Drama|Comedy', 'Adventure|Science Fiction',
       'Adventure|Action|Thriller|Crime|Mystery',
       'Action|Drama|Thriller|Crime|Mystery',
       'Action|Adventure|Drama|History|Western', 'Drama|War|Action',
       'Drama|Action|Romance', 'Drama|Family|Animation|Adventure',
       'Science Fiction|Adventure', 'Fantasy|Comedy|Science Fiction',
       'Adventure|Drama|Action|Romance|Family', 'Action|Comedy|Drama|War',
       'Horror|Comedy|Music', 'Crime|Thriller|Action',
       'Crime|Drama|History', 'History|Action|Drama|Adventure|War',
       'Action|Comedy|Drama|Western', 'History|Drama|Western|Crime',
       'Drama|Animation|Adventure', 'Action|Drama|Horror|Thriller'],
      dtype=object)
```

That explains the issue. There are 840 unique genres because self-reporting the data allows inputers to tag mulitple genres to a movie. To remedy this, only the first descriptive name out of each 840 genres will be pulled. That single word descriptor will be used to identify genres. That change should reduce the unique count of genres.

First, a copy of the original df is placed into a new df. Then, the genres column will be changed in a single word category.

[61]:
```
one_wrd_gen_df = rev_over75_df.copy()
```

[62]:
```
#split genres column by "/"
split_column_gen = ['genres']

#apply split function to the column in df
for c in split_column_gen:
    one_wrd_gen_df[c] = one_wrd_gen_df[c].apply(lambda x: x.split("|")[0])
```

[63]:
```
#find number of genres. s/b significantly reduced from 840 genres
one_wrd_gen_df.nunique()
```

[63]:
```
id                       2700
imdb_id                  2700
budget                    265
revenue                  2657
original_title           2664
director                 1133
runtime                   123
genres                     19
production_companies     2056
release_date             2199
release_year               56
budget_adj               1470
revenue_adj              2695
75th_%tile_budget           2
75th_%tile_revenue          1
dtype: int64
```

Delineating genres down to sinlge words was successful. From initially 840 genres, the df was reduced to just 19 genres.

Below are the names of the 19 genres.

[64]:
```
#from 840 to just 19 genres!
#here are the 19 genres listed
one_wrd_gen_df['genres'].unique()
```

[64]:
```
array(['Action', 'Adventure', 'Western', 'Science Fiction', 'Drama',
       'Family', 'Comedy', 'Crime', 'Romance', 'War', 'Mystery',
       'Thriller', 'Fantasy', 'History', 'Animation', 'Horror', 'Music',
       'Documentary', 'TV Movie'], dtype=object)
```

The below code, showing genre counts, is informative and interesting but ultimatley inconsequential for this research question. However, it could prove to be a useful jumping off point for further analysis.

[87]:

```
#count value of each genre.
one_wrd_gen_df.genres.value_counts()
```

[87]:

```
Comedy             598
Drama              515
Action             503
Adventure          265
Horror             165
Crime              115
Thriller           103
Fantasy             95
Animation           88
Science Fiction     73
Romance             52
Family              36
Mystery             24
Music               20
War                 14
History             14
Documentary         10
Western              9
TV Movie             1
Name: genres, dtype: int64
```

The following code allows for grouping each genre by its mean revenue.

[66]:

```
#find the average revenue for each genre
top_gen_rev_df = one_wrd_gen_df.groupby('genres').revenue.mean()
top_gen_rev_df
```

[66]:

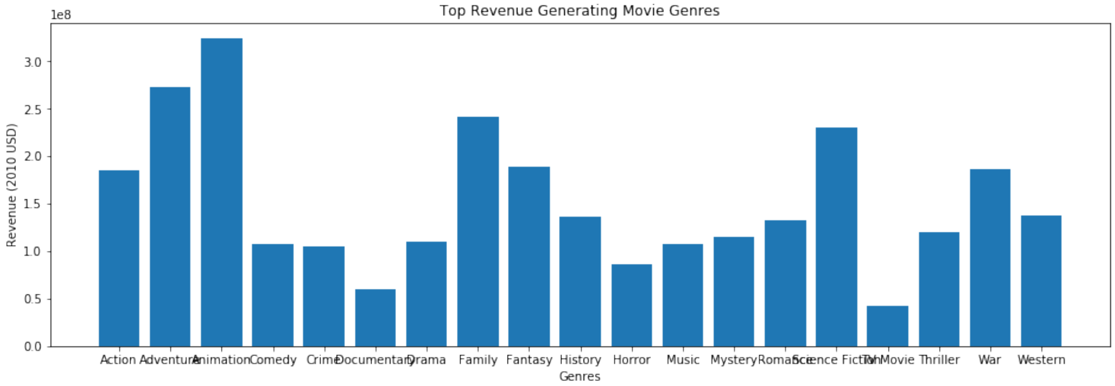
```
genres
Action             1.856718e+08
Adventure          2.727880e+08
Animation          3.244053e+08
Comedy             1.074446e+08
Crime              1.050341e+08
Documentary        5.952969e+07
Drama              1.103658e+08
Family             2.418806e+08
Fantasy            1.891425e+08
History            1.368413e+08
Horror             8.575997e+07
Music              1.072924e+08
Mystery            1.149644e+08
```

```
Romance            1.321886e+08
Science Fiction    2.309256e+08
TV Movie           4.200000e+07
Thriller           1.199019e+08
War                1.868975e+08
Western            1.378732e+08
Name: revenue, dtype: float64
```

Which, of course, allows for plotting the information graphically.

[67]:

```
#plot all the genres by their average revenue
plt.subplots(figsize = (16,5))
plt.bar(top_gen_rev_df.index, top_gen_rev_df)
plt.title('Top Revenue Generating Movie Genres')
plt.xlabel('Genres')
plt.ylabel('Revenue (2010 USD)');
```



So as the bar chart indicates, Animation, Adventure and Family are the movie genres most highly represented in the top 25% of highest revenued movies. As a side note, Science Fiction closely approaches the Family genre as a close 4th place contender.

Finally, the below code will store the newly created data frames back into the original .csv file.

[71]:

```
#store these dataframes in the original csv file
df_mov.to_csv('tmdb-movies.csv', index=False)
bud_over75_df.to_csv('tmdb-movies.csv', index=False)
rev_over75_df.to_csv('tmdb-movies.csv', index=False)
bud_or_rev_over75_df.to_csv('tmdb-movies.csv', index=False)
bud_and_rev_over75_df.to_csv('tmdb-movies.csv', index=False)
one_wrd_gen_df.to_csv('tmdb-movies.csv', index=False)
```

1.1.5 Which Fiscal Quarter's Movie Releases are Most Represented in the Top 25% Highest Revenued Movies?

This question requires the creation of a column that turns the release date info into their respective fiscal quarter.

The first step is copying the original df into a new df.

[72]:

```
fiscal_qtr_df = rev_over75_df.copy()
```

Google searching provided the code necessary to create a new column consisting of fiscal quarter information.

[86]:

```
#Quick Google search to find out how to deliniate dates into fiscal quarters
#Mentor 1-on-1 was the true help on this!
fiscal_qtr_df['fiscal_q'] = pd.PeriodIndex(fiscal_qtr_df.release_date, freq='Q')
fiscal_qtr_df.head(10)
```

[86]:

```
       id    imdb_id     budget     revenue                original_title  \
0  135397  tt0369610  150000000  1513528810                Jurassic World
1   76341  tt1392190  150000000   378436354            Mad Max: Fury Road
2  262500  tt2908446  110000000   295238201                     Insurgent
3  140607  tt2488496  200000000  2068178225  Star Wars: The Force Awakens
4  168259  tt2820852  190000000  1506249360                     Furious 7
5  281957  tt1663202  135000000   532950503                  The Revenant
6   87101  tt1340138  155000000   440603537            Terminator Genisys
7  286217  tt3659388  108000000   595380321                   The Martian
8  211672  tt2293640   74000000  1156730962                       Minions
9  150540  tt2096673  175000000   853708609                    Inside Out

                         director  runtime  \
0                 Colin Trevorrow      124
1                   George Miller      120
2                Robert Schwentke      119
3                     J.J. Abrams      136
4                       James Wan      137
5  Alejandro GonzÃąlez IÃśÃąrritu      156
6                     Alan Taylor      125
7                    Ridley Scott      141
8        Kyle Balda|Pierre Coffin       91
9                     Pete Docter       94

                                      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
5           Western|Drama|Adventure|Thriller
```

```
6  Science Fiction|Action|Thriller|Adventure
7            Drama|Adventure|Science Fiction
8          Family|Animation|Adventure|Comedy
9                    Comedy|Animation|Family

                                production_companies release_date  \
0  Universal Studios|Amblin Entertainment|Legenda...   2015-06-09
1  Village Roadshow Pictures|Kennedy Miller Produ...   2015-05-13
2  Summit Entertainment|Mandeville Films|Red Wago...   2015-03-18
3          Lucasfilm|Truenorth Productions|Bad Robot   2015-12-15
4  Universal Pictures|Original Film|Media Rights ...   2015-04-01
5  Regency Enterprises|Appian Way|CatchPlay|Anony...   2015-12-25
6            Paramount Pictures|Skydance Productions   2015-06-23
7  Twentieth Century Fox Film Corporation|Scott F...   2015-09-30
8      Universal Pictures|Illumination Entertainment   2015-06-17
9  Walt Disney Pictures|Pixar Animation Studios|W...   2015-06-09

   release_year    budget_adj   revenue_adj 75th_%tile_budget  \
0          2015  1.379999e+08  1.392446e+09              over
1          2015  1.379999e+08  3.481613e+08              over
2          2015  1.012000e+08  2.716190e+08              over
3          2015  1.839999e+08  1.902723e+09              over
4          2015  1.747999e+08  1.385749e+09              over
5          2015  1.241999e+08  4.903142e+08              over
6          2015  1.425999e+08  4.053551e+08              over
7          2015  9.935996e+07  5.477497e+08              over
8          2015  6.807997e+07  1.064192e+09              over
9          2015  1.609999e+08  7.854116e+08              over

  75th_%tile_revenue fiscal_q  qtr
0               over   2015Q2    2
1               over   2015Q2    2
2               over   2015Q1    1
3               over   2015Q4    4
4               over   2015Q2    2
5               over   2015Q4    4
6               over   2015Q2    2
7               over   2015Q3    3
8               over   2015Q2    2
9               over   2015Q2    2
```

The following code allows for grouping each fiscal quarter by its mean revenue.

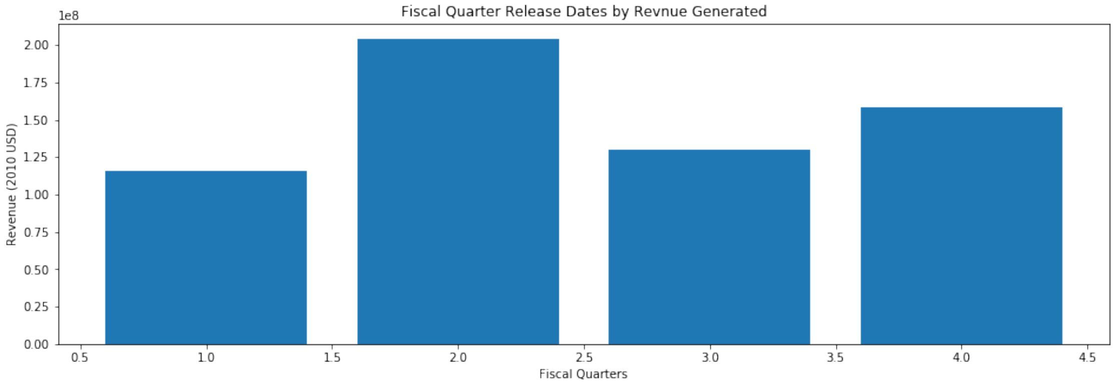
[88]:

```
#find the average revenue for each quarter
top_qtr_rev_df = fiscal_qtr_df.groupby('qtr').revenue.mean()
top_qtr_rev_df
```

[88]:
```
qtr
1    1.156677e+08
2    2.041026e+08
3    1.297636e+08
4    1.585213e+08
Name: revenue, dtype: float64
```

Again, this allows for plotting the information graphically.

[89]:
```
#plot all the fiscal quarters by their average revenue
plt.subplots(figsize = (16,5))
plt.bar(top_qtr_rev_df.index, top_qtr_rev_df)
plt.title('Fiscal Quarter Release Dates by Revnue Generated')
plt.xlabel('Fiscal Quarters')
plt.ylabel('Revenue (2010 USD)');
```



In analysis, though the movie industry speaks of “Summer Blockbuster Season”, the bar charts shows that Q2 (April-Jun) shows the highest revenue generated with Q4 (Oct-Dec) showing the second highest revenue generated during the years under study.

And, finally, the below code stores the newly created dataframes back into the original .csv file.

[91]:
```
fiscal_qtr_df.to_csv('tmdb-movies.csv', index=False)
top_qtr_rev_df.to_csv('tmdb-movies.csv', index=False)
```

```
/Users/steve-o/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:2:
FutureWarning: The signature of `Series.to_csv` was aligned to that of
`DataFrame.to_csv`, and argument 'header' will change its default value from
False to True: please pass an explicit value to suppress this warning.
```

Though, it's beyond my understanding of Python to interpret the warning message received from the above code, I'd assume that there was issue with perhaps the period indexing of the fiscal quarter creation and it appears that it won't be stored back to the .csv file in the way that I intended it to. However, it appears that it will still store it back in a different way.

## Conclusions

This research has provided a few conclusions that, if pursued through more rigorous statistical analysis, would provide a spring board to further study.

From results of studying the first research question, a somewhat positive relation was shown between budget costs and revenues generated. What was not shown (graphically, at least) was a stronger positive relation between higher ended budgeted costs and higher ended revenues generated.

The results from the second question show that the Animation genere rates highly in revenue generation. There isn't really a suprise there. However, what may have been more interesting was that Family movies ranked third. That genre doesn't traditionally come to mind when one thinks of revenue generation.

Finally, as far as fiscal quarter release dates, it's intersting to note that Oct-Dec ranks higher than the second part of summer (Jul-Sept). That result is a bit of a suprise too. Though, to be honest, one often hears of Holiday release season so perhaps that makes a larger impact in generating revenues than one might expect.

In conclusion, much can be gleaned from initial analysis such as this. However, more is there to be analyized rigorously before conclusion can be concretely drawn.

[ ]: