Project: Investigate a Dataset (TMDb_Movies Dataset)

Table of Contents

Introduction

Overview:

This is the TMDb movie data set for data analysis. This data set contains information about 10,000 movies collected from The Movie Database (TMDb), including user ratings and revenue.

The fetures of the data:

- movie_id - A unique identifier for each movie.
- imdb_id - A unique identifier for each movie on IMDB.
- cast - The name of lead and supporting actors.
- director - the director of the movie.
- budget - The budget in which the movie was made.
- genre - The genre of the movie, Action, Comedy ,Thriller etc.
- homepage - A link to the homepage of the movie.
- id - This is infact the movie_id as in the first dataset.
- keywords - The keywords or tags related to the movie.
- original_title - The title of the movie before translation or adaptation.
- overview - A brief description of the movie.
- popularity - A numeric quantity specifying the movie popularity.
- production_companies - The production house of the movie.
- production_countries - The country in which it was produced.
- release_date - The date on which it was released.
- revenue - The worldwide revenue generated by the movie.
- runtime - The running time of the movie in minutes.
- tagline - Movie's tagline.
- vote_average - average ratings the movie recieved.
- budget_adj - shows the budget associated movie in terms of 2010 dollars.
- revenue_adj - shows the revenue associated movie in terms of 2010 dollars.

The used libraries:

- **Numpy** - a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.
- **Pandas** - a library for the Python programming language, it offers data structures and operations for manipulating numerical tables and time series.
- **Matplotlib** - Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy
- **Seaborn** - Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

Question needed to be answered to analyzed the data:

- 1. what's the most and least frquant movie's genres?
- 2. who's the most frequant 10 actors appearancees in the movies?
- 3. what's the most frequant 10 production companies Produce movies?
- 4. who's the most frequant 10 directors in the movies?
- 5. what's the Top and least 10 movies based on the revenue?
- 6. what's the Top and least 10 movies based on the budget?
- 7. what's the Top and least 10 movies based on the profit?
- 8. what's the Top and least 10 movies based on the popularity?

- 9. What's the properties with the movies with high profit?
- * Dose the year of realse associated with high profit?
- * Dose the sesone of realse associated with high profit?
- * Dose the budget associated with high profit??
- * Dose the runtime of the movie associated with high profit? ??
- *Is there any spacific genre associated with high profit?
- * Is there any cast member associated with high profit?
- * Is there any spacific director associated with high profit?
- * Is there any spacific production company associated with high profit?

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

Data Wrangling

load the data and take a look to its columns and cheak the rows before cleaning

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

In [3]:
```
1 movies_df.head()
```

Out[3]:

	id	imdb_id	popularity	budget	revenue	original_title	cast	
0	135397	tt0369610	32.985763	150000000	1513528810	Jurassic World	Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi...	
1	76341	tt1392190	28.419936	150000000	378436354	Mad Max: Fury Road	Tom Hardy|Charlize Theron|Hugh Keays-Byrne|Nic...	
2	262500	tt2908446	13.112507	110000000	295238201	Insurgent	Shailene Woodley|Theo James|Kate Winslet|Ansel...	http://www.tl
3	140607	tt2488496	11.173104	200000000	2068178225	Star Wars: The Force Awakens	Harrison Ford|Mark Hamill|Carrie Fisher|Adam D...	http:/
4	168259	tt2820852	9.335014	190000000	1506249360	Furious 7	Vin Diesel|Paul Walker|Jason Statham|Michelle ...	

5 rows × 21 columns

In [4]:
```
1 movies_df.columns
```

Out[4]:
```
Index(['id', 'imdb_id', 'popularity', 'budget', 'revenue', 'original_title',
       'cast', 'homepage', 'director', 'tagline', 'keywords', 'overview',
       'runtime', 'genres', 'production_companies', 'release_date',
       'vote_count', 'vote_average', 'release_year', 'budget_adj',
       'revenue_adj'],
      dtype='object')
```

In [5]:
```
1 movies_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10866 entries, 0 to 10865
Data columns (total 21 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   id                    10866 non-null  int64  
 1   imdb_id               10856 non-null  object 
 2   popularity            10866 non-null  float64
 3   budget                10866 non-null  int64  
 4   revenue               10866 non-null  int64  
 5   original_title        10866 non-null  object 
 6   cast                  10790 non-null  object 
 7   homepage              2936 non-null   object 
 8   director              10822 non-null  object 
 9   tagline               8042 non-null   object 
 10  keywords              9373 non-null   object 
 11  overview              10862 non-null  object 
 12  runtime               10866 non-null  int64  
 13  genres                10843 non-null  object 
 14  production_companies  9836 non-null   object 
 15  release_date          10866 non-null  object 
 16  vote_count            10866 non-null  int64  
 17  vote_average          10866 non-null  float64
 18  release_year          10866 non-null  int64  
 19  budget_adj            10866 non-null  float64
 20  revenue_adj           10866 non-null  float64
dtypes: float64(4), int64(6), object(11)
memory usage: 1.7+ MB
```

noticed from the previous cell that the data conaints NaN values

In [6]:
```
#calculate NaN values in the data
movies_df.isnull().sum()
```

Out[6]:
```
id                      0
imdb_id                10
popularity              0
budget                  0
revenue                 0
original_title          0
cast                   76
homepage             7930
director               44
tagline              2824
keywords             1493
overview                4
runtime                 0
genres                 23
production_companies  1030
release_date            0
vote_count              0
vote_average            0
release_year            0
budget_adj              0
revenue_adj             0
dtype: int64
```

In [7]:
```
#check for dulicates raws
movies_df.duplicated().sum()
```

Out[7]:
```
1
```

In [8]:
```
movies_df.describe()
```

Out[8]:

	id	popularity	budget	revenue	runtime	vote_count	vote
count	10866.000000	10866.000000	1.086600e+04	1.086600e+04	10866.000000	10866.000000	1086
mean	66064.177434	0.646441	1.462570e+07	3.982332e+07	102.070863	217.389748	
std	92130.136561	1.000185	3.091321e+07	1.170035e+08	31.381405	575.619058	
min	5.000000	0.000065	0.000000e+00	0.000000e+00	0.000000	10.000000	
25%	10596.250000	0.207583	0.000000e+00	0.000000e+00	90.000000	17.000000	
50%	20669.000000	0.383856	0.000000e+00	0.000000e+00	99.000000	38.000000	
75%	75610.000000	0.713817	1.500000e+07	2.400000e+07	111.000000	145.750000	
max	417859.000000	32.985763	4.250000e+08	2.781506e+09	900.000000	9767.000000	

Noticed from the previous cell that the 'runtime', 'budget', 'revenue', 'budget_adj', and 'revenue_adj' have zero values, which is'nt convenient

In [9]:
```
def row_zero(df, col_name):
    """
    check the rows with zero value.

    Arges:
        (pandas datafrma) df- the data that needed to check the zero values
        (str) col_name- the name of the column needed to check the zero valu
    Returns:
        (array) index- array contines the index of the zero values.
        (tuple) shape- contines the number of the the zero values.

    """
    zero = df[col_name] == 0
    index = df[zero].index.values
    shape = df[zero].shape
    return index, shape
```

In [10]:
```
#check the zero values in revenue col.
row_zero(movies_df, 'revenue')
```

Out[10]:
```
(array([   48,    67,    74, ..., 10863, 10864, 10865], dtype=int64),
 (6016, 21))
```

In [11]:
```
#check the zero values in budget col.
row_zero(movies_df, 'budget')
```

Out[11]:
```
(array([   30,    36,    72, ..., 10862, 10863, 10864], dtype=int64),
 (5696, 21))
```

In [12]:
```
#check the zero values in runtime col.
row_zero(movies_df, 'runtime')
```

Out[12]:
```
(array([  92,  334,  410,  445,  486,  595,  616, 1241, 1289, 1293, 1849,
        2315, 2370, 3329, 3794, 3857, 3884, 4063, 4138, 4829, 4944, 5216,
        5695, 5920, 5938, 5992, 6040, 6383, 6552, 6934, 8874], dtype=int64),
 (31, 21))
```

In [13]:
```
#check the zero values in budget_adj col.
row_zero(movies_df, 'budget_adj')
```

Out[13]:
```
(array([   30,    36,    72, ..., 10862, 10863, 10864], dtype=int64),
 (5696, 21))
```

In [14]:
```
#check the zero values in revenue_adj col.
row_zero(movies_df, 'revenue_adj')
```

Out[14]:
```
(array([   48,    67,    74, ..., 10863, 10864, 10865], dtype=int64),
 (6016, 21))
```

Data Cleaning

what we observed from the data wrangling process:

1. unnecessary columns in the data, that aren't important in our data analysis, need to be removed. **('imdb_id', 'id', 'homepage', 'tagline', 'keywords', 'overview', 'revnue_adj', 'budget_adj', 'vote_count', 'vote_average;)**
2. there's NAN need to be handle.
3. there's duplicated row need to be removed.
4. there's zero values in **('runtime', 'budget', 'revenue', 'budget_adj', and 'revenue_adj')** columns need to be handle.
5. changing the data type of 'release_date' column to datetime type.
6. noticed that 'cast', 'genres', and 'production_companies' columns are contine multiple values separated by pipe (|) characters.

In [15]:
```
#Replace the zero values with NAN values and drop the rows from the data.
col_list=['budget', 'revenue', 'runtime']
movies_df[col_list] = movies_df[col_list].replace(0, np.NAN)
movies_df.dropna(subset = col_list, inplace = True)
movies_df.shape
```

Out[15]:
```
(3855, 21)
```

After cleaning the data from the zero values, we only have 3855 rows to be analyzed

In [16]:
```
#Drop the unnecessary columns from our data
movies_df.drop(['imdb_id', 'id', 'homepage', 'tagline', 'keywords', 'overvie
```

check for our columns after drop unnecessary columns.

In [17]:
```
movies_df.columns, movies_df.shape
```

Out[17]:
```
(Index(['popularity', 'budget', 'revenue', 'original_title', 'cast', 'directo
r',
       'runtime', 'genres', 'production_companies', 'release_date',
       'release_year'],
      dtype='object'),
 (3855, 11))
```

After dropping the unnecessary columns, there's only 11 columns are lefted to be analyzed.

In [18]:
```
#Drop the NAN values
movies_df.dropna(inplace = True)
movies_df.shape
```

Out[18]:
```
(3806, 11)
```

After dropping the unnecessary columns, there's only 3806 rows are lefted to be analyzed.

In [19]:
```
#chech for NAN.
movies_df.isnull().sum()
```

Out[19]:
```
popularity             0
budget                 0
revenue                0
original_title         0
cast                   0
director               0
runtime                0
genres                 0
production_companies    0
release_date           0
release_year           0
dtype: int64
```

In [20]:
```
#Drop the duplicated values.
movies_df.drop_duplicates(inplace = True)
movies_df.shape
```

Out[20]:
```
(3805, 11)
```

After dropping the unnecessary columns, there's only 3805 rows are lefted to be analyzed.

In [21]:
```
#Check for duplicated values in the data.
movies_df.duplicated().sum()
```

Out[21]:
```
0
```

In [22]:
```
#Change the data type of 'release_date' column to datetime type
movies_df['release_date'] = pd.to_datetime(movies_df['release_date'])
```

In [23]:
```
#Check the type of 'release_date' colums.
movies_df.release_date.dtypes
```

Out[23]:
```
dtype('<M8[ns]')
```

Creating a new column 'month' from 'release_date' column to help me in my analysis

In [24]:
```
movies_df['month'] = movies_df['release_date'].dt.month
movies_df['month']
```

Out[24]:
```
0         6
1         5
2         3
3        12
4         4
        ..
10822     6
10828     7
10829    12
10835    12
10848     8
Name: month, Length: 3805, dtype: int64
```

Drop 'release_date' column becouse i don't need it any more

In [25]:
```
movies_df.drop(['release_date'], axis=1, inplace=True)
```

for my analysis, i need to creat a 'seson' column to check if the sesone of the release is associat with high renenue

In [26]:
```
seasons = [0,2,5,8,11,12]
labels = ['winter', 'sprint', 'summer', 'fall', 'winter']
movies_df['season'] = pd.cut(movies_df['month'], seasons, labels=labels, ord
movies_df['season']
```

Out[26]:
```
0        summer
1        sprint
2        sprint
3        winter
4        sprint
          ...
10822    summer
10828    summer
10829    winter
10835    winter
10848    summer
Name: season, Length: 3805, dtype: category
Categories (4, object): ['fall', 'sprint', 'summer', 'winter']
```

Drop the month column, which isn't needed anymore in the analysis

In [27]:
```
movies_df.drop(['month'], axis=1, inplace=True)
```

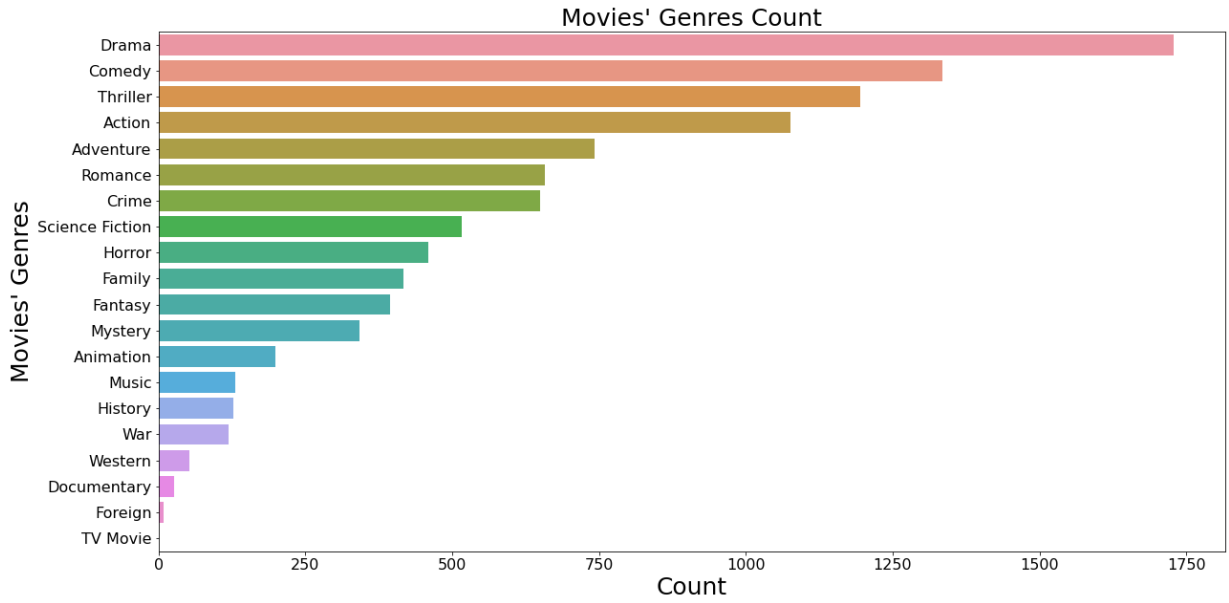
Creating **'profit'** column from **'revenue'** and **'budget'** columns, to know how mach profit did the movies made.

In [28]:
```
movies_df['profit'] = movies_df['revenue'] - movies_df['budget']
```

Exploratory Data Analysis

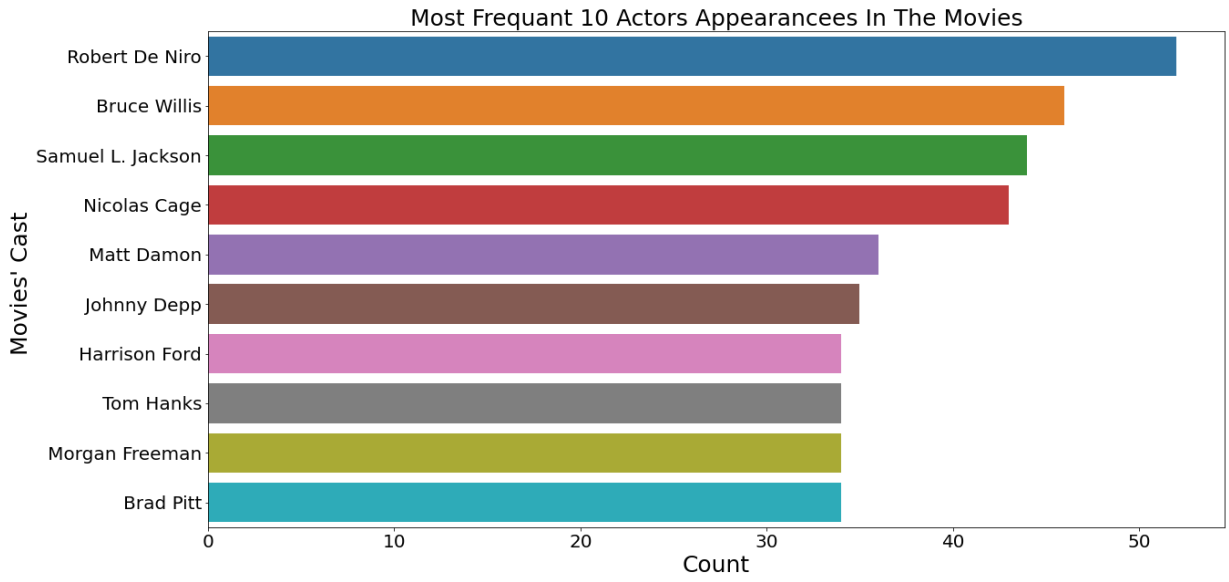
Q1: what's the most and least frquant movie's genres?

In [29]:
```
#split genres column and count each catogory.
genres_df = movies_df.assign(genres=movies_df['genres'].str.split('|')).expl

plt.figure(figsize=(20, 10))
sns.countplot(y='genres',data=genres_df, order = genres_df['genres'].value_c

plt.title('Movies\' Genres Count', fontsize=25)
plt.ylabel('Movies\' Genres', fontsize=25)
plt.xlabel('Count', fontsize=25)

plt.xticks(fontsize=16)
plt.yticks(fontsize=16)
plt.show()
```



Q2: who's the most frequant 10 actors appearancees in the movies?

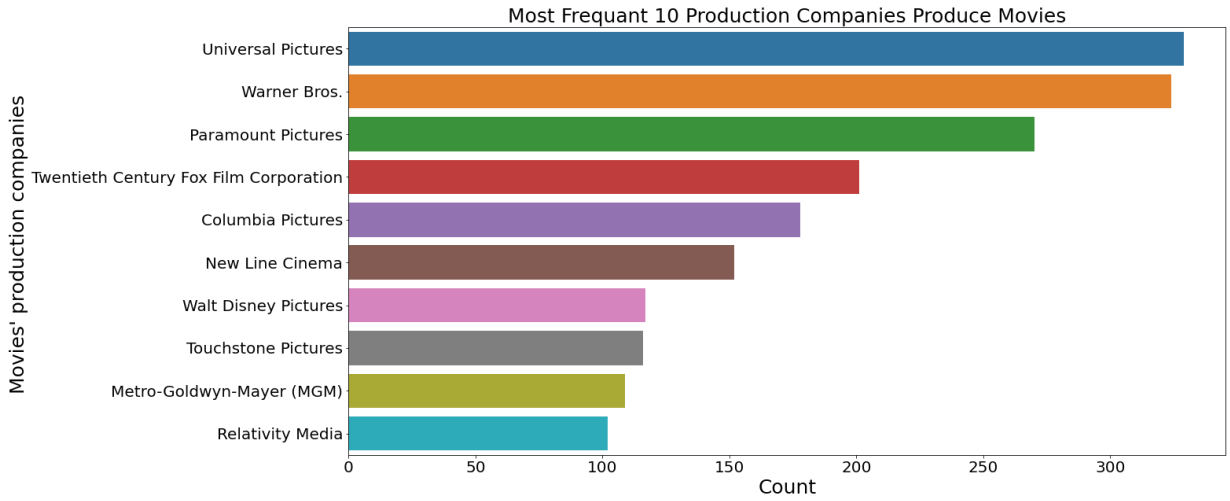
In [30]:
```
#split cast column, count thier appearancees, and plot the most frequant 10
cast_df = movies_df.assign(cast=movies_df['cast'].str.split('|')).explode('c

plt.figure(figsize=(20, 10))
sns.countplot(y='cast', data=cast_df, order = cast_df['cast'].value_counts()

plt.title('Most Frequant 10 Actors Appearancees In The Movies', fontsize=25)
plt.ylabel('Movies\' Cast ', fontsize=25)
plt.xlabel('Count', fontsize=25)

plt.xticks(fontsize=20)
plt.yticks(fontsize=20)
plt.show()

```



Q3: what's the most frequant 10 production companies Produce movies?

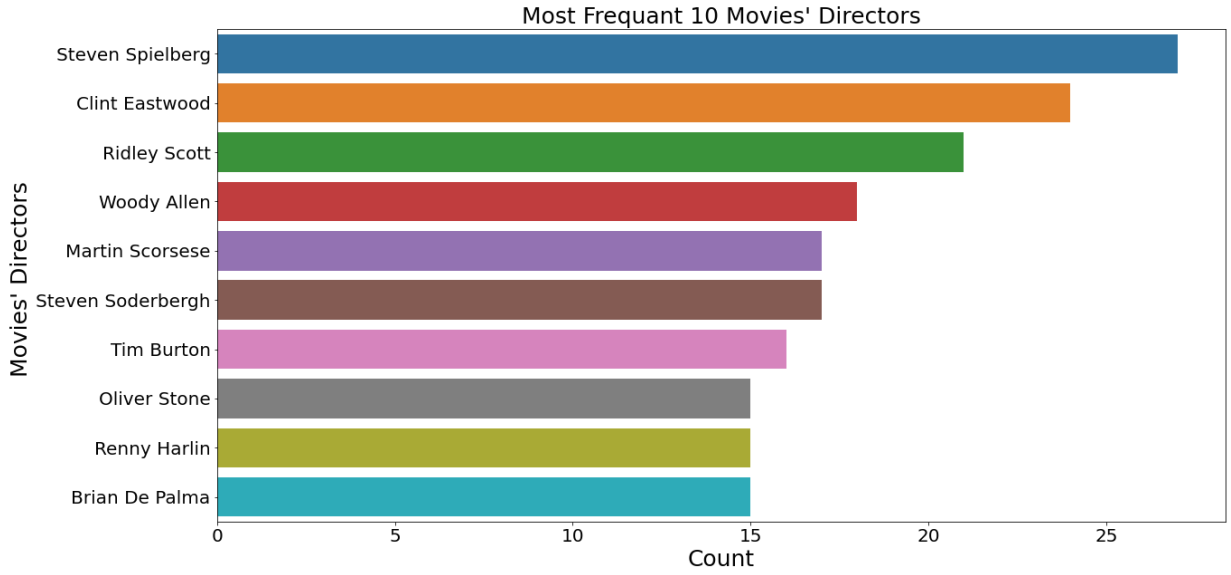
In [31]:

```
#split production_companies column, count the number of movie producing, and
production_companies_df = movies_df.assign(production_companies=movies_df['p

plt.figure(figsize=(20, 10))
sns.countplot(y='production_companies', data=production_companies_df, order

plt.title('Most Frequant 10 Production Companies Produce Movies', fontsize=2
plt.ylabel('Movies\' production companies ', fontsize=25)
plt.xlabel('Count', fontsize=25)

plt.xticks(fontsize=20)
plt.yticks(fontsize=20)
plt.show()


```



Q4: who's the most frequant 10 directors in the movies?

In [32]:
```
#count the directors and plot the most frequant 10 directors.
director_df = movies_df.explode('director')

plt.figure(figsize=(20, 10))
sns.countplot(y='director', data=director_df, order = director_df['director'

plt.title('Most Frequant 10 Movies\' Directors', fontsize=25)

plt.ylabel('Movies\' Directors ', fontsize=25)
plt.xlabel('Count', fontsize=25)

plt.yticks(fontsize=20)
plt.xticks(fontsize=20)
plt.show()


```

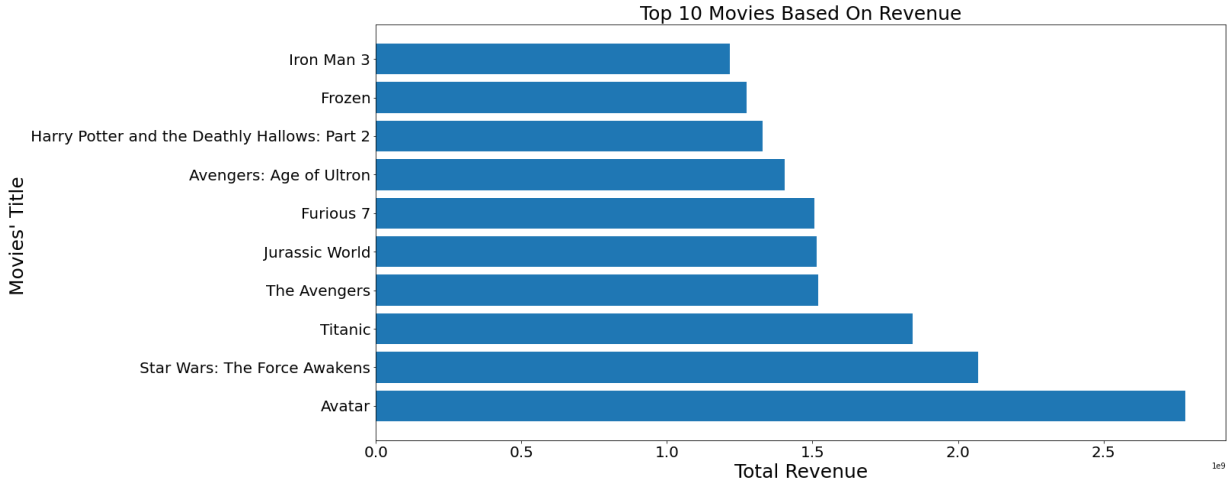


Q5: what's the Top and least 10 movies based on the revenue?

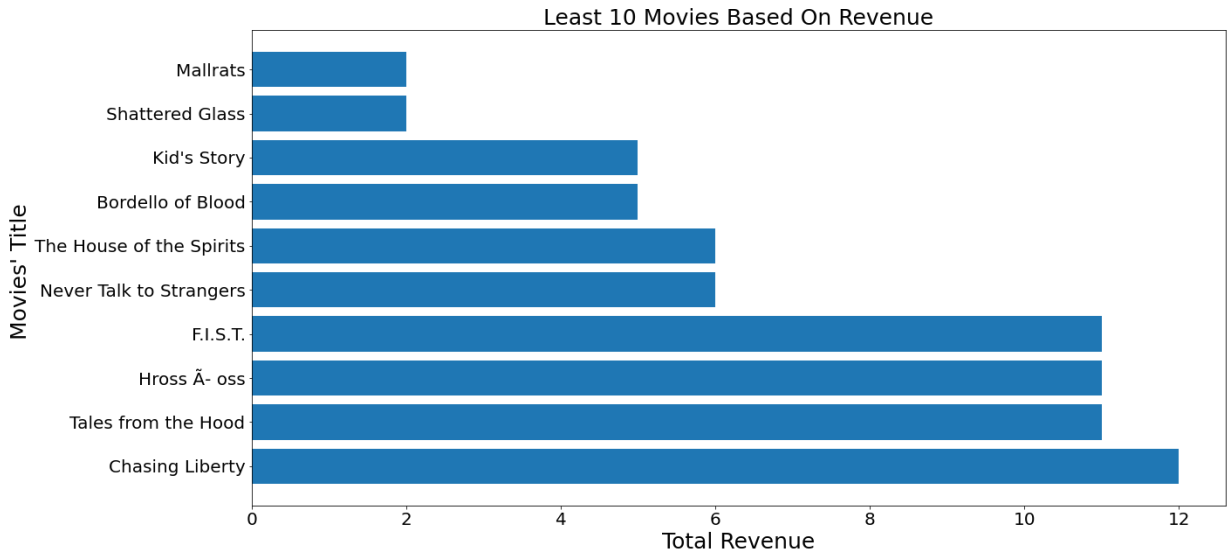
In [33]:
```
def top_10_movies(df, col_name):
    """
    Sort the movies titles and return the the top 10 movies title dapanding

    Return:
        x_axis- dataframe contains the top 10 movies titles.
        y_axis- dataframe contains the top 10 values of a certain column whi
    """
    movies_and_col = df[["original_title", col_name]]

    x_axis = movies_and_col.sort_values(by = col_name, ascending=False).head
    y_axis = movies_and_col.sort_values(by = col_name, ascending=False).head

    return x_axis, y_axis

```

In [34]:
```
def least_10_movies(df, col_name):
    """
    Sort the movies titles and return the the least 10 movies title dapandin

    Return:
        x_axis- dataframe contains the least 10 movies titles.
        y_axis- dataframe contains the least 10 values of a certain column w
    """
    movies_and_col = df[["original_title", col_name]]

    x_axis = movies_and_col.sort_values(by = col_name, ascending=False).tail
    y_axis = movies_and_col.sort_values(by = col_name, ascending=False).tail

    return x_axis, y_axis
```

In [35]:
```
#Apply top_10_movies(df, col_name) function on 'revenue' column.
x_axis = top_10_movies(movies_df, 'revenue')[0]
y_axis = top_10_movies(movies_df, 'revenue')[1]

plt.figure(figsize=(20, 10))
plt.barh(x_axis, y_axis)

plt.title('Top 10 Movies Based On Revenue', fontsize=25)

plt.ylabel('Movies\' Title ', fontsize=25)
plt.xlabel('Total Revenue', fontsize=25)

plt.yticks(fontsize=20)
plt.xticks(fontsize=20)
plt.show()


```



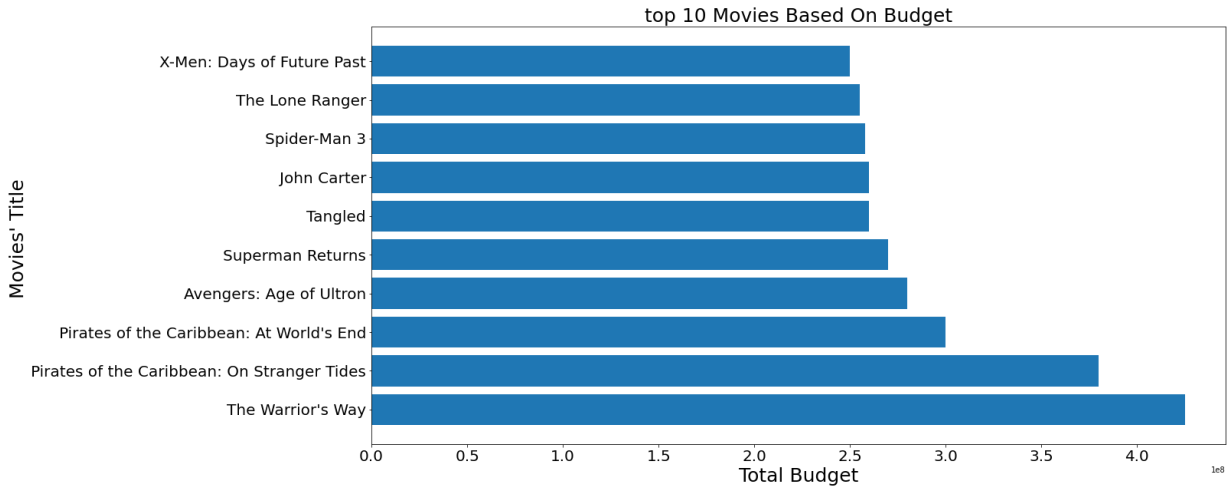
In [36]:
```
#Apply least_10_movies(df, col_name) function on 'revenue' column.

x_axis = least_10_movies(movies_df, 'revenue')[0]
y_axis = least_10_movies(movies_df, 'revenue')[1]

plt.figure(figsize=(20, 10))
plt.barh(x_axis, y_axis)

plt.title('Least 10 Movies Based On Revenue', fontsize=25)

plt.ylabel('Movies\' Title ', fontsize=25)
plt.xlabel('Total Revenue', fontsize=25)

plt.yticks(fontsize=20)
plt.xticks(fontsize=20)
plt.show()

```



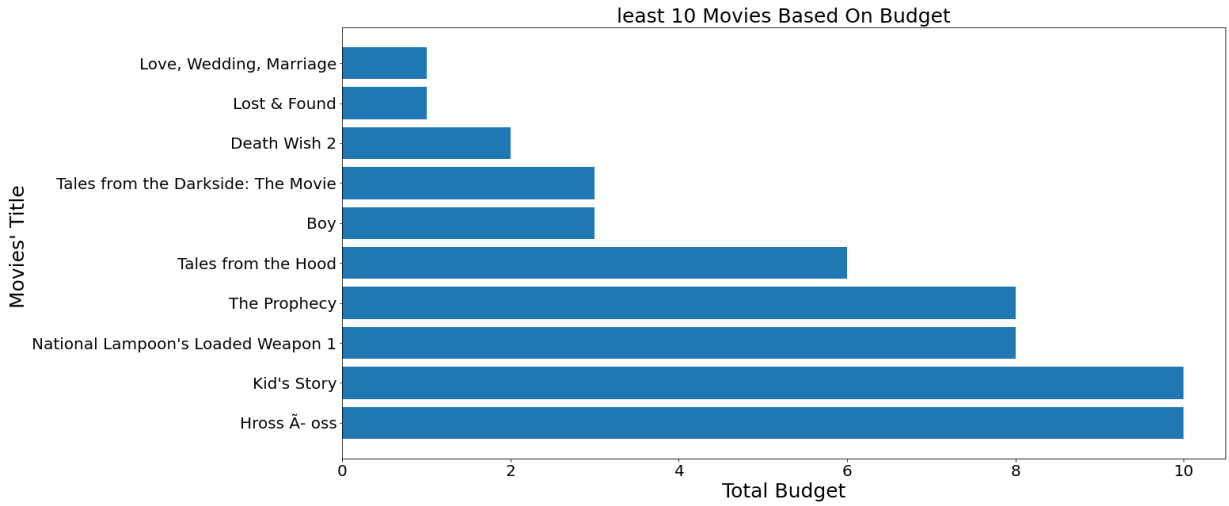
Q6: what's the Top and least 10 movies based on the budget?

In [37]:

```
#Apply top_10_movies(df, col_name) function on 'budget' column.

x_axis = top_10_movies(movies_df, 'budget')[0]
y_axis = top_10_movies(movies_df, 'budget')[1]

plt.figure(figsize=(20, 10))
plt.barh(x_axis, y_axis)

plt.title('top 10 Movies Based On Budget', fontsize=25)

plt.ylabel('Movies\' Title ', fontsize=25)
plt.xlabel('Total Budget', fontsize=25)

plt.yticks(fontsize=20)
plt.xticks(fontsize=20)
plt.show()
```



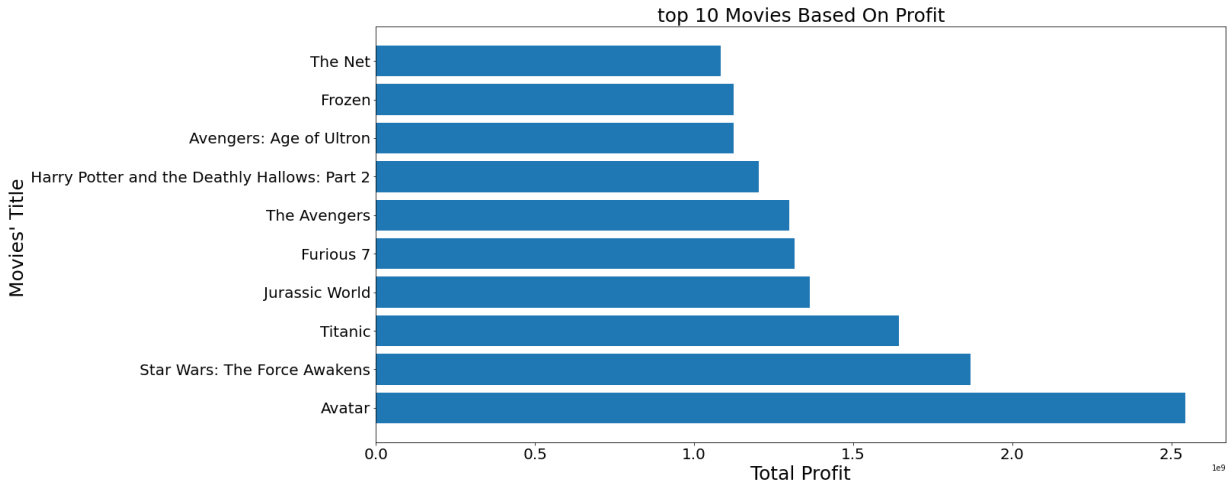
In [38]:
```
#Apply least_10_movies(df, col_name) function on 'budget' column.

x_axis = least_10_movies(movies_df, 'budget')[0]
y_axis = least_10_movies(movies_df, 'budget')[1]

plt.figure(figsize=(20, 10))
plt.barh(x_axis, y_axis)

plt.title('least 10 Movies Based On Budget', fontsize=25)

plt.ylabel('Movies\' Title ', fontsize=25)
plt.xlabel('Total Budget', fontsize=25)

plt.yticks(fontsize=20)
plt.xticks(fontsize=20)
plt.show()
```



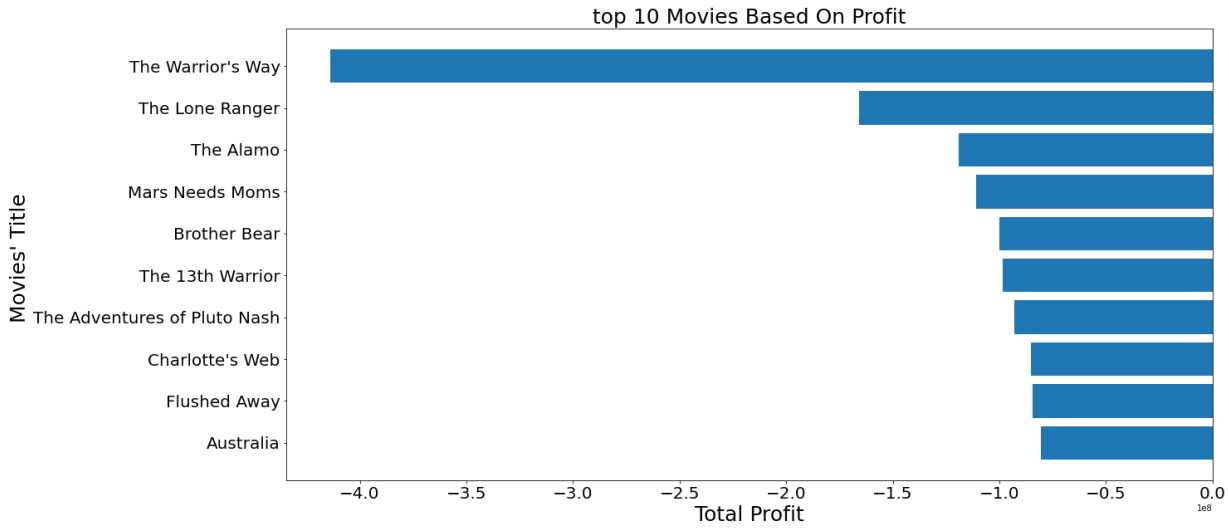
Q7: what's the Top and least 10 movies based on the profit?

In [39]:

```
#Apply top_10_movies(df, col_name) function on 'profit' column.

x_axis = top_10_movies(movies_df, 'profit')[0]
y_axis = top_10_movies(movies_df, 'profit')[1]

plt.figure(figsize=(20, 10))
plt.barh(x_axis, y_axis)

plt.title('top 10 Movies Based On Profit', fontsize=25)

plt.ylabel('Movies\' Title ', fontsize=25)
plt.xlabel('Total Profit', fontsize=25)

plt.yticks(fontsize=20)
plt.xticks(fontsize=20)
plt.show()
```



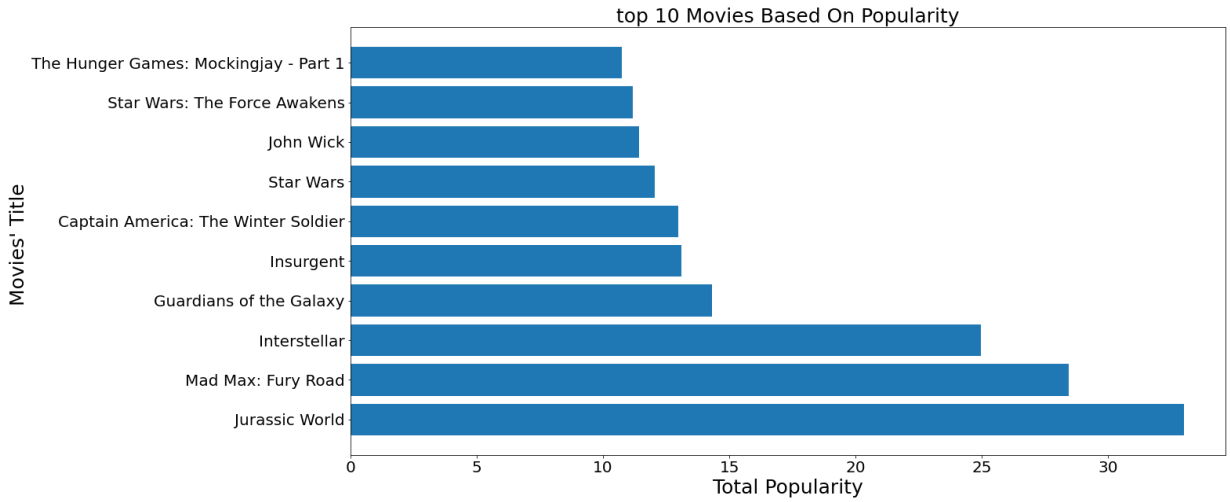
In [40]:
```
#Apply least_10_movies(df, col_name) function on 'profit' column.

x_axis = least_10_movies(movies_df, 'profit')[0]
y_axis = least_10_movies(movies_df, 'profit')[1]

plt.figure(figsize=(20, 10))
plt.barh(x_axis, y_axis)

plt.title('top 10 Movies Based On Profit', fontsize=25)

plt.ylabel('Movies\' Title ', fontsize=25)
plt.xlabel('Total Profit', fontsize=25)

plt.yticks(fontsize=20)
plt.xticks(fontsize=20)
plt.show()
```



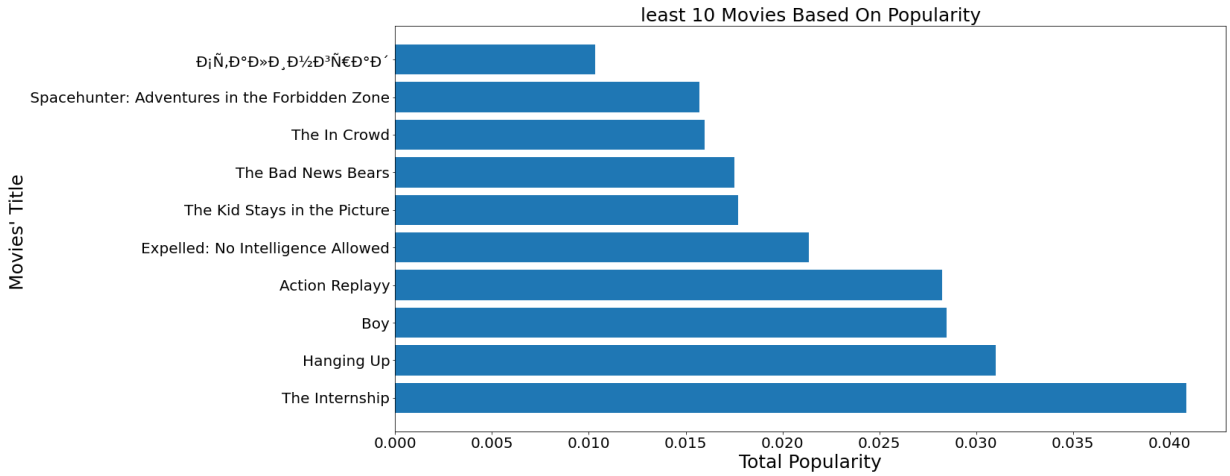
Q8: what's the Top and least 10 movies based on the popularity?

In [41]:

```
#Apply top_10_movies(df, col_name) function on 'popularity' column.

x_axis = top_10_movies(movies_df, 'popularity')[0]
y_axis = top_10_movies(movies_df, 'popularity')[1]

plt.figure(figsize=(20, 10))
plt.barh(x_axis, y_axis)

plt.title('top 10 Movies Based On Popularity', fontsize=25)

plt.ylabel('Movies\' Title ', fontsize=25)
plt.xlabel('Total Popularity', fontsize=25)

plt.yticks(fontsize=20)
plt.xticks(fontsize=20)
plt.show()
```



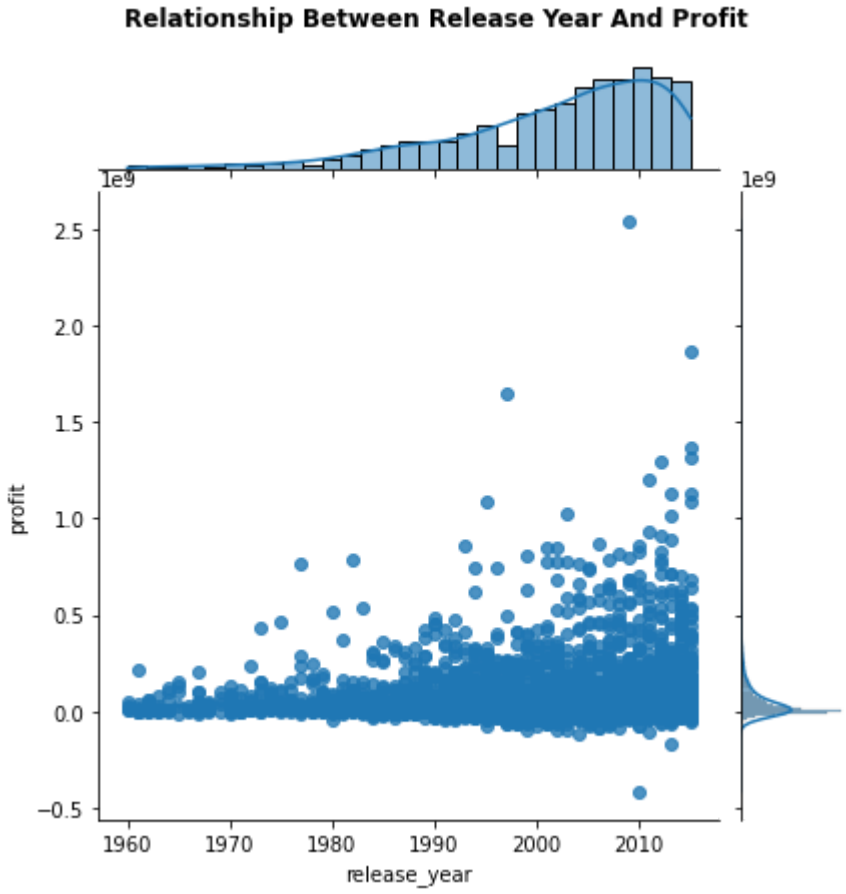
In [42]:
```
#Apply least_10_movies(df, col_name) function on 'popularity' column.

x_axis = least_10_movies(movies_df, 'popularity')[0]
y_axis = least_10_movies(movies_df, 'popularity')[1]

plt.figure(figsize=(20, 10))
plt.barh(x_axis, y_axis)

plt.title('least 10 Movies Based On Popularity', fontsize=25)

plt.ylabel('Movies\' Title ', fontsize=25)
plt.xlabel('Total Popularity', fontsize=25)

plt.yticks(fontsize=20)
plt.xticks(fontsize=20)
plt.show()
```



Q9: What's the properties with the movies with high profit?

1. Dose the year of realse associated with high profit?

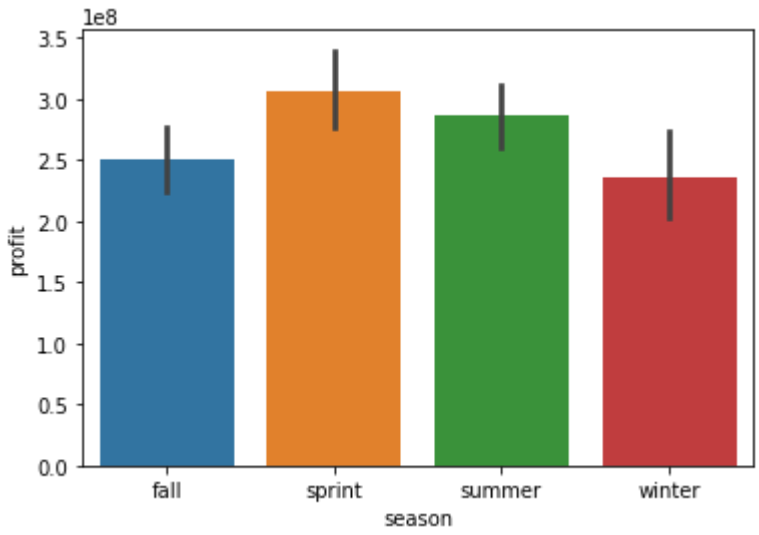
In [43]:
```
#plot a relationship between the profit and the release year.
sns.jointplot(x='release_year', y='profit', data=movies_df,kind='reg');
plt.suptitle("Relationship Between Release Year And Profit".title(), weight=
```



2. Dose the sesone of realse associated with high profit?

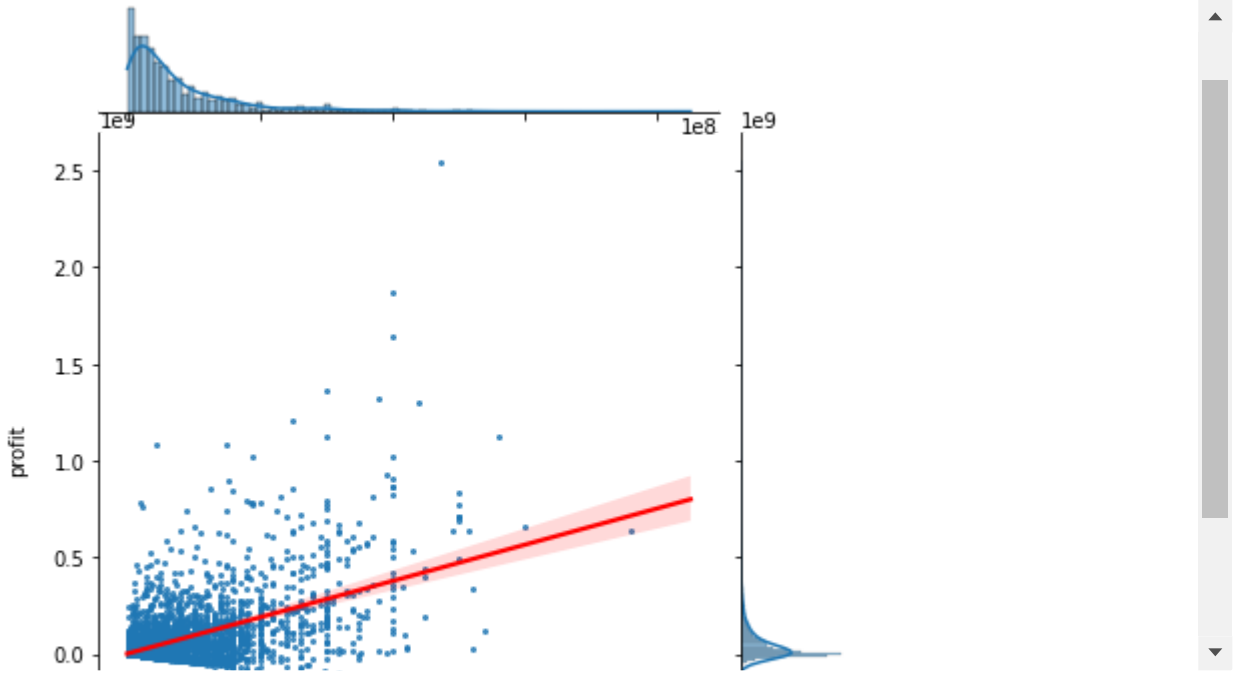
In [44]:
```
#make a new dataframe which include all the raws with profit >= 100 million.
high_profit = movies_df.query('profit >= 100000000')
```

In [45]:
```
#plot the sesone count compared to the total profits.
sns.barplot(x='season',y='profit',data=high_profit);
```



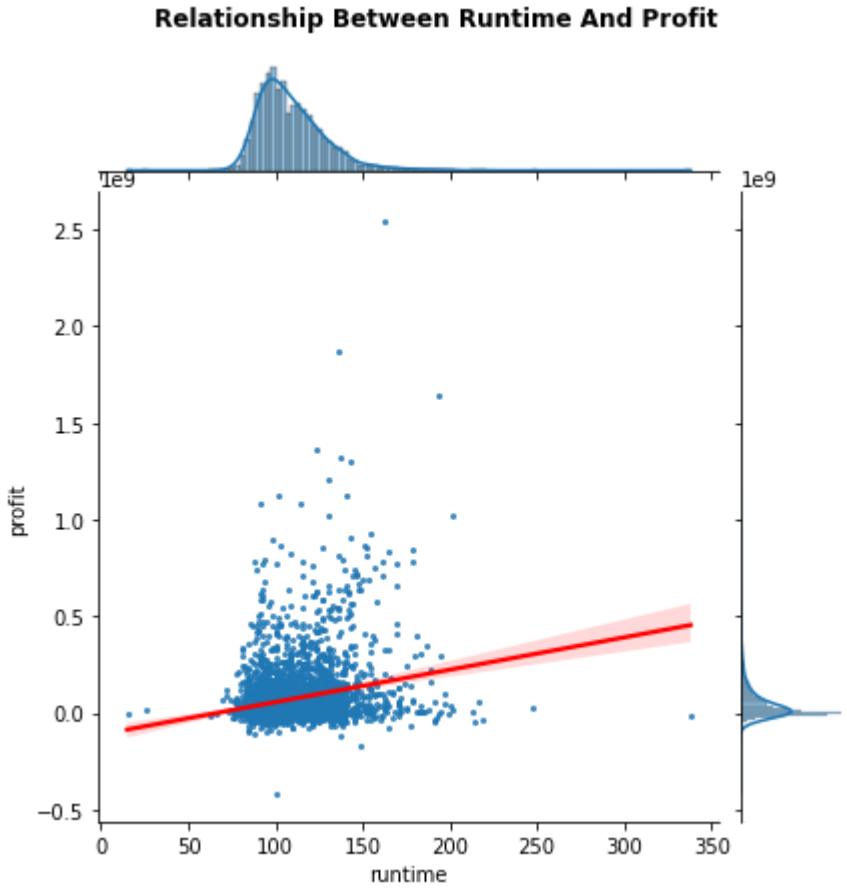
3. Dose the budget associated with high profit?

In [46]:
```
#plot a relationship betwwen budget and profit.
sns.jointplot(x='budget',y='profit',data=movies_df,kind='reg', scatter_kws={
plt.suptitle("Relationship Between Budget And Profit".title(), weight='bold'

```



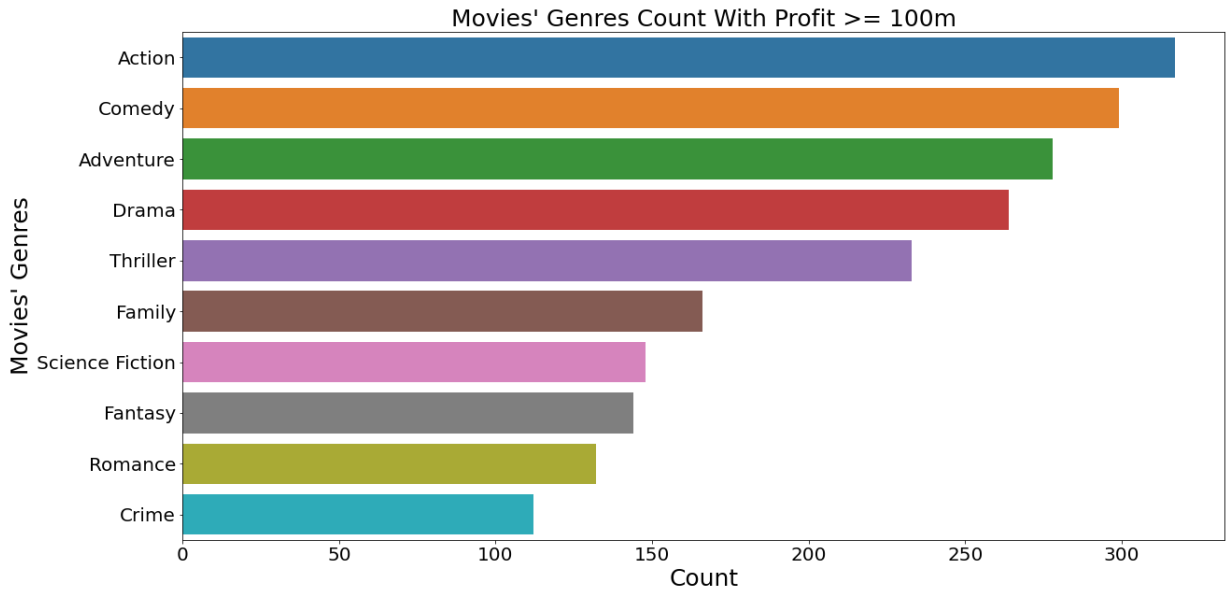
4. Dose the runtime of the movie associated with high profit?

In [47]:
```
#plot a relationship betwwen runtime and profit.
sns.jointplot(x='runtime',y='profit',data=movies_df,kind='reg', scatter_kws=
plt.suptitle("Relationship Between Runtime And Profit".title(), weight='bold
```



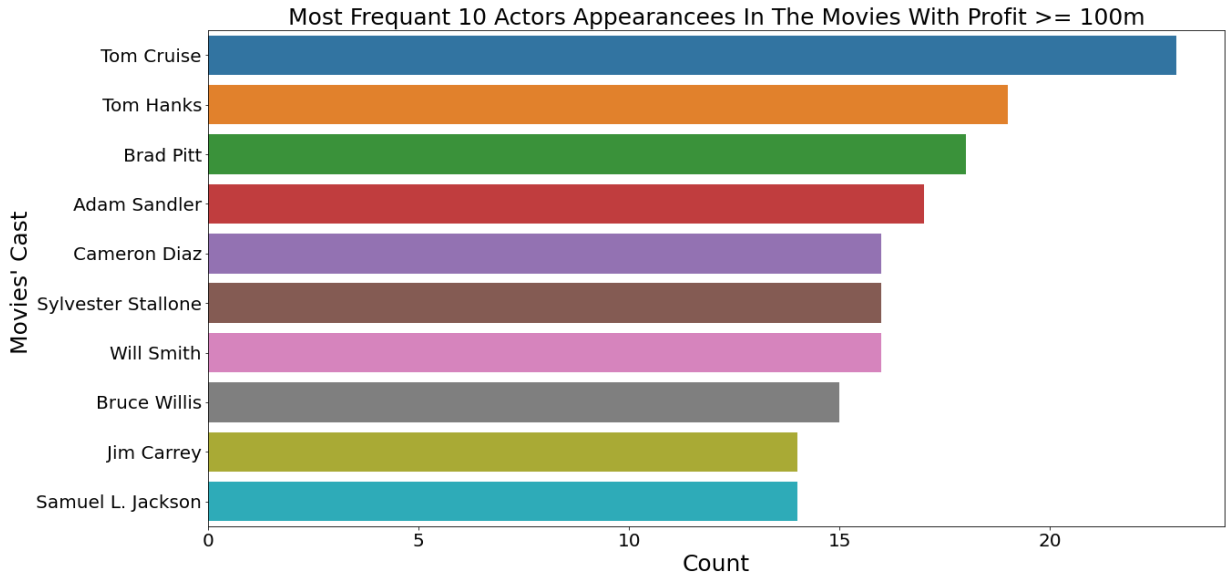
5. Is there any spacific production genre associated with high profit?

In [48]:
```
#plot the count of the most 10 frequant movies' genres with profit >=100 mil
high_profit_genres = high_profit.assign(genres=high_profit['genres'].str.spl

plt.figure(figsize=(20, 10))
sns.countplot(y='genres',data=high_profit_genres, order = high_profit_genres

plt.title('Movies\' Genres Count With Profit >= 100m', fontsize=25)
plt.ylabel('Movies\' Genres', fontsize=25)
plt.xlabel('Count', fontsize=25)

plt.xticks(fontsize=20)
plt.yticks(fontsize=20)
plt.show()

```



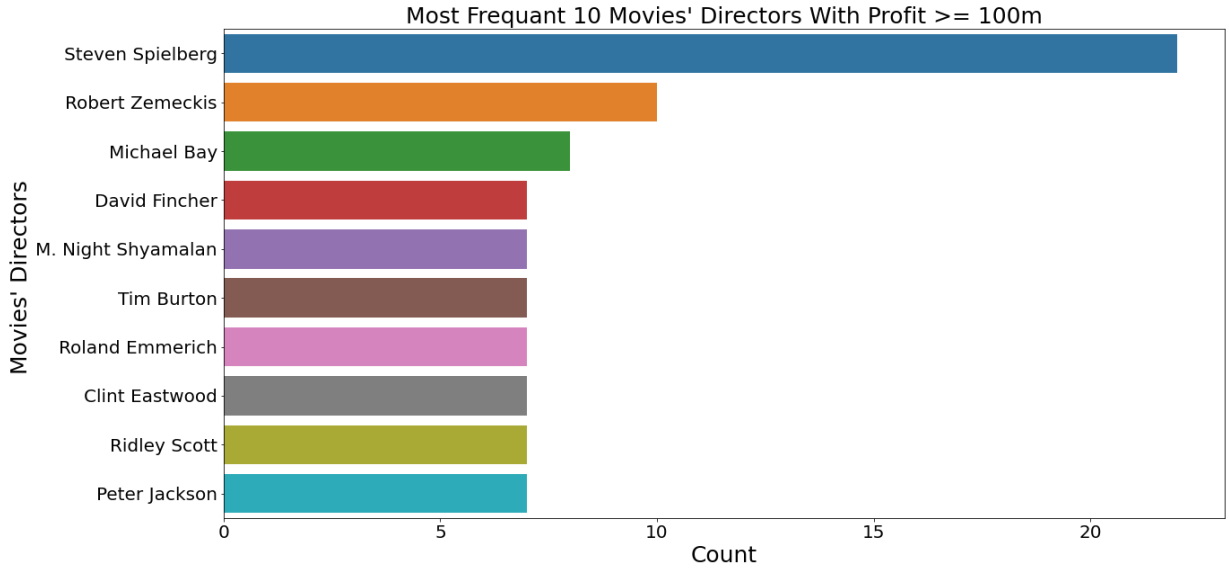
6. Is there any spacific cast member associated with high profit?

In [49]:
```
#plot the count of the most 10 frequant cast members with movies' profit >=1
high_profit_cast = high_profit.assign(cast=high_profit['cast'].str.split('|'

plt.figure(figsize=(20, 10))
sns.countplot(y='cast', data=high_profit_cast, order = high_profit_cast['cas

plt.title('Most Frequant 10 Actors Appearancees In The Movies With Profit >=
plt.ylabel('Movies\' Cast ', fontsize=25)
plt.xlabel('Count', fontsize=25)

plt.xticks(fontsize=20)
plt.yticks(fontsize=20)
plt.show()
```



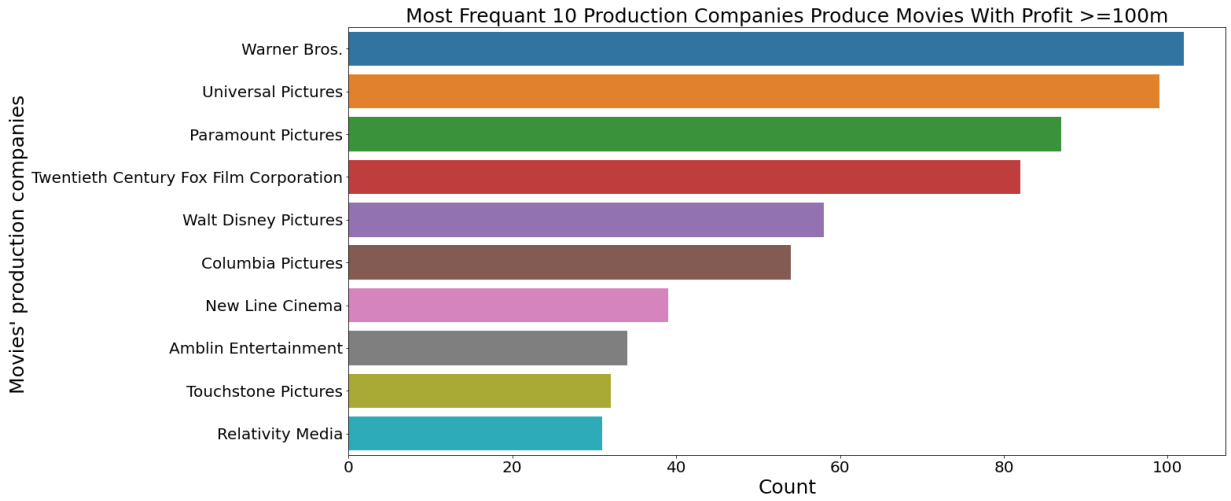
7. Is there any spacific director associated with high profit?

In [50]:
```
#plot the count of the most 10 frequant directors with movies' profit >=100 
high_profit_director = high_profit.explode('director')

plt.figure(figsize=(20, 10))
sns.countplot(y='director', data=high_profit_director, order = high_profit_d

plt.title('Most Frequant 10 Movies\' Directors With Profit >= 100m', fontsiz

plt.ylabel('Movies\' Directors ', fontsize=25)
plt.xlabel('Count', fontsize=25)

plt.yticks(fontsize=20)
plt.xticks(fontsize=20)
plt.show()

```



8. Is there any spacific production company associated with high profit?

In [51]:
```
#plot the count of the most 10 frequant production companies with movies' pr
high_profit_production_companies = high_profit.assign(production_companies=h

plt.figure(figsize=(20, 10))
sns.countplot(y='production_companies', data=high_profit_production_companie

plt.title('Most Frequant 10 Production Companies Produce Movies With Profit
plt.ylabel('Movies\' production companies ', fontsize=25)
plt.xlabel('Count', fontsize=25)

plt.xticks(fontsize=20)
plt.yticks(fontsize=20)
plt.show()

```



Conclusions

Q1. what's the most and least frquant movie's genres?

from the count plot:

- The most frequant movies' genres: Drama
- The least frequant movies' genres: TV Movie

Q2. who's the most frequant 10 actors appearancees in the movies?

from the count plot:

The most frequant 10 actors appearancees in the movies:

1. 'Robert De Niro'
2. 'Bruce Willis'
3. 'Samuel L. Jackson'
4. 'Nicolas Cage'
5. 'Matt Damon'

6. 'Johnny Depp'
7. 'Harrison Ford'
8. 'Brad Pitt'
9. 'Tom Hanks'
10. 'Sylvester Stallone'

Q3. what's the most frequant 10 production companies Produce movies?

from the count plot:

The most frequant 10 production companies produce movies:

1. 'Universal Pictures'
2. 'Warner Bros.'
3. 'Paramount Pictures'
4. 'Twentieth Century Fox Film Corporation'
5. 'Columbia Pictures'
6. 'New Line Cinema'
7. 'Walt Disney Pictures'
8. 'Touchstone Pictures'
9. 'Metro-Goldwyn-Mayer (MGM)'
10. 'Relativity Media'

Q4. who's the most frequant 10 directors in the movies?

from the count plot:

The most frequant 10 directors in the movies

1. 'Steven Spielberg'
2. Clint Eastwood'
3. 'Ridley Scott'
4. 'Woody Allen'
5. 'Steven Soderbergh'
6. 'Martin Scorsese'
7. 'Tim Burton'
8. 'Robert Zemeckis'
9. 'Renny Harlin'
10. 'Oliver Stone'

Q5. what's the Top and least 10 movies based on the revenue?

from the bar chart:

- The top 10 movies based on the revenue:

1. Avatar
2. Star Wars: The Force Awakens
3. Titanic

4. The Avengers
5. Jurassic World
6. Furious 7
7. Avengers: Age of Ultron
8. Harry Potter and the Deathly Hallows: Part 2
9. Frozen
10. Iron Man 3

- The least 10 movies based on the revenue:

1. Chasing Liberty
2. Tales from the Hood
3. Hross Ã oss
4. F.I.S.T.
5. Never Talk to Strangers
6. The House of the Spirits
7. Bordello of Blood
8. Kid's Story
9. Shattered Glass
10. Mallrats

Q6. what's the Top and least 10 movies based on the budget?

from the bar chart:

- The Top 10 movies based on the budget:

1. The Warrior's Way
2. Pirates of the Caribbean: On Stranger Tides
3. Pirates of the Caribbean: At World's End
4. Avengers: Age of Ultron
5. Superman Returns
6. Tangled
7. John Carter
8. Spider-Man 3
9. The Lone Ranger
10. X-Men: Days of Future Pa

- The least 10 movies based on the budget:

1. Hross Ã oss
2. Kid's Story
3. National Lampoon's Loaded Weapon 1
4. The Prophecy
5. Tales from the Hood
6. Boy
7. Tales from the Darkside: The Movie
8. Death Wish 2
9. Lost & Found
10. Love, Wedding, Marriage

Q7. what's the Top and least 10 movies based on the profit?

from th bar chart:

- The top 10 movies based on the profit:

1. Avatar
2. Star Wars: The Force Awakens
3. Titanic
4. Jurassic World
5. Furious 7
6. The Avengers
7. Harry Potter and the Deathly Hallows: Part 2
8. Avengers: Age of Ultron
9. Frozen
10. The Net

- The least 10 movies based on the profit:

1. Australia
2. Flushed Away
3. Charlotte's Web
4. The Adventures of Pluto Nash
5. The 13th Warrior
6. Brother Bear
7. Mars Needs Moms
8. The Alamo
9. The Lone Ranger
10. The Warrior's Way

Q8. what's the Top and least 10 movies based on the popularity?

form the par chart:

- The top 10 movies based on the popularity:

1. Jurassic World
2. Mad Max: Fury Road
3. Interstellar
4. Guardians of the Galaxy
5. Insurgent
6. Captain America: The Winter Soldier
7. Wars
8. John Wick
9. Star Wars: The Force Awakens
10. The Hunger Games: Mockingjay - Part 1

- The least 10 movies based on the popularity:

1. The Internship
2. Hanging Up
3. Boy
4. Action Replayy
5. Expelled: No Intelligence Allowed
6. The Kid Stays in the Picture
7. The Bad News Bears
8. The In Crowd
9. Spacehunter: Adventures in the Forbidden Zone
10. Ð¡Ñ‚Ð°Ð»Ð¸Ð½Ð³Ñ€Ð°Ð´

Q9. What's the properties with the movies with high profit?

Q9.1. Dose the year of realse associated with high profit?

from the scatter plot, we can conclude that pravious last few years have the most profits earned by the movies.

Q9.2. Dose the sesone of realse associated with high profit?

we can conclude from the count plot that spring has the most profits earned by the movies.

Q9.3. Dose the budget associated with high profit?

we can conclude from the scatter plot that the movies with high profits, have a budget in the range (150:200)m

Q9.4. Dose the runtime of the movie associated with high profit? ??

we can conclude from the scatter plot that the movies with highest profits its runtime in a range (100:150) mins.

Q9.5. Is there any spacific genre associated with high profit?

from the count plot:

Action, comedy, adventure, drama, and thriller movies are associated with high profit.

Q9.6. Is there any cast member associated with high profit?

from the count plot:

Tom Cruise', Tom Hanks', 'Brad Pitt', Adam Sandler', 'Will Smith', 'Sylvester Stallone', 'Cameron Diaz', 'Bruce Willis', 'Samuel L. Jackson', and 'Jim Carrey' are associated with high profit

Q9.7. Is there any spacific director associated with high profit?

from the count plot:

'Steven Spielberg', 'Robert Zemeckis', 'Michael Bay', 'Ridley Scott', 'Chris Columbus', 'Peter Jackson', 'David Fincher', 'Clint Eastwood', 'Roland Emmerich', 'Tim Burton' are associated with high profit

Q9.8. Is there any spacific production company associated with high profit?

from the count plot:

'Warner Bros.', 'Universal Pictures', 'Paramount Pictures', 'Twentieth Century Fox Film Corporation', 'Walt Disney Pictures', 'Columbia Pictures','New Line Cinema', 'Amblin Entertainment', 'Touchstone Pictures', 'Relativity Media' are associated with high profit

*** High profit movies factors:**

1. **The year of release:** the movies which are produced in early years has higher profit than ones which are produced in previous. (but it can't be a factor, becouse we can't control it.)
2. **The sesone of realse:** we conclude that the sesone is very effective factor, the movies which was realsed in **Spring** have high profits.
3. **The budget:** you must have a budget rang **(150:200) milion**.
4. **The runtime:** the movie's runtime must be in a rang **(100:150) mins**.
5. **The Genres:** the movie must be **action, comedy, adventure, drama, or thriller**.
6. **The cast:** one or more of these cast must be in the movie, **[Tom Cruise', Tom Hanks', 'Brad Pitt', Adam Sandler', 'Will Smith', 'Sylvester Stallone', 'Cameron Diaz', 'Bruce Willis', 'Samuel L. Jackson', and 'Jim Carrey']** .
7. **The director:** one or more of these directors must be in the movie **['Steven Spielberg', 'Robert Zemeckis', 'Michael Bay', 'Ridley Scott', 'Chris Columbus', 'Peter Jackson', 'David Fincher', 'Clint Eastwood', 'Roland Emmerich', 'Tim Burton']** (**'Steven Spielberg'** is very recommended).
8. **The production company:** one or more of these production companies must be in the movie **['Warner Bros.', 'Universal Pictures', 'Paramount Pictures', 'Twentieth Century Fox Film Corporation', 'Walt Disney Pictures', 'Columbia Pictures','New Line Cinema', 'Amblin Entertainment', 'Touchstone Pictures', 'Relativity Media']**

limitations:

- the data had too many zero values in budget and revenue columns, after cleaning the the raws with zero values, our data become very small, so the analysis could be not very accurate and not completly error free, but our analysis can increace the profit by following the factors of high profits.
- The data in the columns of (Directors, genres, production_companies) are sperated by '|' so i needed to remove '|' before making any visuliztions.

In [ ]:
```
1
```