b4evargru

October 23, 2025

1 Movie Industry Exploratory Data Analysis

1.1 Objective: Investigate the film industry to gain sufficient understanding of what attributes to success and in turn utilize this analysis to create *actionable* recommendations for companies to enter the industry.

1.1.1 Importing necessary libraries and the datasets.

[1]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import datetime as dt
import numpy as np
```

[57]:
```
#Setting the default style for plots
plt.style.use('ggplot')

from matplotlib.pyplot import figure
plt.rcParams['figure.figsize'] = (12,8)

%matplotlib inline
```

[3]:
```
movie_dates_df = pd.read_csv('movie_release_dates.csv', index_col=0)
theaters_df = pd.read_csv('movie_theater_data.csv', index_col=0)
awards_df = pd.read_csv('movie_awards.csv', index_col=0)
actors_df = pd.read_csv('Actors_Table.csv')
directors_df = pd.read_csv('Directors_Table.csv')
imdb_base_df = pd.read_csv('IMDb_base.csv')
imdb_budgets_df = pd.read_csv('IMDb_budgets.csv')
studio_df = pd.read_csv('studiodf.csv')
```

[4]:
```
#First remove any movies that had a $0 domestic gross.
imdb_budgets_df = imdb_budgets_df[imdb_budgets_df['Domestic Gross'] !=0]
```

1.1.2 Previewing the head of each dataframe so we know what data we are working with.

[5]:
```
imdb_budgets_df.head()
```

[5]:
```
                   Movie  Year  IMDb Rating  Runtime  \
0      Avengers: Endgame  2019   8.4  PG-13      181
1                 Avatar  2009   7.8  PG-13      162
2          Black Panther  2018   7.3  PG-13      134
3 Avengers: Infinity War  2018   8.4  PG-13      149
4                Titanic  1997   7.8  PG-13      194

                       Genre  Release Date  Production Budget  \
0    Action, Adventure, Drama  Apr 23, 2019          400000000
1  Action, Adventure, Fantasy  Dec 17, 2009          237000000
2   Action, Adventure, Sci-Fi  Feb 13, 2018          200000000
3   Action, Adventure, Sci-Fi  Apr 25, 2018          300000000
4             Drama, Romance  Dec 18, 1997          200000000

   Domestic Gross  Worldwide Gross
0       858373000       2797800564
1       760507625       2788701337
2       700059566       1346103376
3       678815482       2048359754
4       659363944       2208208395
```

[6]:
```
movie_dates_df.head()
```

[6]:
```
                     movie release_date release_month release_day  \
0               Metropolis   1927-03-06         March      Sunday
1  Dr. Mabuse, the Gambler   1927-08-08        August      Monday
2              The Unknown   1927-06-03          June      Friday
3           The Jazz Singer   1927-10-06       October    Thursday
4                  Chicago   1927-12-23      December      Friday

   release_year
0          1927
1          1927
2          1927
3          1927
4          1927
```

[7]:
```
theaters_df.head()
```

[7]:
```
                       title  max_theaters  year  total_dom_gross($)  \
0              The Lion King          4802  2019           543638043
1          Avengers: Endgame          4662  2019           858373000
2  Spider-Man: Far from Home          4634  2019           390532085
```

```
3              Toy Story 4          4575  2019           434038008
4             It Chapter Two          4570  2019           211593228

         studio
0        Disney
1        Disney
2          Sony
3        Disney
4  Warner Bros.
```

[8]:
```
actors_df.head()
```

[8]:
```
               Movie  Year              value  Release Date  \
0  Avengers: Endgame  2019  Robert Downey Jr.  Apr 23, 2019
1  Avengers: Endgame  2019        Chris Evans  Apr 23, 2019
2  Avengers: Endgame  2019       Mark Ruffalo  Apr 23, 2019
3  Avengers: Endgame  2019    Chris Hemsworth  Apr 23, 2019
4             Avatar  2009    Sam Worthington  Dec 17, 2009

   Production Budget  Domestic Gross  Worldwide Gross
0          400000000       858373000       2797800564
1          400000000       858373000       2797800564
2          400000000       858373000       2797800564
3          400000000       858373000       2797800564
4          237000000       760507625       2788701337
```

[9]:
```
directors_df.head()
```

[9]:
```
                    Movie  Year           value  Release Date  \
0       Avengers: Endgame  2019       Joe Russo  Apr 23, 2019
1       Avengers: Endgame  2019   Anthony Russo  Apr 23, 2019
2                  Avatar  2009  James Cameron  Dec 17, 2009
3           Black Panther  2018    Ryan Coogler  Feb 13, 2018
4  Avengers: Infinity War  2018       Joe Russo  Apr 25, 2018

   Production Budget  Domestic Gross  Worldwide Gross
0          400000000       858373000       2797800564
1          400000000       858373000       2797800564
2          237000000       760507625       2788701337
3          200000000       700059566       1346103376
4          300000000       678815482       2048359754
```

[10]:
```
imdb_base_df.head()
```

[10]:
```
                                         Movie  Year  IMDb Rating  Runtime  \
0  Star Wars: Episode VII - The Force Awakens  2015   7.9  PG-13      138
1                            Avengers: Endgame  2019   8.4  PG-13      181
```

```
2                                      Avatar  2009    7.8  PG-13       162
3                               Black Panther  2018    7.3  PG-13       134
4                      Avengers: Infinity War  2018    8.4  PG-13       149

                     Genre
0   Action, Adventure, Sci-Fi
1    Action, Adventure, Drama
2  Action, Adventure, Fantasy
3   Action, Adventure, Sci-Fi
4   Action, Adventure, Sci-Fi
```

[11]:
```python
studio_df.head()
```

[11]:
```
                                         title           studio  \
0                                  Toy Story 3      Buena Vista
1                   Alice in Wonderland (2010)      Buena Vista
2  Harry Potter and the Deathly Hallows Part 1               WB
3                                    Inception               WB
4                          Shrek Forever After  Pixar/Dreamworks

   domestic_gross foreign_gross  year
0     415000000.0     652000000  2010
1     334200000.0     691300000  2010
2     296000000.0     664300000  2010
3     292600000.0     535700000  2010
4     238700000.0     513900000  2010
```

2 Question 1: What are the most profitable movies and how much should you spend?

Let's calculate profit and profit margin for each of the movies in `imdb_budgets_df` dataframe and add those as new columns.

Here, we'll define profit as `Worldwide Gross-Production Budget`.

It will also be beneficial in our analysis to have uniformity when discussing movie budgets and profits so we will also create an adjusted budget and adjusted profit column to account for inflation.

We will use an average inflation rate of 3.22%.

[12]:
```python
imdb_budgets_df['Profit'] = imdb_budgets_df['Worldwide Gross'] -␣
 ↪imdb_budgets_df['Production Budget']

imdb_budgets_df['Profit_Margin'] = (imdb_budgets_df['Worldwide Gross'] -
                                    imdb_budgets_df['Production Budget'])/
 ↪imdb_budgets_df['Worldwide Gross']
```

[13]:
```
imdb_budgets_df['Adjusted_Budget'] = ((((2020-imdb_budgets_df['Year'])*.
↪0322)+1)*
                                       imdb_budgets_df['Production Budget'])

#Suppressing Scienific Notation
pd.options.display.float_format = '{:.2f}'.format

imdb_budgets_df['Adjusted_Profit'] = (((2020-imdb_budgets_df['Year'])*.
↪0322)+1)*imdb_budgets_df['Profit']
imdb_budgets_df.head()
```

[13]:
```
                    Movie  Year  IMDb Rating  Runtime  \
0       Avengers: Endgame  2019  8.40  PG-13      181
1                  Avatar  2009  7.80  PG-13      162
2           Black Panther  2018  7.30  PG-13      134
3  Avengers: Infinity War  2018  8.40  PG-13      149
4                 Titanic  1997  7.80  PG-13      194

                        Genre  Release Date  Production Budget  \
0    Action, Adventure, Drama  Apr 23, 2019          400000000
1  Action, Adventure, Fantasy  Dec 17, 2009          237000000
2   Action, Adventure, Sci-Fi  Feb 13, 2018          200000000
3   Action, Adventure, Sci-Fi  Apr 25, 2018          300000000
4             Drama, Romance  Dec 18, 1997          200000000

   Domestic Gross  Worldwide Gross      Profit  Profit_Margin  \
0       858373000       2797800564  2397800564           0.86
1       760507625       2788701337  2551701337           0.92
2       700059566       1346103376  1146103376           0.85
3       678815482       2048359754  1748359754           0.85
4       659363944       2208208395  2008208395           0.91

   Adjusted_Budget  Adjusted_Profit
0     412880000.00    2475009742.16
1     320945400.00    3455513950.57
2     212880000.00    1219912433.41
3     319320000.00    1860954122.16
4     348120000.00    3495487532.34
```

For this question we are specifically looking at profitable movies. We'll create a separate dataframe called `profitable_movies_df` where the `Profit` column is greater than 0. We will then sort by `Adjusted_Profit` to rank movies in terms of profitability.

[14]:
```
profitable_movies_df = imdb_budgets_df.loc[imdb_budgets_df['Profit'] > 0]
profitable_ranked_df = profitable_movies_df.sort_values(by=['Adjusted_Profit'],␣
↪ascending=False)
```

```
profitable_ranked_df.reset_index(inplace=True) #Modify the DataFrame in place␣
 ↪(do not create a new object).
profitable_ranked_df.head()
```

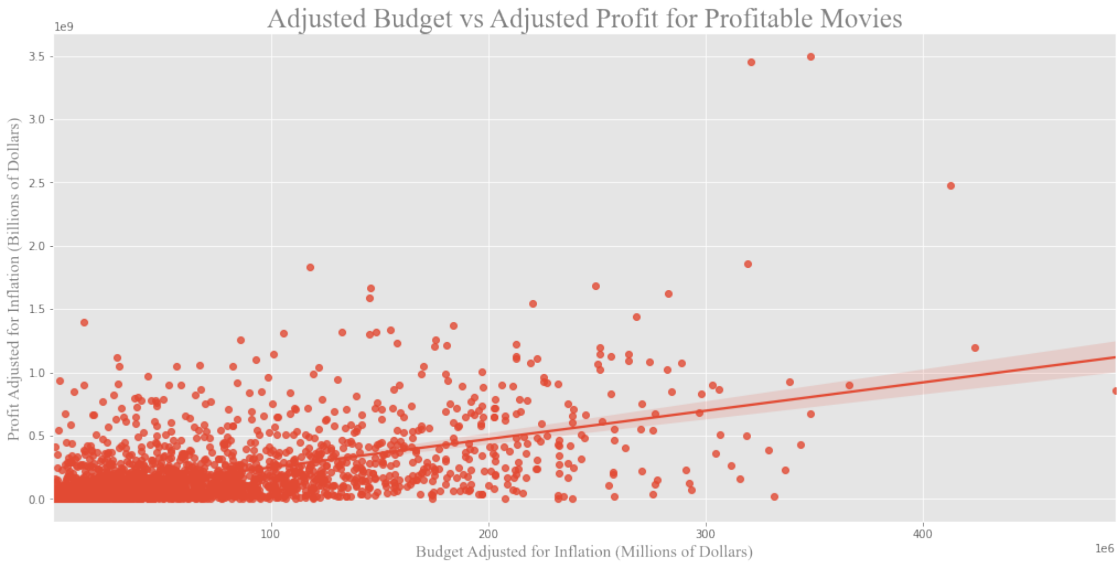
[14]:

```
   index                    Movie  Year  IMDb Rating  Runtime  \
0      4                  Titanic  1997  7.80  PG-13      194
1      1                   Avatar  2009  7.80  PG-13      162
2      0        Avengers: Endgame  2019  8.40  PG-13      181
3      3   Avengers: Infinity War  2018  8.40  PG-13      149
4     28            Jurassic Park  1993  8.10  PG-13      127

                        Genre  Release Date  Production Budget  \
0             Drama, Romance  Dec 18, 1997          200000000
1  Action, Adventure, Fantasy  Dec 17, 2009          237000000
2    Action, Adventure, Drama  Apr 23, 2019          400000000
3   Action, Adventure, Sci-Fi  Apr 25, 2018          300000000
4   Action, Adventure, Sci-Fi  Jun 11, 1993           63000000

   Domestic Gross  Worldwide Gross       Profit  Profit_Margin  \
0       659363944       2208208395  2008208395           0.91
1       760507625       2788701337  2551701337           0.92
2       858373000       2797800564  2397800564           0.86
3       678815482       2048359754  1748359754           0.85
4       402523348       1045627627   982627627           0.94

   Adjusted_Budget  Adjusted_Profit
0     348120000.00    3495487532.34
1     320945400.00    3455513950.57
2     412880000.00    2475009742.16
3     319320000.00    1860954122.16
4     117772200.00    1836924085.91
```

Now that we've got our profitable movie data, let's take a look at adjusted profit versus adjusted budget for each of the movies in the dataframe.

[15]:

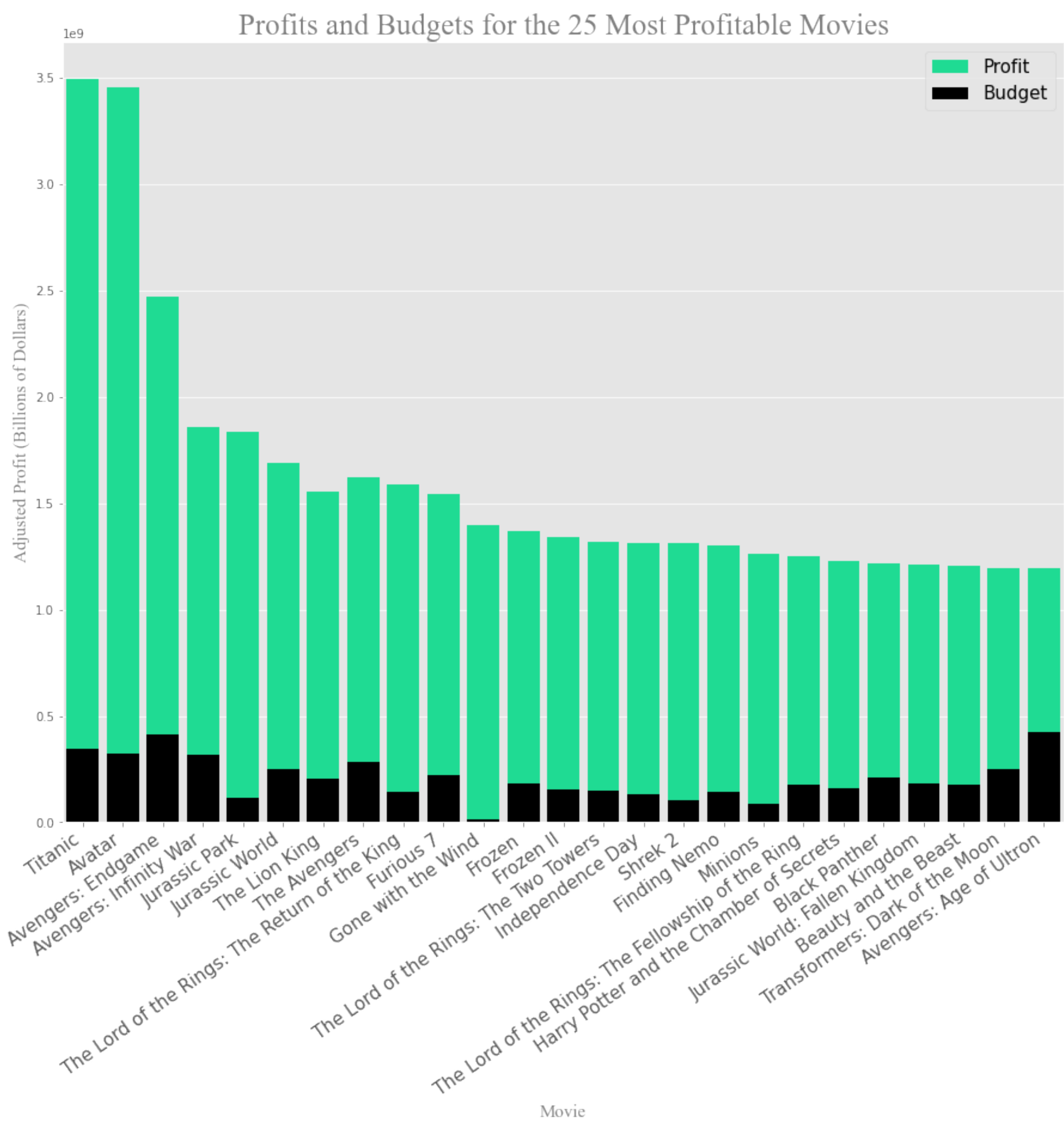
```
ax1 = sns.lmplot(x='Adjusted_Budget', y='Adjusted_Profit',␣
 ↪data=profitable_ranked_df, height=7, aspect=2)
plt.xlabel('Budget Adjusted for Inflation (Millions of Dollars)', fontdict =␣
 ↪{'fontname': 'Times New Roman', 'color': 'gray', 'fontsize' : '15'})
#setting x-axis label
plt.ticklabel_format(axis='x', style='sci', scilimits=(6,6))
plt.ylabel('Profit Adjusted for Inflation (Billions of Dollars)',fontdict =␣
 ↪{'fontname': 'Times New Roman', 'color': 'gray', 'fontsize' : '15'})
plt.title('Adjusted Budget vs Adjusted Profit for Profitable Movies', fontdict␣
 ↪= {'fontname': 'Times New Roman', 'color': 'gray', 'fontsize' : '25'})
plt.savefig('BudgetVProfit', dpi = 300);
```



This scatter plot is helpful in beginning to understand how much money should be budgeted for a movie. The positive trend line indicates that an increase in the budget will result in an increase in profit.

Let's take a look at the most successful movies so that we can get a better idea of what the budget should be.

[16]:
```
plt.figure(figsize=(15,12))
sns.barplot(x=profitable_ranked_df.loc[0:25, 'Movie'],y=profitable_ranked_df.
 ↪loc[0:25, 'Adjusted_Profit'],
            color='mediumspringgreen', label='Profit', ci=None)
sns.barplot(x=profitable_ranked_df.loc[0:25, 'Movie'],y=profitable_ranked_df.
 ↪loc[0:25, 'Adjusted_Budget'],
            color='black', label='Budget', ci=None)
plt.xlabel('Movie', fontdict = {'fontname': 'Times New Roman', 'color': 'gray',␣
 ↪'fontsize' : '15'})
plt.title("Profits and Budgets for the 25 Most Profitable Movies", fontdict =␣
 ↪{'fontname': 'Times New Roman', 'color': 'gray', 'fontsize' : '25'})
plt.ylabel('Adjusted Profit (Billions of Dollars)', fontdict = {'fontname':␣
 ↪'Times New Roman', 'color': 'gray', 'fontsize' : '15'})
plt.xticks(rotation=35, horizontalalignment='right', fontsize=15)
plt.legend(loc='best', fontsize=15)
plt.savefig('ProfitBudgetTop25', dpi=300);
```



[17]:
```
profitable_movies_df['Adjusted_Budget'].describe()
```

[17]:
```
count        2836.00
mean     60689139.20
std      63199464.86
min         10606.40
25%      16608850.00
50%      38684100.00
75%      82247150.00
max     488834200.00
Name: Adjusted_Budget, dtype: float64
```

[18]:
```
profitable_movies_df.loc[0:24, 'Adjusted_Budget'].describe()
```

[18]:
```
count          25.00
mean    242777774.40
std      80698866.89
min     106064000.00
25%     180635000.00
50%     225760000.00
75%     282960000.00
max     423765000.00
Name: Adjusted_Budget, dtype: float64
```

[19]:
```
profitable_movies_df['Profit_Margin'].describe()
```

[19]:
```
count   2836.00
mean       0.62
std        0.24
min        0.00
25%        0.47
50%        0.67
75%        0.81
max        1.00
Name: Profit_Margin, dtype: float64
```

[20]:
```
profitable_movies_df.loc[0:24, 'Profit_Margin'].describe()
```

[20]:
```
count   25.00
mean     0.85
std      0.05
min      0.74
25%      0.81
50%      0.85
75%      0.87
max      0.93
Name: Profit_Margin, dtype: float64
```

[21]:
```
len(profitable_ranked_df.loc[profitable_ranked_df['Profit_Margin'] > 0.5])
```

[21]:
```
2041
```

Clearly the most successful 25 movies have both incredible profits and profit margins. Titanic (1997), Avatar, and Avengers: Endgame are the most successful movies in terms of sheer profit.

So how do we know what to spend? We need to think about what sort of profit margin we want to see. 2043 out of 2841 total profitable movies have a profit margin over 50%. That's good news as it indicates that we can be more aggressive in choosing a threshold for the profit margin. The top 25 movies have a median profit margin of 84.9% with a median budget of \$225,760,000. When looking at all of our profitable movies, the profit margin drops significantly to 67.1% and the budget drops significantly to \$38,676,000. We use the median to describe our data here as the mean will be skewed by outlier data.

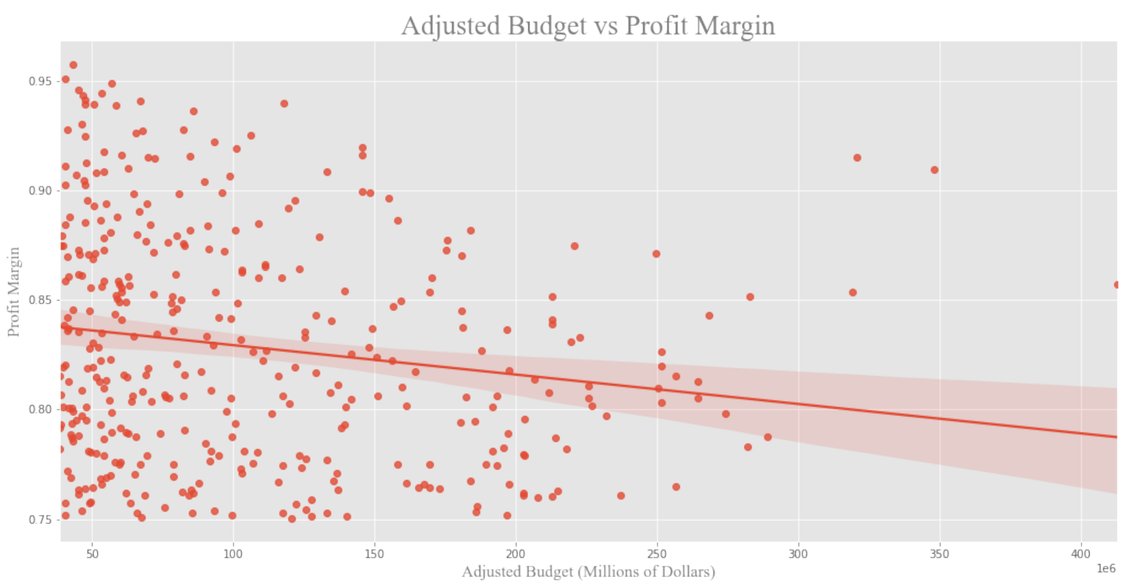
Let's filter the data with a profit margin of 75% or greater and a budget greater than $38,676,000.

[23]:
```
filtered_df = profitable_ranked_df.loc[(profitable_ranked_df['Profit_Margin']␣
↪>= 0.75) &
                                (profitable_ranked_df['Adjusted_Budget'] > 38676000)]
len(filtered_df)
```

[23]:
```
374
```

After filtering we still have 374 movies left upon which to draw conclusions.

[32]:
```
ax2 = sns.lmplot(x='Adjusted_Budget', y='Profit_Margin', data=filtered_df,␣
↪height=7, aspect=2)
plt.xlabel('Adjusted Budget (Millions of Dollars)', fontdict = {'fontname':␣
↪'Times New Roman', 'color': 'gray', 'fontsize' : '15'})
plt.ticklabel_format(axis='x', style='sci', scilimits=(6,6))
plt.ylabel('Profit Margin', fontdict = {'fontname': 'Times New Roman', 'color':␣
↪'gray', 'fontsize' : '15'})
plt.title('Adjusted Budget vs Profit Margin', fontdict = {'fontname': 'Times␣
↪New Roman', 'color': 'gray', 'fontsize' : '25'})
plt.savefig('BudgetVMargin', dpi=300);
```



[25]:
```
filtered_df.describe()
```

[25]:
```
         index    Year   IMDb  Runtime  Production Budget  Domestic Gross  \
count   374.00  374.00 374.00   374.00             374.00          374.00
```

```
mean    391.53 2004.97   7.01    118.60          77814178.13      193378841.67
std     378.20   10.81   0.90     24.02          57570152.51      127088965.57
min       0.00 1956.00   3.30     79.00          13500000.00       19019882.00
25%     111.25 1998.00   6.40    100.00          35000000.00      106948347.75
50%     279.50 2007.00   7.00    116.00          55000000.00      162801999.50
75%     550.50 2014.00   7.70    131.75         100000000.00      242081446.50
max    2424.00 2020.00   9.00    228.00         400000000.00      858373000.00

       Worldwide Gross         Profit  Profit_Margin  Adjusted_Budget  \
count           374.00         374.00         374.00           374.00
mean      484994903.63   407180725.50           0.83     105858522.51
std       377690264.14   329994078.69           0.05      66272237.80
min        69995385.00    54995385.00           0.75      38685000.00
25%       217288435.75   176354400.25           0.78      53471100.00
50%       350937609.00   299062980.00           0.82      82249300.00
75%       636084264.50   513979301.75           0.87     139654600.00
max      2797800564.00  2551701337.00           0.96     412880000.00

       Adjusted_Profit
count           374.00
mean      562879114.94
std       413114307.71
min       123209844.42
25%       274861614.08
50%       449229900.01
75%       719591073.46
max      3495487532.34
```

We examine the data in a scatter plot again to see if we can determine trends. Our data is much more spread out when comparing profit margin and budget. The trend line in this plot is negative which cautions against spending too much money as we may potentially hurt our profit margin. Looking at the filtered data, we have a median budget of $82,249,300 and a median profit margin of 81.9%.

Question 1 Conclusion: We recommend that our Company should budget approximately $82,250,000 to make a movie. This should correlate with a profit margin above 80%.

3 Question 2: Which movie genres are most commonly produced and does quantity equate to higher net profits?

[27]:
```
#Create a genre table that separates each value in the genre column in their␣
↪own rows.
imdb_budgets_df['Genre'] = imdb_budgets_df['Genre'].str.split(', ')
imdb_budgets_df1 = imdb_budgets_df['Genre'].apply(pd.Series)

imdb_budgets_df2 = pd.merge(imdb_budgets_df, imdb_budgets_df1, right_index =␣
↪True, left_index = True)
```

```
imdb_budgets_df3 = imdb_budgets_df2.drop(['Genre'], axis = 1)

genre_budgets_df = imdb_budgets_df3.melt(id_vars=['Movie', 'Year'],␣
↪value_vars=[0, 1, 2] ,var_name = ['X'])
genre_budgets_df = pd.merge(genre_budgets_df, imdb_budgets_df)
genre_budgets_df = genre_budgets_df.drop(['Genre', 'X'], axis=1)
genre_budgets_df = genre_budgets_df.drop_duplicates()
genre_budgets_df = genre_budgets_df.rename(columns={'value': 'Genre'})
genre_budgets_df = genre_budgets_df.dropna()
```

[28]:
```
#Do a count of all movies grouped by genre.
m_by_genre = genre_budgets_df.groupby('Genre', as_index=False)['Movie'].count().
↪sort_values(by='Movie', ascending=False)
```

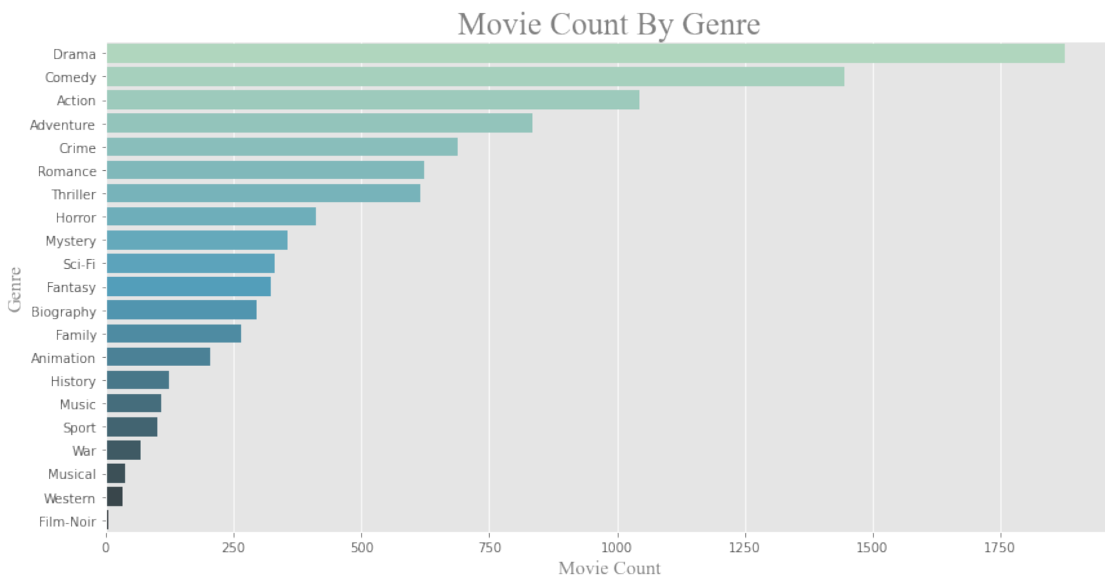
[29]:
```
m_by_genre
```

[29]:

	Genre	Movie
6	Drama	1876
4	Comedy	1444
0	Action	1045
1	Adventure	834
5	Crime	689
15	Romance	622
18	Thriller	615
11	Horror	410
14	Mystery	356
16	Sci-Fi	330
8	Fantasy	324
3	Biography	294
7	Family	265
2	Animation	205
10	History	123
12	Music	109
17	Sport	100
19	War	68
13	Musical	39
20	Western	32
9	Film-Noir	6

[31]:
```
#Plot the above findings.
plt.figure(figsize=(14,7))
ax3 = sns.barplot(x=m_by_genre['Movie'], y=m_by_genre['Genre'],␣
↪palette='GnBu_d')
plt.xlabel('Movie Count', fontdict = {'fontname': 'Times New Roman', 'color':␣
↪'gray', 'fontsize' : '15'})
```

```
plt.ylabel('Genre', fontdict = {'fontname': 'Times New Roman', 'color': 'gray',␣
    ↪'fontsize' : '15'})
plt.title('Movie Count By Genre', fontdict = {'fontname': 'Times New Roman',␣
    ↪'color': 'gray', 'fontsize' : '25'})
plt.savefig('CountGenre', dpi=300);
```



We can see that drama, comedy, and action dominate the quantity of movie genres but does this necessarily mean these are the most profitable genres? In order to determine this we will once again group each genre but this time we are going to take a look at the average net profit for each.

[35]:
```
#Once again group the movies by genre, showing the average net profit and␣
    ↪profit margin for each.
p_by_genre = genre_budgets_df.groupby('Genre',␣
    ↪as_index=False)[['Adjusted_Profit', 'Profit_Margin']].median().
    ↪sort_values(by='Adjusted_Profit', ascending=False)
```

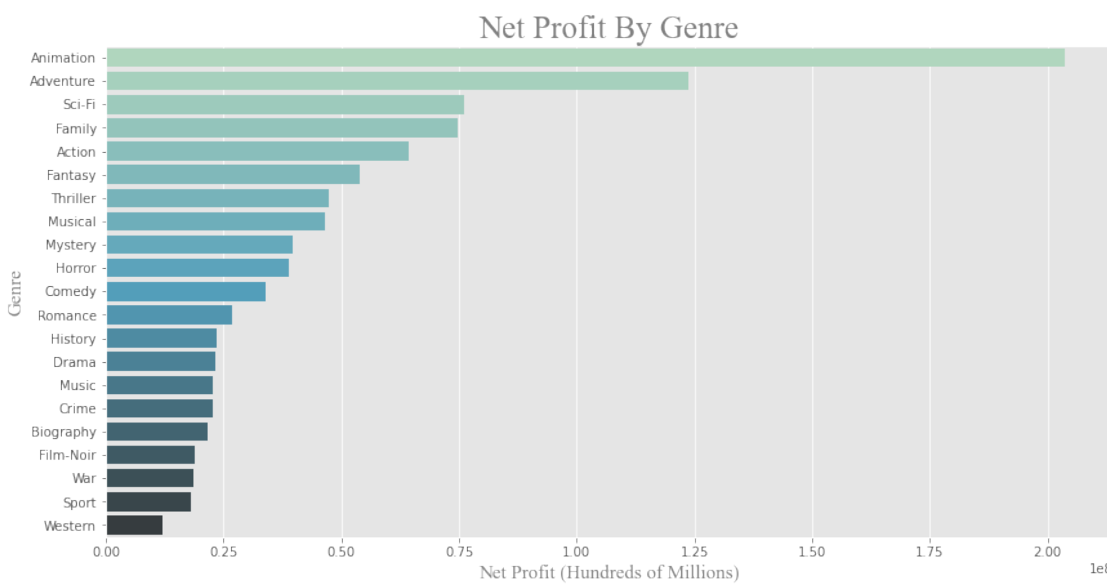
[36]:
```
p_by_genre
```

[36]:

	Genre	Adjusted_Profit	Profit_Margin
2	Animation	203606574.36	0.68
1	Adventure	123795016.96	0.61
16	Sci-Fi	76199115.79	0.60
7	Family	74621544.29	0.58
0	Action	64332532.19	0.52
8	Fantasy	54057582.24	0.54
18	Thriller	47338952.53	0.60
13	Musical	46631897.60	0.65
14	Mystery	39634323.82	0.61

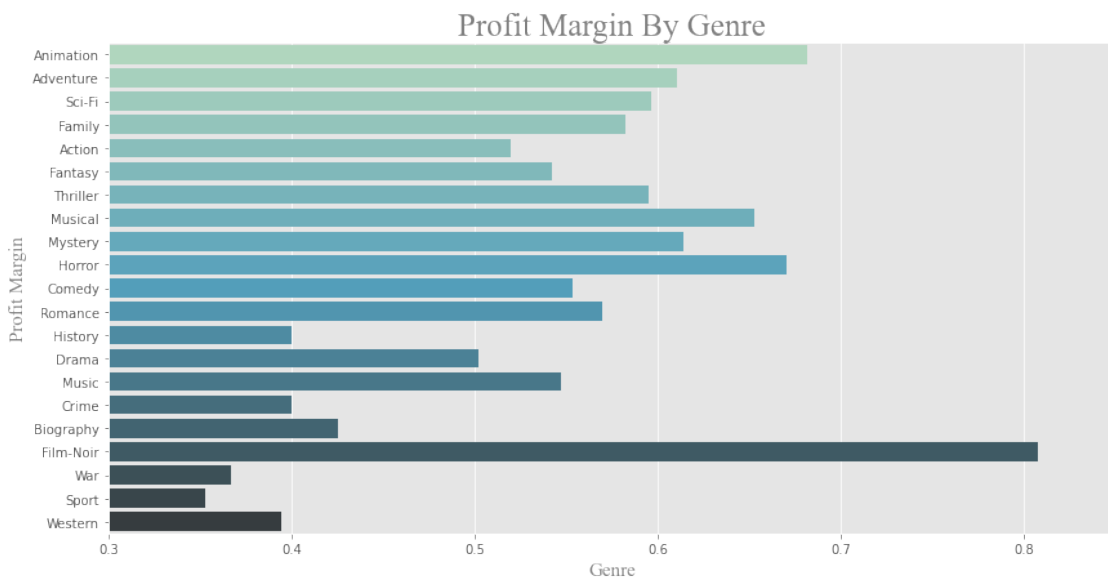
11	Horror	38963349.12	0.67
4	Comedy	33917454.39	0.55
15	Romance	26739545.09	0.57
10	History	23435554.73	0.40
6	Drama	23258412.08	0.50
12	Music	22774962.29	0.55
5	Crime	22752334.82	0.40
3	Biography	21750633.96	0.43
9	Film-Noir	18766783.04	0.81
19	War	18653512.63	0.37
17	Sport	17950554.99	0.35
20	Western	12037135.33	0.39

[37]:
```
#Plot the above findings.
plt.figure(figsize=(14,7))
ax4 = sns.barplot(x=p_by_genre['Adjusted_Profit'], y=p_by_genre['Genre'], palette='GnBu_d')
plt.xlabel('Net Profit (Hundreds of Millions)', fontdict = {'fontname': 'Times New Roman', 'color': 'gray', 'fontsize' : '15'})
plt.ylabel('Genre',fontdict = {'fontname': 'Times New Roman', 'color': 'gray', 'fontsize' : '15'})
plt.title('Net Profit By Genre', fontdict = {'fontname': 'Times New Roman', 'color': 'gray', 'fontsize' : '25'})
plt.savefig('NetProfitGenre', dpi=300);
```



[38]:
```
plt.figure(figsize=(14,7))
```

```
ax5 = sns.barplot(x=p_by_genre['Profit_Margin'], y=p_by_genre['Genre'],␣
↪palette='GnBu_d')
plt.xlabel('Genre', fontdict = {'fontname': 'Times New Roman', 'color': 'gray',␣
↪'fontsize' : '15'})
plt.ylabel('Profit Margin', fontdict = {'fontname': 'Times New Roman', 'color':␣
↪'gray', 'fontsize' : '15'})
plt.title('Profit Margin By Genre', fontdict = {'fontname': 'Times New Roman',␣
↪'color': 'gray', 'fontsize' : '25'})
plt.xlim(0.3, 0.85)
plt.savefig('ProfitMarginGenre', dpi=300);
```



Interesting, although they are not the most commonly released genres; animation, adventure, and sci-fi typically have the most success in terms of median net profit. We can also see that Animation has a desirable profit margin along with horror and musicals. Note: although Film Noir leads with a .8+ profit margin this is based on 6 movies and has to be disregarded due to the small sample size.

Lastly, of what percentage of the total net profit from all genres does each genre account?

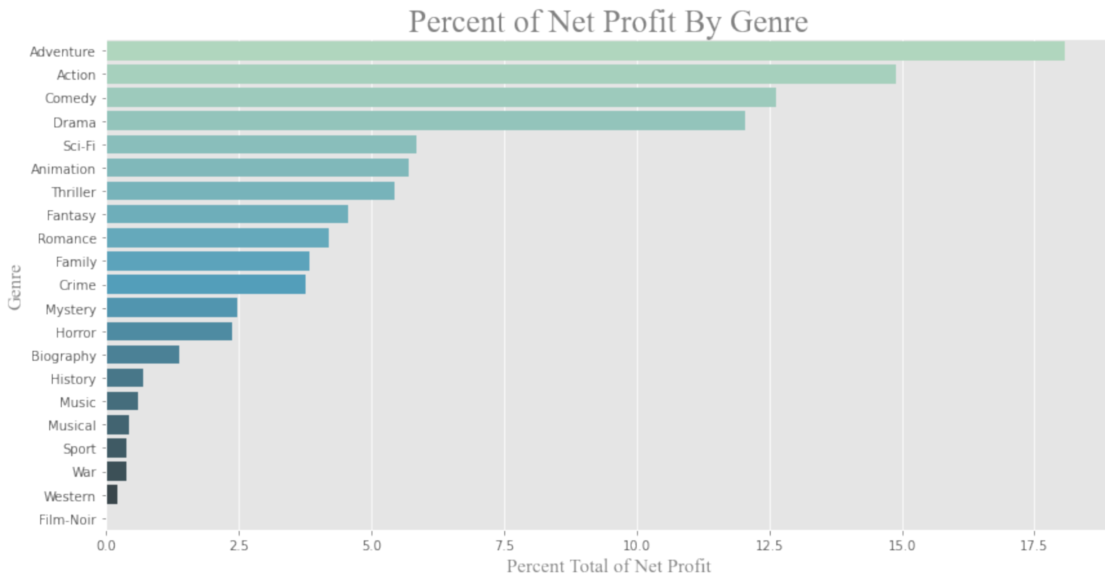
[39]:
```
#Grouped by genre, find the percent total of the net profit for each.
per_by_genre = genre_budgets_df.groupby(['Genre'], ␣
↪as_index=False)['Adjusted_Profit'].sum().sort_values(by='Adjusted_Profit',␣
↪ascending=False)
per_by_genre['Percent Total of Net Profit'] = (per_by_genre['Adjusted_Profit']/
↪per_by_genre['Adjusted_Profit'].sum()*100).round(2)
per_by_genre
```

[39]:

	Genre	Adjusted_Profit	Percent Total of Net Profit
1	Adventure	217335741708.40	18.07
0	Action	178930045524.32	14.88
4	Comedy	151922895671.69	12.63
6	Drama	144990041873.71	12.05
16	Sci-Fi	70465612908.78	5.86
2	Animation	68720987812.40	5.71
18	Thriller	65442236225.98	5.44
8	Fantasy	54797139085.80	4.56
15	Romance	50510744180.92	4.20
7	Family	46040638020.14	3.83
5	Crime	45194406614.69	3.76
14	Mystery	29903244700.35	2.49
11	Horror	28800384751.85	2.39
3	Biography	16776660619.24	1.39
10	History	8429562660.69	0.70
12	Music	7439929226.68	0.62
13	Musical	5228065825.20	0.43
17	Sport	4620549486.84	0.38
19	War	4619522490.02	0.38
20	Western	2551516786.77	0.21
9	Film-Noir	153313504.88	0.01

[40]:

```
#Plot the above findings.
plt.figure(figsize=(14,7))
ax6 = sns.barplot(x=per_by_genre['Percent Total of Net Profit'],
 ↪y=per_by_genre['Genre'], palette='GnBu_d')
plt.xlabel('Percent Total of Net Profit', fontdict = {'fontname': 'Times New
 ↪Roman', 'color': 'gray', 'fontsize' : '15'})
plt.ylabel('Genre', fontdict = {'fontname': 'Times New Roman', 'color': 'gray',
 ↪'fontsize' : '15'})
plt.title('Percent of Net Profit By Genre', fontdict = {'fontname': 'Times New
 ↪Roman', 'color': 'gray', 'fontsize' : '25'})
plt.savefig('PercentProfitGenre');
```



Now we can see that adventure, action, comedy and drama make up the lionshare of the overall net profits from all movies. However, from our recent observations we know there are also major opportunities in the animation and sci-fi markets due to lower saturation but high average net profits. We will soon determine which genres are most successful during which months.

Question 2 Conclusion: We recommend that our Company should focus their efforts on the top 6 most profitable movie genres: Adventure, Action, Comedy, Drama, Sci-Fi and Animation. A further recommendation to focus on Sci-Fi and Animation due to less competition and a higher opportunity to profit.

4 Question 3: What is the best time of the year to release a movie?

[41]:
```
#Convert the Release Date field to type datetime.
imdb_budgets_df['Release Date'] = pd.to_datetime(imdb_budgets_df['Release␣
↪Date'])
```

[42]:
```
#Add a new column called month, displaying only the month from the release date.
dateData =  [x.strftime('%B') for x in imdb_budgets_df['Release Date']]
imdb_budgets_df['Month'] = dateData
```

Let's first start by determing which months see the most movie releases.

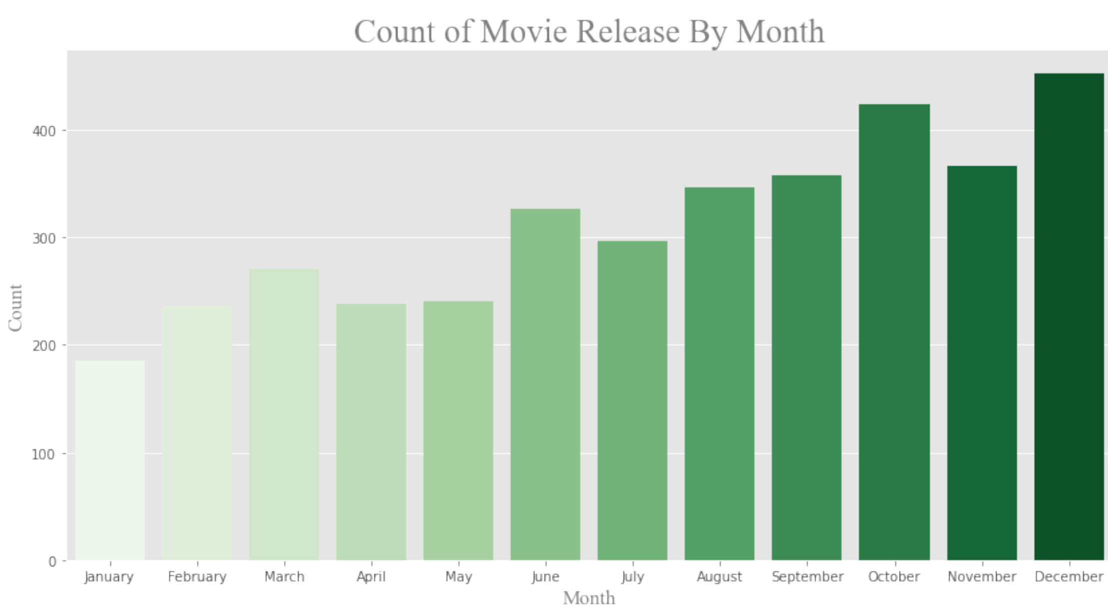
[43]:
```
#Count the total number of movies and group by month.
m_by_month = imdb_budgets_df.groupby(['Month'], as_index=False)['Movie'].
↪count().sort_values(by='Movie', ascending=False)
m_by_month
```

[43]:

	Month	Movie
2	December	452
10	October	424
9	November	366
11	September	358
1	August	346
6	June	327
5	July	296
7	March	270
8	May	241
0	April	238
3	February	236
4	January	186

[44]:

```
#Plot the above findings in order by month.
plt.figure(figsize=(14,7))
ax7 = sns.countplot(x=imdb_budgets_df['Month'], palette='Greens',
                    order=['January', 'February', 'March', 'April', 'May',␣
 ↪'June', 'July', 'August', 'September', 'October', 'November', 'December'])
plt.xlabel('Month', fontdict = {'fontname': 'Times New Roman', 'color': 'gray',␣
 ↪'fontsize' : '15'})
plt.ylabel('Count', fontdict = {'fontname': 'Times New Roman', 'color': 'gray',␣
 ↪'fontsize' : '15'})
plt.title('Count of Movie Release By Month', fontdict = {'fontname': 'Times New␣
 ↪Roman', 'color': 'gray', 'fontsize' : '25'})
plt.savefig('CountbyMonth', dpi=300);
```



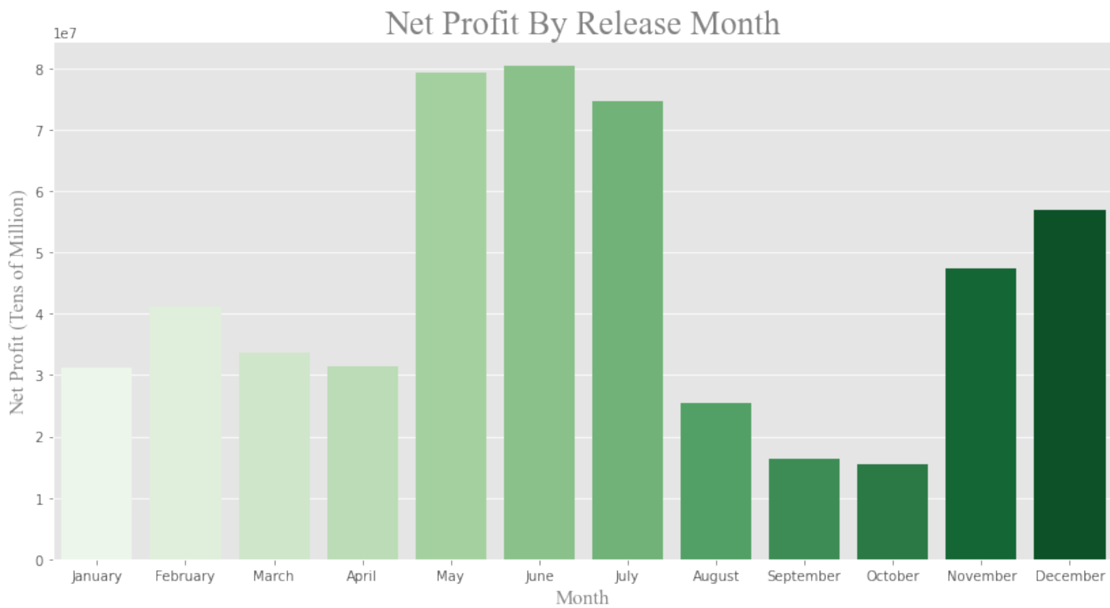
As you can see Decemeber and October lead the way in terms of sheer quantity of movies but does this suggest a higher level of profitability? Next we will look into the average net income by movie for each month.

[45]:
```
#Once again group the movies by month, showing the average net profit for each.
p_by_month = imdb_budgets_df.groupby('Month',␣
↪as_index=False)[['Adjusted_Profit', 'Profit_Margin']].median().
↪sort_values(by='Adjusted_Profit', ascending=False)
p_by_month
```

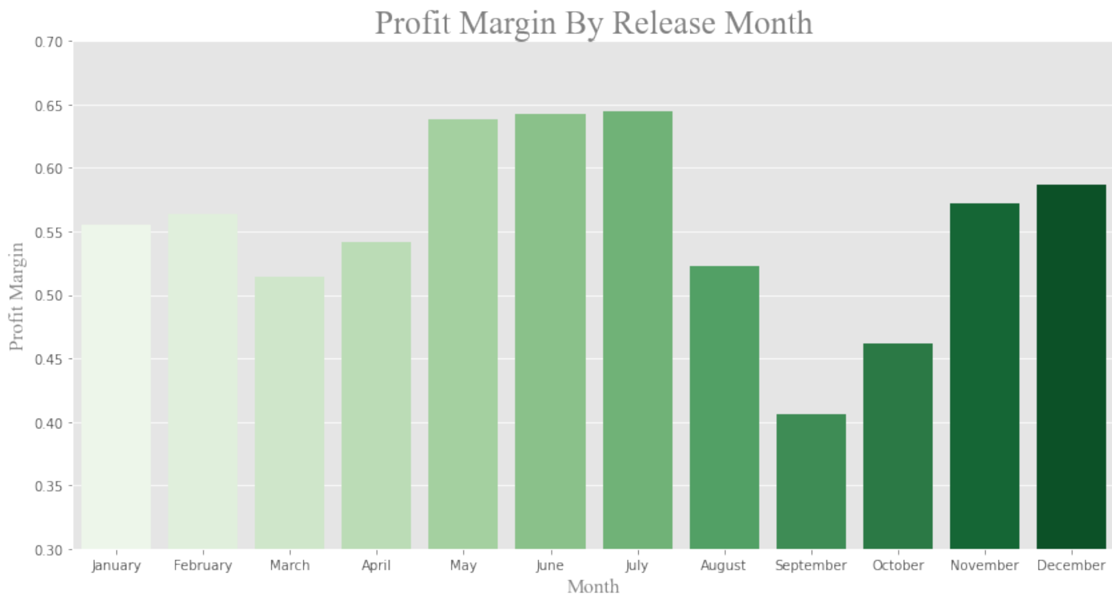
[45]:

	Month	Adjusted_Profit	Profit_Margin
6	June	80327640.00	0.64
8	May	79372161.65	0.64
5	July	74716618.14	0.64
2	December	56823086.46	0.59
9	November	47476647.51	0.57
3	February	41089454.38	0.56
7	March	33645813.78	0.51
0	April	31435638.57	0.54
4	January	31132342.98	0.56
1	August	25383311.33	0.52
11	September	16430952.78	0.41
10	October	15579534.04	0.46

[46]:
```
#Plot your above findings in order by month.
plt.figure(figsize=(14,7))
ax8 = sns.barplot(x=p_by_month['Month'], y=p_by_month['Adjusted_Profit'],␣
↪palette='Greens',
                  order=['January', 'February', 'March', 'April', 'May',␣
↪'June', 'July', 'August', 'September', 'October', 'November', 'December'])
plt.xlabel('Month', fontdict = {'fontname': 'Times New Roman', 'color': 'gray',␣
↪'fontsize' : '15'})
plt.ylabel('Net Profit (Tens of Million)', fontdict = {'fontname': 'Times New␣
↪Roman', 'color': 'gray', 'fontsize' : '15'})
plt.title('Net Profit By Release Month', fontdict = {'fontname': 'Times New␣
↪Roman', 'color': 'gray', 'fontsize' : '25'})
plt.savefig('ProfitbyMonth', dpi=300);
```



[47]:
```
plt.figure(figsize=(14,7))
ax9 = sns.barplot(x=p_by_month['Month'], y=p_by_month['Profit_Margin'],␣
↪palette='Greens',
                  order=['January', 'February', 'March', 'April', 'May',␣
↪'June', 'July', 'August', 'September', 'October', 'November', 'December'])
plt.xlabel('Month', fontdict = {'fontname': 'Times New Roman', 'color': 'gray',␣
↪'fontsize' : '15'})
plt.ylabel('Profit Margin', fontdict = {'fontname': 'Times New Roman', 'color':␣
↪'gray', 'fontsize' : '15'})
plt.title('Profit Margin By Release Month', fontdict = {'fontname': 'Times New␣
↪Roman', 'color': 'gray', 'fontsize' : '25'})
plt.ylim(0.3, 0.7)
plt.savefig('MarginByMonth', dpi=300);
```



Interestingly, May, June and July shoot to the top in terms of both median net profit and profit margin. It appears that the summer months tend to result in greater success, perhaps as a result of an influx of children and their parents during summer break. Now as previously mentioned, let's dig a little further and see which genre tends to do the best in which month.

[49]:
```
#Convert the Release Date field to type datetime
#Add a new column called month, displaying only the month from the release date.
genre_budgets_df['Release Date'] = pd.to_datetime(genre_budgets_df['Release␣
↪Date'])
genreDate =  [x.strftime('%B') for x in genre_budgets_df['Release Date']]
genre_budgets_df['Month'] = genreDate
```

[50]:
```
#Create a new table called month_genre consisting of Genre, Month, Net Profit,␣
↪and Release Date
month_genre = genre_budgets_df[['Genre', 'Month', 'Adjusted_Profit', 'Release␣
↪Date']]
#Group by Genre and Month, displaying the average Net Profit for each␣
↪combination.
month_genre = month_genre.groupby(['Genre', 'Month'],␣
↪as_index=False)['Adjusted_Profit'].mean().sort_values(by='Adjusted_Profit',␣
↪ascending=False)
```

[51]:
```
#Slice the top six most profitable genres from above.
Adventure_df = month_genre.loc[month_genre['Genre'].str.contains('Adventure')]
Action_df = month_genre.loc[month_genre['Genre'].str.contains('Action')]
Comedy_df = month_genre.loc[month_genre['Genre'].str.contains('Comedy')]
Drama_df = month_genre.loc[month_genre['Genre'].str.contains('Drama')]
```

```python
Scifi_df = month_genre.loc[month_genre['Genre'].str.contains('Sci-Fi')]
Animation_df = month_genre.loc[month_genre['Genre'].str.contains('Animation')]
```

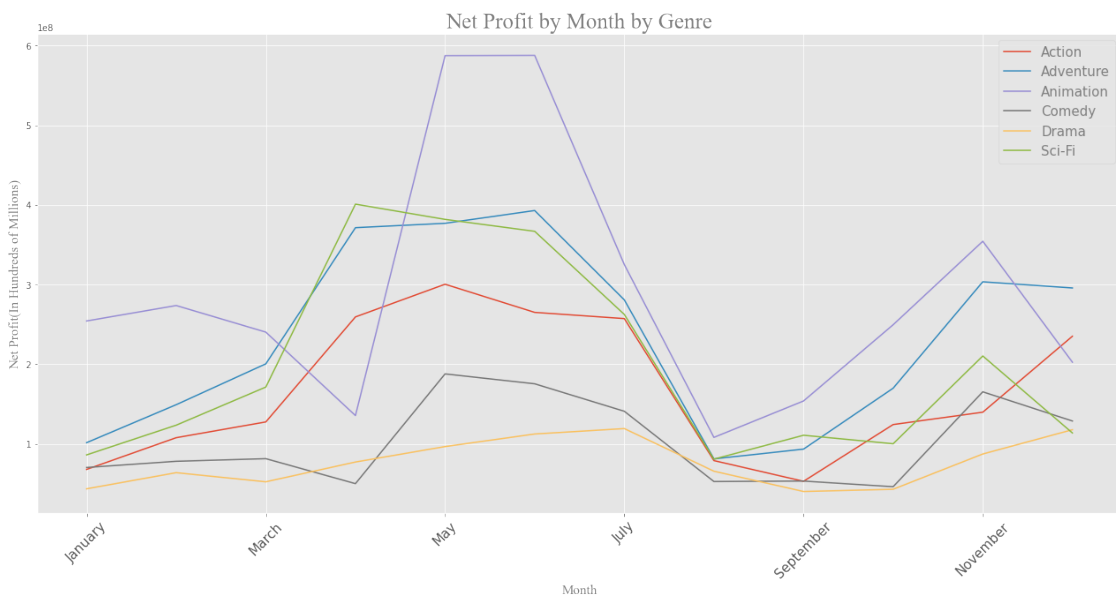
[52]:
```python
#Concatenate the six new tables into one new table.
genre_concat = [Adventure_df, Action_df, Comedy_df, Drama_df, Scifi_df, Animation_df]
month_genre_df = pd.concat(genre_concat)
```

[53]:
```python
#Create a table of the months in order.
months_in_order = ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', 'September', 'October', 'November', 'December']
#Create a pivot table of month_genre_df, use the month_in_order table to reindex the pivot table.
month_genre_pivoted = month_genre_df.pivot(index='Month', columns='Genre', values='Adjusted_Profit').reindex(months_in_order)
```

[54]:
```python
month_genre_pivoted
```

[54]:
```
Genre             Action     Adventure     Animation        Comedy         Drama  \
Month
January      67911226.86  101480251.68  254304586.21   70321717.64   43539017.01
February    107741220.58  149172991.22  273699863.40   78129901.96   63807537.49
March       127548996.11  200474749.59  240295152.35   81411129.63   52348133.09
April       259392394.58  371426341.09  135514583.52   50050513.61   77199294.63
May         300431780.23  376946029.72  587476204.76  187839907.64   96590740.22
June        265101499.32  392963586.66  587763663.68  175416615.42  112382070.55
July        257293527.76  280812330.30  325184250.83  140927144.14  119198995.62
August       78993517.46   81128041.19  108115881.94   52702618.10   65637106.34
September    52980175.19   93388465.69  153847514.52   53288686.20   40194497.00
October     124257794.43  169896169.96  249582645.96   46177500.88   42992650.53
November    139749410.88  303503861.24  354381890.29  165340406.04   87265604.53
December    235113158.91  295732977.48  202553251.30  128699177.32  117758948.19

Genre             Sci-Fi
Month
January      86131136.28
February    123463145.04
March       171335731.24
April       400992743.36
May         381838680.03
June        366873462.47
July        262513716.23
August       80812011.13
September   110804792.63
October     100120506.83
November    210336333.85
December    113695722.89
```

[63]:
```
#Visualize the top 6 most profitable genre's by month
ax10 = month_genre_pivoted.plot(kind='line', figsize=(22, 10), rot=0)
plt.legend(labelcolor='grey', loc='best', prop={'size': 15})
plt.xlabel('Month', fontdict = {'fontname': 'Times New Roman', 'color': 'gray',␣
↪'fontsize' : '15'})
plt.ylabel('Net Profit(In Hundreds of Millions)', fontdict = {'fontname':␣
↪'Times New Roman', 'color': 'gray', 'fontsize' : '15'})
plt.title('Net Profit by Month by Genre', fontdict = {'fontname': 'Times New␣
↪Roman', 'color': 'gray', 'fontsize' : '25'})
plt.xticks(fontsize=15, rotation=45)
plt.savefig('ProfitbyMonthbyGenre', dpi=300);
```



We can see that each genre follows the same basic pattern, with the summer months proving to be the most profitable time to release a movie. Some further analysis shows that releasing an animation movie in particular during the summer months will have the greatest potential for high net profits. On the other hand drama, although fluctuates slightly with the months, tends to have no impact based on release date. When considering what aspects go into creating a successful movie, it's clear that one must take take into account the impact of a well timed release date.

Question 3 Conclusion: We recommend that our Company release the bulk of their movies, especially Animation, during the summer months. Adventure, Drama and Comedy movies would see similar success if released in November, but the recommendation remains to focus on summer.

5 Question 4: Now that we've got a better understanding of what attributes to a successful movie, which actors and directors tend to add the most value?

In this section we are going to take a look at the average net profit across all movies. From there we want to determine which actors and directors consistently appear in movies where the net profit substantially exceeds the average. We will represent this in a field called Value Above Replacement(VAR). To further simplify this concept; if across all movies the average net profit is 100 dollars and the average net profit of movies from 'Actor: X' is 200 dollars he/she would have a VAR of 2. This number represents X times over the average. To eliminate outliers we will look at actors who appear in 10 or more movies and directors who work in 5 or more.

[64]:
```
#Similar to the imdb_budget_df table let's start by adjusting for inflation.
actors_df['Production Budget'] = (((2020-actors_df['Year'])*.
 ↪0322)+1)*actors_df['Production Budget']
actors_df['Worldwide Gross'] = (((2020-actors_df['Year'])*.
 ↪0322)+1)*actors_df['Worldwide Gross']
actors_df['Domestic Gross'] = (((2020-actors_df['Year'])*.
 ↪0322)+1)*actors_df['Domestic Gross']
```

[65]:
```
#Calculate Net Profit and Profit Margin
actors_df['Net Profit'] = actors_df['Worldwide Gross'] - actors_df['Production␣
 ↪Budget']
actors_df['Profit Margin'] = actors_df['Net Profit'] / actors_df['Worldwide␣
 ↪Gross']
```

[66]:
```
#Let's filter the actors_df table to only include actors that appeared in 10 or␣
 ↪more movies
actor_counts = actors_df['value'].value_counts()
actor_list = actor_counts[actor_counts >= 10].index.tolist()
actors_df = actors_df[actors_df['value'].isin(actor_list)]
```

[67]:
```
#Calculate VAR, which is the average Net Profit by actor divided by average Net␣
 ↪Profit for all movies.
actor_total = actors_df.groupby(['value'], as_index=False)['Net Profit'].
 ↪mean().sort_values(by='Net Profit', ascending=False)
actor_total['VAR'] = (actor_total['Net Profit']/actor_total['Net Profit'].
 ↪mean())
```

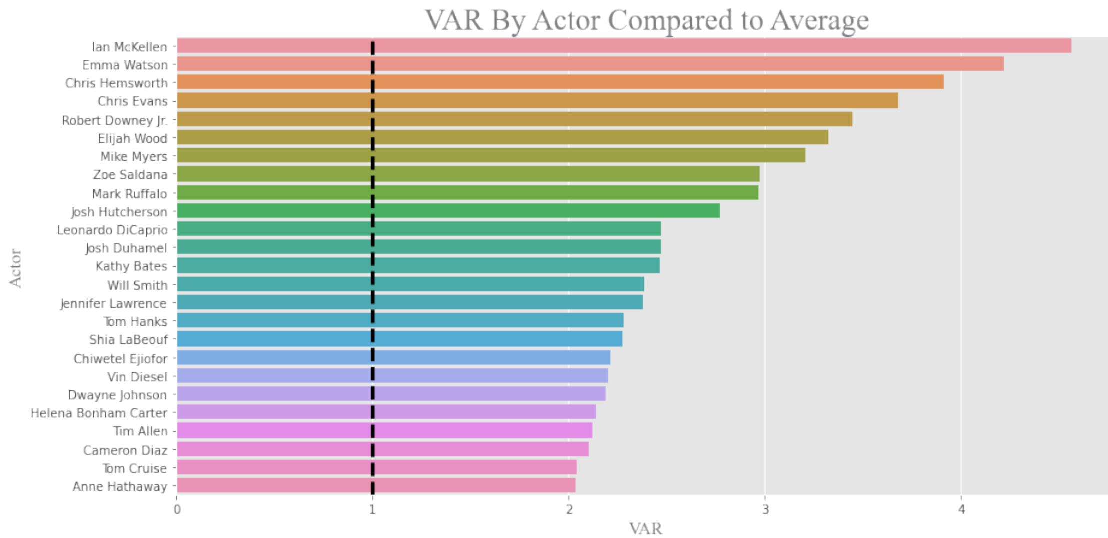
[68]:
```
#Create new table consisting of top 25 actors by VAR.
top_actors = actor_total.head(25)
top_actors
```

[68]:

	value	Net Profit	VAR
113	Ian McKellen	642641141.05	4.56
88	Emma Watson	594070330.59	4.22

```
48         Chris Hemsworth 550993070.74 3.91
47             Chris Evans 518397913.83 3.68
262      Robert Downey Jr. 484884995.15 3.44
82             Elijah Wood 468414890.65 3.33
227             Mike Myers 451615981.41 3.21
324            Zoe Saldana 418413981.69 2.97
205           Mark Ruffalo 418051684.80 2.97
166        Josh Hutcherson 389946768.85 2.77
197      Leonardo DiCaprio 347929775.33 2.47
164           Josh Duhamel 347668686.44 2.47
178            Kathy Bates 347201332.37 2.47
316             Will Smith 336002549.53 2.39
138      Jennifer Lawrence 334744177.49 2.38
299              Tom Hanks 320791739.32 2.28
285           Shia LaBeouf 320522135.54 2.28
45        Chiwetel Ejiofor 311862722.21 2.21
308             Vin Diesel 309819051.08 2.20
78          Dwayne Johnson 308538514.10 2.19
108   Helena Bonham Carter 301712229.56 2.14
296              Tim Allen 298679367.49 2.12
36            Cameron Diaz 295720384.55 2.10
298             Tom Cruise 287290600.79 2.04
13           Anne Hathaway 286762937.70 2.04
```

[70]:
```
#Plot above finding and label the average of 1 with a black line.
plt.figure(figsize=(14,7))
ax11 = sns.barplot(x=top_actors['VAR'], y=top_actors['value'])
plt.axvline(1, ls='--', color='black', linewidth=3)
plt.xlabel('VAR', fontdict = {'fontname': 'Times New Roman', 'color': 'gray',␣
 ↪'fontsize' : '15'})
plt.ylabel('Actor', fontdict = {'fontname': 'Times New Roman', 'color': 'gray',␣
 ↪'fontsize' : '15'})
plt.title('VAR By Actor Compared to Average', fontdict = {'fontname': 'Times␣
 ↪New Roman', 'color': 'gray', 'fontsize' : '25'})
plt.savefig('VARActor', dpi=300);
```



Wow, from this list we can see that all of these actors consistently appear in very profitable movies; anywhere from two times the norm to four and a half times the norm. When casting a movie this is a good short-list from where to start making calls.

[71]:
```
#Adjust directors table for inflation.
directors_df['Production Budget'] = (((2020-directors_df['Year'])*.
↪0322)+1)*directors_df['Production Budget']
directors_df['Worldwide Gross'] = (((2020-directors_df['Year'])*.
↪0322)+1)*directors_df['Worldwide Gross']
directors_df['Domestic Gross'] = (((2020-directors_df['Year'])*.
↪0322)+1)*directors_df['Domestic Gross']
```

[72]:
```
#Calucalte Net Profit and Profit Margin.
directors_df['Net Profit'] = directors_df['Worldwide Gross'] -␣
↪directors_df['Production Budget']
directors_df['Profit Margin'] = directors_df['Net Profit'] /␣
↪directors_df['Worldwide Gross']
```

[73]:
```
#Let's filter the actors_df table to only include actors that appeared in 5 or␣
↪more movies.
director_counts = directors_df['value'].value_counts()
director_list = director_counts[director_counts >= 5].index.tolist()
directors_df = directors_df[directors_df['value'].isin(director_list)]
```

[74]:
```
#Calculate VAR, which is the average Net Profit by director divided by average␣
↪Net Profit for all movies.
director_total = directors_df.groupby(['value'], as_index=False)['Net Profit'].
↪mean().sort_values(by='Net Profit', ascending=False)
```

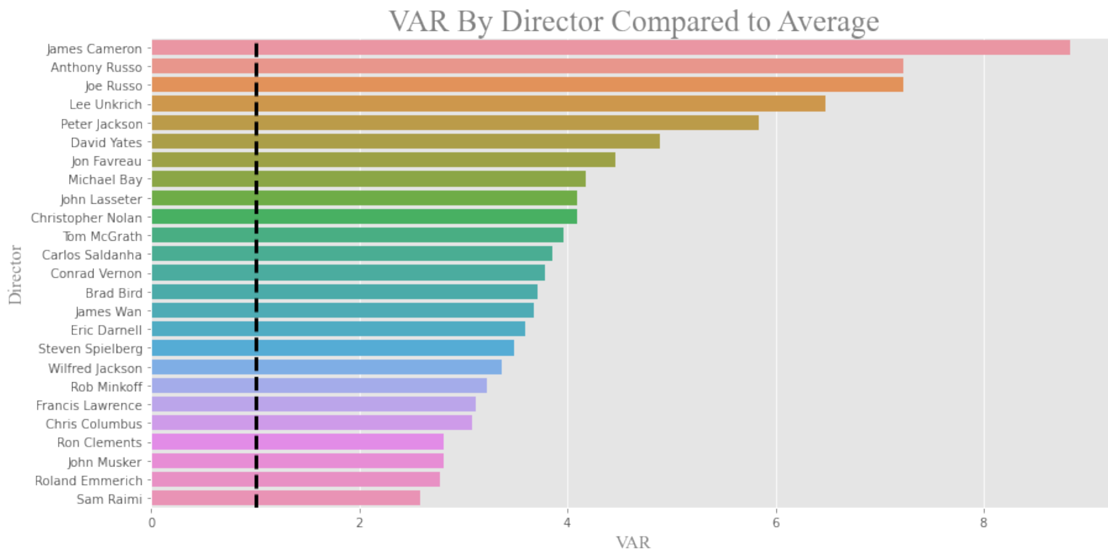
```
director_total['VAR'] = (director_total['Net Profit']/actor_total['Net Profit'].
 ↪mean())
```

[75]:
```
#Create new table consisting of top 25 directors by VAR.
top_directors = director_total.head(25)
top_directors
```

[75]:

	value	Net Profit	VAR
78	James Cameron	1244750157.55	8.84
11	Anthony Russo	1017389415.62	7.22
89	Joe Russo	1017389415.62	7.22
115	Lee Unkrich	912067911.25	6.48
148	Peter Jackson	821878024.53	5.84
50	David Yates	688135205.04	4.89
104	Jon Favreau	628704113.52	4.46
129	Michael Bay	588804626.49	4.18
96	John Lasseter	577254528.66	4.10
31	Christopher Nolan	576508914.30	4.09
194	Tom McGrath	558026757.25	3.96
27	Carlos Saldanha	542327603.19	3.85
34	Conrad Vernon	533554799.18	3.79
19	Brad Bird	522918604.82	3.71
82	James Wan	517843475.89	3.68
58	Eric Darnell	506570978.60	3.60
188	Steven Spielberg	490403244.69	3.48
200	Wilfred Jackson	473675805.64	3.36
160	Rob Minkoff	453631830.01	3.22
62	Francis Lawrence	439117499.61	3.12
29	Chris Columbus	434315443.48	3.08
171	Ron Clements	396185896.16	2.81
101	John Musker	396185896.16	2.81
169	Roland Emmerich	391218701.44	2.78
175	Sam Raimi	364101893.22	2.59

[76]:
```
#Plot above finding and label the average of 1 with a black line.
plt.figure(figsize=(14,7))
ax12 = sns.barplot(x=top_directors['VAR'], y=top_directors['value'])
plt.axvline(1, ls='--', color='black', linewidth=3)
plt.xlabel('VAR', fontdict = {'fontname': 'Times New Roman', 'color': 'gray',␣
 ↪'fontsize' : '15'})
plt.ylabel('Director', fontdict = {'fontname': 'Times New Roman', 'color':␣
 ↪'gray', 'fontsize' : '15'})
plt.title('VAR By Director Compared to Average', fontdict = {'fontname': 'Times␣
 ↪New Roman', 'color': 'gray', 'fontsize' : '25'})
plt.savefig('VARDirector', dpi=300);
```



It appears that the most significant value added comes from the directors chair. James Cameron movies on average make almost nine times the amount of the average movie, this emphasizes what great leadership represents on a set. If we wanted to further investigate which actors and directors make the most impact it would be important to determine which genre of movies they appear in most.

Question 4 Conclusion: We recommend that our Comapany focus their cast and crew search to individuals who consistently score at least 1.0 on the VAR score. We can, with a high level of confidence, conclude that these individuals will elevate the overall production.

6 Question 5: How much should you spend on a movie to win an Oscar?

In order to answer this question we'll first need to join the `imdb_budgets_df` dataframe and the `awards_df` dataframe. As there may be movies with duplicate titles, we set the indices of both dataframes to the movie name and year so that matching data is correctly joined.

[77]:
```
imdb_budgets_df.set_index(['Movie','Year'], inplace=True)
awards_df.set_index(['film_name', 'film_year'], inplace=True)
```

[78]:
```
budgets_and_awards = imdb_budgets_df.join(awards_df, how='inner', on=['Movie',␣
↪'Year'])
budgets_and_awards.head()
```

[78]:
```
                          IMDb Rating  Runtime                          Genre  \
Movie           Year
Avatar          2009   7.80   PG-13      162    [Action, Adventure, Fantasy]
Black Panther   2018   7.30   PG-13      134     [Action, Adventure, Sci-Fi]
Titanic         1997   7.80   PG-13      194               [Drama, Romance]
```

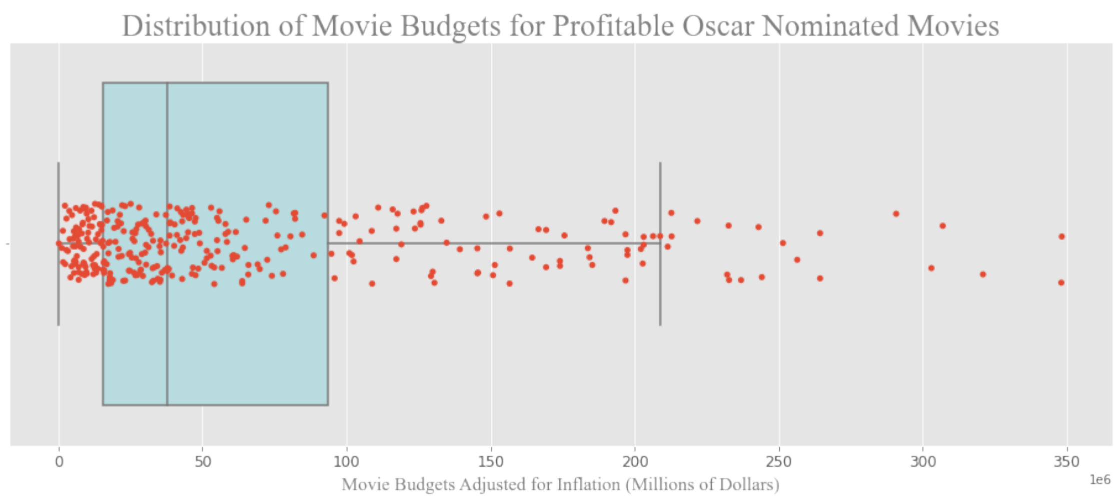
```
The Dark Knight 2008   9.00  PG-13      152          [Action, Crime, Drama]
Toy Story 4     2019   7.80      G      100  [Animation, Adventure, Comedy]

                     Release Date  Production Budget  Domestic Gross  \
Movie           Year
Avatar          2009   2009-12-17          237000000       760507625
Black Panther   2018   2018-02-13          200000000       700059566
Titanic         1997   1997-12-18          200000000       659363944
The Dark Knight 2008   2008-07-11          185000000       533720947
Toy Story 4     2019   2019-06-20          200000000       434038008

                      Worldwide Gross      Profit  Profit_Margin  \
Movie           Year
Avatar          2009       2788701337  2551701337           0.92
Black Panther   2018       1346103376  1146103376           0.85
Titanic         1997       2208208395  2008208395           0.91
The Dark Knight 2008       1000742751   815742751           0.82
Toy Story 4     2019       1073394813   873394813           0.81

                      Adjusted_Budget  Adjusted_Profit     Month  awards_won  \
Movie           Year
Avatar          2009     320945400.00    3455513950.57  December           3
Black Panther   2018     212880000.00    1219912433.41  February           3
Titanic         1997     348120000.00    3495487532.34  December          11
The Dark Knight 2008     256484000.00    1130945749.99      July           2
Toy Story 4     2019     206440000.00     901518125.98      June           1

                      awards_nominated  win_rate
Movie           Year
Avatar          2009                 9      0.33
Black Panther   2018                 7      0.43
Titanic         1997                14      0.79
The Dark Knight 2008                 8      0.25
Toy Story 4     2019                 2      0.50
```

We've successfully joined the two dataframes. Let's filter the dataframe to include movies where the profit is greater than 0.

[79]:
```python
nominated_movies_df = budgets_and_awards.loc[budgets_and_awards['Profit'] > 0]
```

[82]:
```python
plt.figure(figsize=(16,6))
sns.boxplot(x='Adjusted_Budget', data=nominated_movies_df, showfliers=False,␣
 ↪color='powderblue')
sns.stripplot(x='Adjusted_Budget', data=nominated_movies_df)
plt.ticklabel_format(axis='x', style='sci', scilimits=(6,6))
plt.xticks(fontsize=12)
```

```
plt.xlabel('Movie Budgets Adjusted for Inflation (Millions of Dollars)',␣
 ↪fontdict = {'fontname': 'Times New Roman', 'color': 'gray', 'fontsize' :␣
 ↪'15'});
plt.title('Distribution of Movie Budgets for Profitable Oscar Nominated␣
 ↪Movies', fontdict = {'fontname': 'Times New Roman', 'color': 'gray',␣
 ↪'fontsize' : '25'})
plt.savefig('Oscar_Nominated', dpi=300);
```



[83]:
```
nominated_movies_df['Adjusted_Budget'].describe()
```

[83]:
```
count          331.00
mean      66479336.13
std       72497186.73
min         212790.00
25%       15425660.00
50%       37816500.00
75%       93598000.00
max      348300000.00
Name: Adjusted_Budget, dtype: float64
```

By looking at the distribution of movie budgets we see that the majority of data is clustered in an area below \$100 million dollars.

We need to take this a step further as the above distribution includes movies that were nominated and won awards as well as movies that did not win awards. In order to properly answer our question we must win an Oscar.

We could filter by win rate and exclude those movies that did not win anything, however our data would still include movies that were nominated in a single category and won. This would skew the win rate as there would be several movies with a win rate of 100%. Let's take a look at the mean and median win rate to establish a threshold for award nominations.

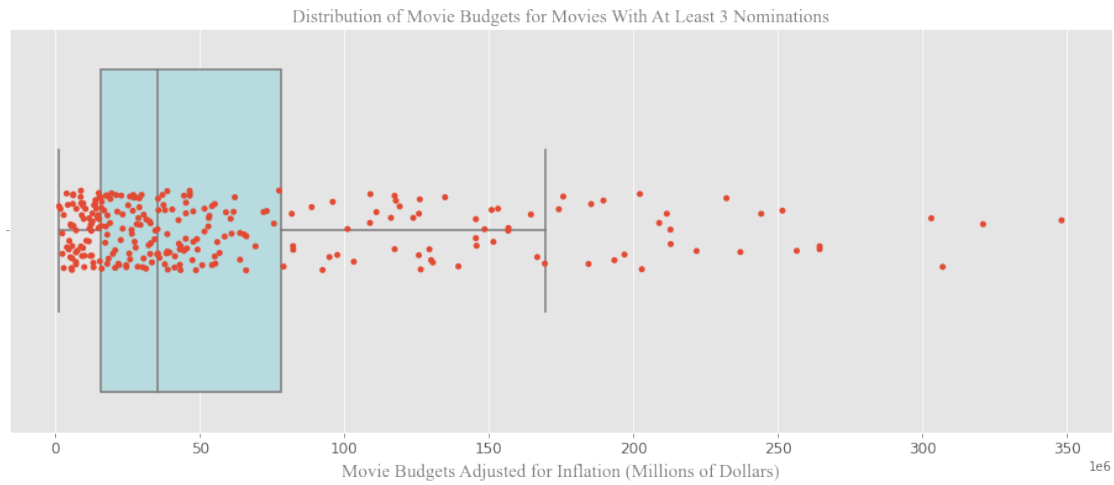
[84]:
```
nominated_movies_df['win_rate'].describe()
#Let's be conservative for win rate and use the median win rate
#That means we would need to be nominated for at least 3 awards in order to win␣
↪1 award.
```

[84]:
```
count    330.00
mean       0.45
std        0.28
min        0.00
25%        0.25
50%        0.39
75%        0.60
max        1.00
Name: win_rate, dtype: float64
```

The mean win rate is 44.8% but as we mentioned is skewed by those movies with only 1 nomination. The median win rate is 39.2% which should be less skewed by the data and is a more conservative number. Using the median win rate of 39.2%, our movie would need to be nominated for at least 3 awards in order to get at least one win. 3 nominations will be the cutoff.

[85]:
```
nominated_over_three = nominated_movies_df.
↪loc[nominated_movies_df['awards_nominated'] >= 3]
print(len(nominated_over_three))
plt.figure(figsize=(16,6))
sns.boxplot(x=nominated_over_three['Adjusted_Budget'], showfliers=False,␣
↪color='powderblue')
sns.stripplot(x='Adjusted_Budget', data=nominated_over_three)
plt.ticklabel_format(axis='x', style='sci', scilimits=(6,6))
plt.xticks(fontsize=12)
plt.xlabel('Movie Budgets Adjusted for Inflation (Millions of Dollars)',␣
↪fontdict = {'fontname': 'Times New Roman', 'color': 'gray', 'fontsize' :␣
↪'15'})
plt.title('Distribution of Movie Budgets for Movies With At Least 3␣
↪Nominations', fontdict = {'fontname': 'Times New Roman', 'color': 'gray',␣
↪'fontsize' : '15'})
plt.savefig('3_Nominations', dpi=300);
```

```
263
```




```
[86]: nominated_over_three['Adjusted_Budget'].describe()
```

```
[86]: count          263.00
      mean      62404651.14
      std       69126844.12
      min        1224990.00
      25%       15482900.00
      50%       35465000.00
      75%       78132000.00
      max      348120000.00
      Name: Adjusted_Budget, dtype: float64
```

It's important to note that the box plot of the `nominated_over_three` dataframe has shrunk! This means that our filter has decreased our interquartile range for the movie budget. Since this range is smaller there should be less variability in the middle of the data set. Since we have adjusted budgets that are extreme outliers, it is best to use the median as the primary measure of central tendency. The median adjusted budget for this data is \$35,465,000.

Question 5 Conclusion: Our Company should spend at least $35,465,000 in order to make an Oscar-winning movie.

It is also worth noting that the 75th percentile of the adjusted budget for movies with at least three nominations is $78,132,000. This is close to our recommendation of a \$82 million budget for a profitable movie with a profit margin of approximately 80%.

7 Question 6: What impact, if any, does runtime and movie rating have on Net Profit, Profit Margin and IMDb rating?

Let's first start by analyzing the ratings. We want to include only movies rated G, PG, PG-13 or R.

[87]:
```python
rating_counts = imdb_budgets_df['Rating'].value_counts()
rating_list = rating_counts[rating_counts >= 50].index.tolist()
rating_df = imdb_budgets_df[imdb_budgets_df['Rating'].isin(rating_list)]
```

[88]:
```python
rating_df = rating_df.reset_index()
rating_df.head()
```

[88]:
```
                    Movie  Year  IMDb Rating  Runtime  \
0       Avengers: Endgame  2019  8.40  PG-13      181
1                  Avatar  2009  7.80  PG-13      162
2           Black Panther  2018  7.30  PG-13      134
3  Avengers: Infinity War  2018  8.40  PG-13      149
4                 Titanic  1997  7.80  PG-13      194

                             Genre Release Date  Production Budget  \
0     [Action, Adventure, Drama]   2019-04-23          400000000
1   [Action, Adventure, Fantasy]   2009-12-17          237000000
2    [Action, Adventure, Sci-Fi]   2018-02-13          200000000
3    [Action, Adventure, Sci-Fi]   2018-04-25          300000000
4              [Drama, Romance]   1997-12-18          200000000

   Domestic Gross  Worldwide Gross      Profit  Profit_Margin  \
0       858373000       2797800564  2397800564           0.86
1       760507625       2788701337  2551701337           0.92
2       700059566       1346103376  1146103376           0.85
3       678815482       2048359754  1748359754           0.85
4       659363944       2208208395  2008208395           0.91

   Adjusted_Budget  Adjusted_Profit     Month
0     412880000.00    2475009742.16     April
1     320945400.00    3455513950.57  December
2     212880000.00    1219912433.41  February
3     319320000.00    1860954122.16     April
4     348120000.00    3495487532.34  December
```

[89]:
```python
#Count the total number of movies and group by month.
rating_count = rating_df.groupby(['Rating'], as_index=False)['Movie'].count().
 ↪sort_values(by='Movie', ascending=False)
rating_count
```

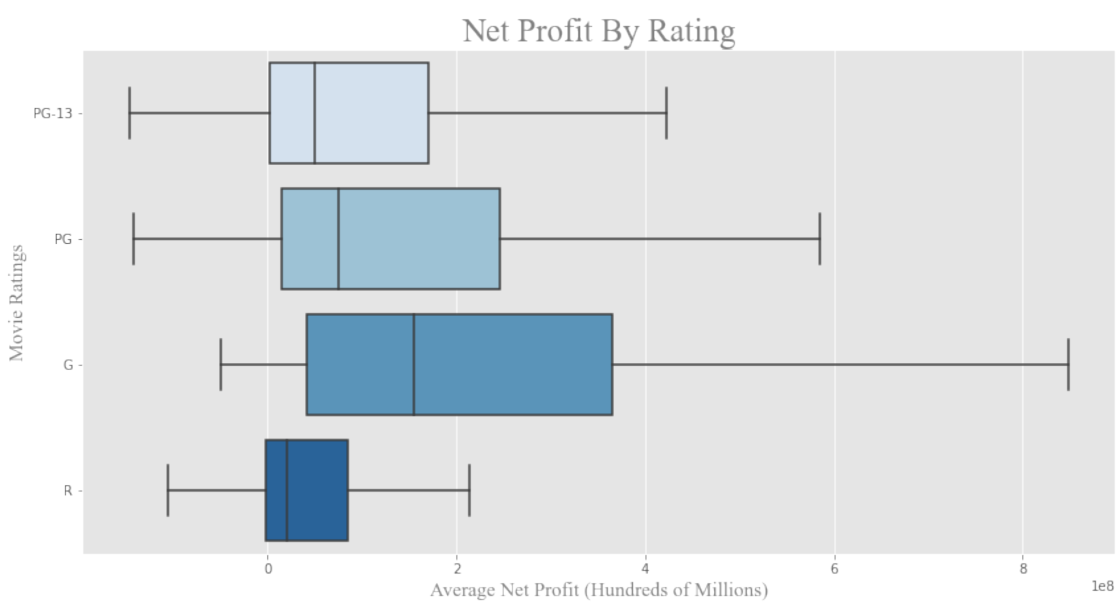
[89]:
```
  Rating  Movie
3      R   1631
2  PG-13   1339
1     PG    590
0      G     93
```

[90]:
```
#Group by Rating let's determine which has the highest median net profit and␣
↪profit margin.
rating_df2 = rating_df.groupby(['Rating'], as_index=False)[['Adjusted_Profit',␣
↪'Profit_Margin', 'IMDb']].median().sort_values(by='Adjusted_Profit',␣
↪ascending=False)
rating_df2
```

[90]:

	Rating	Adjusted_Profit	Profit_Margin	IMDb
0	G	154376810.04	0.76	7.10
1	PG	75404192.25	0.62	6.50
2	PG-13	49565772.61	0.55	6.30
3	R	20402474.98	0.51	6.60

[91]:
```
# Plot your above findings
plt.figure(figsize=(14,7))
ax13 = sns.boxplot( y=rating_df["Rating"], x=rating_df["Adjusted_Profit"],␣
↪showfliers=False, palette='Blues')
plt.xlabel('Average Net Profit (Hundreds of Millions)', fontdict = {'fontname':␣
↪'Times New Roman', 'color': 'gray', 'fontsize' : '15'})
plt.ylabel('Movie Ratings', fontdict = {'fontname': 'Times New Roman', 'color':␣
↪'gray', 'fontsize' : '15'})
plt.title('Net Profit By Rating', fontdict = {'fontname': 'Times New Roman',␣
↪'color': 'gray', 'fontsize' : '25'})
plt.savefig('ProfitbyRating', dpi=300);
```



As you can see, G and PG rated movies tend to perform best and account for the smallest market share. This, like the animation genre, is another opportunity to enter the market in a highly profitable arena with fewer competitors. It would be interesting to see a breakdown of total net profit by genre by rating to get a better idea of which rating and genres go best together.

[92]:
```python
# First drop the rating column from genre_budgets_df and genre from rating_df
genre_rating_df = genre_budgets_df.drop(['Rating'], axis=1)
rating_df = rating_df.drop(['Genre'], axis=1)
```

[93]:
```python
# Merge the genre_rating_df table and rating_df table
genre_rating_df = pd.merge(genre_rating_df, rating_df)
```

[94]:
```python
#Slice the top six most profitable genres.
Adv_df = genre_rating_df.loc[genre_rating_df['Genre'].str.contains('Adventure')]
Act_df = genre_rating_df.loc[genre_rating_df['Genre'].str.contains('Action')]
Com_df = genre_rating_df.loc[genre_rating_df['Genre'].str.contains('Comedy')]
Dra_df = genre_rating_df.loc[genre_rating_df['Genre'].str.contains('Drama')]
Sci_df = genre_rating_df.loc[genre_rating_df['Genre'].str.contains('Sci-Fi')]
Ani_df = genre_rating_df.loc[genre_rating_df['Genre'].str.contains('Animation')]

genre_concat = [Adv_df, Act_df, Com_df, Dra_df, Sci_df, Ani_df]
genre_rating = pd.concat(genre_concat)
```

[95]:
```python
# Create a pivot table from genre_rating
gr_df = genre_rating.groupby(['Genre', 'Rating'], ␣
↪as_index=False)['Adjusted_Profit'].sum().sort_values(by='Adjusted_Profit',␣
↪ascending=False)
gr_pivoted = gr_df.pivot(index='Genre', columns='Rating',␣
↪values='Adjusted_Profit')
```

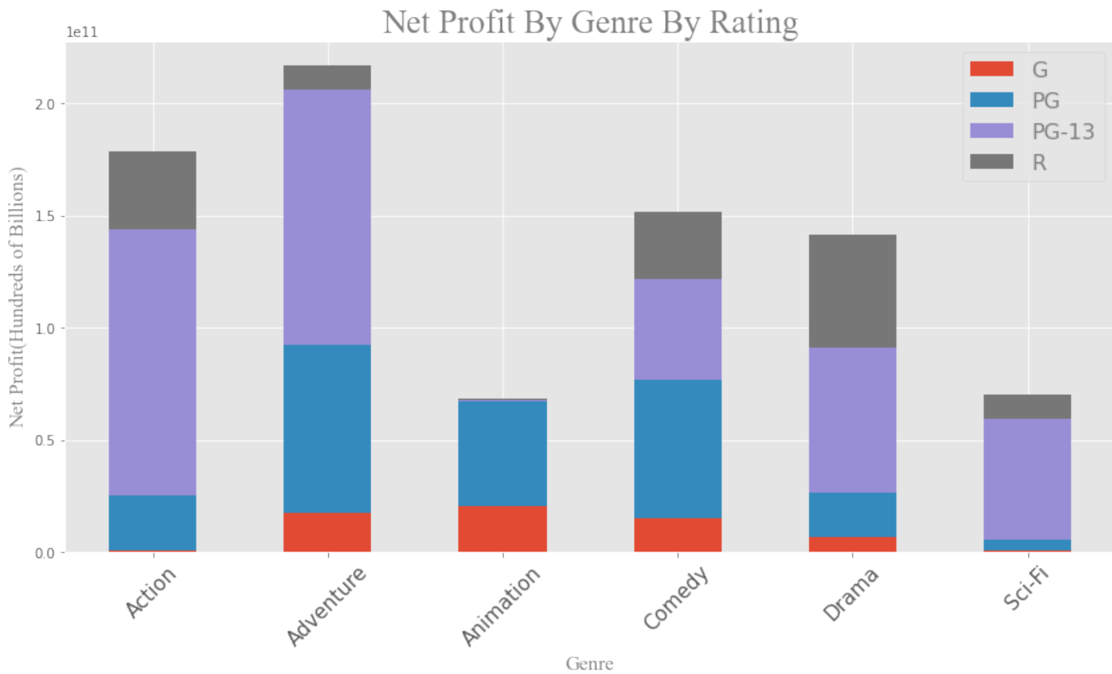
[96]:
```python
# Preview the table.
gr_pivoted
```

[96]:

Rating	G	PG	PG-13	R
Genre				
Action	476713962.52	24806502581.61	118476527154.35	34527820240.94
Adventure	17497561206.41	74656830471.14	114180501731.83	10663312187.82
Animation	20451774875.23	46792514260.78	682637577.33	120368587.97
Comedy	14989898831.46	61733858474.80	44722618139.99	30095649966.62
Drama	6452247472.37	19785801203.02	64695667306.22	50557666303.54
Sci-Fi	575199818.94	4693467863.02	54045363674.82	11072810424.26

[99]:
```python
# Plot the above findings.
ax14 = gr_pivoted.plot(kind='bar', stacked=True, figsize=(14,7))
plt.legend(labelcolor='grey', prop={'size': 16})
plt.xlabel('Genre', fontdict = {'fontname': 'Times New Roman', 'color': 'gray',␣
↪'fontsize' : '15'})
plt.ylabel('Net Profit(Hundreds of Billions)', fontdict = {'fontname': 'Times␣
↪New Roman', 'color': 'gray', 'fontsize' : '15'})
```

```
plt.title('Net Profit By Genre By Rating', fontdict = {'fontname': 'Times New Roman', 'color': 'gray', 'fontsize' : '25'})
plt.xticks(fontsize=15, rotation=45)
plt.savefig('ProfitbyGenrebyRating');
```



As one could have probably guessed, animation is almost entirely made up of G and PG rated movies. We can see that for most other genres, the bulk of their total net profits come from PG-13 rated movies. From this we can focus on which rating to aim for in each genre to evoke the most success.

Now let's shift our focus to the film's runtime. Does movie length have an impact in terms of success?

[100]:
```
# Create a new table with runtime, net profit and profit margin.
runtime_df = imdb_budgets_df[['Runtime', 'Adjusted_Profit', 'Profit_Margin']]
runtime_df
```

[100]:

Movie	Year	Runtime	Adjusted_Profit	Profit_Margin
Avengers: Endgame	2019	181	2475009742.16	0.86
Avatar	2009	162	3455513950.57	0.92
Black Panther	2018	134	1219912433.41	0.85
Avengers: Infinity War	2018	149	1860954122.16	0.85
Titanic	1997	194	3495487532.34	0.91
…		…	…	…
The Misfits	1961	125	12179160.00	0.51

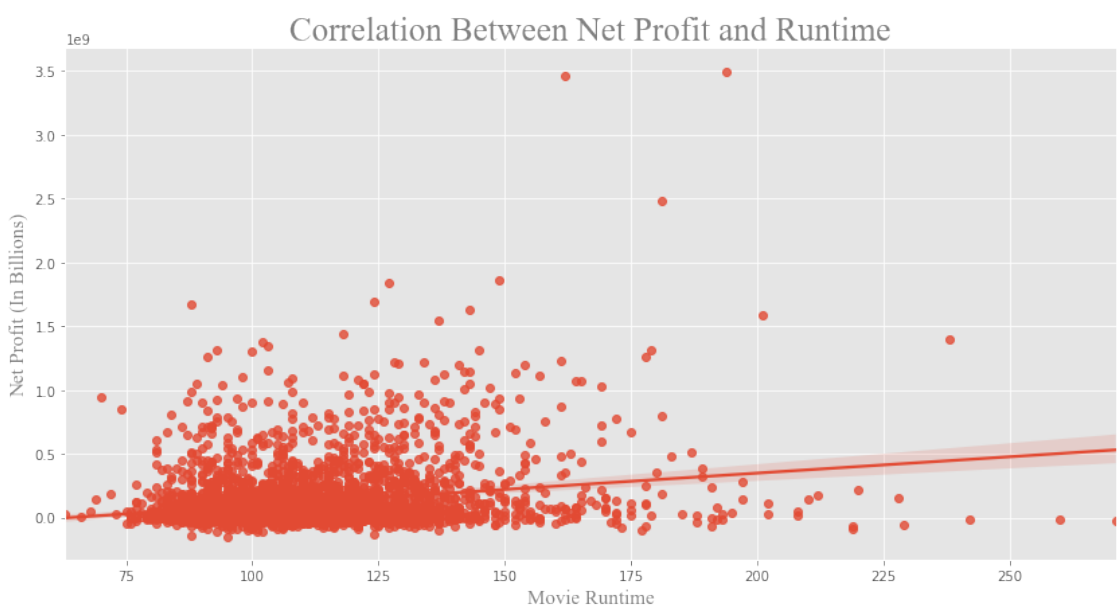
```
Judgment at Nuremberg  1961       179       20298600.00            0.70
The Wrong Man          1956       105        2448640.00            0.40
The Trouble with Harry 1955        99       17939400.00            0.83
Niagara                1953        92        3946750.00            0.50

[3740 rows x 3 columns]
```

[101]:
```python
# Let's start by taking a look at the correlation between runtime and net
 ↪profit/profit margin.
pearsoncorr = runtime_df.corr(method='pearson')
pearsoncorr
```

[101]:

	Runtime	Adjusted_Profit	Profit_Margin
Runtime	1.00	0.22	0.05
Adjusted_Profit	0.22	1.00	0.05
Profit_Margin	0.05	0.05	1.00

[102]:
```python
# Plot the correlation.
plt.figure(figsize=(14,7))
ax15 = sns.regplot(x='Runtime', y='Adjusted_Profit', data=imdb_budgets_df)
plt.xlabel('Movie Runtime', fontdict = {'fontname': 'Times New Roman', 'color':
 ↪'gray', 'fontsize' : '15'})
plt.ylabel('Net Profit (In Billions)', fontdict = {'fontname': 'Times New
 ↪Roman', 'color': 'gray', 'fontsize' : '15'})
plt.title('Correlation Between Net Profit and Runtime', fontdict = {'fontname':
 ↪'Times New Roman', 'color': 'gray', 'fontsize' : '25'})
plt.savefig('CorrProfitRuntime', dpi=300);
```



Although there is a small positive correlation of .223 showing that the long the runtime the higher the net profit, it's incredibly minute. With that in mind, we can take from this that, typically, it is not important to keep a movie above or below a cetain time threshold.

Question 6 Conclusion: We recommend that our Company take into consideration the rating of the movie based on the genre and target audience. If making animation movies, it is wise to stick to a G or PG rating, otherwise PG-13 is the sweetspot. In terms of runtime, there is little correlation in terms of overall profitability.

8 Question 7: Sticking to our analysis of Net Profit and Profit Margin, what should our Company determine to be the baseline for sustainable success?

We have an understanding of what goes into a successful movie but let's determine what our Comapny should expect in terms of profitability if they expect to compete with the other top movie studios.

[103]:
```
# Merge studio_df and imdb_budgets_df
studiobudgets_df = pd.merge(studio_df, imdb_budgets_df, left_on = 'title',␣
 ↪right_on='Movie')
studiobudgets_df.head()
```

[103]:
```
                        title                 studio  domestic_gross  \
0                 Toy Story 3            Buena Vista    415000000.00
1                   Inception                     WB    292600000.00
2          Shrek Forever After      Pixar/Dreamworks    238700000.00
3  The Twilight Saga: Eclipse  Sumbadhat Productions    300500000.00
4                  Iron Man 2              Paramount    312400000.00

  foreign_gross  year  IMDb Rating  Runtime                          Genre  \
0     652000000  2010  8.30      G      103  [Animation, Adventure, Comedy]
1     535700000  2010  8.80  PG-13      148     [Action, Adventure, Sci-Fi]
2     513900000  2010  6.30     PG       93  [Animation, Adventure, Comedy]
3     398000000  2010  5.00  PG-13      124     [Adventure, Drama, Fantasy]
4     311500000  2010  7.00  PG-13      124     [Action, Adventure, Sci-Fi]

  Release Date  Production Budget  Domestic Gross  Worldwide Gross     Profit  \
0   2010-06-18          200000000       415004880       1068879522  868879522
1   2010-07-16          160000000       292576195        832551961  672551961
2   2010-05-21          165000000       238736787        756244673  591244673
3   2010-06-30           68000000       300531751        706102828  638102828
4   2010-05-07          170000000       312433331        621156389  451156389

   Profit_Margin  Adjusted_Budget  Adjusted_Profit Month
0           0.81     264400000.00    1148658728.08  June
1           0.81     211520000.00     889113692.44  July
2           0.78     218130000.00     781625457.71   May
```

```
3           0.90      89896000.00     843571938.62   June
4           0.73     224740000.00     596428746.26    May
```

[237]:
```
# Let's remove some unnecessary fields.
studiobudgets_df.drop(columns = {'title', 'domestic_gross', 'Domestic Gross',␣
 ↪'foreign_gross', 'year', 'Production Budget', 'Worldwide Gross', 'Profit'},␣
 ↪inplace = True)
studiobudgets_df.rename(columns = {'studio':'Studio','Worldwide Gross ':
 ↪'Worldwide Gross' }, inplace = True)
studiobudgets_df.head()
```

[237]:
```
                  Studio  IMDb Rating  Runtime  \
0            Buena Vista   8.3      G      103
1                     WB   8.8  PG-13      148
2      Pixar/Dreamworks   6.3     PG       93
3  Sumbadhat Productions   5.0  PG-13      124
4              Paramount   7.0  PG-13      124

                             Genre Release Date  Profit_Margin  \
0  [Animation, Adventure, Comedy]   2010-06-18       0.812888
1     [Action, Adventure, Sci-Fi]   2010-07-16       0.807820
2  [Animation, Adventure, Comedy]   2010-05-21       0.781817
3     [Adventure, Drama, Fantasy]   2010-06-30       0.903697
4     [Action, Adventure, Sci-Fi]   2010-05-07       0.726317

   Adjusted_Budget  Adjusted_Profit Month
0      264400000.0     1.148659e+09  June
1      211520000.0     8.891137e+08  July
2      218130000.0     7.816255e+08   May
3       89896000.0     8.435719e+08  June
4      224740000.0     5.964287e+08   May
```

[254]:
```
# Group by studio, find median and filter to top 25 by Adjusted Profit
profit_by_studiodf = studiobudgets_df.groupby('Studio').median()
profit_by_studiodf = profit_by_studiodf.reset_index()
profit_by_studiodf = profit_by_studiodf.nlargest(25, 'Adjusted_Profit')
profit_by_studiodf
```

[254]:
```
                         Studio  IMDb  Runtime  Profit_Margin  \
51                          UTV  8.45    141.5       0.958798
37             Pixar/Dreamworks  6.70     94.0       0.716170
9                   Buena Vista  7.10    117.0       0.667056
28                         MBox  7.80    158.0       0.624019
48                       Strand  6.50    112.0       0.741792
45                         Sony  6.30    105.0       0.658692
35                    Paramount  6.40    110.0       0.639187
20                          Fox  6.35    106.0       0.644465
```

```
52                    Universal  6.20   108.0        0.686945
54                           WB  6.60   113.5        0.542261
15           Eros International  7.10   160.0        0.836702
55               Wein/Dimension  5.90    96.0        0.750298
44                  Screen Gems  5.80   103.0        0.698444
27  Lionsgate/Summit Entertainment  6.55   110.0        0.606561
32                         Neon  7.50   119.0        0.795529
46                Sony Pictures  6.70   112.0        0.664717
25                    Lionsgate  6.15   103.5        0.601290
49         Sumbadhat Productions  6.60   100.0        0.446140
19               Focus Features  6.90   108.0        0.484553
56            Weinstein Company  7.20   106.5        0.694665
43                          STX  6.40   104.0        0.528697
40             Relativity Media  6.25   105.5        0.506080
10                          CBS  6.60   102.0        0.591352
50                  The Orchard  7.90   101.0        0.891707
47        Sony Pictures Classics  7.20   109.0        0.600112

    Adjusted_Budget  Adjusted_Profit
51       33747300.0     6.921112e+08
37      182352000.0     4.921191e+08
9       176565000.0     1.928538e+08
28      116082000.0     1.926625e+08
48       50796000.0     1.459292e+08
45       65796000.0     1.296401e+08
35       53053600.0     1.270562e+08
20       65785200.0     1.171804e+08
52       47728000.0     1.081619e+08
54       66914000.0     8.010906e+07
15       10170820.0     5.211316e+07
55       27474000.0     5.093755e+07
44       30121600.0     5.004866e+07
27       44584000.0     4.695959e+07
32       12062600.0     4.693164e+07
46       26847000.0     4.222879e+07
25       32376500.0     3.662573e+07
49       41273600.0     3.615531e+07
19       20121600.0     3.370892e+07
56       18381000.0     3.362412e+07
43       32898000.0     3.331053e+07
40       33053600.0     2.923035e+07
10       21926600.0     2.688925e+07
50        2822000.0     2.323702e+07
47        8041120.0     1.587142e+07
```

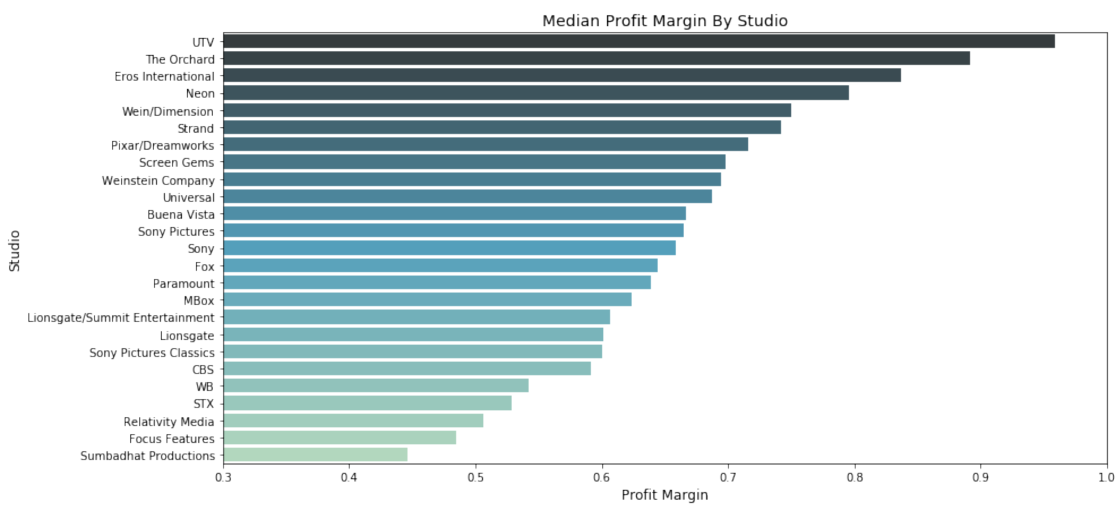
[253]:
```
# Let's take a look at the average of these median values.
profit_by_studiodf.describe()
```

[253]:

	IMDb	Runtime	Profit_Margin	Adjusted_Budget	Adjusted_Profit
count	25.00000	25.000000	25.000000	2.500000e+01	2.500000e+01
mean	6.76600	112.180000	0.663049	4.883893e+07	1.134278e+08
std	0.64108	16.751169	0.122761	4.612474e+07	1.557719e+08
min	5.80000	94.000000	0.446140	2.822000e+06	1.587142e+07
25%	6.35000	103.500000	0.600112	2.192660e+07	3.370892e+07
50%	6.60000	108.000000	0.658692	3.305360e+07	5.004866e+07
75%	7.10000	112.000000	0.716170	5.305360e+07	1.270562e+08
max	8.45000	160.000000	0.958798	1.823520e+08	6.921112e+08

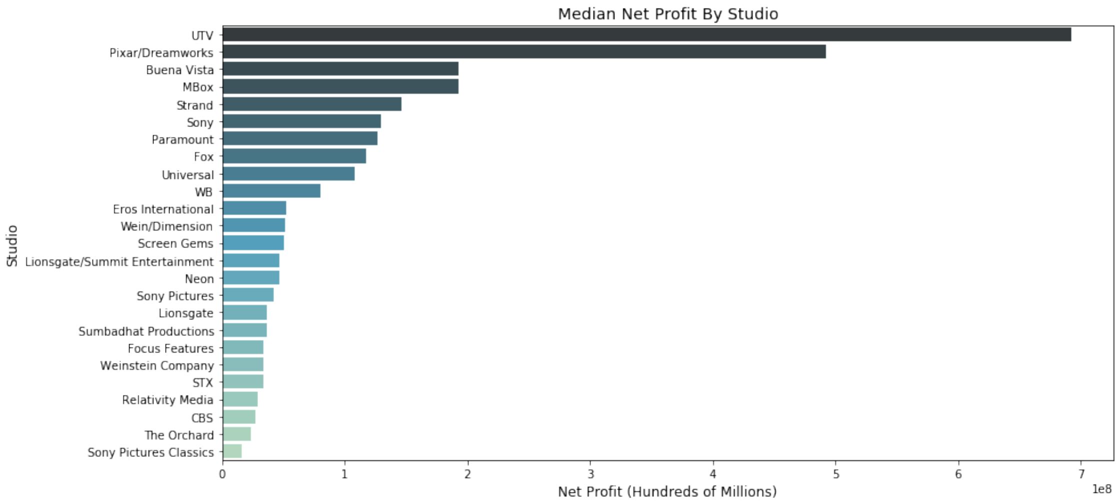
We can see that if we want to strive to be in the top half of this elite list of movie studios we need to have a profit margin of 66% and a net profit of 50 million per movie.

[256]:
```
#Plot the above findings.
plt.figure(figsize=(14,7))
ax16 = sns.barplot(x=profit_by_studiodf['Profit_Margin'],␣
 ↪y=profit_by_studiodf['Studio'],
                   order=profit_by_studiodf.sort_values('Profit_Margin',␣
 ↪ascending=False).Studio, palette='GnBu_d')
plt.xlabel('Profit Margin', fontsize=12)
plt.ylabel('Studio', fontsize=12)
plt.title('Median Profit Margin By Studio', fontsize=14)
plt.xlim(0.3, 1.0)
plt.savefig('ProfitMarginStudio')
```



[255]:
```
#Plot the above findings.
plt.figure(figsize=(14,7))
ax16 = sns.barplot(y=profit_by_studiodf['Studio'],␣
 ↪x=profit_by_studiodf['Adjusted_Profit'], palette='GnBu_d')
plt.xlabel('Net Profit (Hundreds of Millions)', fontsize=12)
```

```
plt.ylabel('Studio', fontsize=12)
plt.title('Median Net Profit By Studio', fontsize=14)
plt.savefig('NetProfitStudio');
```



We can see from the graph above that the major players in the studio industry have profit margins ranging from 24% to 95%. That's quite a large range to define success. However, the top 25 studios shown are many of the studios that we often recognize when we go to the movies. As we've done previously, we use the median profit margin of the top 25 as a target for success among major studios. That profit margin is 66%. In the next analysis we'll take a closer look at some of these major studios to see what metrics we should try to mimic. Let's also keep this in mind as we go into our next analysis: UTV which has the greatest profit margin of all the studios is a subsidiary of Disney.

Question 7 Conclusion: Microsoft should aim for a profit margin of 66% and a net profit of slightly over 50 million per movie to compete with the top existing studios.

9 Question 8: Based on the success of current competitors, which should we look to for best practices?

We need to add a column to the `theaters_df` dataframe to calculate the money grossed per theater for a given movie. Then we can group by studio.

[106]:
```
theaters_df['dollars_per_theater'] = theaters_df['total_dom_gross($)'] /␣
↪theaters_df['max_theaters']
theaters_df.head()
```

[106]:
```
                  title  max_theaters  year  total_dom_gross($)  \
0         The Lion King          4802  2019           543638043
1     Avengers: Endgame          4662  2019           858373000
2  Spider-Man: Far from Home     4634  2019           390532085
```

```
3             Toy Story 4          4575  2019           434038008
4          It Chapter Two          4570  2019           211593228

         studio  dollars_per_theater
0        Disney            113210.75
1        Disney            184121.19
2          Sony             84275.37
3        Disney             94871.70
4  Warner Bros.             46300.49
```

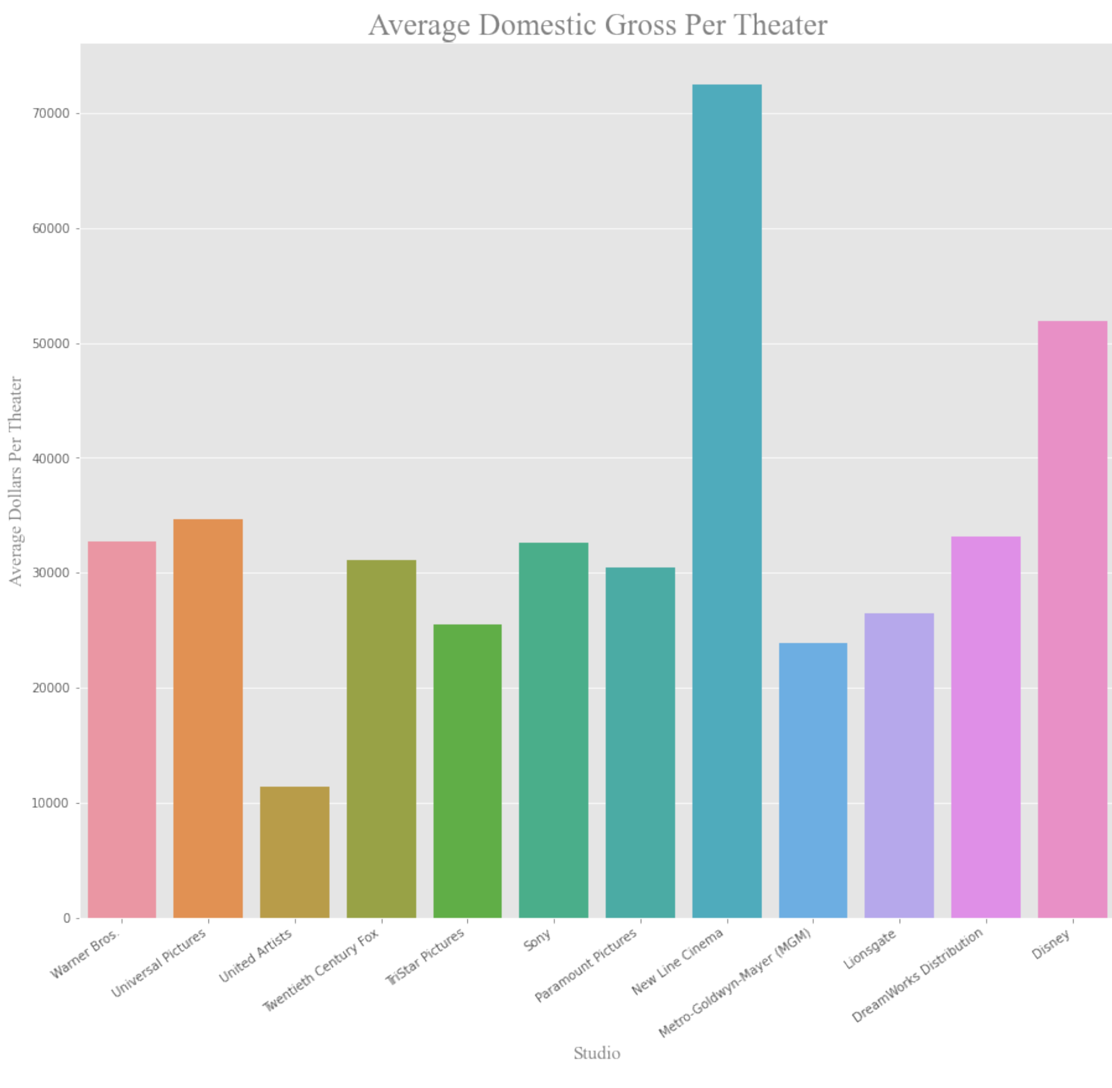
[107]:

```
#Let's see what the average is for max number of theaters and for gross per␣
 ↪theater for each studio
average_theaters = theaters_df.groupby('studio').mean()
average_theaters_ranked = average_theaters.
 ↪sort_values(by=['studio'],ascending=False)
average_theaters_ranked.reset_index(inplace=True)
average_theaters
```

[107]:

```
                           max_theaters    year  total_dom_gross($)  \
studio
Disney                          3682.32 2010.59        202617891.97
DreamWorks Distribution         3408.26 2002.95        118198315.42
Lionsgate                       3356.24 2014.47         95268293.14
Metro-Goldwyn-Mayer (MGM)       3259.14 2004.00         78437576.64
New Line Cinema                 3410.57 2001.86        249718149.29
Paramount Pictures              3466.71 2010.71        108614912.30
Sony                            3478.36 2010.56        116677932.63
TriStar Pictures                3146.00 2014.00         80703217.29
Twentieth Century Fox           3493.98 2011.21        111009777.12
United Artists                  3124.00 2003.00         35667218.00
Universal Pictures              3488.41 2011.96        124914179.39
Warner Bros.                    3535.03 2011.59        120355240.25

                           dollars_per_theater
studio
Disney                                51856.14
DreamWorks Distribution               33102.06
Lionsgate                             26485.34
Metro-Goldwyn-Mayer (MGM)             23829.21
New Line Cinema                       72518.24
Paramount Pictures                    30508.47
Sony                                  32626.67
TriStar Pictures                      25546.75
Twentieth Century Fox                 31119.14
United Artists                        11417.16
Universal Pictures                    34679.48
Warner Bros.                          32678.01
```

```
[115]: plt.figure(figsize=(15,13))
ax16 = sns.barplot(x='studio', y='dollars_per_theater',␣
 ↪data=average_theaters_ranked)
plt.xlabel('Studio', fontdict = {'fontname': 'Times New Roman', 'color':␣
 ↪'gray', 'fontsize' : '15'})
plt.title("Average Domestic Gross Per Theater", fontdict = {'fontname': 'Times␣
 ↪New Roman', 'color': 'gray', 'fontsize' : '25'});
plt.ylabel('Average Dollars Per Theater', fontdict = {'fontname': 'Times New␣
 ↪Roman', 'color': 'gray', 'fontsize' : '15'});
plt.xticks(rotation=35, horizontalalignment='right')
plt.savefig('DomesticPerTheater', dpi=300);
```



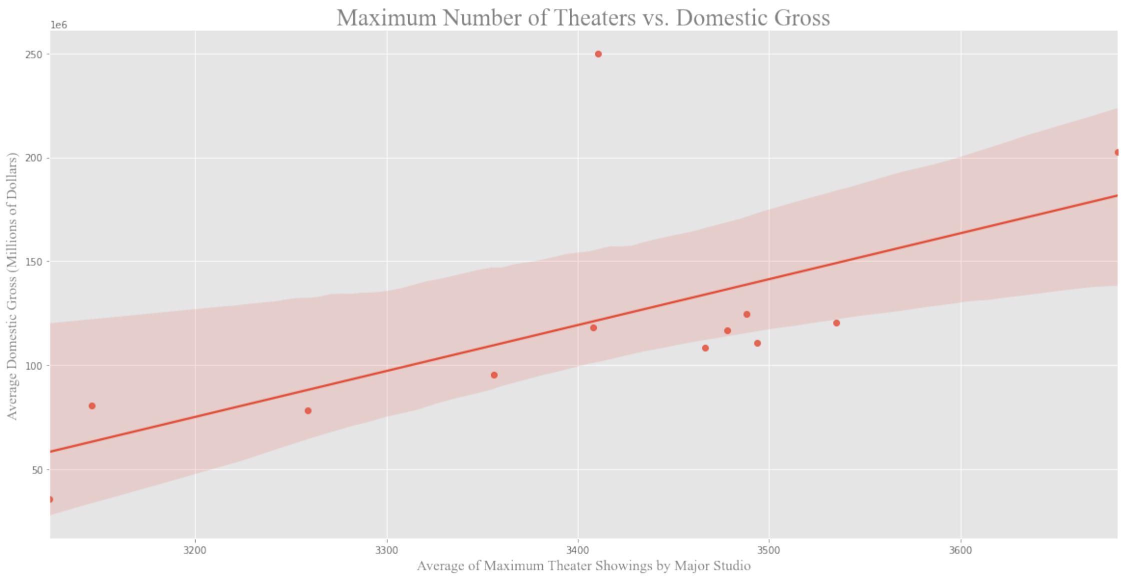
In the bar plot above, Disney and New Line Cinema stand out. We need to double check that there are an appropriate number of movies by each of these studios before jumping to conclusions.

[116]:
```
theaters_df['studio'].value_counts()
```

[116]:
```
Warner Bros.                  208
Twentieth Century Fox         165
Disney                        147
Universal Pictures            136
Sony                          135
Paramount Pictures            112
Lionsgate                      49
DreamWorks Distribution        19
Metro-Goldwyn-Mayer (MGM)      14
New Line Cinema                 7
TriStar Pictures                7
United Artists                  1
Name: studio, dtype: int64
```

We can see that New Line Cinema only has 7 movies in this dataframe which means that their average domestic gross per theater is going to be skewed. Disney is certainly still a possibility and we should also consider Warner Bros. and Twentieth Century Fox.

[118]:
```
ax17 = sns.lmplot(x='max_theaters', y='total_dom_gross($)',␣
 ↪data=average_theaters, height=8, aspect=2)
plt.ticklabel_format(axis='y', style='sci', scilimits=(6,6))
plt.xlabel('Average of Maximum Theater Showings by Major Studio', fontdict =␣
 ↪{'fontname': 'Times New Roman', 'color': 'gray', 'fontsize' : '15'})
plt.ylabel('Average Domestic Gross (Millions of Dollars)', fontdict =␣
 ↪{'fontname': 'Times New Roman', 'color': 'gray', 'fontsize' : '15'})
plt.title('Maximum Number of Theaters vs. Domestic Gross', fontdict =␣
 ↪{'fontname': 'Times New Roman', 'color': 'gray', 'fontsize' : '25'})
plt.savefig('TheatersVGross', dpi=300);
```



The scatter plot shows a positive trend between the average number of theaters and the average domestic gross. The sole outlier is New Line Cinemas due to how few movies they are associated with in our dataframe. Disney is farthest to the right and above the trend line further proving that they should be a strong consideration.

We'll join the theater and awards dataframes so that we can see which studios have the best win rate at the Oscars.

[119]:
```
theaters_df.set_index(['title', 'year'], inplace=True)
```

[120]:
```
theaters_and_awards = theaters_df.join(awards_df, how='inner', on=['title',␣
    ↪'year'])
```

[121]:
```
theaters_and_awards.groupby('studio').count()
```

[121]:
```
                          max_theaters  total_dom_gross($)  \
studio
Disney                              22                  22
DreamWorks Distribution              4                   4
New Line Cinema                      2                   2
Paramount Pictures                   7                   7
Sony                                 6                   6
Twentieth Century Fox                4                   4
Universal Pictures                   6                   6
Warner Bros.                        15                  15

                          dollars_per_theater  awards_won  awards_nominated  \
studio
Disney                                     22          22                22
DreamWorks Distribution                     4           4                 4
New Line Cinema                             2           2                 2
Paramount Pictures                          7           7                 7
Sony                                        6           6                 6
Twentieth Century Fox                       4           4                 4
Universal Pictures                          6           6                 6
Warner Bros.                               15          15                15

                          win_rate
studio
Disney                          22
DreamWorks Distribution          4
New Line Cinema                  2
Paramount Pictures               7
Sony                             6
Twentieth Century Fox            4
Universal Pictures               6
```

```
Warner Bros.                  15
```

[122]:
```python
theaters_and_awards.groupby('studio').mean()
```

[122]:
```
                          max_theaters  total_dom_gross($)  \
studio
Disney                         3818.73        305217242.45
DreamWorks Distribution        3444.25        153223630.75
New Line Cinema                3662.50        358408603.00
Paramount Pictures             3564.86        140835427.57
Sony                           3653.67        237842295.67
Twentieth Century Fox          3501.75        136874930.25
Universal Pictures             3338.83        149344665.00
Warner Bros.                   3831.60        234055876.80

                          dollars_per_theater  awards_won  awards_nominated  \
studio
Disney                               78797.61        1.36              3.00
DreamWorks Distribution              44447.63        2.00              4.25
New Line Cinema                      97814.75        6.50              8.50
Paramount Pictures                   38930.82        1.00              3.71
Sony                                 64720.23        1.17              3.17
Twentieth Century Fox                38404.79        2.25              6.00
Universal Pictures                   44970.82        1.33              3.33
Warner Bros.                         60023.04        2.67              5.87

                          win_rate
studio
Disney                        0.60
DreamWorks Distribution       0.60
New Line Cinema               0.67
Paramount Pictures            0.45
Sony                          0.54
Twentieth Century Fox         0.43
Universal Pictures            0.51
Warner Bros.                  0.56
```

Unfortunately, the joining of the dataframes only left us with 66 common movies. We would prefer to have more data to be more confident in establishing trends. We will consider the average number of theaters and average win rate to make a determination. Disney is associated with 22 movies in our joined dataframe while Warner Bros. is associated with 15. Warner. Bros does have a higher average for the number of theaters, however Disney has a noticeable $18,000 advantage in average domestic gross per theater. Disney also has the higher win rate for Oscars at nearly 60%.

Question 8 Conclusion: Our Company should research Disney's best practices and try to build off the success of this well established studio.

10 Conclusion

While there are many other factors that we could consider in a future analysis we feel that the following 8 conclusions will result in a successful business venture as our Comapany enters the movie industry.

1. I recommend that we should budget approximately $82,250,000 to make a movie. This should correlate with a profit margin above 80%.
2. I recommend that we should focus their efforts on the top 6 most profitable movie genres: Adventure, Action, Comedy, Drama, Sci-Fi and Animation. A further recommendation to focus on Sci-Fi and Animation due to less competition and a higher opportunity to profit.
3. I recommend that we release the bulk of their movies, especially Animation, during the summer months. Adventure, Drama and Comedy movies would see similar success if released in November, but the recommendation remains to focus on summer.
4. I recommend that we focus their cast and crew search to individuals who consistently score at least 1.0 on the VAR score. We can, with a high level of confidence, conclude that these individuals will elevate the overall production.
5. We should spend at least $35,465,000 in order to make an Oscar-winning movie.
6. I recommend that we take into consideration the rating of the movie based on the genre and target audience. If making animation movies, it is wise to stick to a G or PG rating, otherwise PG-13 is the sweetspot. In terms of runtime, there is little correlation in terms of overall profitability.
7. We should aim for a profit margin of 66% and a net profit of slightly over 50 million per movie to compete with the top existing studios.
8. We should research Disney's best practices and try to build off the success of this well established studio.