Investigate [The Movie Database (TMDb)] to find out the best investment options

"Data Driven Business Study"

By Data Scientist : Khaled Saad
Date : 23-01-2023

Table of Contents

- Introduction

- Data Wrangling
- Exploratory Data Analysis
- Conclusions
- Refrences

Introduction

> I am considering myself as an investor how want to invest his money in the filmmaking industry and now i want to make a data-driven decision to select which is the best genre of movies to invest in and who is the best movie director i should invest with him So i decided to invistagate " The Movie Database (TMDb) " which is avialble on Kaggle

Data Wrangling

> Now I will do the following steps:
>
> 1. load in the data
> 2. Exploe the data to decide if it needs cleaning or it needs some modifications
> 3. Do the required cleanliness
> 4. Do the required modifications (adding or droping raws and columns)

Step No. (0) : importing the required packages to do the job

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

#this magic ward is essntial to plot in-line
%matplotlib inline
```

Step No. (1) : loading the data

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

Step No. (2) : what is our data looks like ?

```
In [3]: print("\n",'*'*100 ,"\n")
        print("The size of the data looks like following: \n \n" , my_raw_data.size)
        print("\n",'*'*100 ,"\n")
        print("The shape of the data looks like following: \n \n" , my_raw_data.shape)
        print("\n",'*'*100 ,"\n")
        print("The info of the data looks like following: \n \n", my_raw_data.info())
        print("\n",'*'*100 ,"\n")
        my_raw_data
```

```
 ****************************************************************************************************

The size of the data looks like following:

 228186

 ****************************************************************************************************

The shape of the data looks like following:

 (10866, 21)

 ****************************************************************************************************

<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
The info of the data looks like following:

 None
```

****

Out[3]:

	id	imdb_id	popularity	budget	revenue	original_title	cast	homepage	directo
0	135397	tt0369610	32.985763	150000000	1513528810	Jurassic World	Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi...	http://www.jurassicworld.com/	Coli Trevorrow
1	76341	tt1392190	28.419936	150000000	378436354	Mad Max: Fury Road	Tom Hardy|Charlize Theron|Hugh Keays-Byrne|Nic...	http://www.madmaxmovie.com/	George Mille
2	262500	tt2908446	13.112507	110000000	295238201	Insurgent	Shailene Woodley|Theo James|Kate Winslet|Ansel...	http://www.thedivergentseries.movie/#insurgent	Rober Schwentke
3	140607	tt2488496	11.173104	200000000	2068178225	Star Wars: The Force Awakens	Harrison Ford|Mark Hamill|Carrie Fisher|Adam D...	http://www.starwars.com/films/star-wars-episod...	J.J. Abram
4	168259	tt2820852	9.335014	190000000	1506249360	Furious 7	Vin Diesel|Paul Walker|Jason Statham|Michelle ...	http://www.furious7.com/	James War
...	...	...	...	...	...	...	...	...	..
10861	21	tt0060371	0.080598	0	0	The Endless Summer	Michael Hynson|Robert August|Lord 'Tally Ho' B...	NaN	Bruce Brow
10862	20379	tt0060472	0.065543	0	0	Grand Prix	James Garner|Eva Marie Saint|Yves Montand|Tosh...	NaN	Joh Frankenheime

	id	imdb_id	popularity	budget	revenue	original_title	cast	homepage	directo
10863	39768	tt0060161	0.065141	0	0	Beregis Avtomobilya	Innokentiy Smoktunovskiy|Oleg Efremov|Georgi Z...	NaN	Elda Ryazano
10864	21449	tt0061177	0.064317	0	0	What's Up, Tiger Lily?	Tatsuya Mihashi|Akiko Wakabayashi|Mie Hama|Joh...	NaN	Woody Alle
10865	22293	tt0060666	0.035919	19000	0	Manos: The Hands of Fate	Harold P. Warren|Tom Neyman|John Reynolds|Dian...	NaN	Harold P Warre

In [4]:
```
print("\n",'*'*100 ,"\n")
print("The first rows of the data looks like following: \n \n")
print("\n",'*'*100 ,"\n")
print(my_raw_data.head())
print("\n",'*'*100 ,"\n")
print("\n",'*'*100 ,"\n")
print("The last rows of the data looks like following: \n \n")
print("\n",'*'*100 ,"\n")
print(my_raw_data.tail())
print("\n",'*'*100 ,"\n")

x = 5

while True:
        explr = input("If you would like to explore more data type 'yes' and press enter or just press enter to skip ")
        if explr.lower() == "yes":
                x += 5
                print("-"*100)
                print("\n As per your selection you are watching the rows from row no. ", x-4,"to row no.",x," \n \n \n ", my_raw_
                print("-"*100)
        else:
            break
```

```
 *****************************************************************************************************

The first rows of the data looks like following:



 *****************************************************************************************************

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

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

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

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

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

                                            overview runtime  \
0  Twenty-two years after the events of Jurassic ...     124   
1  An apocalyptic story set in the furthest reach...     120   
2  Beatrice Prior must confront her inner demons ...     119   
```

```
3  Thirty years after defeating the Galactic Empi...     136
4  Deckard Shaw seeks revenge against Dominic Tor...     137

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

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

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

[5 rows x 21 columns]

 ****************************************************************************************************


 ****************************************************************************************************

The last rows of the data looks like following:



 ****************************************************************************************************

          id    imdb_id  popularity  budget  revenue  \
10861     21  tt0060371    0.080598       0        0
10862  20379  tt0060472    0.065543       0        0
10863  39768  tt0060161    0.065141       0        0
10864  21449  tt0061177    0.064317       0        0
10865  22293  tt0060666    0.035919   19000        0

                 original_title  \
10861        The Endless Summer
10862                Grand Prix
```

```
10863      Beregis Avtomobilya
10864    What's Up, Tiger Lily?
10865  Manos: The Hands of Fate

                                                    cast homepage  \
10861  Michael Hynson|Robert August|Lord 'Tally Ho' B...      NaN
10862  James Garner|Eva Marie Saint|Yves Montand|Tosh...      NaN
10863  Innokentiy Smoktunovskiy|Oleg Efremov|Georgi Z...      NaN
10864  Tatsuya Mihashi|Akiko Wakabayashi|Mie Hama|Joh...      NaN
10865  Harold P. Warren|Tom Neyman|John Reynolds|Dian...      NaN

                director                                            tagline  \
10861        Bruce Brown                                                NaN
10862  John Frankenheimer  Cinerama sweeps YOU into a drama of speed and ...
10863     Eldar Ryazanov                                                NaN
10864        Woody Allen                            WOODY ALLEN STRIKES BACK!
10865   Harold P. Warren       It's Shocking! It's Beyond Your Imagination!

       ...                                          overview runtime  \
10861  ...  The Endless Summer, by Bruce Brown, is one of ...      95
10862  ...  Grand Prix driver Pete Aron is fired by his te...     176
10863  ...  An insurance agent who moonlights as a carthie...      94
10864  ...  In comic Woody Allen's film debut, he took the...      80
10865  ...  A family gets lost on the road and stumbles up...      74

                         genres  \
10861               Documentary
10862  Action|Adventure|Drama
10863            Mystery|Comedy
10864             Action|Comedy
10865                    Horror

                                  production_companies release_date  \
10861                                    Bruce Brown Films      6/15/66
10862  Cherokee Productions|Joel Productions|Douglas ...     12/21/66
10863                                              Mosfilm       1/1/66
10864                              Benedict Pictures Corp.      11/2/66
10865                                            Norm-Iris     11/15/66

      vote_count  vote_average  release_year     budget_adj  revenue_adj
10861         11           7.4          1966       0.000000          0.0
10862         20           5.7          1966       0.000000          0.0
10863         11           6.5          1966       0.000000          0.0
10864         22           5.4          1966       0.000000          0.0
10865         15           1.5          1966  127642.279154          0.0

[5 rows x 21 columns]
```

```
 ****************************************************************************************************

If you would like to explore more data type 'yes' and press enter or just press enter to skip
```

Step (3) : Tray to simplify the data by droping the un-useful data

```
In [5]: my_data_one = my_raw_data.filter(['release_year','genres', 'original_title', 'director', 'popularity', 'budget', 'revenue'], axis=
```

Step No. (4) : let us see our data after simplification

```
In [6]: print("\n",'*'*100 ,"\n")
        print("The size of the data looks like following: \n \n" , my_data_one.size)
        print("\n",'*'*100 ,"\n")
        print("The shape of the data looks like following: \n \n" , my_data_one.shape)
        print("\n",'*'*100 ,"\n")
        print("The info of the data looks like following: \n \n", my_data_one.info())
        print("\n",'*'*100 ,"\n")
        my_data_one
```

```
 ****************************************************************************************************

The size of the data looks like following:

 76062

 ****************************************************************************************************

The shape of the data looks like following:

 (10866, 7)

 ****************************************************************************************************

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10866 entries, 0 to 10865
Data columns (total 7 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   release_year    10866 non-null  int64  
 1   genres          10843 non-null  object 
 2   original_title  10866 non-null  object 
 3   director        10822 non-null  object 
 4   popularity      10866 non-null  float64
 5   budget          10866 non-null  int64  
 6   revenue         10866 non-null  int64  
dtypes: float64(1), int64(3), object(3)
memory usage: 594.4+ KB
The info of the data looks like following:

 None

 ****************************************************************************************************
```

Out[6]:

	release_year	genres	original_title	director	popularity	budget	revenue
0	2015	Action|Adventure|Science Fiction|Thriller	Jurassic World	Colin Trevorrow	32.985763	150000000	1513528810
1	2015	Action|Adventure|Science Fiction|Thriller	Mad Max: Fury Road	George Miller	28.419936	150000000	378436354
2	2015	Adventure|Science Fiction|Thriller	Insurgent	Robert Schwentke	13.112507	110000000	295238201
3	2015	Action|Adventure|Science Fiction|Fantasy	Star Wars: The Force Awakens	J.J. Abrams	11.173104	200000000	2068178225
4	2015	Action|Crime|Thriller	Furious 7	James Wan	9.335014	190000000	1506249360
...	...	...	...	...	...	...	...
10861	1966	Documentary	The Endless Summer	Bruce Brown	0.080598	0	0
10862	1966	Action|Adventure|Drama	Grand Prix	John Frankenheimer	0.065543	0	0
10863	1966	Mystery|Comedy	Beregis Avtomobilya	Eldar Ryazanov	0.065141	0	0
10864	1966	Action|Comedy	What's Up, Tiger Lily?	Woody Allen	0.064317	0	0
10865	1966	Horror	Manos: The Hands of Fate	Harold P. Warren	0.035919	19000	0

10866 rows × 7 columns

Step (5) : let us simplify the work by adding new valuable column "The Profit" from the budget and revenue

In [7]:
```
my_data_one['profit'] = my_data_one['revenue'] - my_data_one['budget']
```

Step (6) : and again let us tke a look on our data

In [8]:
```
print("\n",'*'*100 ,"\n")
print("The size of the data looks like following: \n \n" , my_data_one.size)
print("\n",'*'*100 ,"\n")
print("The shape of the data looks like following: \n \n" , my_data_one.shape)
print("\n",'*'*100 ,"\n")
print("The info of the data looks like following: \n \n", my_data_one.info())
print("\n",'*'*100 ,"\n")
my_data_one
```

```
 ****************************************************************************************************

The size of the data looks like following:

 86928

 ****************************************************************************************************

The shape of the data looks like following:

 (10866, 8)

 ****************************************************************************************************

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10866 entries, 0 to 10865
Data columns (total 8 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   release_year    10866 non-null  int64  
 1   genres          10843 non-null  object 
 2   original_title  10866 non-null  object 
 3   director        10822 non-null  object 
 4   popularity      10866 non-null  float64
 5   budget          10866 non-null  int64  
 6   revenue         10866 non-null  int64  
 7   profit          10866 non-null  int64  
dtypes: float64(1), int64(4), object(3)
memory usage: 679.2+ KB
The info of the data looks like following:

 None

 ****************************************************************************************************
```

Out[8]:

	release_year	genres	original_title	director	popularity	budget	revenue	profit
0	2015	Action|Adventure|Science Fiction|Thriller	Jurassic World	Colin Trevorrow	32.985763	150000000	1513528810	1363528810
1	2015	Action|Adventure|Science Fiction|Thriller	Mad Max: Fury Road	George Miller	28.419936	150000000	378436354	228436354
2	2015	Adventure|Science Fiction|Thriller	Insurgent	Robert Schwentke	13.112507	110000000	295238201	185238201
3	2015	Action|Adventure|Science Fiction|Fantasy	Star Wars: The Force Awakens	J.J. Abrams	11.173104	200000000	2068178225	1868178225
4	2015	Action|Crime|Thriller	Furious 7	James Wan	9.335014	190000000	1506249360	1316249360
...	...	...	...	...	...	...	...	...
10861	1966	Documentary	The Endless Summer	Bruce Brown	0.080598	0	0	0
10862	1966	Action|Adventure|Drama	Grand Prix	John Frankenheimer	0.065543	0	0	0
10863	1966	Mystery|Comedy	Beregis Avtomobilya	Eldar Ryazanov	0.065141	0	0	0
10864	1966	Action|Comedy	What's Up, Tiger Lily?	Woody Allen	0.064317	0	0	0
10865	1966	Horror	Manos: The Hands of Fate	Harold P. Warren	0.035919	19000	0	-19000

10866 rows × 8 columns

In [9]:

```
print("\n",'*'*100 ,"\n")
print("The first rows of the data looks like following: \n \n")
print("\n",'*'*100 ,"\n")
print(my_data_one.head())
print("\n",'*'*100 ,"\n")
print("\n",'*'*100 ,"\n")
print("The last rows of the data looks like following: \n \n")
print("\n",'*'*100 ,"\n")
print(my_data_one.tail())
print("\n",'*'*100 ,"\n")

x = 5

while True:
        explr = input("If you would like to explore more data type 'yes' and press enter or just press enter to skip ")
        if explr.lower() == "yes":
                x += 5
                print("-"*100)
                print("\n As per your selection you are watching the rows from row no. ", x-4,"to row no.",x," \n \n \n ", my_data
                print("-"*100)
        else:
            break
```

```
 ****************************************************************************************************

The first rows of the data looks like following:



 ****************************************************************************************************

   release_year                                     genres  \
0          2015  Action|Adventure|Science Fiction|Thriller
1          2015  Action|Adventure|Science Fiction|Thriller
2          2015         Adventure|Science Fiction|Thriller
3          2015   Action|Adventure|Science Fiction|Fantasy
4          2015                      Action|Crime|Thriller

                original_title          director  popularity     budget  \
0               Jurassic World   Colin Trevorrow   32.985763  150000000
1           Mad Max: Fury Road     George Miller   28.419936  150000000
2                    Insurgent  Robert Schwentke   13.112507  110000000
3  Star Wars: The Force Awakens       J.J. Abrams   11.173104  200000000
4                    Furious 7         James Wan    9.335014  190000000

      revenue      profit
0  1513528810  1363528810
1   378436354   228436354
2   295238201   185238201
3  2068178225  1868178225
4  1506249360  1316249360

 ****************************************************************************************************


 ****************************************************************************************************

The last rows of the data looks like following:



 ****************************************************************************************************

       release_year                  genres           original_title  \
10861          1966             Documentary        The Endless Summer
10862          1966  Action|Adventure|Drama                Grand Prix
10863          1966          Mystery|Comedy       Beregis Avtomobilya
10864          1966           Action|Comedy    What's Up, Tiger Lily?
10865          1966                  Horror  Manos: The Hands of Fate
```

```
                director  popularity  budget  revenue  profit
10861        Bruce Brown    0.080598       0        0       0
10862  John Frankenheimer    0.065543       0        0       0
10863      Eldar Ryazanov    0.065141       0        0       0
10864         Woody Allen    0.064317       0        0       0
10865    Harold P. Warren    0.035919   19000        0  -19000

 ****************************************************************************************************

If you would like to explore more data type 'yes' and press enter or just press enter to skip
```

Step No. (7) : now let us make a copy of our simplified data to start clean if from the duplication and empty values

In [10]:
```python
my_data_two = my_data_one.copy()
```

In [11]:
```python
my_data_two
```

Out[11]:

	release_year	genres	original_title	director	popularity	budget	revenue	profit
0	2015	Action|Adventure|Science Fiction|Thriller	Jurassic World	Colin Trevorrow	32.985763	150000000	1513528810	1363528810
1	2015	Action|Adventure|Science Fiction|Thriller	Mad Max: Fury Road	George Miller	28.419936	150000000	378436354	228436354
2	2015	Adventure|Science Fiction|Thriller	Insurgent	Robert Schwentke	13.112507	110000000	295238201	185238201
3	2015	Action|Adventure|Science Fiction|Fantasy	Star Wars: The Force Awakens	J.J. Abrams	11.173104	200000000	2068178225	1868178225
4	2015	Action|Crime|Thriller	Furious 7	James Wan	9.335014	190000000	1506249360	1316249360
...	...	...	...	...	...	...	...	...
10861	1966	Documentary	The Endless Summer	Bruce Brown	0.080598	0	0	0
10862	1966	Action|Adventure|Drama	Grand Prix	John Frankenheimer	0.065543	0	0	0
10863	1966	Mystery|Comedy	Beregis Avtomobilya	Eldar Ryazanov	0.065141	0	0	0
10864	1966	Action|Comedy	What's Up, Tiger Lily?	Woody Allen	0.064317	0	0	0
10865	1966	Horror	Manos: The Hands of Fate	Harold P. Warren	0.035919	19000	0	-19000

10866 rows × 8 columns

Step No. (8) : There are many budget and revenue data are missed which will lead to inacurate profit data - which is a very vital information for our last decision -, so now we will remove any row with missed budget or revenue data

```
In [12]: my_data_two.drop(my_data_two[my_data_two['budget'] == 0].index , inplace = True)
```

```
In [13]: my_data_two
```

Out[13]:

	release_year	genres	original_title	director	popularity	budget	revenue	profit
0	2015	Action|Adventure|Science Fiction|Thriller	Jurassic World	Colin Trevorrow	32.985763	150000000	1513528810	1363528810
1	2015	Action|Adventure|Science Fiction|Thriller	Mad Max: Fury Road	George Miller	28.419936	150000000	378436354	228436354
2	2015	Adventure|Science Fiction|Thriller	Insurgent	Robert Schwentke	13.112507	110000000	295238201	185238201
3	2015	Action|Adventure|Science Fiction|Fantasy	Star Wars: The Force Awakens	J.J. Abrams	11.173104	200000000	2068178225	1868178225
4	2015	Action|Crime|Thriller	Furious 7	James Wan	9.335014	190000000	1506249360	1316249360
...	...	...	...	...	...	...	...	...
10835	1966	Action|Adventure|Drama|War|Romance	The Sand Pebbles	Robert Wise	0.299911	12000000	20000000	8000000
10841	1966	Western	The Shooting	Monte Hellman	0.264925	75000	0	-75000
10848	1966	Adventure|Science Fiction	Fantastic Voyage	Richard Fleischer	0.207257	5115000	12000000	6885000
10855	1966	Comedy|Family|Mystery|Romance	The Ghost & Mr. Chicken	Alan Rafkin	0.141026	700000	0	-700000
10865	1966	Horror	Manos: The Hands of Fate	Harold P. Warren	0.035919	19000	0	-19000

5170 rows × 8 columns

```
In [14]: my_data_two.drop(my_data_two[my_data_two['revenue'] == 0].index , inplace = True)
```

```
In [15]: my_data_two
```

Out[15]:

	release_year	genres	original_title	director	popularity	budget	revenue	profit
0	2015	Action|Adventure|Science Fiction|Thriller	Jurassic World	Colin Trevorrow	32.985763	150000000	1513528810	1363528810
1	2015	Action|Adventure|Science Fiction|Thriller	Mad Max: Fury Road	George Miller	28.419936	150000000	378436354	228436354
2	2015	Adventure|Science Fiction|Thriller	Insurgent	Robert Schwentke	13.112507	110000000	295238201	185238201
3	2015	Action|Adventure|Science Fiction|Fantasy	Star Wars: The Force Awakens	J.J. Abrams	11.173104	200000000	2068178225	1868178225
4	2015	Action|Crime|Thriller	Furious 7	James Wan	9.335014	190000000	1506249360	1316249360
...	...	...	...	...	...	...	...	...
10822	1966	Drama	Who's Afraid of Virginia Woolf?	Mike Nichols	0.670274	7500000	33736689	26236689
10828	1966	Mystery|Thriller	Torn Curtain	Alfred Hitchcock	0.402730	3000000	13000000	10000000
10829	1966	Action|Western	El Dorado	Howard Hawks	0.395668	4653000	6000000	1347000
10835	1966	Action|Adventure|Drama|War|Romance	The Sand Pebbles	Robert Wise	0.299911	12000000	20000000	8000000
10848	1966	Adventure|Science Fiction	Fantastic Voyage	Richard Fleischer	0.207257	5115000	12000000	6885000

3855 rows × 8 columns

Step No. (9) : remove any other missed values

In [16]:
```
my_data_two.dropna(inplace = True)
```

In [17]:
```
my_data_two
```

Out[17]:

	release_year	genres	original_title	director	popularity	budget	revenue	profit
0	2015	Action|Adventure|Science Fiction|Thriller	Jurassic World	Colin Trevorrow	32.985763	150000000	1513528810	1363528810
1	2015	Action|Adventure|Science Fiction|Thriller	Mad Max: Fury Road	George Miller	28.419936	150000000	378436354	228436354
2	2015	Adventure|Science Fiction|Thriller	Insurgent	Robert Schwentke	13.112507	110000000	295238201	185238201
3	2015	Action|Adventure|Science Fiction|Fantasy	Star Wars: The Force Awakens	J.J. Abrams	11.173104	200000000	2068178225	1868178225
4	2015	Action|Crime|Thriller	Furious 7	James Wan	9.335014	190000000	1506249360	1316249360
...	...	...	...	...	...	...	...	...
10822	1966	Drama	Who's Afraid of Virginia Woolf?	Mike Nichols	0.670274	7500000	33736689	26236689
10828	1966	Mystery|Thriller	Torn Curtain	Alfred Hitchcock	0.402730	3000000	13000000	10000000
10829	1966	Action|Western	El Dorado	Howard Hawks	0.395668	4653000	6000000	1347000
10835	1966	Action|Adventure|Drama|War|Romance	The Sand Pebbles	Robert Wise	0.299911	12000000	20000000	8000000
10848	1966	Adventure|Science Fiction	Fantastic Voyage	Richard Fleischer	0.207257	5115000	12000000	6885000

3854 rows × 8 columns

Step No. (10) : remove any duplicated values

In [18]:
```
my_data_two.drop_duplicates(inplace = True)
```

In [19]:
```
my_data_two
```

Out[19]:

	release_year	genres	original_title	director	popularity	budget	revenue	profit
0	2015	Action|Adventure|Science Fiction|Thriller	Jurassic World	Colin Trevorrow	32.985763	150000000	1513528810	1363528810
1	2015	Action|Adventure|Science Fiction|Thriller	Mad Max: Fury Road	George Miller	28.419936	150000000	378436354	228436354
2	2015	Adventure|Science Fiction|Thriller	Insurgent	Robert Schwentke	13.112507	110000000	295238201	185238201
3	2015	Action|Adventure|Science Fiction|Fantasy	Star Wars: The Force Awakens	J.J. Abrams	11.173104	200000000	2068178225	1868178225
4	2015	Action|Crime|Thriller	Furious 7	James Wan	9.335014	190000000	1506249360	1316249360
...	...	...	...	...	...	...	...	...
10822	1966	Drama	Who's Afraid of Virginia Woolf?	Mike Nichols	0.670274	7500000	33736689	26236689
10828	1966	Mystery|Thriller	Torn Curtain	Alfred Hitchcock	0.402730	3000000	13000000	10000000
10829	1966	Action|Western	El Dorado	Howard Hawks	0.395668	4653000	6000000	1347000
10835	1966	Action|Adventure|Drama|War|Romance	The Sand Pebbles	Robert Wise	0.299911	12000000	20000000	8000000
10848	1966	Adventure|Science Fiction	Fantastic Voyage	Richard Fleischer	0.207257	5115000	12000000	6885000

3853 rows × 8 columns

Warning: about 65% of the data was not good (either missed or duplicated) and was deleted

Only 35% of the data is valid for the analysis which is a real limitation

---

Step No. (11) : now let take another look on our final data-set

In [20]:

```
print("\n",'*'*100 ,"\n")
print("The size of the data looks like following: \n \n" , my_data_two.size)
print("\n",'*'*100 ,"\n")
print("The shape of the data looks like following: \n \n" , my_data_two.shape)
print("\n",'*'*100 ,"\n")
print("The info of the data looks like following: \n \n", my_data_two.info())
print("\n",'*'*100 ,"\n")
```

```
 ****************************************************************************************************

The size of the data looks like following:

 30824

 ****************************************************************************************************

The shape of the data looks like following:

 (3853, 8)

 ****************************************************************************************************

<class 'pandas.core.frame.DataFrame'>
Int64Index: 3853 entries, 0 to 10848
Data columns (total 8 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   release_year    3853 non-null   int64  
 1   genres          3853 non-null   object 
 2   original_title  3853 non-null   object 
 3   director        3853 non-null   object 
 4   popularity      3853 non-null   float64
 5   budget          3853 non-null   int64  
 6   revenue         3853 non-null   int64  
 7   profit          3853 non-null   int64  
dtypes: float64(1), int64(4), object(3)
memory usage: 270.9+ KB
The info of the data looks like following:

 None

 ****************************************************************************************************
```

In [21]:

```python
print("\n",'*'*100 ,"\n")
print("The first rows of the data looks like following: \n \n")
print("\n",'*'*100 ,"\n")
print(my_data_two.head())
print("\n",'*'*100 ,"\n")
print("\n",'*'*100 ,"\n")
print("The last rows of the data looks like following: \n \n")
print("\n",'*'*100 ,"\n")
print(my_data_two.tail())
print("\n",'*'*100 ,"\n")
```

```
x = 5

while True:
        explr = input("If you would like to explore more data type 'yes' and press enter or just press enter to skip ")
        if explr.lower() == "yes":
                x += 5
                print("-"*100)
                print("\n As per your selection you are watching the rows from row no. ", x-4,"to row no.",x," \n \n \n ", my_data
                print("-"*100)
        else:
            break
```

```
 ****************************************************************************************************

The first rows of the data looks like following:



 ****************************************************************************************************

   release_year                                     genres  \
0          2015  Action|Adventure|Science Fiction|Thriller   
1          2015  Action|Adventure|Science Fiction|Thriller   
2          2015         Adventure|Science Fiction|Thriller   
3          2015   Action|Adventure|Science Fiction|Fantasy   
4          2015                      Action|Crime|Thriller   

                original_title          director  popularity     budget  \
0               Jurassic World   Colin Trevorrow   32.985763  150000000   
1           Mad Max: Fury Road     George Miller   28.419936  150000000   
2                    Insurgent  Robert Schwentke   13.112507  110000000   
3  Star Wars: The Force Awakens       J.J. Abrams   11.173104  200000000   
4                    Furious 7         James Wan    9.335014  190000000   

      revenue      profit  
0  1513528810  1363528810  
1   378436354   228436354  
2   295238201   185238201  
3  2068178225  1868178225  
4  1506249360  1316249360  

 ****************************************************************************************************


 ****************************************************************************************************

The last rows of the data looks like following:



 ****************************************************************************************************

       release_year                                genres  \
10822          1966                                 Drama   
10828          1966                      Mystery|Thriller   
10829          1966                        Action|Western   
10835          1966  Action|Adventure|Drama|War|Romance   
10848          1966           Adventure|Science Fiction   
```

```
                   original_title           director  popularity  \
10822  Who's Afraid of Virginia Woolf?       Mike Nichols    0.670274
10828                     Torn Curtain   Alfred Hitchcock    0.402730
10829                        El Dorado       Howard Hawks    0.395668
10835                  The Sand Pebbles        Robert Wise    0.299911
10848                  Fantastic Voyage  Richard Fleischer    0.207257

         budget   revenue    profit
10822   7500000  33736689  26236689
10828   3000000  13000000  10000000
10829   4653000   6000000   1347000
10835  12000000  20000000   8000000
10848   5115000  12000000   6885000

 ****************************************************************************************************

If you would like to explore more data type 'yes' and press enter or just press enter to skip
```

In [22]:
```python
print("\n",'*'*100 ,"\n")
print("The description of the data looks like following: \n \n")
print("\n",'*'*100 ,"\n")
print(my_data_two.describe())
print("\n",'*'*100 ,"\n")
```

```
 ****************************************************************************************************

The description of the data looks like following:



 ****************************************************************************************************

       release_year   popularity        budget       revenue        profit
count   3853.000000  3853.000000  3.853000e+03  3.853000e+03  3.853000e+03
mean    2001.259278     1.191825  3.721227e+07  1.077117e+08  7.049944e+07
std       11.283517     1.475258  4.221035e+07  1.765554e+08  1.506356e+08
min     1960.000000     0.001117  1.000000e+00  2.000000e+00 -4.139124e+08
25%     1995.000000     0.462609  1.000000e+07  1.360940e+07 -1.324619e+06
50%     2004.000000     0.797723  2.400000e+07  4.480678e+07  2.003320e+07
75%     2010.000000     1.368403  5.000000e+07  1.242721e+08  8.172336e+07
max     2015.000000    32.985763  4.250000e+08  2.781506e+09  2.544506e+09

 ****************************************************************************************************
```

In [23]:
```python
my_data_two.sort_values(by=['profit'], ascending = False)
```

Out[23]:

	release_year	genres	original_title	director	popularity	budget	revenue	profit
1386	2009	Action|Adventure|Fantasy|Science Fiction	Avatar	James Cameron	9.432768	237000000	2781505847	2544505847
3	2015	Action|Adventure|Science Fiction|Fantasy	Star Wars: The Force Awakens	J.J. Abrams	11.173104	200000000	2068178225	1868178225
5231	1997	Drama|Romance|Thriller	Titanic	James Cameron	4.355219	200000000	1845034188	1645034188
0	2015	Action|Adventure|Science Fiction|Thriller	Jurassic World	Colin Trevorrow	32.985763	150000000	1513528810	1363528810
4	2015	Action|Crime|Thriller	Furious 7	James Wan	9.335014	190000000	1506249360	1316249360
...	...	...	...	...	...	...	...	...
4970	2003	Animation|Adventure|Family|Fantasy	Brother Bear	Aaron Blaise|Robert Walker	1.653031	100000000	250	-99999750
3484	2011	Adventure|Animation|Family	Mars Needs Moms	Simon Wells	0.921653	150000000	38992758	-111007242
7031	2004	Western|History|War	The Alamo	John Lee Hancock	0.948560	145000000	25819961	-119180039
5508	2013	Action|Adventure|Western	The Lone Ranger	Gore Verbinski	1.214510	255000000	89289910	-165710090
2244	2010	Adventure|Fantasy|Action|Western|Thriller	The Warrior's Way	Sngmoo Lee	0.250540	425000000	11087569	-413912431

3853 rows × 8 columns

Exploratory Data Analysis

> Now our data set are ready to give us the answers to our questions

Research Question 1 : who is the best movie director to invest my money with ?

In [24]:
```
#make a list of years

years = my_data_two['release_year']
years_sorted = years.drop_duplicates().sort_values()
print(years_sorted, '\n\n\n', 'No. of years : ',years_sorted.value_counts().sum())
```

10141	1960
10110	1961
9849	1962
10438	1963
9881	1964
10689	1965
10822	1966
10398	1967
9719	1968
10724	1969
10648	1970
9923	1971
7269	1972
10593	1973
9758	1974
9805	1975
10173	1976
1329	1977
10755	1978
7825	1979
7309	1980
8375	1981
8888	1982
7987	1983
7882	1984
6081	1985
10472	1986
9594	1987
9449	1988
9179	1989
9978	1990
9316	1991
8242	1992
10220	1993
4177	1994
8067	1995
8457	1996
5231	1997
8969	1998
2409	1999
8661	2000
2633	2001
3911	2002
4949	2003
6962	2004
6190	2005
6554	2006

```
7387    2007
2875    2008
1386    2009
1919    2010
3372    2011
4361    2012
5422    2013
629     2014
0       2015
Name: release_year, dtype: int64


 No. of years :  56
```

In [25]:
```
#make list of directors

directors = my_data_two['director'].drop_duplicates().sort_values()
print(directors)
```

```
3621                   FrÃ©dÃ©ric Jardin
3235                     A.R. Murugadoss
4970          Aaron Blaise|Robert Walker
8241                        Aaron Norris
6668       Aaron Seltzer|Jason Friedberg
                         ...
1398                         Zack Snyder
5746                      Zal Batmanglij
7268          Zana Briski|Ross Kauffman
2303                   Ãlex de la Iglesia
5133                    Ã‰mile Gaudreault
Name: director, Length: 1713, dtype: object
```

Sub-Question (1) : who is the most profitable directors ?

In [26]:
```
#find out the 5 directors with the maximum commulative profit

index1 = 0
max_profit = 0
profit_director_dict = {}

while index1 < 1713 :
    director_profit = my_data_two[my_data_two['director'] == directors.iloc[index1]]['profit'].sum()
    profit_director_dict[directors.iloc[index1]] = director_profit
    if director_profit > max_profit :
        max_profit = director_profit
        profit_director = directors.iloc[index1]
    index1 += 1
```

```
print( 'The director "',profit_director, '" has a comulative profit equals to "', max_profit, '$" which is the largest comulative
profit_director_df = pd.DataFrame.from_dict(profit_director_dict, orient='index').sort_values(by = [0], ascending=False).iloc[0:5]
print( 'and here is a list of the 5 highest comulative profit directors :',profit_director_df)
```

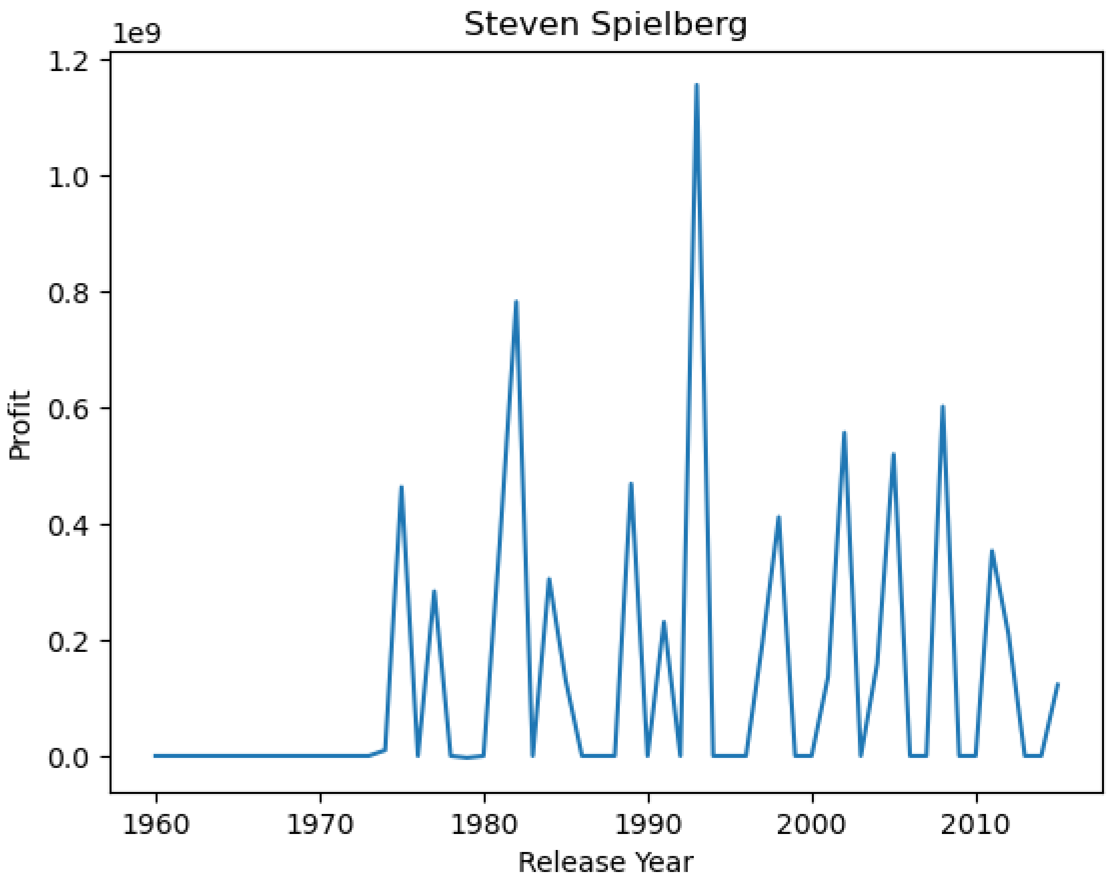
```
The director " Steven Spielberg " has a comulative profit equals to " 7467063772 $" which is the largest comulative profit ever

and here is a list of the 5 highest comulative profit directors :                              0
Steven Spielberg  7467063772
Peter Jackson     5197244659
James Cameron     5081994863
Michael Bay       3557208171
David Yates       3379295625
```

Who is the most active director over years ?

In [27]:
```
def director_growth(fam_director) :

    """
    this function is called to find out how the cumulative profit
    of a director is distributed over years
    """

    s = []
    t = 0

    while t < 56:

        i = years_sorted.iloc[t]
        a = my_data_two[my_data_two['release_year'] == i ]
        b = a[a['director'] == fam_director]
        c = b['profit'].sum()
        s.append(c)
        t += 1

    return(s)
```

In [28]:
```
def plt_director(fam_director_name, fam_director_list) :

    """
    this function is called to plot how the cumulative profit
    of a director is distributed over years
    """


    plt.plot(years_sorted, fam_director_list)
    plt.xlabel('Release Year')
    plt.ylabel('Profit')
    plt.title(fam_director_name)
    plt.show()
```

In [29]:
```
steven = director_growth('Steven Spielberg')
steven2 = plt_director('Steven Spielberg', steven)
```



This figure shows us how the comulative profit of "steven spleberg" is distributed over years

---

In [30]:
```
my_data_two[my_data_two['director'] == 'Steven Spielberg'].sort_values(['profit'])
```

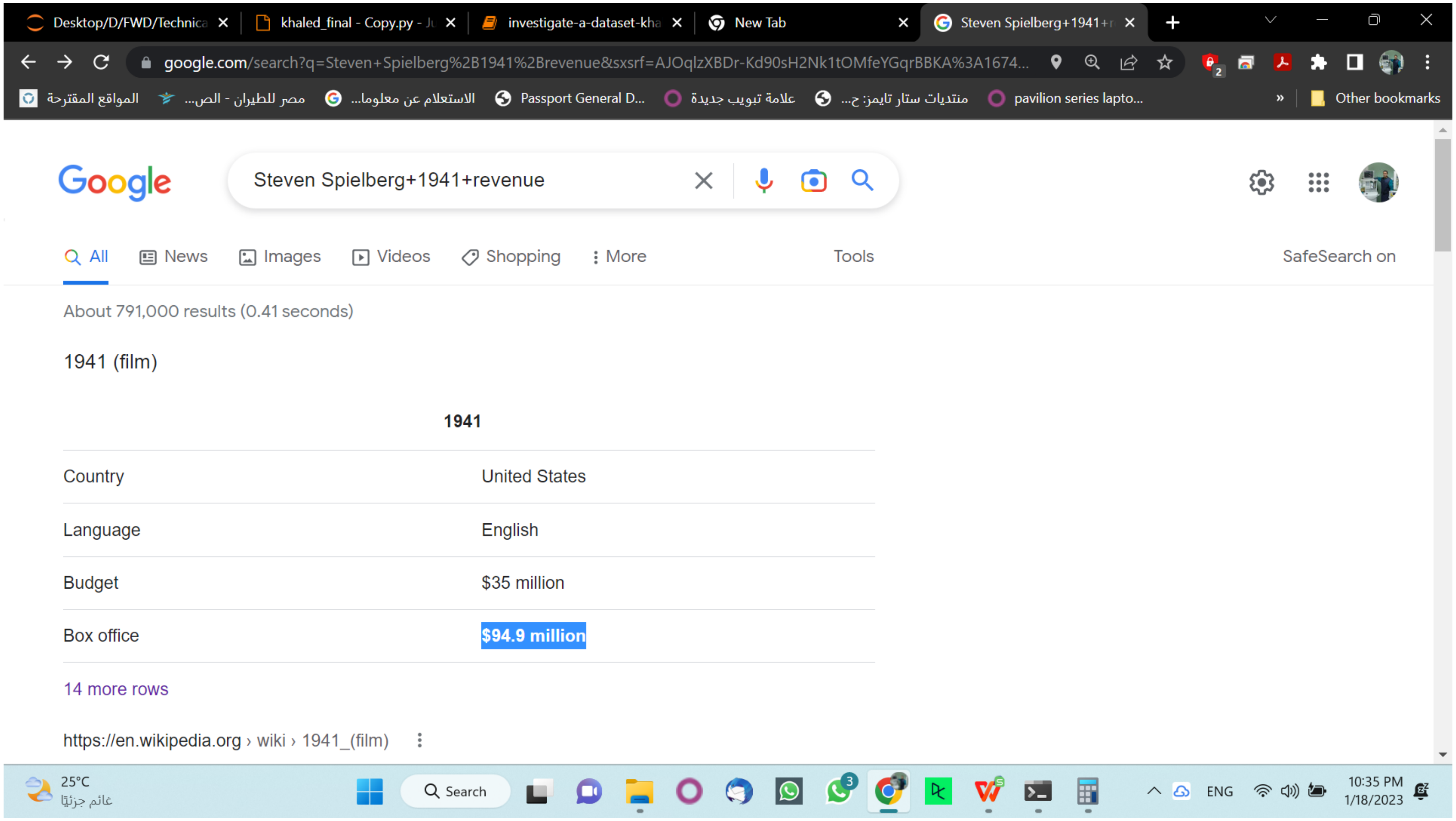
Out[30]:

	release_year	genres	original_title	director	popularity	budget	revenue	profit
7851	1979	Action|Comedy	1941	Steven Spielberg	0.387797	35000000	31755742	-3244258
9770	1974	Action|Crime|Drama	The Sugarland Express	Steven Spielberg	0.415866	3000000	12800000	9800000
5387	1997	Drama|History|Mystery	Amistad	Steven Spielberg	0.221360	36000000	74000000	38000000
9219	1989	Fantasy|Drama|Romance	Always	Steven Spielberg	0.494235	31000000	74134790	43134790
6265	2005	Drama|Action|History|Thriller	Munich	Steven Spielberg	0.869394	70000000	130358911	60358911
3414	2011	Drama|War	War Horse	Steven Spielberg	1.592819	66000000	177584879	111584879
33	2015	Thriller|Drama	Bridge of Spies	Steven Spielberg	3.648210	40000000	162610473	122610473
6094	1985	Drama	The Color Purple	Steven Spielberg	1.012186	15000000	146292009	131292009
2638	2001	Drama|Science Fiction|Adventure	A.I. Artificial Intelligence	Steven Spielberg	2.971372	100000000	235926552	135926552
5391	1997	Adventure|Action|Science Fiction	The Lost World: Jurassic Park	Steven Spielberg	0.210550	73000000	229074524	156074524
6988	2004	Comedy|Drama	The Terminal	Steven Spielberg	1.682492	60000000	219417255	159417255
4425	2012	Drama|War	Lincoln	Steven Spielberg	1.312488	65000000	275293450	210293450
9318	1991	Adventure|Fantasy|Comedy|Family	Hook	Steven Spielberg	2.326917	70000000	300854823	230854823
3397	2011	Adventure|Animation|Action|Family|Mystery	The Adventures of Tintin	Steven Spielberg	2.234300	130000000	371940071	241940071
3921	2002	Action|Thriller|Science Fiction|Mystery	Minority Report	Steven Spielberg	2.103595	102000000	358372926	256372926
1334	1977	Science Fiction|Drama	Close Encounters of the Third Kind	Steven Spielberg	1.104816	20000000	303788635	283788635
10222	1993	Drama|History|War	Schindler's List	Steven Spielberg	2.377288	22000000	321265768	299265768

	release_year	genres	original_title	director	popularity	budget	revenue	profit
3918	2002	Drama|Crime	Catch Me If You Can	Steven Spielberg	2.973115	52000000	352114312	300114312
7883	1984	Adventure|Action	Indiana Jones and the Temple of Doom	Steven Spielberg	2.556799	28000000	333000000	305000000
8375	1981	Adventure|Action	Raiders of the Lost Ark	Steven Spielberg	4.578300	18000000	389925971	371925971
8974	1998	Drama|History|War	Saving Private Ryan	Steven Spielberg	2.170136	70000000	481840909	411840909
9180	1989	Adventure|Action	Indiana Jones and the Last Crusade	Steven Spielberg	3.536655	48000000	474171806	426171806
6205	2005	Adventure|Thriller|Science Fiction	War of the Worlds	Steven Spielberg	1.844731	132000000	591739379	459739379
9806	1975	Horror|Thriller|Adventure	Jaws	Steven Spielberg	2.563191	7000000	470654000	463654000
2879	2008	Adventure|Action	Indiana Jones and the Kingdom of the Crystal S...	Steven Spielberg	3.161670	185000000	786636033	601636033
8889	1982	Science Fiction|Adventure|Family|Fantasy	E.T. the Extra-Terrestrial	Steven Spielberg	2.900556	10500000	792910554	782410554
10223	1993	Adventure|Science Fiction	Jurassic Park	Steven Spielberg	2.204926	63000000	920100000	857100000

Warning : I think that our data set is not accurate enuogh to make money decisions and this is another limitation

---


Steven Spielberg+1941+revenue

About 791,000 results (0.41 seconds)

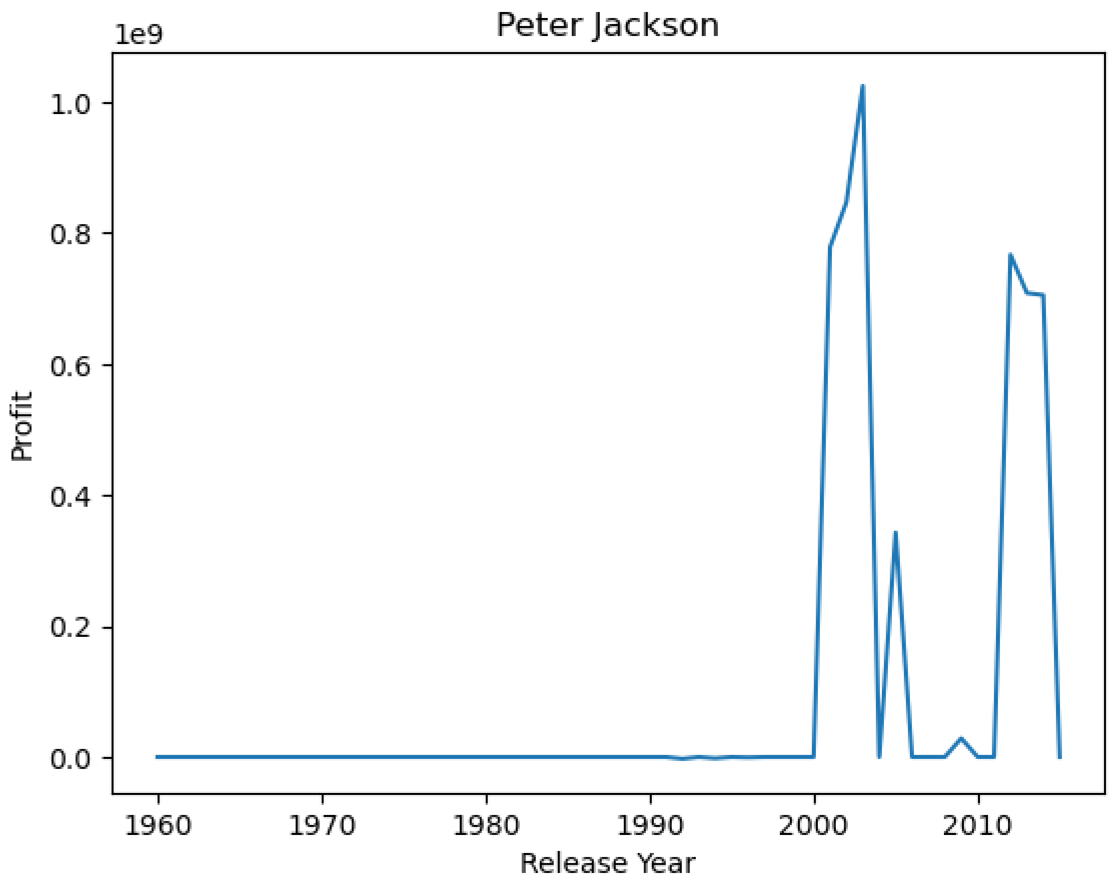
1941 (film)

	1941
Country	United States
Language	English
Budget	$35 million
Box office	$94.9 million

14 more rows

https://en.wikipedia.org › wiki › 1941_(film)

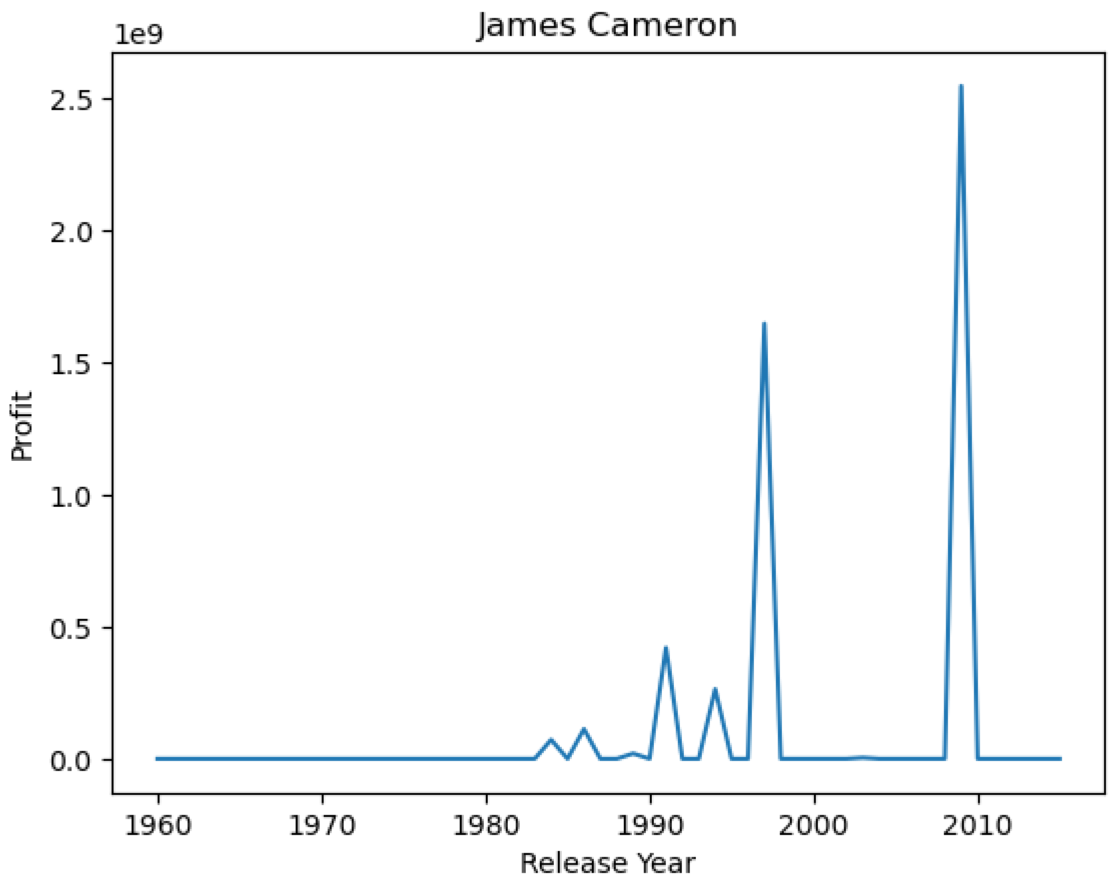
In [31]:
```python
peter = director_growth('Peter Jackson')
peter2 = plt_director('Peter Jackson', peter)
```



This figure shows us how the comulative profit of "peter jackson" is distributed over years

---

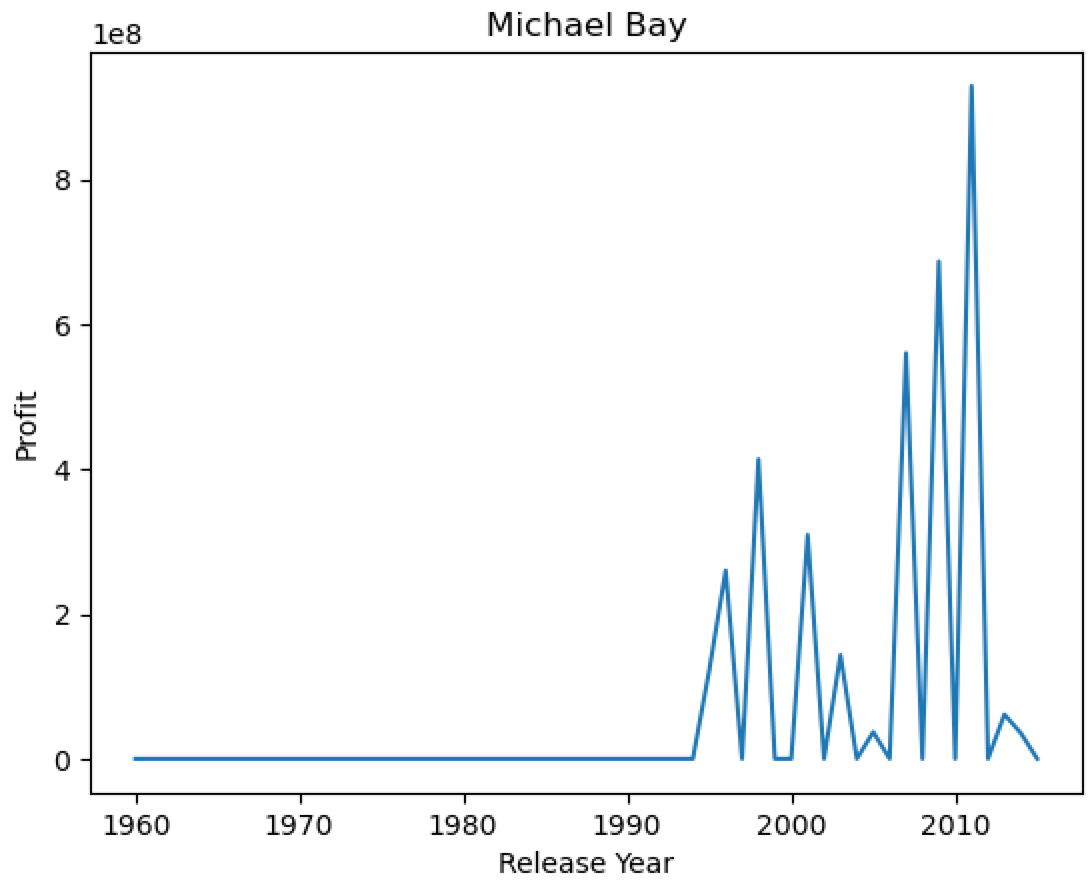
In [32]:
```
james = director_growth('James Cameron')
james2 = plt_director('James Cameron', james)
```



This figure shows us how the comulative profit of "james cameron" is distributed over years

---

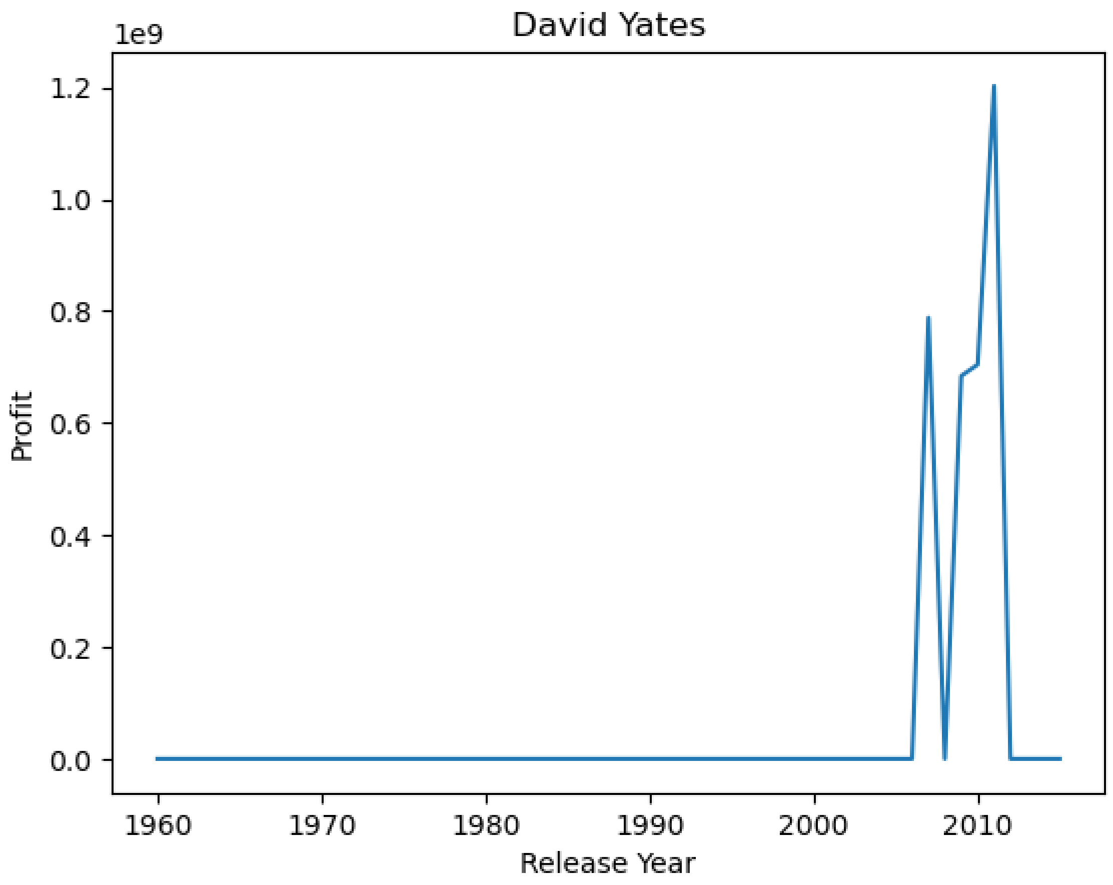
In [33]:
```
michael = director_growth('Michael Bay')
michael2 = plt_director('Michael Bay', michael)
```



This figure shows us how the comulative profit of "michael bay" is distributed over years

---

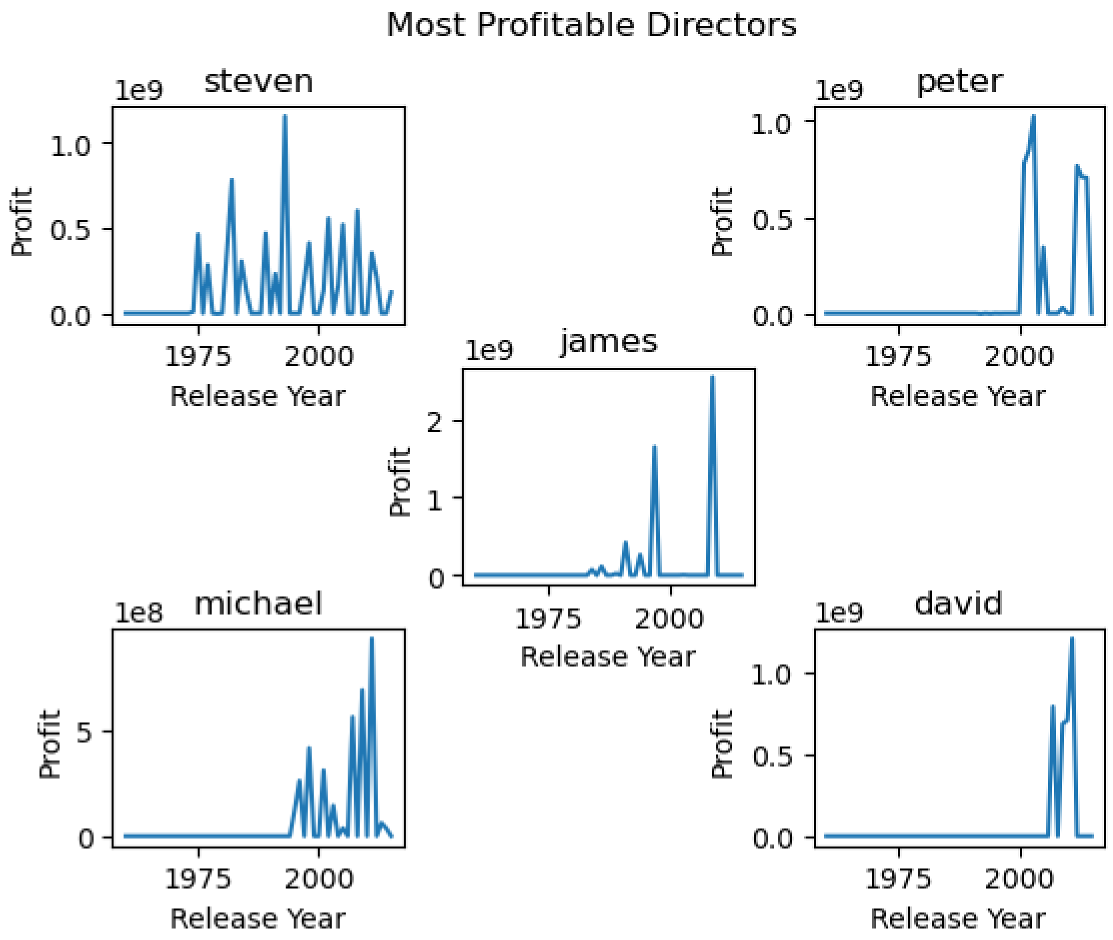
In [34]:
```
david = director_growth('David Yates')
david2 = plt_director('David Yates', david)
```



This figure shows us how the comulative profit of "David yates" is distributed over years

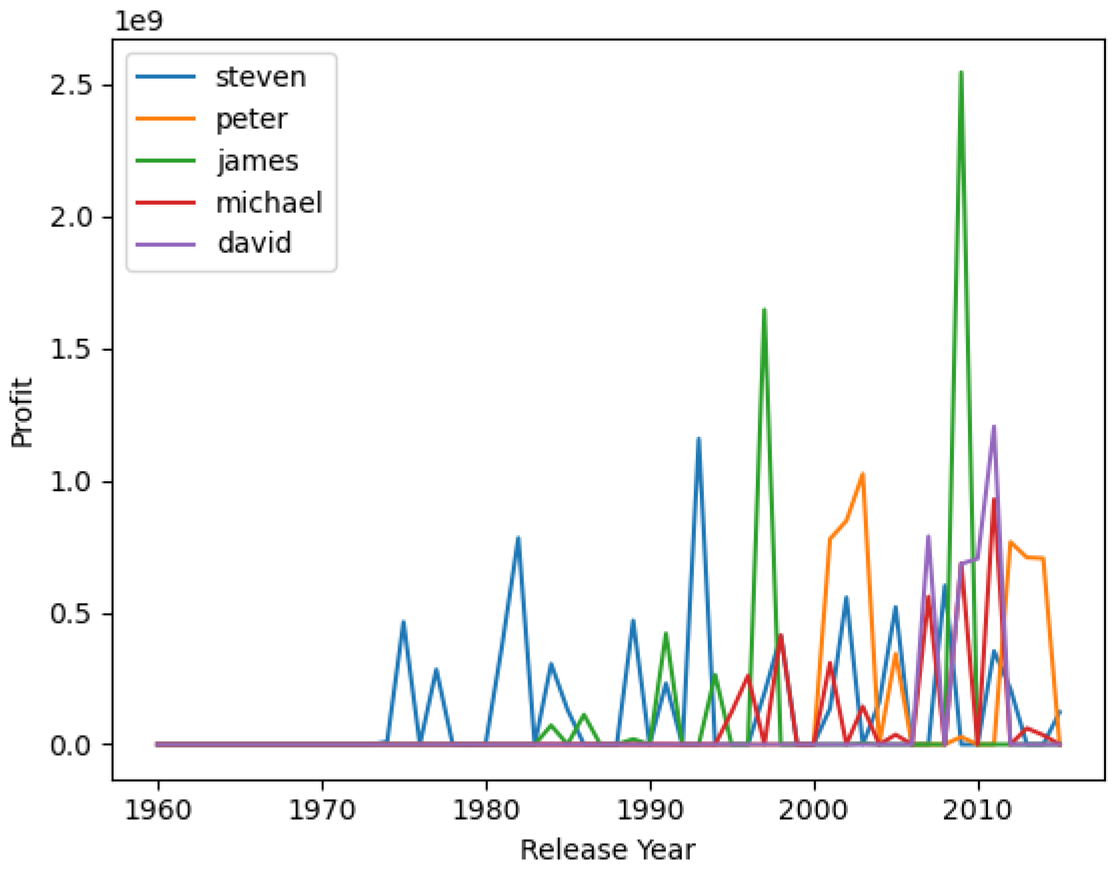
---

In [35]:
```
plt.subplot(3,3,1)
plt.plot(years_sorted, steven)
plt.xlabel('Release Year')
plt.ylabel('Profit')
plt.title('steven')

plt.subplot(3,3,3)
plt.plot(years_sorted, peter)
plt.xlabel('Release Year')
plt.ylabel('Profit')
plt.title('peter')

plt.subplot(3,3,5)
plt.plot(years_sorted, james)
plt.xlabel('Release Year')
plt.ylabel('Profit')
plt.title('james')

plt.subplot(3,3,7)
plt.plot(years_sorted, michael)
plt.xlabel('Release Year')
plt.ylabel('Profit')
plt.title('michael')

plt.subplot(3,3,9)
plt.plot(years_sorted, david)
plt.xlabel('Release Year')
plt.ylabel('Profit')
plt.title('david')

plt.suptitle("Most Profitable Directors")
plt.show()
```



```
In [36]: plt.plot(years_sorted, steven, label = 'steven')

         plt.plot(years_sorted, peter, label = 'peter')

         plt.plot(years_sorted, james, label = 'james')

         plt.plot(years_sorted, michael, label = 'michael')

         plt.plot(years_sorted, david, label = 'david')

         plt.xlabel('Release Year')
         plt.ylabel('Profit')

         plt.legend()

         plt.show()
```

```
print("This figure show comparison between how the comulative profits of the 5 highest profit dirctors")
```



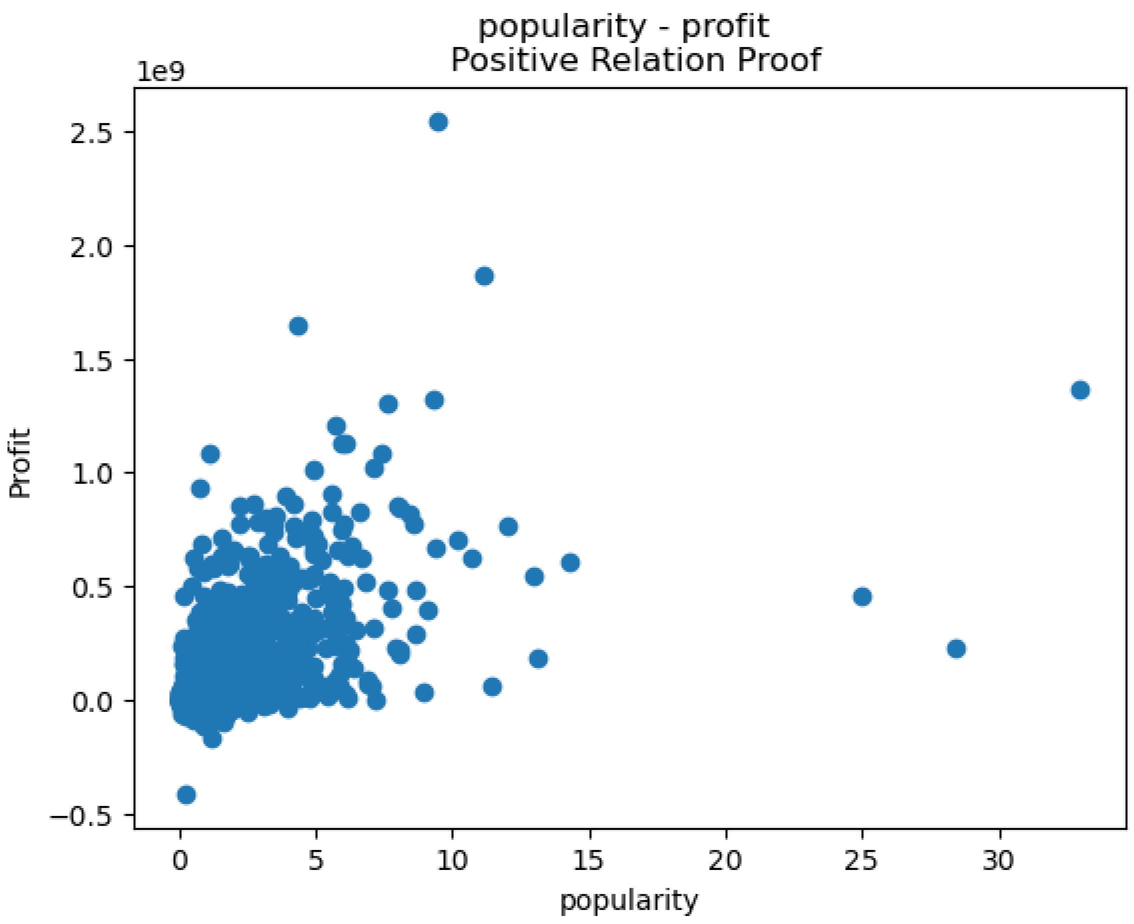
```
This figure show comparison between how the comulative profits of the 5 highest profit dirctors
```

> From this figure, I see that although "steven spielberg" has the highest cumulative profit but "james cameron" has the highest sparks over years

Sub-Question (2) : who is the most popular directors ?

> at the beging, let us take a look at the relation between popularity and profit

In [37]:
```
plt.scatter(my_data_two['popularity'], my_data_two['profit'])
plt.xlabel('popularity')
plt.ylabel('Profit')
plt.title('popularity - profit \n Positive Relation Proof' )
plt.show()
```



> It is obvious that there is a positive relation between popularity and profit

In [38]:
```
index2 = 0
max_popularity = 0
popular_director_dict = {}


while index2 < 1713 :
    director_popularity = my_data_two[my_data_two['director'] == directors.iloc[index2]]['popularity'].max()
    popular_director_dict[directors.iloc[index2]] = director_popularity
    if director_popularity > max_popularity :
        max_popularity = director_popularity
        popular_director = directors.iloc[index2]
    index2 += 1


popular_director_df = pd.DataFrame.from_dict(popular_director_dict, orient='index').sort_values(by = [0], ascending=False).iloc[0:

print("Top 5 directors gained max popularity")
popular_director_df
```

```
Top 5 directors gained max popularity
```

Out[38]:

	0
Colin Trevorrow	32.985763
George Miller	28.419936
Christopher Nolan	24.949134
James Gunn	14.311205
Robert Schwentke	13.112507

In [39]:

```
index2 = 0
max_popularity = 0
popular_director_dict = {}


while index2 < 1713 :
    director_popularity = my_data_two[my_data_two['director'] == directors.iloc[index2]]['popularity'].mean()
    popular_director_dict[directors.iloc[index2]] = director_popularity
    if director_popularity > max_popularity :
        max_popularity = director_popularity
        popular_director = directors.iloc[index2]
    index2 += 1


popular_director_df = pd.DataFrame.from_dict(popular_director_dict, orient='index').sort_values(by = [0], ascending=False).iloc[0:

print("Top 5 directors gained highest mean popularity")

popular_director_df
```

```
Top 5 directors gained highest mean popularity
```

Out[39]:

	0
Colin Trevorrow	16.696886
Joe Russo|Anthony Russo	12.971027
Chad Stahelski|David Leitch	11.422751
Don Hall|Chris Williams	8.691294
Morten Tyldum	8.110711

> "Colin Trevorrow" gained the highest popularity ever

Research Question 2 : what is the best movie genre to invest my money in ?

Sub-Question (1) : what is the most profitable movie genre ?

In [40]:
```
#prepare a list of genres

genres = my_data_two['genres'].drop_duplicates()
print(genres)
```

```
0              Action|Adventure|Science Fiction|Thriller
2                     Adventure|Science Fiction|Thriller
3               Action|Adventure|Science Fiction|Fantasy
4                                  Action|Crime|Thriller
5                       Western|Drama|Adventure|Thriller
                               ...
10780           Horror|Thriller|Science Fiction|Mystery
10788    Adventure|Family|Fantasy|Music|Science Fiction
10791                         Action|Drama|Horror|Thriller
10793                            Adventure|Animation|Drama
10835                    Action|Adventure|Drama|War|Romance
Name: genres, Length: 1053, dtype: object
```

In [41]:
```
index3 = 0
max_genres = 0
profit_genres_dict = {}

while index3 < 1053 :
    genres_profit = my_data_two[my_data_two['genres'] == genres.iloc[index3]]['profit'].sum()
    profit_genres_dict[genres.iloc[index3]] = genres_profit
    if genres_profit > max_genres :
        max_genres = genres_profit
        profit_genres = genres.iloc[index3]
    index3 += 1

profit_genres_df = pd.DataFrame.from_dict(profit_genres_dict, orient='index').sort_values(by = [0], ascending=False).iloc[0:5]
print("Top 5 Genres ")
profit_genres_df
```

```
Top 5 Genres
```

Out[41]:

	0
Comedy	12183078642
Drama	9050102799
Comedy|Romance	7822616677
Adventure|Fantasy|Action	5820583556
Action|Adventure|Science Fiction	4832602017

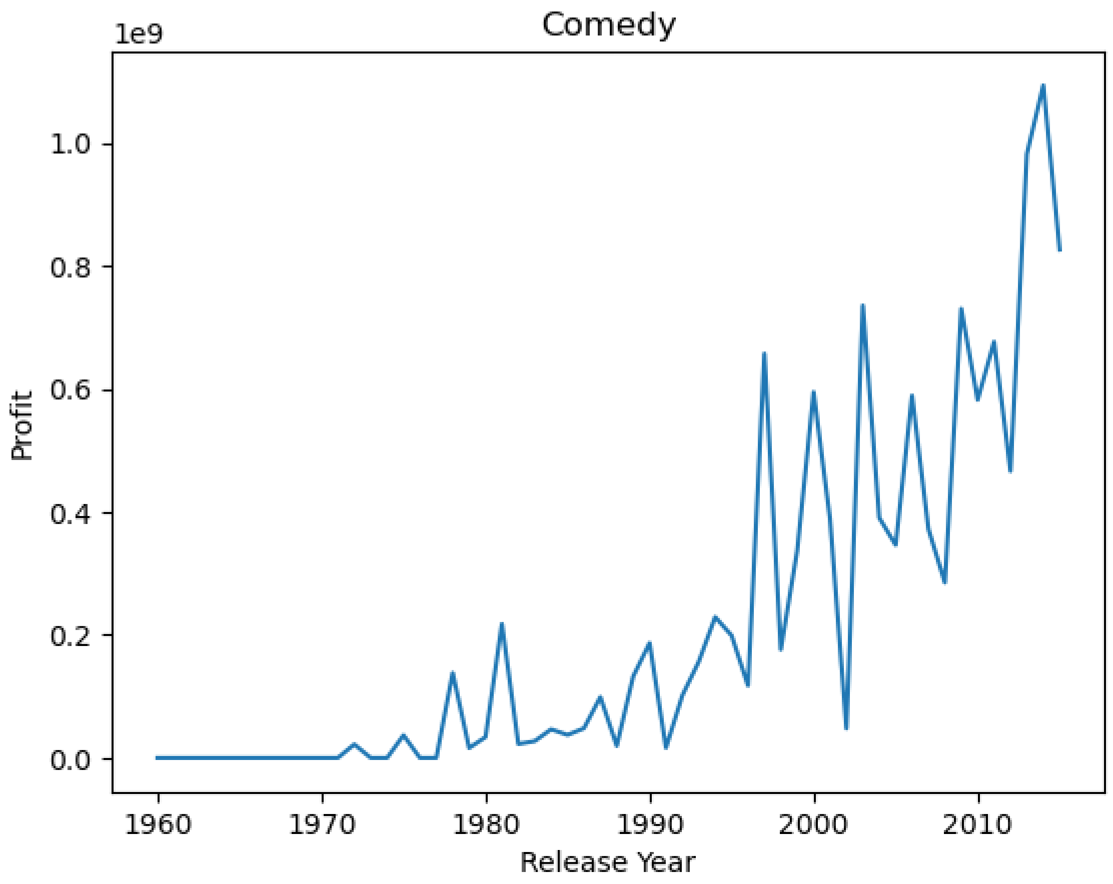
In [42]:

```
def genres_growth(fam_genres) :

    """
    a function to find out the profit of genres over years
    """

    s = []
    t = 0

    while t < 56:

        i = years_sorted.iloc[t]
        a = my_data_two[my_data_two['release_year'] == i ]
        b = a[a['genres'] == fam_genres]
        c = b['profit'].sum()
        s.append(c)
        t += 1

    return(s)
```

In [43]:

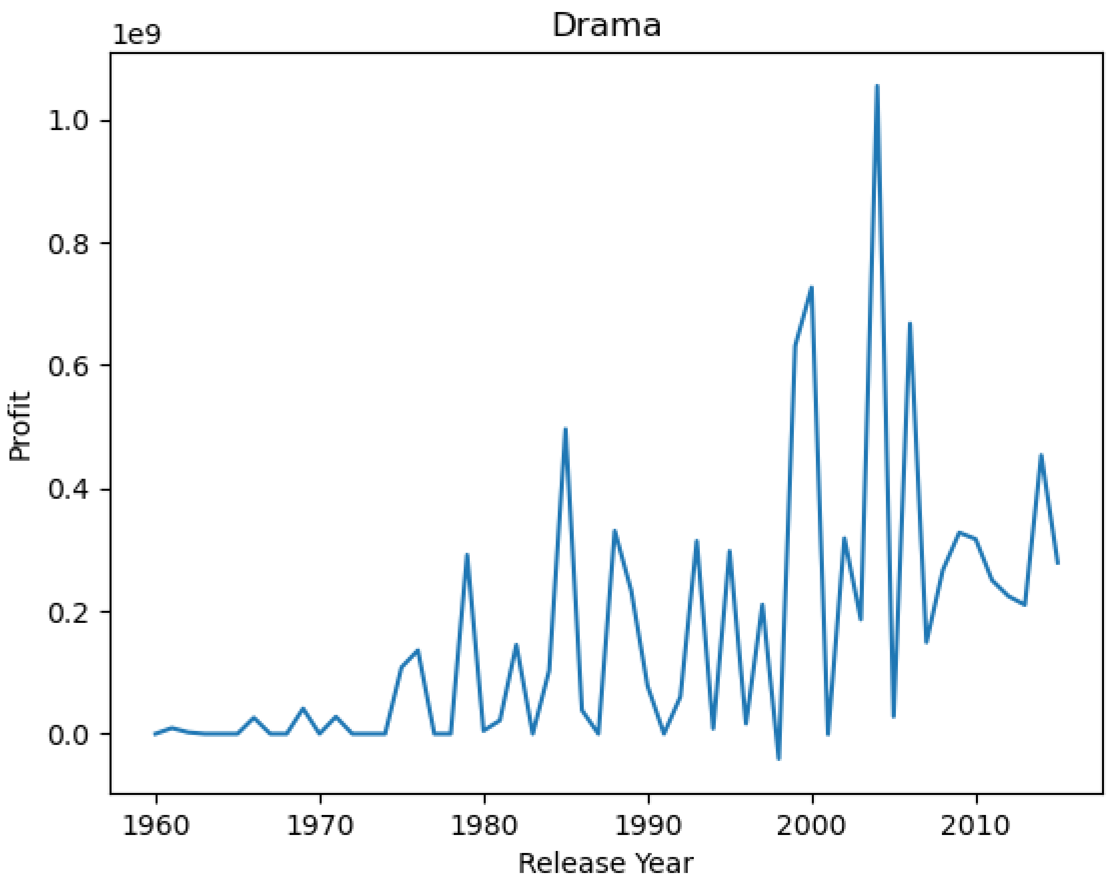
```
def plt_director(fam_genres_name, fam_genres_list) :

    """
    a function to plot the profit of genres over years
    """
    plt.plot(years_sorted, fam_genres_list)
    plt.xlabel('Release Year')
    plt.ylabel('Profit')
    plt.title(fam_genres_name)
    plt.show()
```

In [44]:
```
comedy = genres_growth('Comedy')
comedy2 = plt_director('Comedy', comedy)
```



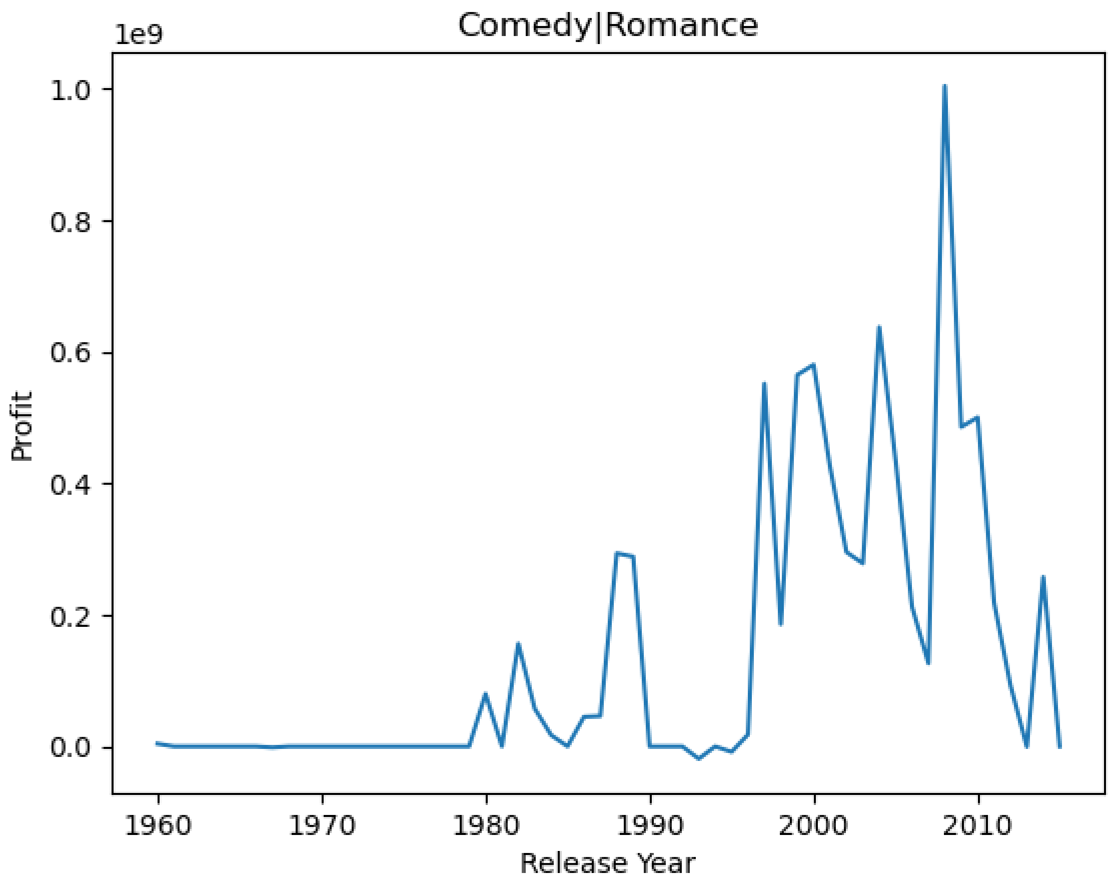
> This figure show that the profit of comedy films increase year after year

In [45]:
```
drama = genres_growth('Drama')
Drama2 = plt_director('Drama', drama)
```



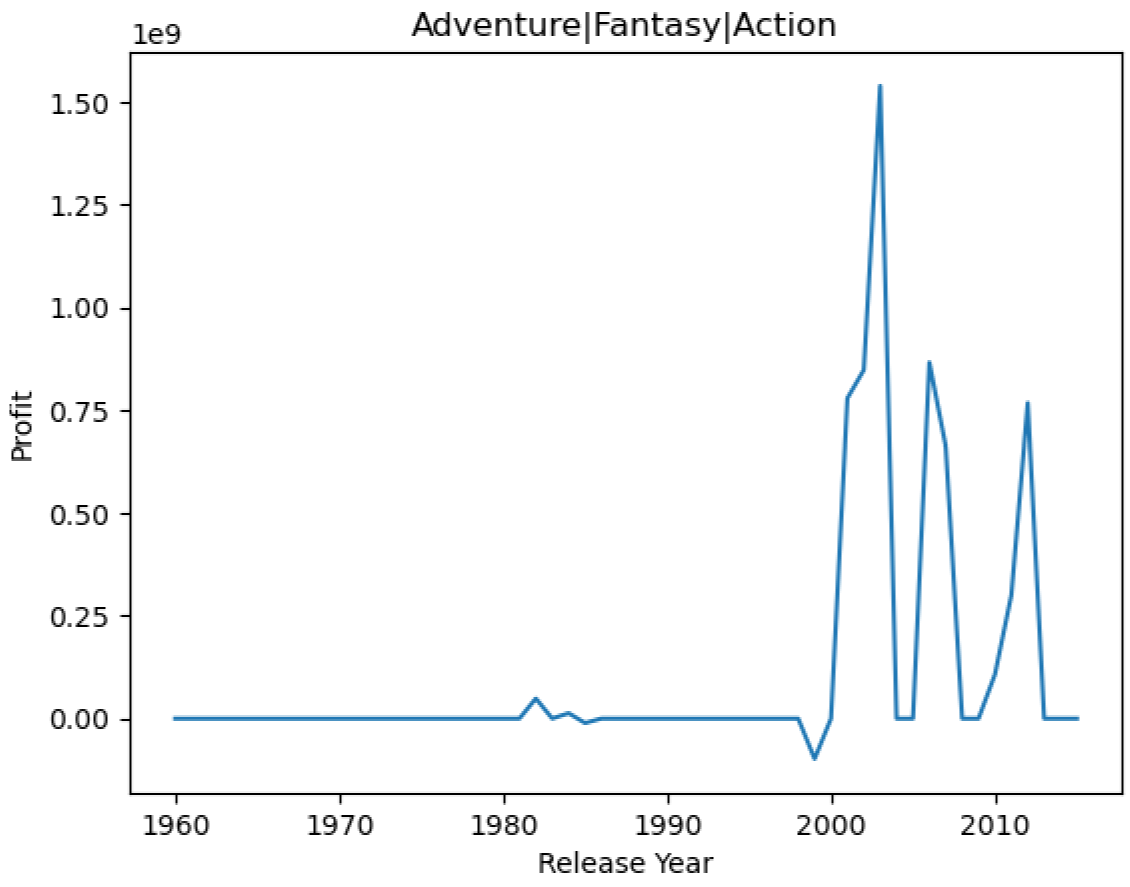
> This figure show that the profit of drama films increase year after year

In [46]:
```
ComedyRomance = genres_growth('Comedy|Romance')
ComedyRomance2 = plt_director('Comedy|Romance', ComedyRomance)
```



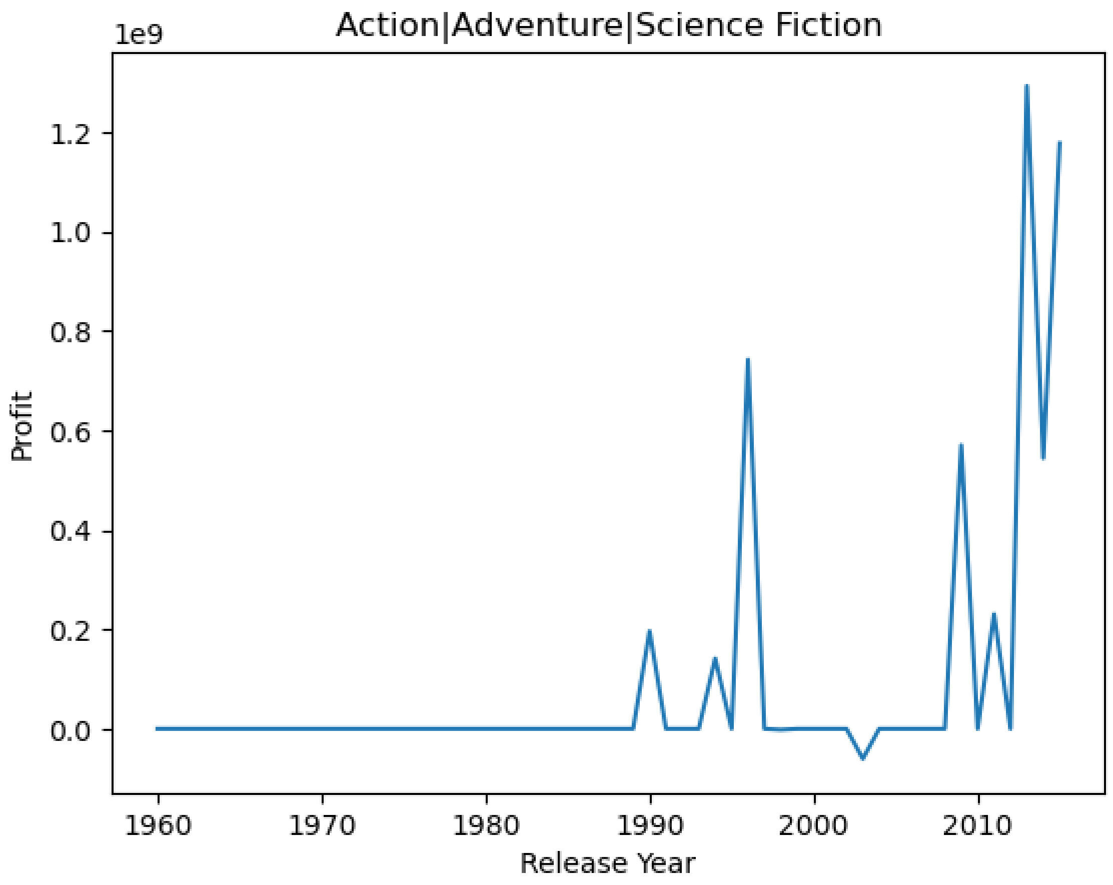
> This figure show that the profit of comedy or romance films increase year after year but not in the last years

In [47]:
```
AdventureFantasyAction = genres_growth('Adventure|Fantasy|Action')
AdventureFantasyAction2 = plt_director('Adventure|Fantasy|Action', AdventureFantasyAction)
```



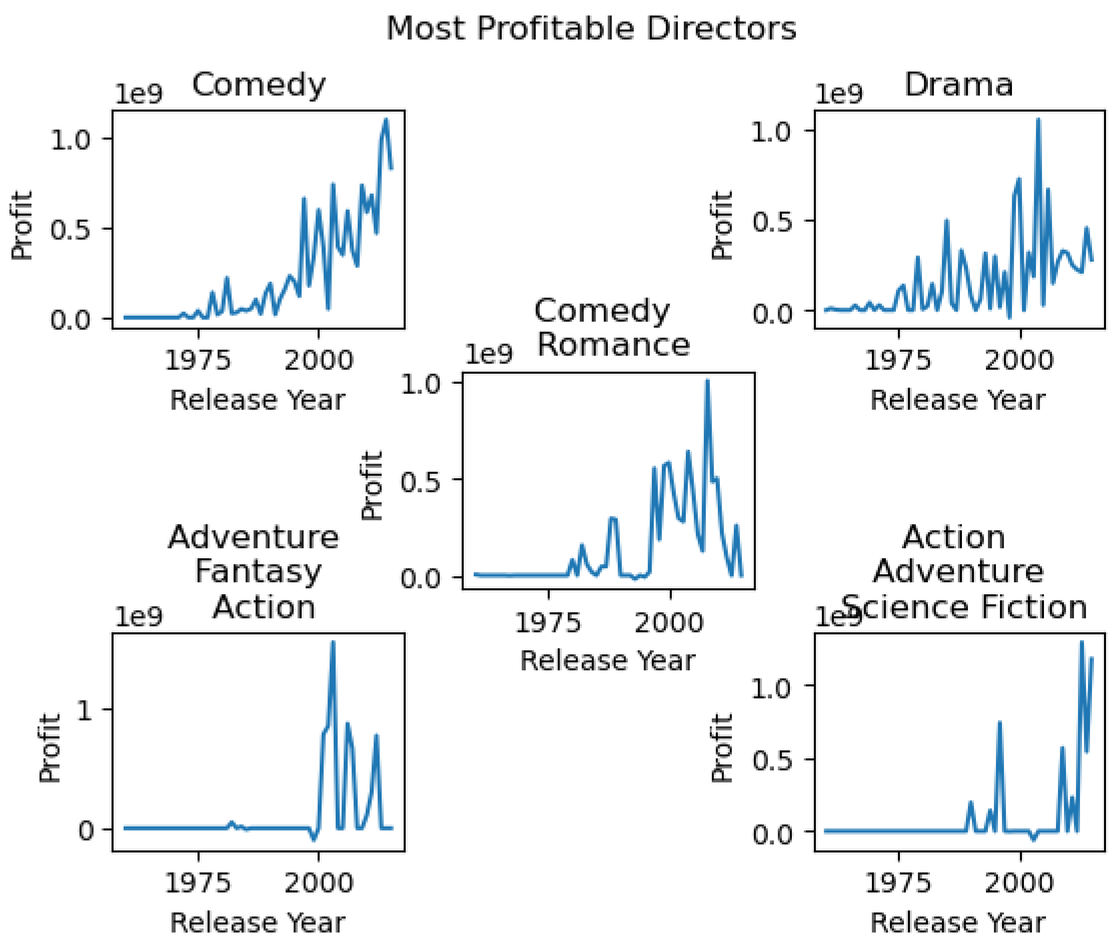
> This figure show that the profit of dventure or fantasy or action films increase year after year but not in the last years

```
In [48]: Action_Adventure_ScienceFiction = genres_growth('Action|Adventure|Science Fiction')
         Action_Adventure_ScienceFiction2 = plt_director('Action|Adventure|Science Fiction', Action_Adventure_ScienceFiction)
```

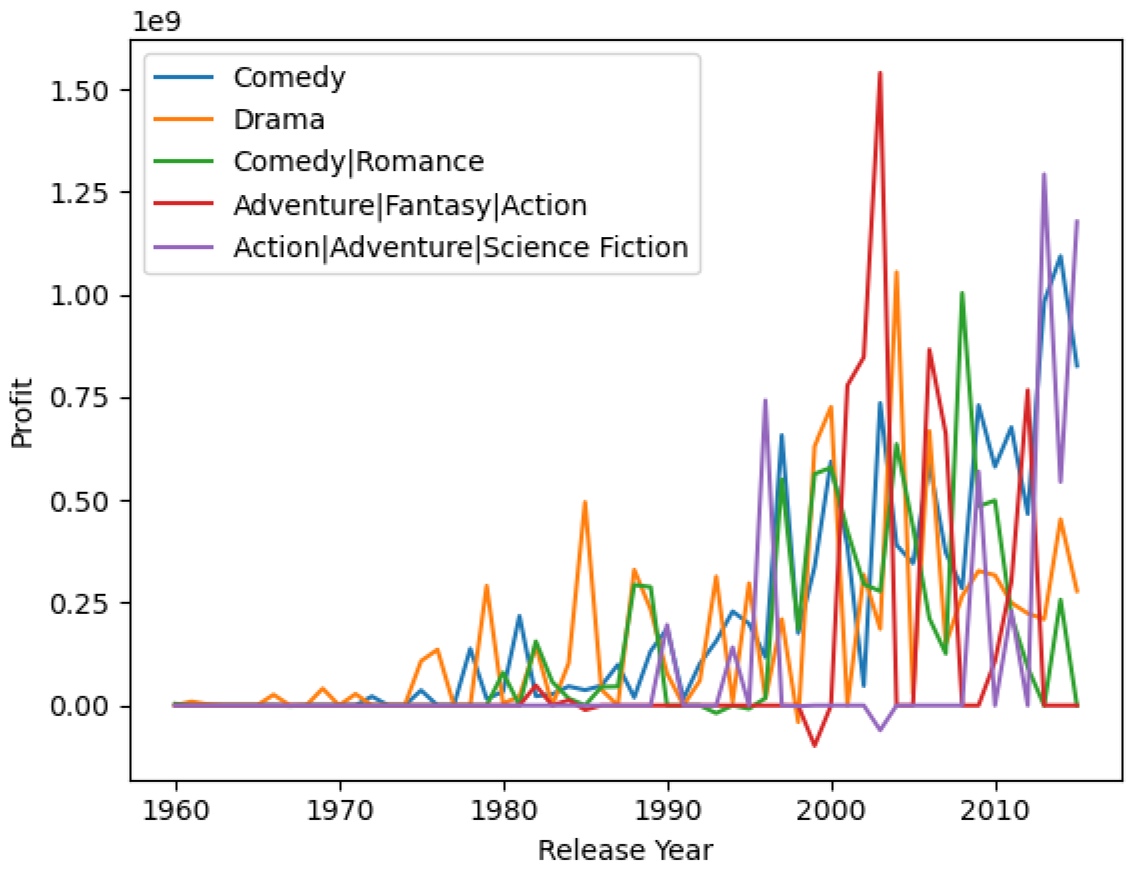


> This figure show that the profit of action or adventure or science fiction films increase year after year, and here is the future

In [49]:
```
plt.subplot(3,3,1)
plt.plot(years_sorted, comedy)
plt.xlabel('Release Year')
plt.ylabel('Profit')
plt.title('Comedy')

plt.subplot(3,3,3)
plt.plot(years_sorted, drama)
plt.xlabel('Release Year')
plt.ylabel('Profit')
plt.title('Drama')

plt.subplot(3,3,5)
plt.plot(years_sorted, ComedyRomance)
plt.xlabel('Release Year')
plt.ylabel('Profit')
plt.title('Comedy \n Romance')

plt.subplot(3,3,7)
plt.plot(years_sorted, AdventureFantasyAction)
plt.xlabel('Release Year')
plt.ylabel('Profit')
plt.title('Adventure \n Fantasy \n Action')

plt.subplot(3,3,9)
plt.plot(years_sorted, Action_Adventure_ScienceFiction)
plt.xlabel('Release Year')
plt.ylabel('Profit')
plt.title('Action \n Adventure \n Science Fiction')

plt.suptitle("Most Profitable Directors")
plt.show()
```



In [50]:
```python
plt.plot(years_sorted, comedy, label = 'Comedy')

plt.plot(years_sorted, drama, label = 'Drama')

plt.plot(years_sorted, ComedyRomance, label = 'Comedy|Romance')

plt.plot(years_sorted, AdventureFantasyAction, label = 'Adventure|Fantasy|Action')

plt.plot(years_sorted, Action_Adventure_ScienceFiction, label = 'Action|Adventure|Science Fiction')

plt.xlabel('Release Year')
plt.ylabel('Profit')

plt.legend()

plt.show()

print("here is a comparison between distributed profit of top 5 genres over years")
```



```
here is a comparison between distributed profit of top 5 genres over years
```

> The future of profit is for "action , adventure, science fiction" movies

Sub-Question (2) : what is the most popular movie genre ?

In [51]:
```
index4 = 0
popular_gen_dict = {}


while index4 < 1053 :
    gen_popularity = my_data_two[my_data_two['genres'] == genres.iloc[index4]]['popularity'].max()
    popular_gen_dict[genres.iloc[index4]] = gen_popularity
    index4 += 1


popular_gen_df = pd.DataFrame.from_dict(popular_gen_dict, orient='index').sort_values(by = [0], ascending=False).iloc[0:5]

print("Top 5 genres according max popularity")
popular_gen_df
```

```
Top 5 genres according max popularity
```

Out[51]:

	0
Action|Adventure|Science Fiction|Thriller	32.985763
Adventure|Drama|Science Fiction	24.949134
Action|Science Fiction|Adventure	14.311205
Adventure|Science Fiction|Thriller	13.112507
Action|Adventure|Science Fiction	12.971027

In [52]:
```
index4 = 0
popular_gen_dict = {}


while index4 < 1053 :
    gen_popularity = my_data_two[my_data_two['genres'] == genres.iloc[index4]]['popularity'].mean()
    popular_gen_dict[genres.iloc[index4]] = gen_popularity
    index4 += 1


popular_gen_df = pd.DataFrame.from_dict(popular_gen_dict, orient='index').sort_values(by = [0], ascending=False).iloc[0:5]
```

```
print("Top 5 genres according mean popularity")

popular_gen_df
```

```
Top 5 genres according mean popularity
```

Out[52]:

	0
Adventure|Drama|Science Fiction	24.949134
Adventure|Science Fiction|Thriller	13.112507
Action|Adventure|Science Fiction|Fantasy	11.173104
Action|Adventure|Science Fiction|Thriller	10.968490
Science Fiction|Adventure|Thriller	10.739009

In [108…

```
def decision(decision_director, decision_genres) :

    """
    this function to find the profit over years for a specific director wirh specific genre

    """

    s = []
    t = 0

    while t < 56:

        i = years_sorted.iloc[t]
        a = my_data_two[my_data_two['release_year'] == i ]
        b = a.loc[(a['director'] == decision_director)  & (a['genres'] == decision_genres) ]
        c = b['profit'].sum()
        s.append(c)
        t += 1
    print(s)
    return(s)
```

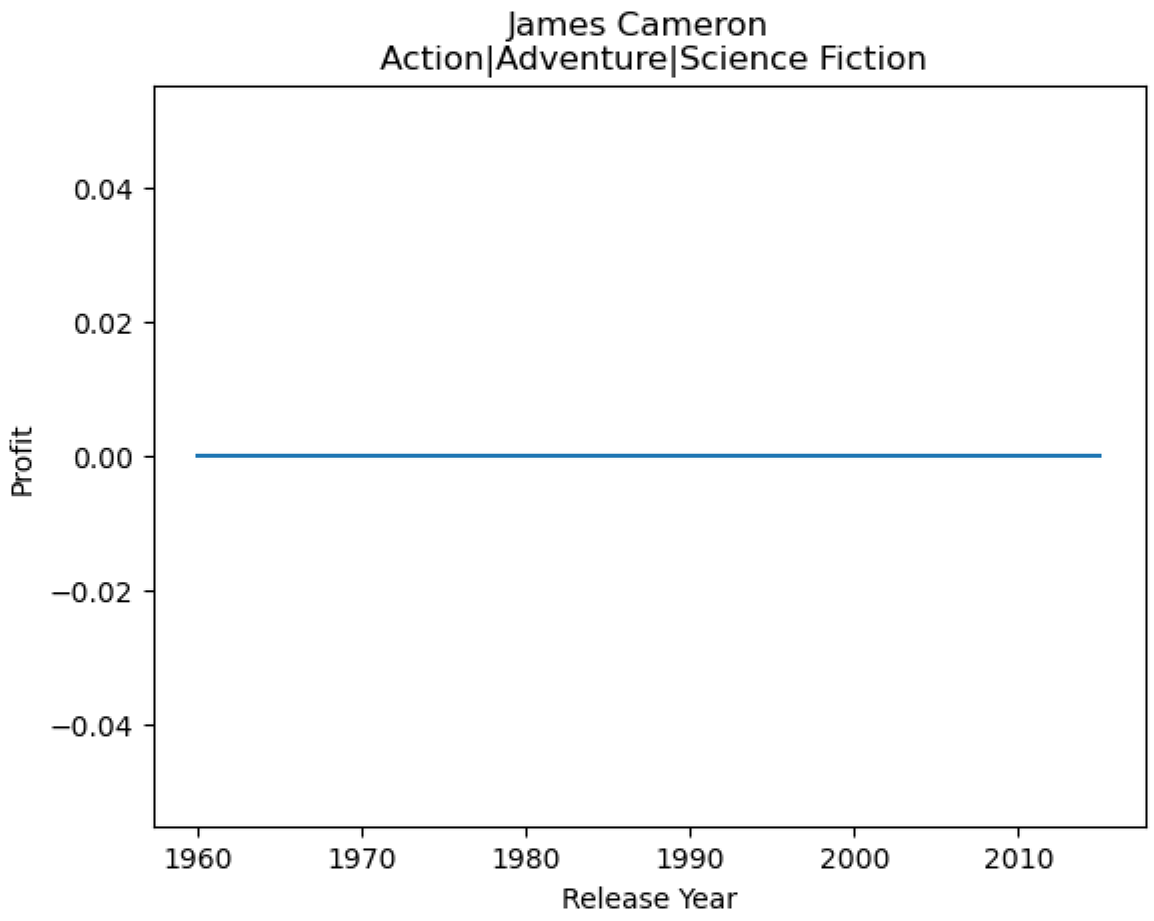
In [109…

```
def plt_decision(decision_name, decision_list) :

    """
    this function to plot the profit over years for a specific director wirh specific genre

    """

    plt.plot(years_sorted, decision_list)
    plt.xlabel('Release Year')
    plt.ylabel('Profit')
    plt.title(decision_name)
    plt.show()
```

In [110…

```
decision_a = decision('James Cameron', 'Action|Adventure|Science Fiction')
decision_aa = plt_decision( 'James Cameron  \n Action|Adventure|Science Fiction', decision_a)
```

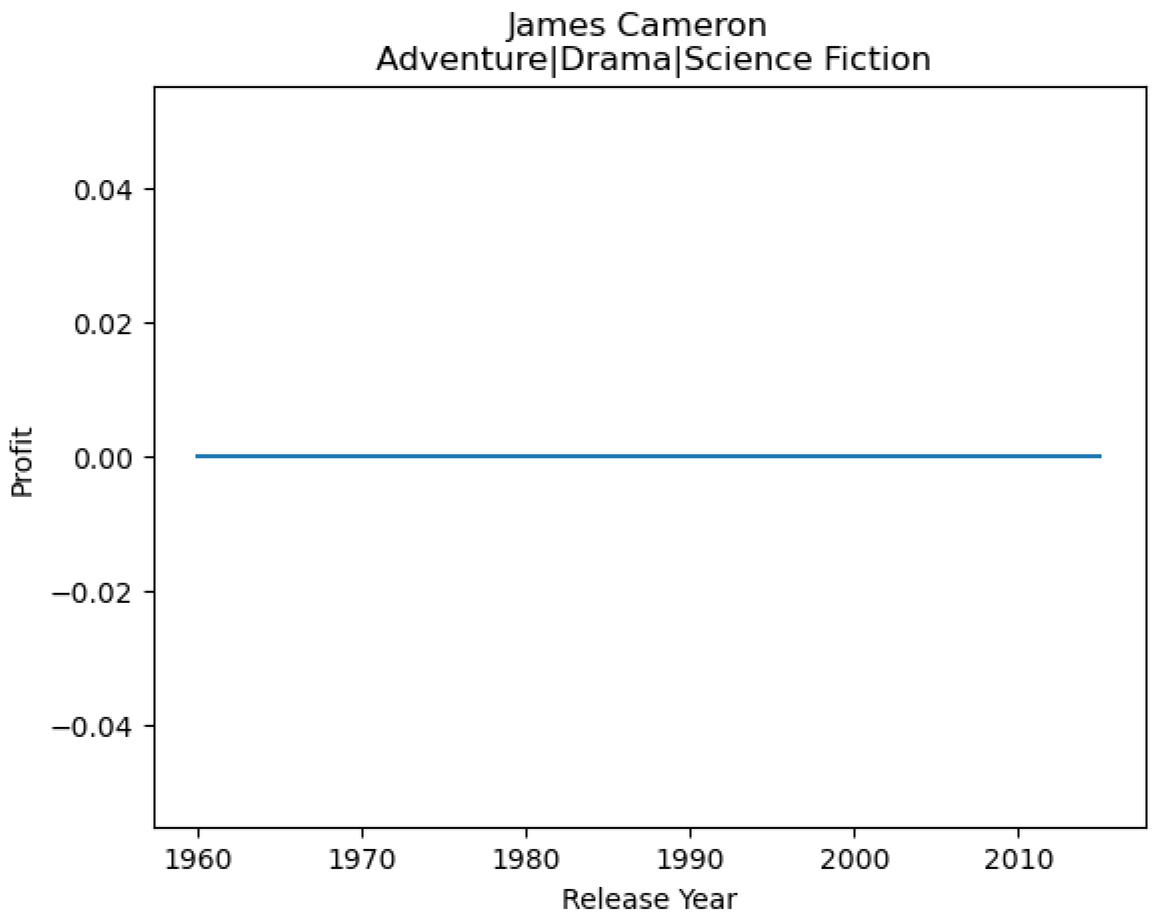
```
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```



In [111…

```
decision_a = decision('James Cameron', 'Adventure|Drama|Science Fiction')
decision_aa = plt_decision( 'James Cameron  \n Adventure|Drama|Science Fiction', decision_a)
```

```
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```



> Unfortunatily,James Cameron has no movies history of "Action|Adventure|Science Fiction" or "Adventure|Drama|Science Fiction" genre
> But let us check the genres of movies he had directed before we might find some similar genres

In [112…

```
my_data_two.loc[my_data_two['director'] == 'James Cameron'].sort_values('genres')
```

Out[112]:

	release_year	genres	original_title	director	popularity	budget	revenue	profit
1386	2009	Action|Adventure|Fantasy|Science Fiction	Avatar	James Cameron	9.432768	237000000	2781505847	2544505847
4186	1994	Action|Thriller	True Lies	James Cameron	1.843243	115000000	378882411	263882411
7882	1984	Action|Thriller|Science Fiction	The Terminator	James Cameron	4.831966	6400000	78371200	71971200
9317	1991	Action|Thriller|Science Fiction	Terminator 2: Judgment Day	James Cameron	3.584406	100000000	520000000	420000000
9189	1989	Adventure|Action|Thriller|Science Fiction	The Abyss	James Cameron	1.691080	70000000	90000098	20000098
10472	1986	Horror|Action|Thriller|Science Fiction	Aliens	James Cameron	2.485419	18500000	131060248	112560248

> Great,James Cameron has a great history of directing movies of similar genres

In [113…

```
#let's exclude 'Ghosts of the Abyss' because it is not similar genre

my_data_two.drop(my_data_two[my_data_two['original_title'] == 'Ghosts of the Abyss' ].index, inplace = True)
```

In [114…

```
James_similar_genres = my_data_two.loc[my_data_two['director'] == 'James Cameron'].sort_values('release_year')
```

In [115…

```
James_similar_genres
```

Out[115]:

	release_year	genres	original_title	director	popularity	budget	revenue	profit
7882	1984	Action|Thriller|Science Fiction	The Terminator	James Cameron	4.831966	6400000	78371200	71971200
10472	1986	Horror|Action|Thriller|Science Fiction	Aliens	James Cameron	2.485419	18500000	131060248	112560248
9189	1989	Adventure|Action|Thriller|Science Fiction	The Abyss	James Cameron	1.691080	70000000	90000098	20000098
9317	1991	Action|Thriller|Science Fiction	Terminator 2: Judgment Day	James Cameron	3.584406	100000000	520000000	420000000
4186	1994	Action|Thriller	True Lies	James Cameron	1.843243	115000000	378882411	263882411
1386	2009	Action|Adventure|Fantasy|Science Fiction	Avatar	James Cameron	9.432768	237000000	2781505847	2544505847

> James Cameron similar genres

In [116…

```
James_similar_genres.describe()
```

Out[116]:

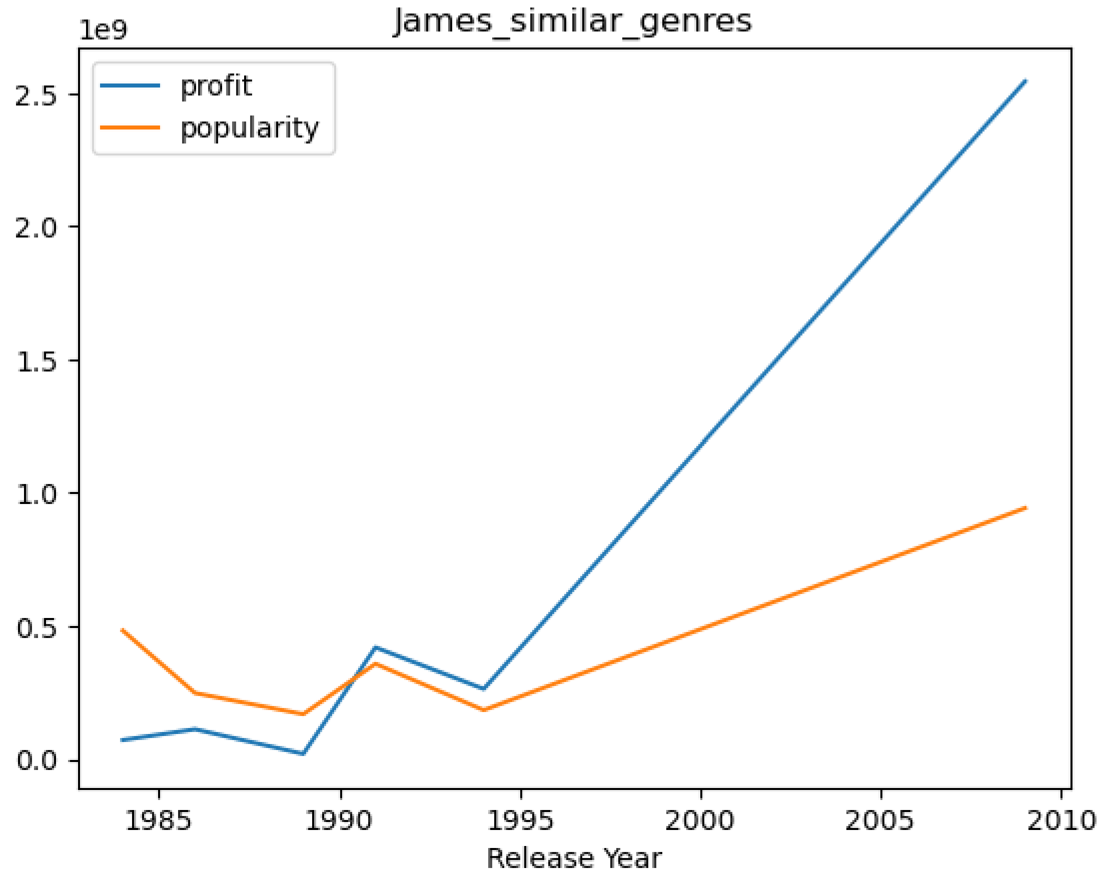
	release_year	popularity	budget	revenue	profit
count	6.000000	6.000000	6.000000e+00	6.000000e+00	6.000000e+00
mean	1992.166667	3.978147	9.115000e+07	6.633033e+08	5.721533e+08
std	8.975894	2.920742	8.345930e+07	1.052836e+09	9.772039e+08
min	1984.000000	1.691080	6.400000e+06	7.837120e+07	2.000010e+07
25%	1986.750000	2.003787	3.137500e+07	1.002651e+08	8.211846e+07
50%	1990.000000	3.034912	8.500000e+07	2.549713e+08	1.882213e+08
75%	1993.250000	4.520076	1.112500e+08	4.847206e+08	3.809706e+08
max	2009.000000	9.432768	2.370000e+08	2.781506e+09	2.544506e+09

In [117…

```
def plt_decision_s(decision_s, years_s, profit_s, popularity_s) :

    """
    this function to plot the profit and popularity over years for a specific director wirh similar genres

    """

    plt.plot(years_s, profit_s, label = 'profit')
    plt.plot(years_s, popularity_s, label = 'popularity')
    plt.xlabel('Release Year')
    plt.title(decision_s)
    plt.legend()
    plt.show()
```

In [118…

```
#recall the plot fuction


a = James_similar_genres['release_year']
b = James_similar_genres['profit']
c = James_similar_genres['popularity']*100000000 #the popularity value is too small compared to profit and should be magnified to

decision_aa = plt_decision_s( 'James_similar_genres', a, b, c )
```

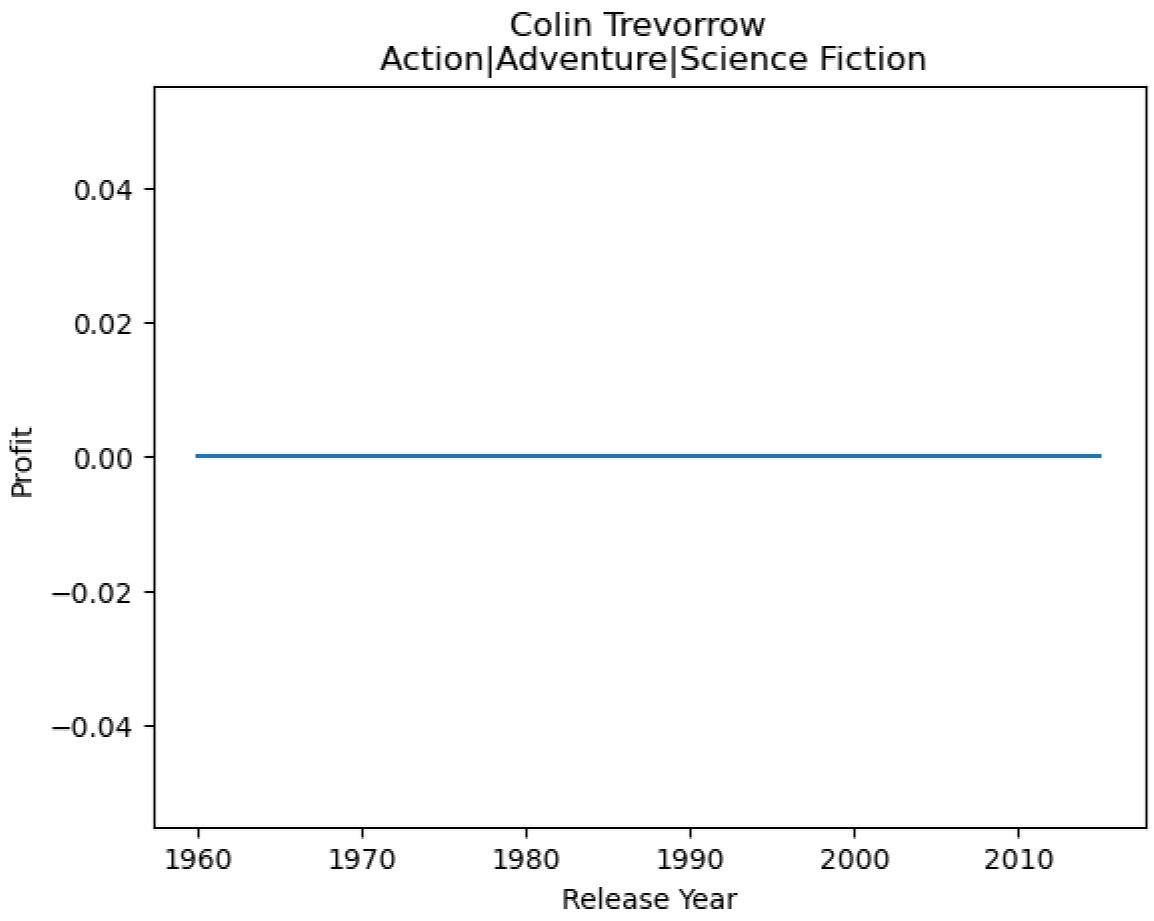


> James Cameron similar genres profit and popularity over years, Really Great History

In [121…

```
decision_a = decision('Colin Trevorrow', 'Action|Adventure|Science Fiction')
decision_aa = plt_decision( 'Colin Trevorrow  \n Action|Adventure|Science Fiction', decision_a)
```

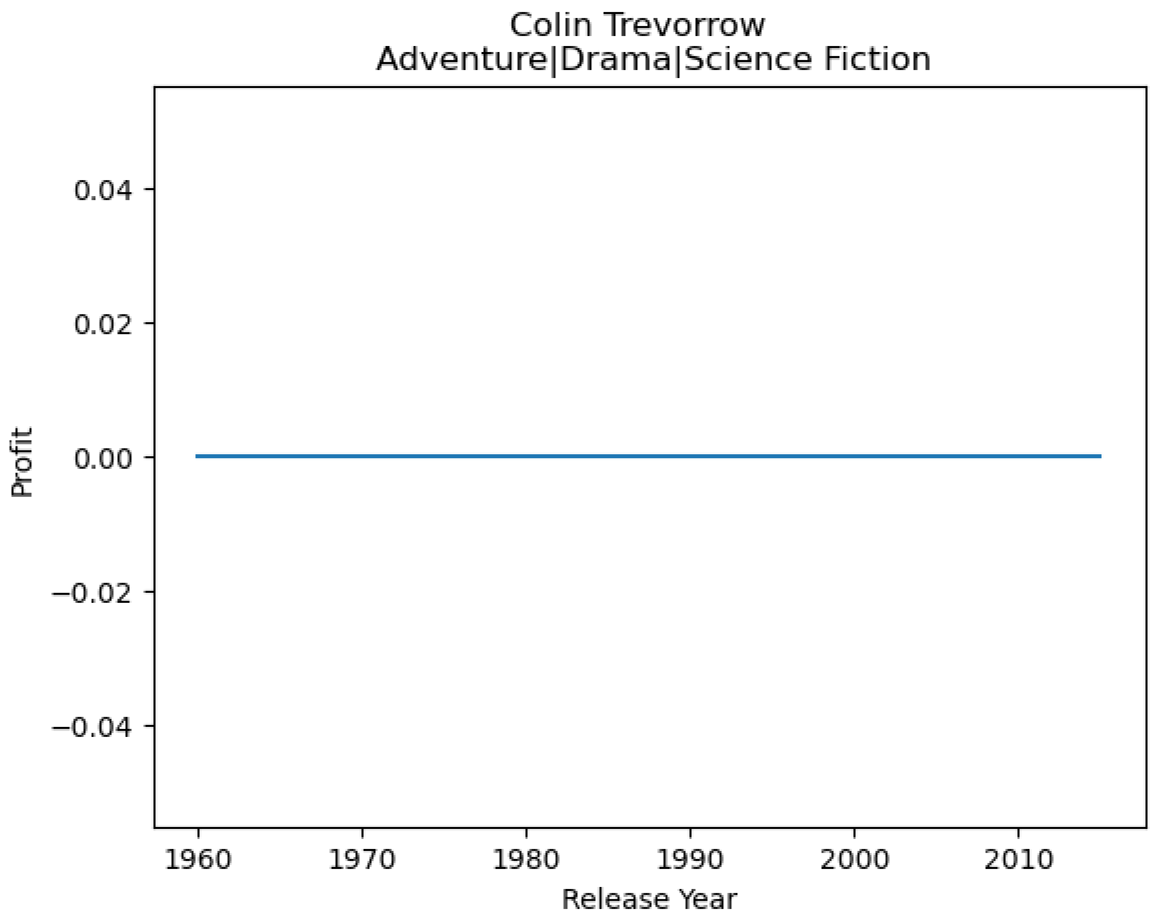
```
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```



In [122…

```
decision_a = decision('Colin Trevorrow', 'Adventure|Drama|Science Fiction')
decision_aa = plt_decision( 'Colin Trevorrow  \n Adventure|Drama|Science Fiction', decision_a)
```

```
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```



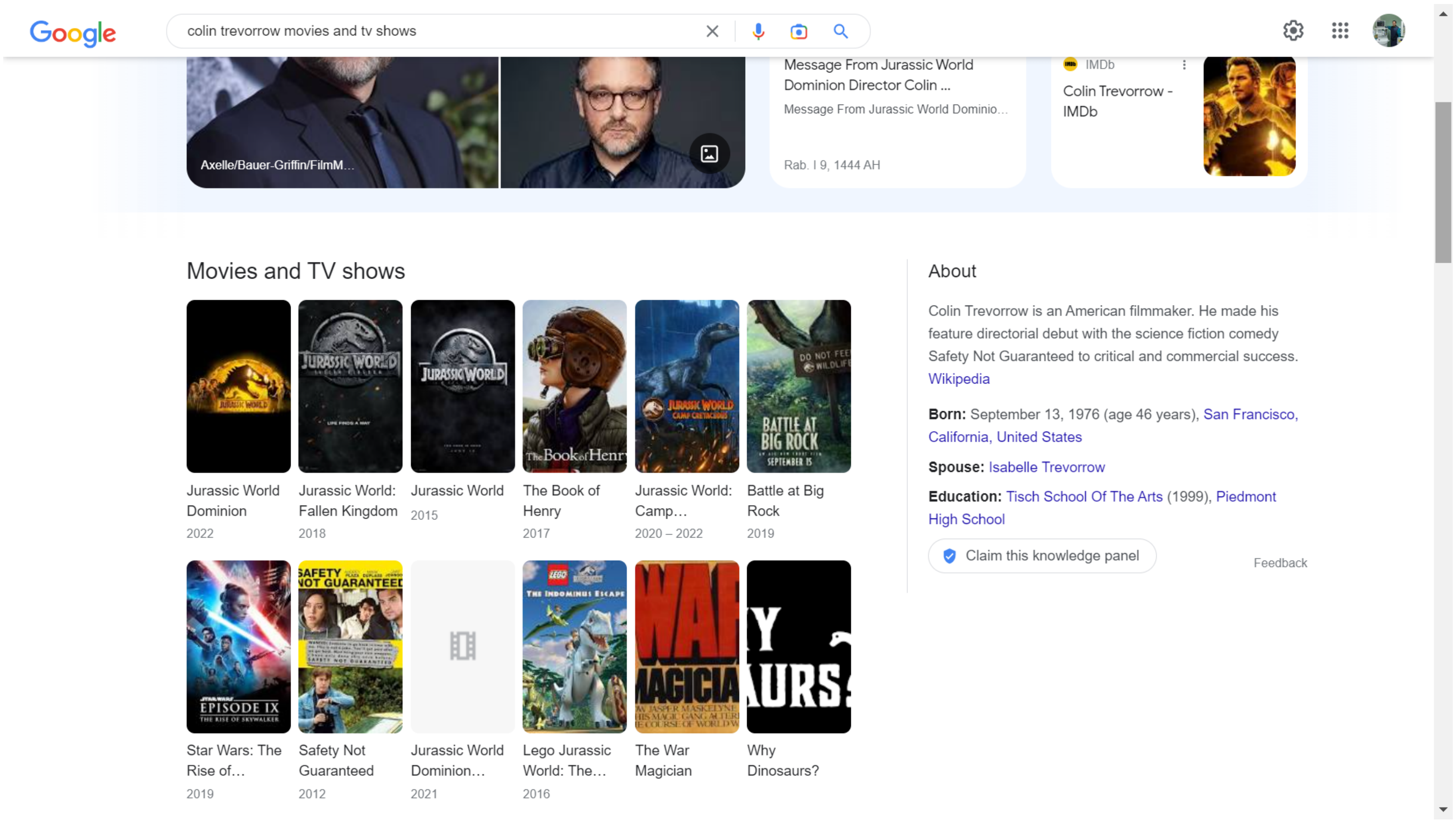
> Unfortunatily,Colin Trevorrow has no movies history of "Action|Adventure|Science Fiction" or "Adventure|Drama|Science Fiction" genre But let us check the genres of movies he had directed before we might find some similar genres

```
In [124…  my_data_two.loc[my_data_two['director'] == 'Colin Trevorrow']
```

Out[124]:

	release_year	genres	original_title	director	popularity	budget	revenue	profit
0	2015	Action|Adventure|Science Fiction|Thriller	Jurassic World	Colin Trevorrow	32.985763	150000000	1513528810	1363528810
4604	2012	Comedy|Romance|Science Fiction|Drama	Safety Not Guaranteed	Colin Trevorrow	0.408010	750000	4007792	3257792

> Unfortunately, we do not have a lot of data about Colin Trevorrow works but one of his works is "Jurassic World" which is one of the most popular and profitable movies over the whole history


colin trevorrow movies and tv shows

Axelle/Bauer-Griffin/FilmM…

Message From Jurassic World Dominion Director Colin ...

Message From Jurassic World Dominio…

Rab. I 9, 1444 AH

IMDb

Colin Trevorrow - IMDb

Movies and TV shows

Jurassic World Dominion	2022	
Jurassic World: Fallen Kingdom	2018	
Jurassic World	2015	
The Book of Henry	2017	
Jurassic World: Camp…	2020 – 2022	
Battle at Big Rock	2019	
Star Wars: The Rise of…	2019	
Safety Not Guaranteed	2012	
Jurassic World Dominion…	2021	
Lego Jurassic World: The…	2016	
The War Magician		
Why Dinosaurs?		

About

Colin Trevorrow is an American filmmaker. He made his feature directorial debut with the science fiction comedy Safety Not Guaranteed to critical and commercial success. Wikipedia

Born: September 13, 1976 (age 46 years), San Francisco, California, United States

Spouse: Isabelle Trevorrow

Education: Tisch School Of The Arts (1999), Piedmont High School

Claim this knowledge panel

Feedback

> Colin Trevorrow is one of the greatest directors ever but unfortunately the data we have - specially atfer cleaning - does not contain most of his works

Conclusions

The purpose of this business study was to make a data-driven decision of which genre of movies should i invest my money in and who is the best movie director should i invest my money with and after investigating TMDb i found the following:

> For the movie director i would select one of the following:
>
> 1. James Cameron (regarding to profit, he is one of the highest profits directors and is more active over years)
> 2. Colin Trevorrow (regarding to popularity he is one of the most popular ever)
>
> For the movie genre i would select one of the following:
>
> 1. Action|Adventure|Science Fiction (regarding to profit, this genre is one of the highest profits directors and is more active over years)
> 2. Adventure|Drama|Science Fiction (regarding to popularity this genre is one of the most popular ever)

More analysis

> in the future we can consider also the "run time" of the movie to find out which is more profitable the short or long films

---

limitations

---

1. 65% of the raw data was cleaned and only 35% of the data was valid for the analysis which may lead to inaccurate or biased resultes

2. I googled some data from TMDb and unfortunatily it was not accurate enough to make money decisions

> Unfortunaltly niether James Cameron or Colin Trevorrow have a previous works of these exact genres "Action|Adventure|Science Fiction" and "Adventure|Drama|Science Fiction" but they had great works with similar genres
> ***however I think that a movie directed by both "James Cameron and Colin Trevorrow" of the genre "Action|Adventure|Drama|Science Fiction" would be a great investment opportunity***

References

> https://pydata.org/ || https://stackoverflow.com/ || https://www.geeksforgeeks.org/|| https://medium.com/