Movie Industry Analysis

Overview

The purpose of this project is to study the movie industry and generate advice on how a new entrant to the industry might create and follow a strategy that is aligned with corporate interests and will assure success.

Business understanding

A large, multinational, technology company has engaged us to explore industry movie trends and identify what type of films are successful at the Box Office and to help them formulate a film strategy that is relevant and optimal for their business concerns.

Data understanding

This project pursues an analysis of objective data that ties film characteristics and attributes to a film's financial performance. The profitability of a movie was analyzed with respect to:
- The director
- Production budget
- Length of a movie

Two databases were chosen for analysis:
[The Internet Movie Database (IMDb) (https://www.imdb.com/)](https://www.imdb.com/)
Provides information on movie title, year of release, running time (length), director, actors, etc.
[The Numbers (TN) (https://www.the-numbers.com/)](https://www.the-numbers.com/)
Provides information on a movie's financial performance including release date , production budget and domestic and worldwide gross.

Data assumptions and preparation

We distinguished movies with the same title by the year in which it was released.
The exact duplicate entries were removed.
In calculation of average rating, many movies had too few. We only looked at movies that had more than 500 votes.

The TN database contained financial information and had the movie name in common with the IMDb. It had also a full release date as opposed to only the release year in IMDb. We calculated the release year in the TN database as a separate column and then used the movie name and release year to join these two databases. Note: The TN database addition of the Release Year column was carried out in a separate Jupyter Notebook called

"Exploration_tn_movie_budgets.ipynb" inthe folder called "Study". The enhanced dataset was exported as .csv file to teh data folder called "zippedData" and is called "NzModified_tn.movie_budgets.csv". This .csv file is imported in this notebook.

The resultant data set contains information on a little over 2,000 movies which is sufficient to identify trends.
Profit was calculated as:
- Profit = (Domestic_Gross + Worldwide_Gross) – Production_Budget
- Profit Percentage = Profit / Production_Cost * 100.
This Profit Percentage is a key measure which we use to judge success.

Initial exploration of databases

All exporation and initial analyses were carried out in Jupyter Notebooks in the folder called "Study".
**This notebook contains only code pertinent to data included in the final analysis and presentation.

In [1]:
```
import pandas as pd
import sqlite3
from matplotlib import pyplot as plt
import seaborn as sns
```

Connect to im.db database.

Join movie_basics, directors, persons and movie_ratings tables and with Number of votes greater than 500.

In [2]:
```
conn = sqlite3.connect('./zippedData/im.db')
```

In [3]:

```
query = """
SELECT DISTINCT mb.primary_title AS "Movie Name", mb.start_year AS "Relea

FROM movie_basics mb

    LEFT JOIN directors d
        ON mb.movie_id = d.movie_id
    LEFT JOIN persons p
        ON d.person_id = p.person_id
    LEFT JOIN movie_ratings mr
        ON mb.movie_id = mr.movie_id

WHERE mr.numvotes > 500

ORDER BY mr.averagerating DESC

"""
imdb_df = pd.read_sql_query(query, conn)
len(imdb_df)
```

Out[3]: 15595

In [4]:

```
imdb_df.head()
```

Out[4]:

	Movie Name	Release Year	Length	Genres	Director	Avg Rating	Number of Votes
0	Once Upon a Time ... in Hollywood	2019	159.0	Comedy,Drama	Quentin Tarantino	9.7	5600
1	Eghantham	2018	125.0	Drama	Arsel Arumugam	9.7	639
2	Yeh Suhaagraat Impossible	2019	92.0	Comedy	Abhinav Thakur	9.6	624
3	Ananthu V/S Nusrath	2018	149.0	Comedy,Drama,Family	Sudheer Shanbhogue	9.6	808
4	Ekvtime: Man of God	2018	132.0	Biography,Drama,History	Nikoloz Khomasuridze	9.6	2604

Read in csv file derived from tn_movie database

As mentioned above, the following csv file was created in a separate Jupyter Notebook called "Exploration_tn_movie_budgets". It has an added column containing just the release year as opposed to the full date.

In [5]:

```
tn_df = pd.read_csv('./zippedData/NzModified_tn.movie_budgets.csv')
len(tn_df)
```

Out[5]: 5782

In [6]:
```
tn_movie_list = tn_df['movie']
tn_df['movie'][0]
```

Out[6]:
```
'Avatar'
```

In [7]:
```
imdb_df.keys()
```

Out[7]:
```
Index(['Movie Name', 'Release Year', 'Length', 'Genres', 'Director',
       'Avg Rating', 'Number of Votes'],
      dtype='object')
```

In [8]:
```
imdb_df['Movie Name'].head()
```

Out[8]:
```
0    Once Upon a Time ... in Hollywood
1                            Eghantham
2              Yeh Suhaagraat Impossible
3                  Ananthu V/S Nusrath
4                  Ekvtime: Man of God
Name: Movie Name, dtype: object
```

Merge imdb and tn dataframes with inner join on movie names

In [9]:
```
merged_df = imdb_df.merge(tn_df, left_on="Movie Name", right_on="movie",
```

In [10]:
```
merged_df.head()
```

Out[10]:

	Movie Name	Release Year	Length	Genres	Director	Avg Rating	Number of Votes	Unnamed: 0
0	Frankenstein	2011	130.0	Drama	Danny Boyle	9.0	1832	1302
1	Frankenstein	2015	89.0	Horror,Sci-Fi,Thriller	Bernard Rose	5.1	2089	1302
2	Inception	2010	148.0	Action,Adventure,Sci-Fi	Christopher Nolan	8.8	1841066	137
3	Coriolanus	2014	192.0	Drama,History,War	Tim Van Someren	8.7	1347	3698
4	Coriolanus	2011	123.0	Drama,Thriller,War	Ralph Fiennes	6.1	29654	3698

In [11]:
```
merged_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2112 entries, 0 to 2111
Data columns (total 15 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Movie Name         2112 non-null   object 
 1   Release Year       2112 non-null   int64  
 2   Length             2111 non-null   float64
 3   Genres             2112 non-null   object 
 4   Director           2112 non-null   object 
 5   Avg Rating         2112 non-null   float64
 6   Number of Votes    2112 non-null   int64  
 7   Unnamed: 0         2112 non-null   int64  
 8   id                 2112 non-null   int64  
 9   release_date       2112 non-null   object 
 10  movie              2112 non-null   object 
 11  production_budget  2112 non-null   object 
 12  domestic_gross     2112 non-null   object 
 13  worldwide_gross    2112 non-null   object 
 14  year               2112 non-null   int64  
dtypes: float64(2), int64(5), object(8)
memory usage: 264.0+ KB
```

This is a significantly reduced data set compared to the imdb database but still has sufficient number of records to derive trends.

Sort merged data by "domestic_gross' in descending order

In [12]:
```
merged_df.sort_values(by=['domestic_gross'], ascending=False)
```

Out[12]:

	Movie Name	Release Year	Length	Genres	Director	Avg Rating	Number of Votes	Un
385	Christopher Robin	2018	104.0	Adventure,Animation,Comedy	Marc Forster	7.3	52737	
1401	Hercules	2014	98.0	Action,Adventure,Fantasy	Brett Ratner	6.0	137287	
1022	Olympus Has Fallen	2013	119.0	Action,Thriller	Antoine Fuqua	6.5	235443	
1521	The Green Hornet	2011	119.0	Action,Comedy,Crime	Michel Gondry	5.8	148622	
1179	Date Night	2010	88.0	Comedy,Crime,Romance	Shawn Levy	6.3	144683	
...	...	...	...	...	...	...	...	
1932	The Veil	2016	93.0	Horror	Phil Joanou	4.8	6895	
1933	The Veil	2017	93.0	Action,Adventure,Sci-Fi	Brent Ryan Green	3.5	1236	
1691	Survivor	2015	96.0	Action,Crime,Thriller	James McTeigue	5.6	28614	
1935	Dawn Patrol	2014	88.0	Drama,Thriller	Daniel Petrie Jr.	4.8	615	
1602	Queen of the Desert	2015	128.0	Adventure,Biography,Drama	Werner Herzog	5.7	8529	

2112 rows × 15 columns

Gross and budget columns contain string values. Convert them to floats.

In [13]:
```
merged_df['float_production_budget'] = merged_df['production_budget'].re
merged_df['float_domestic_gross'] = merged_df['domestic_gross'].replace(
merged_df['float_worldwide_gross'] = merged_df['worldwide_gross'].replac
```

Calculate profit percentage and create 'profit percent' column
Profit = (Domestic_Gross + Worldwide_Gross) – Production_Budget
Profit Percentage = Profit / Production_Cost * 100

In [14]:
```
1 merged_df['profit percent'] = (merged_df['float_domestic_gross'] + merge
```

In [15]:
```
1 merged_df.head()
```

Out[15]:

	Movie Name	Release Year	Length	Genres	Director	Avg Rating	Number of Votes	Unnamed: 0
0	Frankenstein	2011	130.0	Drama	Danny Boyle	9.0	1832	1302
1	Frankenstein	2015	89.0	Horror,Sci-Fi,Thriller	Bernard Rose	5.1	2089	1302
2	Inception	2010	148.0	Action,Adventure,Sci-Fi	Christopher Nolan	8.8	1841066	137
3	Coriolanus	2014	192.0	Drama,History,War	Tim Van Someren	8.7	1347	3698
4	Coriolanus	2011	123.0	Drama,Thriller,War	Ralph Fiennes	6.1	29654	3698

Create new df of merged_df grouped by "Director"

Create a new series (column) of the count of movies directed by each director.
Calculate the means of each element in the newly grouped df called director_means.
Add the count series to the director_means df.

In [16]:

```
director_count = merged_df.groupby(by='Director')['Director'].count()
director_means = merged_df.groupby(by='Director')[['Avg Rating', 'Number

director_means['count'] = director_count
director_means
```

Out[16]:

	Avg Rating	Number of Votes	float_production_budget	float_domestic_gross	float_worldwide_gr
Director					
Aaron Hann	6.0	30645.0	2000000.0	10024.0	1002
Aaron Seltzer	3.4	43984.0	20000000.0	36661504.0	8142498
Aaron T. Wells	3.5	2230.0	500000.0	0.0	
Abby Kohn	5.4	39936.0	32000000.0	48795601.0	9155379
Abdolreza Kahani	7.0	903.0	4000000.0	0.0	6318
...	...	...	...	...	
Zackary Adler	5.0	1723.0	2500000.0	0.0	
Zak Forsman	5.2	846.0	50000.0	0.0	
Zal Batmanglij	6.7	33095.5	3317500.0	1341332.0	172870
Zhigang Yang	7.1	581.0	70000000.0	55011732.0	9497354
Zsófia Szilágyi	7.2	501.0	15000000.0	13843771.0	5916869

1426 rows × 7 columns

Sort of Directors by Count of Movie Releases

Sort director_means df by count to list in order of most prolific directors. Only top 20 directors in sorted list will be used in the profit percentage and busget charts below.

In [17]:

```
director_sort_by_count_df = director_means.sort_values(by=['count', 'Avg
```

In [18]:

```
director_sort_by_count_df.index
```

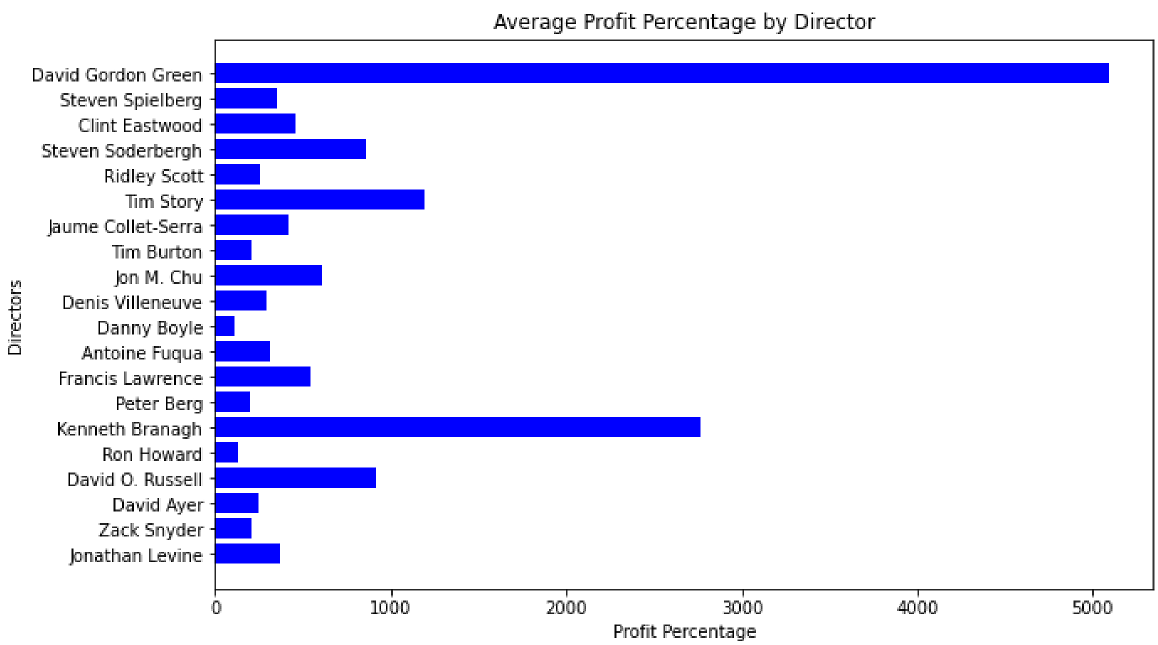
Out[18]:

```
Index(['David Gordon Green', 'Steven Spielberg', 'Clint Eastwood',
       'Steven Soderbergh', 'Ridley Scott', 'Tim Story', 'Jaume Collet-Serr
a',
       'Tim Burton', 'Jon M. Chu', 'Denis Villeneuve',
       ...
       'Matthew R. Anderson', 'Jamie Buckner', 'Timothy Woodward Jr.',
       'Glenn Ciano', 'David Winning', 'David DeCoteau', 'Kaizad Gustad',
       'Frédéric Auburtin', 'Justin Price', 'Lawrence Kasanoff'],
      dtype='object', name='Director', length=1426)
```

In [19]:

```
fig, ax = plt.subplots(figsize=(10,6))

ax.barh(director_sort_by_count_df.index[0:20], director_sort_by_count_df
ax.invert_yaxis()
ax.set_title("Average Profit Percentage by Director")
ax.set_xlabel('Profit Percentage')
ax.set_ylabel('Directors')
plt.savefig('./Images//Profit_Percentage_by_Director_for_High_Movie_Count
```

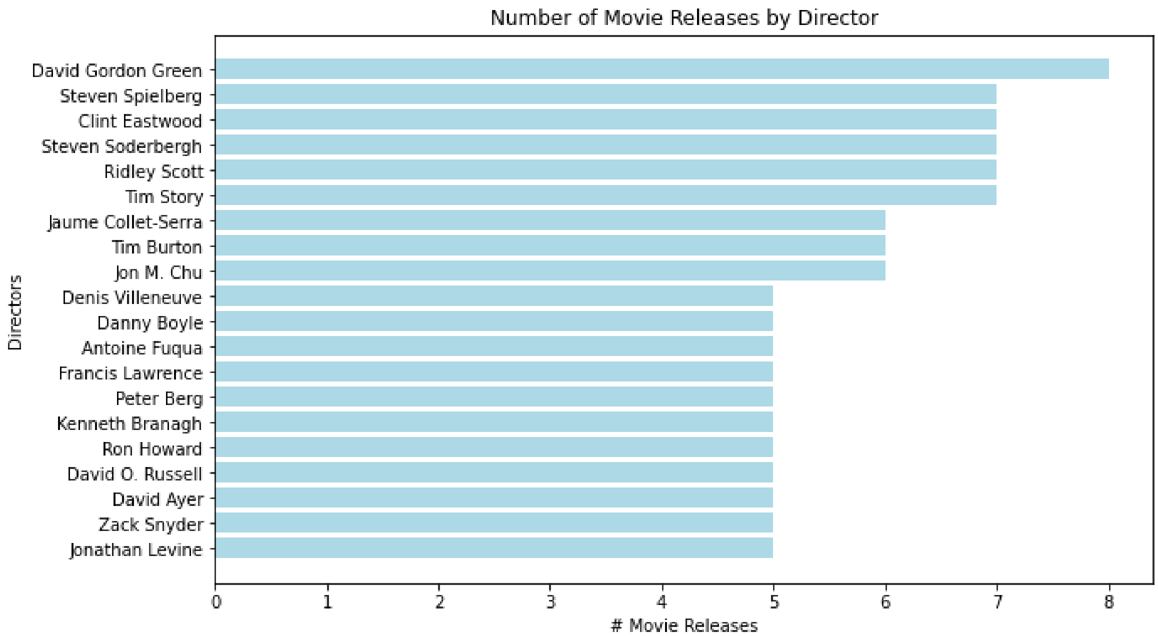


This chart shows that of the top 20 directors by movie count, all of them were very profitable exceeding 100% and couple in the range of thousands of percent. Clearly, there is a correlation between directors with a track record and their profitability.

In [20]:

```
fig, ax = plt.subplots(figsize=(10,6))

ax.barh(director_sort_by_count_df.index[0:20], director_sort_by_count_df
ax.invert_yaxis()
ax.set_title("Number of Movie Releases by Director")
ax.set_xlabel('# Movie Releases')
ax.set_ylabel('Directors')
plt.savefig('./Images/High_Count_of_Movie_Releases_by_Director.png',bbox_
```

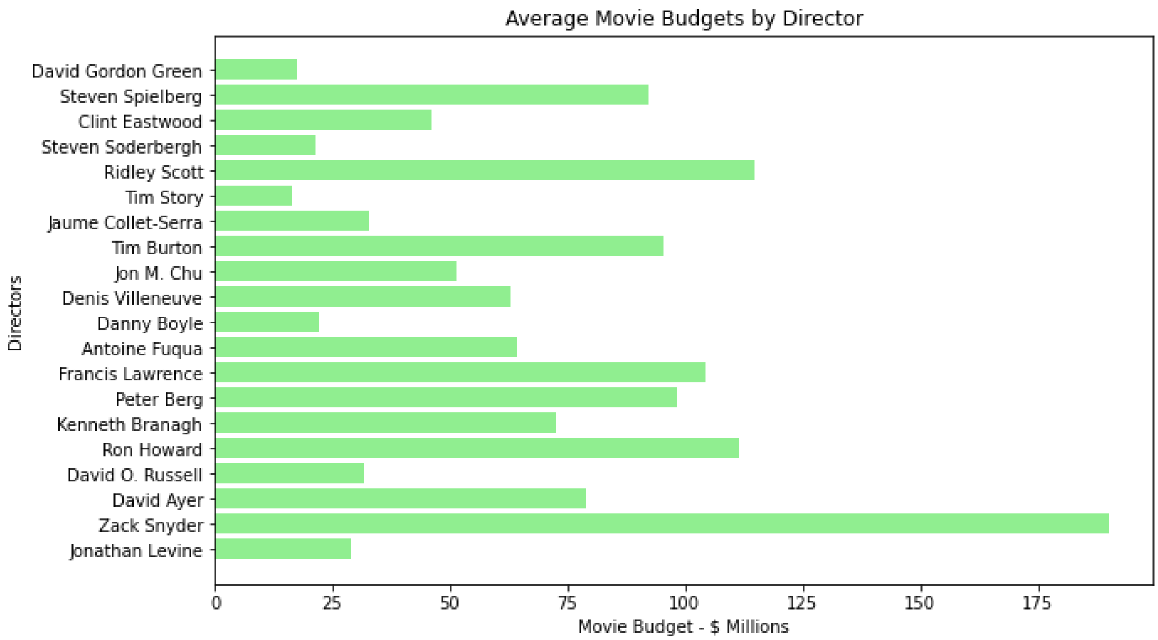


This chart was generated to show the track record of the top directors. As is apparent, they all have made multiple movies.

In [21]:

```
fig, ax = plt.subplots(figsize=(10,6))

ax.barh(director_sort_by_count_df.index[0:20], director_sort_by_count_df
ax.invert_yaxis()
ax.set_title("Average Movie Budgets by Director")
ax.set_xlabel('Movie Budget - $ Millions')
ax.set_ylabel('Directors')
plt.savefig('./Images/Movie_Budgets_of_Directors_with_High_Count_of_Movie
```



The previous charts raised the question as to why some directors can be phenominally profitable and others who are known for making well received movies have not earned as high a profit. This chart along with some general knowledge about the directors provides a clue. David Gordon Green is known for making horror movies. Horror movies do not require a high budget because they do not generally require elaborate special effects and since much happens at night or atleast in the dark, the movie sets do not need to be as detailed or intricate. The likes of Steven Spielberg and Ridley Scott are known for Sci Fi movies which definitely require elaborate special effects and the creation of movie sets reflecting the otherworldly environments that are central to the movies. ALmost by definition a Sci Fi

has to require a higher budget.
This project did not focus on genre but this data is implying that profitability by genre should be separately investigated.

Sort of Directors by Profitability

Take the previously sorted director_means df and sort it by 'profit percent'. The chart will show most profitable directors regardless of number of movies made.

In [22]:

```
director_sort_by_profit_df = director_means.sort_values(by=['profit perc
director_sort_by_profit_df.head(20)
```

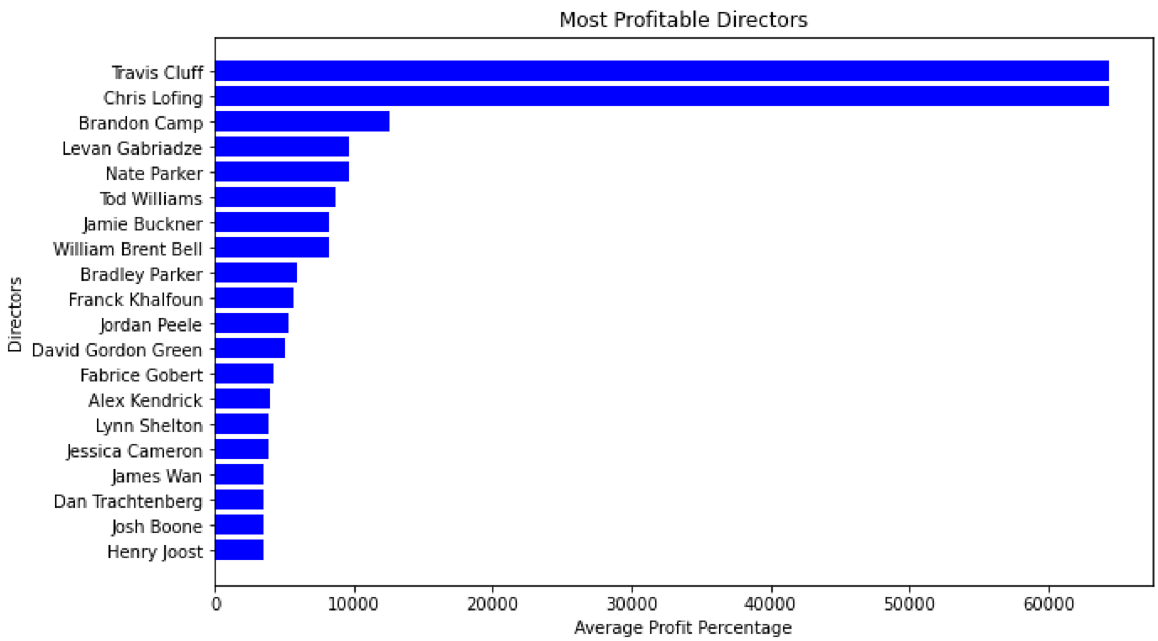
Out[22]:

Director	Avg Rating	Number of Votes	float_production_budget	float_domestic_gross	float_world
Travis Cluff	4.200000	17763.000	100000.0	2.276441e+07	4.
Chris Lofing	4.200000	17763.000	100000.0	2.276441e+07	4.
Brandon Camp	6.400000	2779.000	500000.0	3.155956e+07	3.
Levan Gabriadze	5.600000	62043.000	1000000.0	3.278964e+07	6.
Nate Parker	6.400000	18442.000	5055000.0	1.293078e+07	1.
Tod Williams	5.700000	93122.000	3000000.0	8.475291e+07	1.
Jamie Buckner	2.900000	557.000	5000000.0	1.381416e+08	2.
William Brent Bell	5.100000	51239.500	5500000.0	4.454125e+07	8.
Bradley Parker	5.000000	60304.000	1000000.0	1.811964e+07	4.
Franck Khalfoun	6.100000	32534.000	350000.0	1.000000e+07	1.
Jordan Peele	7.400000	251492.500	12500000.0	1.755238e+08	2.
David Gordon Green	6.312500	63319.125	17290625.0	4.018337e+07	5.
Fabrice Gobert	6.400000	971.000	5000000.0	6.726884e+07	1.
Alex Kendrick	6.750000	14651.000	2500000.0	5.115617e+07	5.
Lynn Shelton	6.700000	24780.000	120000.0	1.597486e+06	3.
Jessica Cameron	4.800000	554.000	3500000.0	4.141102e+07	9.
James Wan	7.175000	312458.000	92875000.0	2.198695e+08	7.
Dan Trachtenberg	7.200000	260383.000	5000000.0	7.208300e+07	1.
Josh Boone	7.700000	315135.000	12000000.0	1.248724e+08	3.
Henry Joost	5.633333	82293.000	10000000.0	6.550426e+07	1.

In [23]:

```
fig, ax = plt.subplots(figsize=(10,6))

ax.barh(director_sort_by_profit_df.index[0:20], director_sort_by_profit_(
ax.invert_yaxis()
ax.set_title("Most Profitable Directors")
ax.set_xlabel('Average Profit Percentage')
ax.set_ylabel('Directors')
plt.savefig('./Images/Most_Profitable_Directors.png')
```

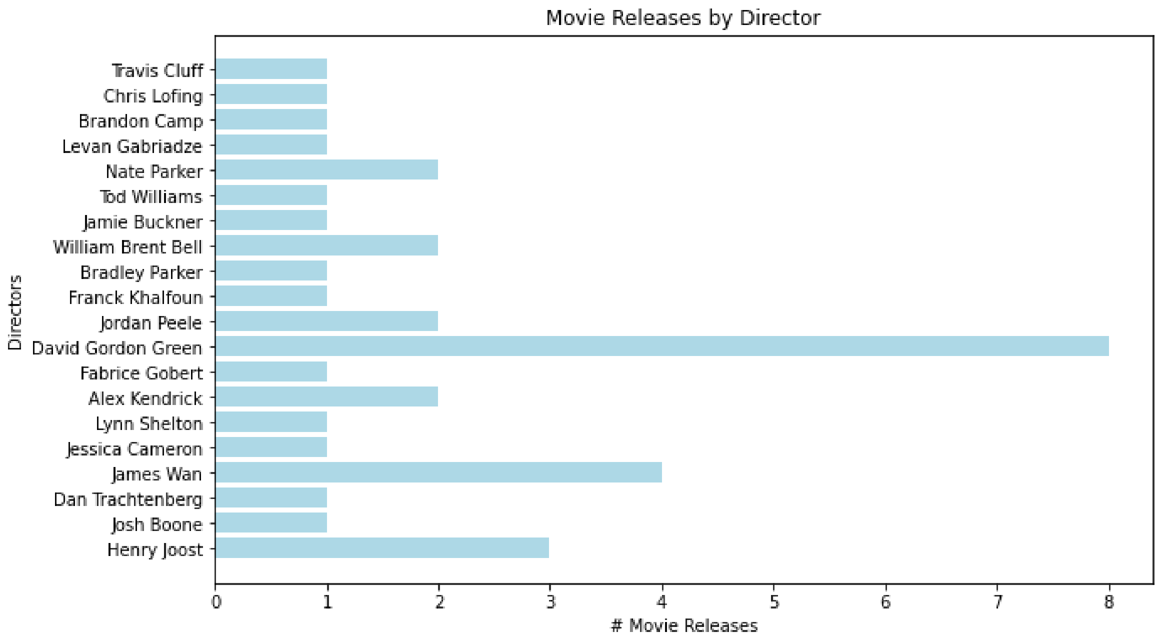


Just looking at profitability without sorting by movie counts, there are many directors that are showing profitability in the thousands of percent, some well over 10,000 percent. However, the number of movies that they have made is limited. Let's also generate the movie count for these most profitable directors. See the next chart below.

In [24]:

```
fig, ax = plt.subplots(figsize=(10,6))

ax.barh(director_sort_by_profit_df.index[0:20], director_sort_by_profit_(
ax.invert_yaxis()
ax.set_title("Movie Releases by Director")
ax.set_xlabel('# Movie Releases')
ax.set_ylabel('Directors')
plt.savefig('./Images/Count_of_Movie_Releases_by_Most_Profitable_Director
```



So, many of these highly profitable directors have only made one movie. Perhaps, there was some political, societal or environmental anomaly that brought focus on their single movie and it generated a huge percentage of profitability. A single point of data does not indicate a trend. They may continue to make profitable movies or they may be relegated in history as One-Hit Wonders.

Movie Length vs Profitability

Sort of original merged_df by Movie Length

This analysis was initiated with the question as to whether other factors like the length of the movie may correlate with profit or lack thereof. This following sorted data set will be used to look at the effect of movie length on profitability.

In [25]:
```
1 merged_sort_by_Length = merged_df.sort_values(by=['Length'], ascending=Fa
2 merged_sort_by_Length
```

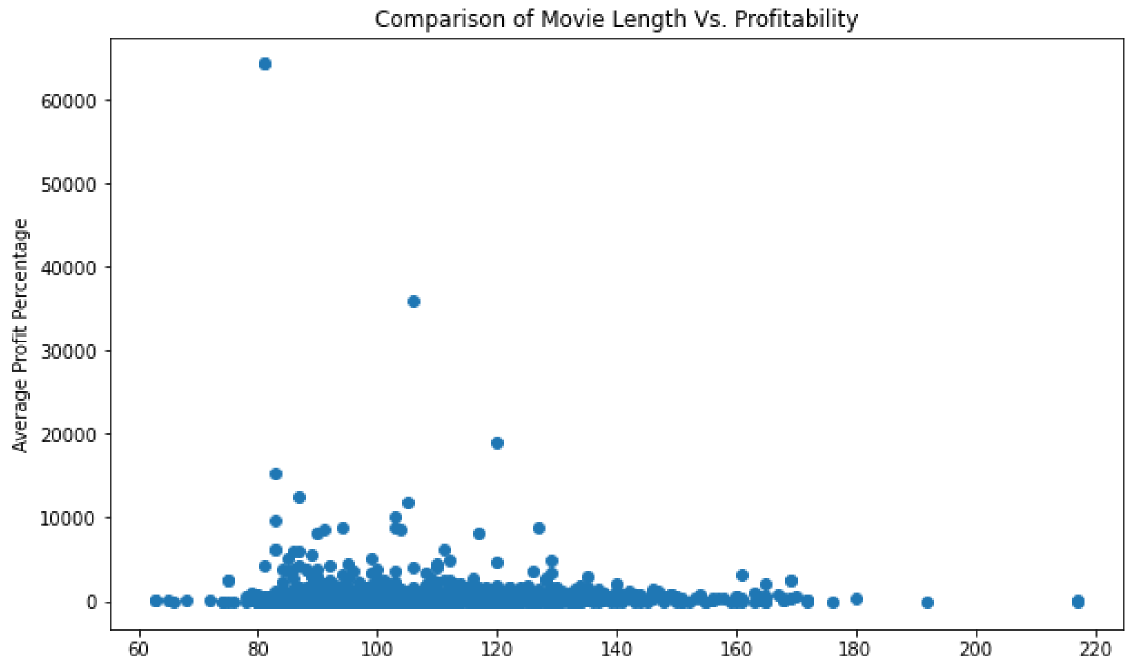
Out[25]:

	Movie Name	Release Year	Length	Genres	Director	Avg Rating	Number of Votes
5	Hamlet	2015	217.0	Drama	Robin Lough	8.6	1587
6	Hamlet	2015	217.0	Drama	Robin Lough	8.6	1587
3	Coriolanus	2014	192.0	Drama,History,War	Tim Van Someren	8.7	1347
30	The Wolf of Wall Street	2013	180.0	Biography,Crime,Drama	Martin Scorsese	8.2	1035358
762	Jab Tak Hai Jaan	2012	176.0	Drama,Romance	Yash Chopra	6.8	48364
...	...	...	...	...	...	...	...
1802	Aroused	2013	66.0	Documentary	Deborah Anderson	5.3	596
714	Unstoppable	2013	65.0	Documentary	Darren Doane	4.3	551
414	Winnie the Pooh	2011	63.0	Adventure,Animation,Comedy	Stephen J. Anderson	7.2	19605
415	Winnie the Pooh	2011	63.0	Adventure,Animation,Comedy	Don Hall	7.2	19605
1889	The Bachelor	2016	NaN	Comedy,Romance	Antonis Sotiropoulos	5.1	895

2112 rows × 19 columns

In [26]:
```
fig, ax = plt.subplots(figsize=(10,6))

ax.scatter(x=merged_sort_by_Length['Length'],y=merged_sort_by_Length['pro
ax.set_title("Comparison of Movie Length Vs. Profitability")
ax.set_xlabel('Movie Length (Mins)')
ax.set_ylabel('Average Profit Percentage')
plt.savefig('./Images/Movie_Length_Vs_Profitability.png',bbox_inches='tig
```



The top few high profit movies (outliers) are skewing the results. Let's only look at movies under 10K% profit by not including the first 10 movies in the sorted list.

In [27]:

```
1 merged_sort_by_Profit_Percent_under_10k = merged_df.sort_values(by=['pro
2 merged_sort_by_Profit_Percent_under_10k.head(20)
```

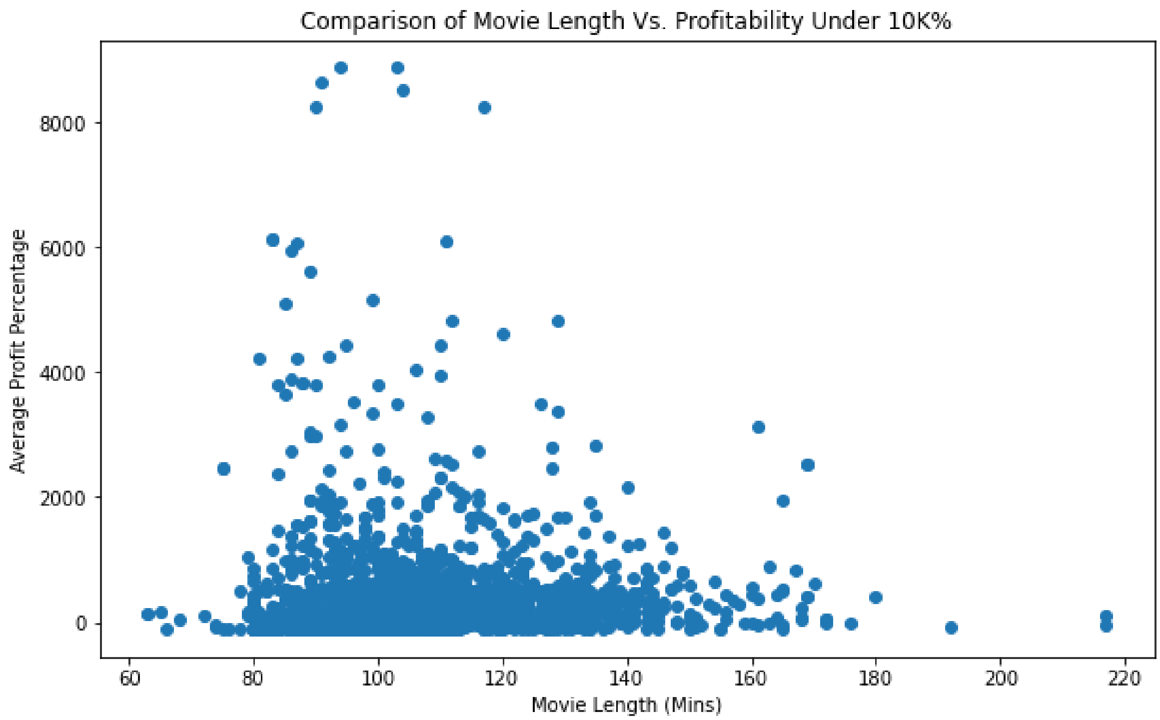
Out[27]:

	Movie Name	Release Year	Length	Genres	Director	Avg Rating	Number of Votes	U
252	Home	2016	103.0	Drama	Fien Troch	7.2	811	
255	Home	2015	94.0	Adventure,Animation,Comedy	Tim Johnson	6.6	85831	
1593	Paranormal Activity 2	2010	91.0	Horror	Tod Williams	5.7	93122	
152	Get Out	2017	104.0	Horror,Mystery,Thriller	Jordan Peele	7.7	400474	
381	Split	2016	90.0	Comedy,Romance,Sport	Jamie Buckner	2.9	557	
380	Split	2016	117.0	Horror,Thriller	M. Night Shyamalan	7.3	358543	
1525	Paranormal Activity 3	2011	83.0	Horror,Mystery,Thriller	Henry Joost	5.8	85689	
1526	Paranormal Activity 3	2011	83.0	Horror,Mystery,Thriller	Ariel Schulman	5.8	85689	
268	Moonlight	2016	111.0	Drama	Barry Jenkins	7.4	227964	
1662	The Last Exorcism	2010	87.0	Drama,Horror,Thriller	Daniel Stamm	5.6	45815	
1892	Chernobyl Diaries	2012	86.0	Horror,Mystery,Thriller	Bradley Parker	5.0	60304	
1354	Maniac	2012	89.0	Horror,Thriller	Franck Khalfoun	6.1	32534	
1765	Annabelle	2014	99.0	Horror,Mystery,Thriller	John R. Leonetti	5.4	122039	
1586	The Purge	2013	85.0	Horror,Thriller	James DeMonaco	5.7	183549	
436	Beauty and the Beast	2014	112.0	Drama,Fantasy,Romance	Christophe Gans	6.4	18100	
434	Beauty and the Beast	2017	129.0	Family,Fantasy,Musical	Bill Condon	7.2	238325	
974	War Room	2015	120.0	Drama	Alex Kendrick	6.5	11716	
904	You're Next	2011	95.0	Action,Comedy,Horror	Adam Wingard	6.6	79451	
750	Sinister	2012	110.0	Horror,Mystery,Thriller	Scott Derrickson	6.8	198345	
729	A Ghost Story	2017	92.0	Drama,Fantasy,Romance	David Lowery	6.8	46280	

In [28]:

```
fig, ax = plt.subplots(figsize=(10,6))

ax.scatter(x=merged_sort_by_Profit_Percent_under_10k['Length'],y=merged_s
ax.set_title("Comparison of Movie Length Vs. Profitability Under 10K%")
ax.set_xlabel('Movie Length (Mins)')
ax.set_ylabel('Average Profit Percentage')
plt.savefig('./Images/Movie_Length_Vs_Profitability_Under_10K.png',bbox_i
```



The multithousand percent profit movies are still skewing the results. Let's only look at movies under 2K% profit by not including the first 81 movies in the sorted list. This will get a better view of the remaining 2000+ movies.

In [29]:

```
merged_sort_by_Profit_Percent_under_2k = merged_df.sort_values(by=['prof
merged_sort_by_Profit_Percent_under_2k.head()
```

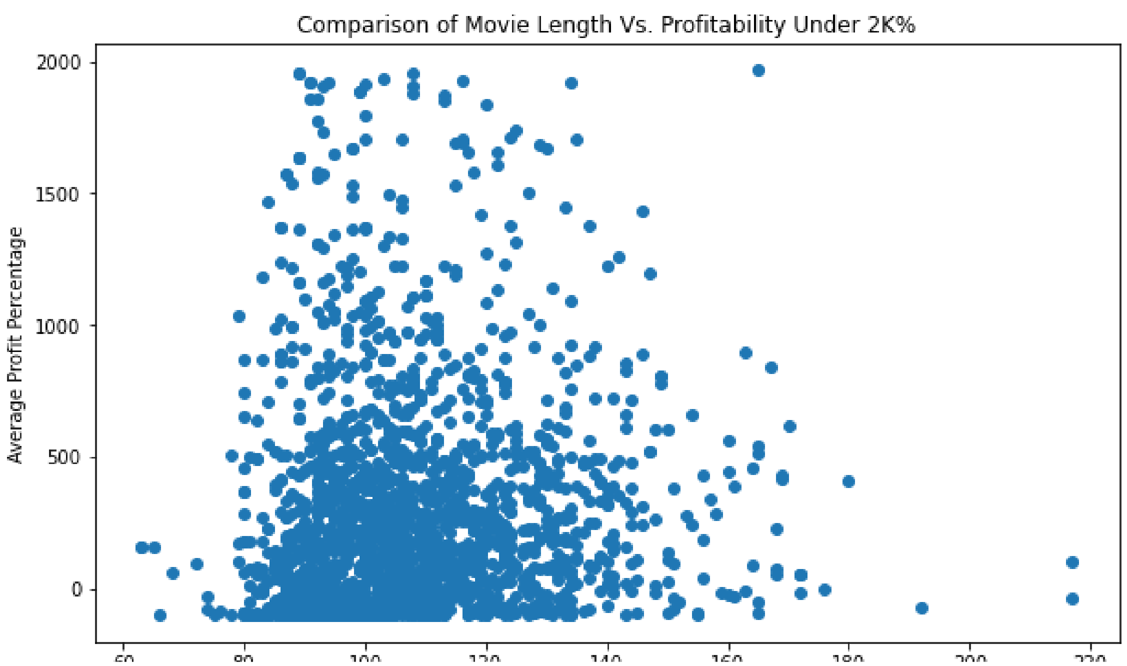
Out[29]:

	Movie Name	Release Year	Length	Genres	Director	Avg Rating	Number of Votes	Unnamed: 0
95	Boyhood	2014	165.0	Drama	Richard Linklater	7.9	315584	4484
263	Kevin Hart: Laugh at My Pain	2011	89.0	Comedy,Documentary	Tim Story	7.4	5081	5374
262	Kevin Hart: Laugh at My Pain	2011	89.0	Comedy,Documentary	Leslie Small	7.4	5081	5374
539	The Gift	2015	108.0	Drama,Mystery,Thriller	Joel Edgerton	7.1	123834	4262
1021	The Purge: Anarchy	2014	103.0	Action,Horror,Sci-Fi	James DeMonaco	6.5	126203	3770

In [30]:

```
fig, ax = plt.subplots(figsize=(10,6))

ax.scatter(x=merged_sort_by_Profit_Percent_under_2k['Length'],y=merged_s
ax.set_title("Comparison of Movie Length Vs. Profitability Under 2K%")
ax.set_xlabel('Movie Length (Mins)')
ax.set_ylabel('Average Profit Percentage')
plt.savefig('./Images/Movie_Length_Vs_Profitability_Under_2K.png',bbox_i
```



Looking at the vast majority of the movies in our original data set, we do not see a strong correlation. There is some loose indication that most movies are under 2 to 2.5 hours. High profits can be generated in a movie of only 1.5 hours. So if the story does not require a long movie there is no need to stretch the movie into that longer timeframe.

Recommendations

We offer the following 3 recommendations:
****- Work with Directors with track records of consistent profit.**
- Give NEW directors a smaller budget to prove themselves.
- Need to investigate further the effect of Genre on budgets and therefore profitability.

Other Observations

Microsoft has:
- Diverse product lines.
- Sells internationally.
- Has a large international employee base.

Movie decisions may be driven by:
- Product placement considerations.
- Overseas markets.

US Market prefers shorter movies. Need to study other markets.

Next Steps

Investigate non-US markets.
Identify opportunities for product placement.
Formulate marketing strategies to drive product placement.
Engage movie industry drivers (producers, directors, writers, etc.) to consider how to move Microsoft interests further.

In [ ]:

```
1
```