Final Project Submission

- Student name: Lucy Chepkemoi Ruto
- Student pace: Part-time
- Scheduled project review date/time: 5th November 2023 at 8:00 P.M
- Instructor name: Everlyn Asiko/ Samwel Jane/ Samuel Mwangi/ Mildred Jepkosgei/ Veronica Isiaho

Microsoft Coorporation movie

Overview

This project explores the types of films that are currently doing the well at the box office and needs of the Microsoft Corporation (https://www.microsoft.com/) which wants to set-up a new movie studio. The Microsoft corporation can use these insights to decide what type of films to create.

Business Problem

Microsoft sees all the big companies creating original video content and they want to get in on the fun. They have decided to create a new movie studio, but they don’t know anything about creating movies. You are charged with exploring what types of films are currently doing the best at the box office. You must then translate those findings into actionable insights that the head of Microsoft's new movie studio can use to help decide what type of films to create.

In [1]:
```
# Importing Standard packages
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import prep as p
import datetime
import seaborn as sns
```

In [2]:
```
#Loading datasets required for analysis
title_basics = pd.read_csv('./Data/title.basics.csv')
movie_budgets = pd.read_csv('./Data/tn.movie_budgets.csv')
title_crew = pd.read_csv('./Data/title.crew.csv')
title_ratings = pd.read_csv('./Data/title.ratings.csv')
name_basics = pd.read_csv('./Data/name.basics.csv')
```

Data Exploration and Cleaning

Cleaning and exploring title_basics dataframe

In [3]:
```
title_basics.shape
```

Out[3]:
```
(146144, 6)
```

In [4]:
```
title_basics.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 146144 entries, 0 to 146143
Data columns (total 6 columns):
 #   Column           Non-Null Count   Dtype  
---  ------           --------------   -----  
 0   tconst           146144 non-null  object 
 1   primary_title    146144 non-null  object 
 2   original_title   146123 non-null  object 
 3   start_year       146144 non-null  int64  
 4   runtime_minutes  114405 non-null  float64
 5   genres           140736 non-null  object 
dtypes: float64(1), int64(1), object(4)
memory usage: 6.7+ MB
```

In [5]:
```
title_basics.duplicated(subset='tconst').value_counts()
```

Out[5]:
```
False    146144
dtype: int64
```

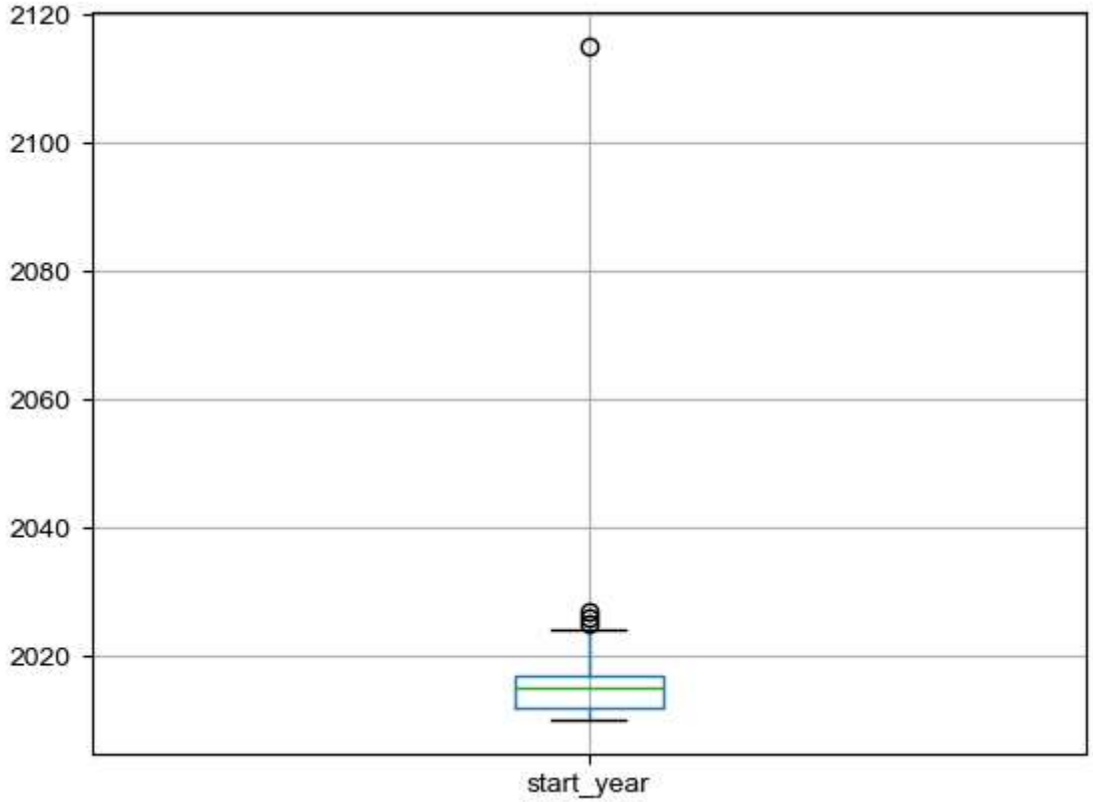
In [6]:
```
title_basics.sort_values(by="start_year", axis=0, ascending=False, inplace=True)
title_basics.nlargest(40,columns=('start_year'))
```

Out[6]:

	tconst	primary_title	original_title	start_year	runtime_minutes	genres
89506	tt5174640	100 Years	100 Years	2115	NaN	Drama
96592	tt5637536	Avatar 5	Avatar 5	2027	NaN	Action,Adventure,Fantasy
2949	tt10300398	Untitled Star Wars Film	Untitled Star Wars Film	2026	NaN	Fantasy
52213	tt3095356	Avatar 4	Avatar 4	2025	NaN	Action,Adventure,Fantasy
105187	tt6149054	Fantastic Beasts and Where to Find Them 5	Fantastic Beasts and Where to Find Them 5	2024	NaN	Adventure,Family,Fantasy
2948	tt10300396	Untitled Star Wars Film	Untitled Star Wars Film	2024	NaN	NaN
2483	tt10255736	Untitled Marvel Project	Untitled Marvel Project	2023	NaN	Action
16337	tt1757678	Avatar 3	Avatar 3	2023	NaN	Action,Adventure,Drama
2906	tt10298848	Untitled Disney Live-Action Project	Untitled Disney Live-Action Project	2023	NaN	NaN
106865	tt6258542	Wraith of the Umbra and Eidolon II	Wraith of the Umbra and Eidolon II	2023	NaN	Adventure,Drama,Fantasy
111226	tt6495056	Untitled Illumination Entertainment Project	Untitled Illumination Entertainment Project	2023	NaN	NaN
821	tt10042446	Untitled Disney Marvel Film	Untitled Disney Marvel Film	2022	NaN	Action
2765	tt10288908	The Weary Traveler	The Weary Traveler	2022	NaN	Horror
2947	tt10300394	Untitled Star Wars Film	Untitled Star Wars Film	2022	NaN	NaN
130618	tt8097030	Untitled Pixar Animation Project	Untitled Pixar Animation Project	2022	NaN	Animation
4382	tt10407798	Nepotism Kingdom	Nepotism Kingdom	2022	120.0	Action,Drama,Thriller
105186	tt6149052	Fantastic Beasts and Where to Find Them 4	Fantastic Beasts and Where to Find Them 4	2022	NaN	Adventure,Family,Fantasy
2905	tt10298840	Untitled Disney Animation Project	Untitled Disney Animation Project	2022	NaN	NaN
129337	tt7974630	Untitled Star Wars Project	Untitled Star Wars Project	2022	NaN	Sci-Fi
130616	tt8097016	Untitled Disney Marvel Film	Untitled Disney Marvel Film	2022	NaN	Action
135475	tt8582042	Lost in Time	Lost in Time	2022	NaN	NaN
112960	tt6637082	Untitled Illumination Entertainment Project	Untitled Illumination Entertainment Project	2022	NaN	Animation

	tconst	primary_title	original_title	start_year	runtime_minutes	genres
134557	tt8483834	Rasa Krida	Rasa Krida	2022	89.0	Documentary
113854	tt6718170	Untitled Illumination Entertainment Project	Untitled Illumination Entertainment Project	2022	NaN	Animation
1330	tt10108854	The Hunchback of the Lighthouse	The Hunchback of the Lighthouse	2022	120.0	Drama
5340	tt10462154	La plus précieuse des marchandises	La plus précieuse des marchandises	2022	NaN	Animation
82608	tt4766078	Girl of Ashima	Girl of Ashima	2022	NaN	Animation
3008	tt10303934	Mi Asesino Favorito	Mi Asesino Favorito	2022	NaN	Comedy,Crime
3014	tt10304134	Untitled Disney Live Action Project	Untitled Disney Live Action Project	2022	NaN	NaN
3015	tt10304138	Untitled Disney Live Action Project	Untitled Disney Live Action Project	2022	NaN	NaN
3016	tt10304140	Untitled Disney Live Action Project	Untitled Disney Live Action Project	2022	NaN	NaN
3017	tt10304142	Untitled Disney Live Action Project	Untitled Disney Live Action Project	2022	NaN	NaN
3018	tt10304194	Untitled Disney Live Action Project	Untitled Disney Live Action Project	2022	NaN	NaN
2904	tt10298810	Untitled Pixar Animation Project	Untitled Pixar Animation Project	2022	NaN	NaN
144423	tt9663764	Aquaman 2	Aquaman 2	2022	NaN	Action,Sci-Fi
61851	tt3566834	Minecraft the First Movie	Minecraft	2022	NaN	Action,Adventure,Documentary
2465	tt10251718	Corazones en Llamas 5	Corazones en Llamas 5	2022	NaN	Action
4903	tt10442852	Triple OG	Triple OG	2022	NaN	Action
138515	tt8912936	DC Super Pets	Super Pets	2022	NaN	Action,Animation,Comedy
4451	tt10411090	Drain Baby	Drain Baby	2022	NaN	Comedy,Drama,Fantasy

In [7]:
```
boxplot = title_basics.boxplot(column=['start_year'])
sns.set_style('darkgrid')
sns.set_context('talk')
```



In [8]:
```
#Removing outlier years in the title_basics DataFrame
clean_title_basics = title_basics.loc[title_basics['start_year'] <= 2023]
clean_title_basics
```

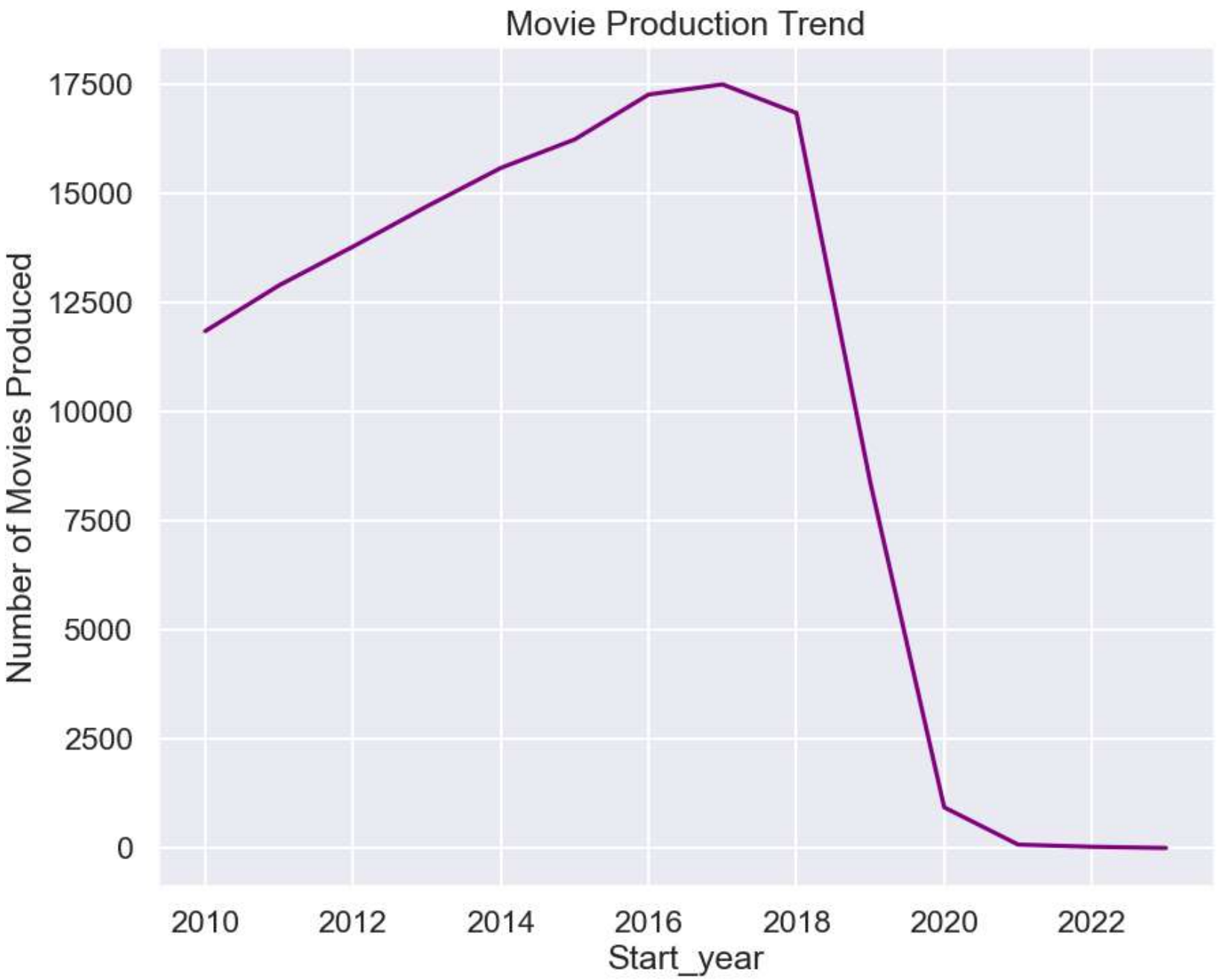
Out[8]:

	tconst	primary_title	original_title	start_year	runtime_minutes	genres
2483	tt10255736	Untitled Marvel Project	Untitled Marvel Project	2023	NaN	Action
16337	tt1757678	Avatar 3	Avatar 3	2023	NaN	Action,Adventure,Drama
2906	tt10298848	Untitled Disney Live-Action Project	Untitled Disney Live-Action Project	2023	NaN	NaN
106865	tt6258542	Wraith of the Umbra and Eidolon II	Wraith of the Umbra and Eidolon II	2023	NaN	Adventure,Drama,Fantasy
111226	tt6495056	Untitled Illumination Entertainment Project	Untitled Illumination Entertainment Project	2023	NaN	NaN
...	...	...	...	...	...	...
74712	tt4264626	Civil War Life: Shot to Pieces	Civil War Life: Shot to Pieces	2010	79.0	Documentary
14471	tt1716746	Heinrich Kieber - Datendieb	Heinrich Kieber - Datendieb	2010	52.0	Documentary
74692	tt4263706	Mushrooms of America	Mushrooms of America	2010	46.0	Adventure,Comedy,Documentary
118065	tt7059624	Zamana	Zamana	2010	140.0	Drama
94000	tt5475580	A Boy and A Girl	A Boy and A Girl	2010	NaN	Romance

146138 rows × 6 columns

In [9]:
```
clean_title_basics.groupby(['start_year'])['primary_title'].count().plot(kind='line',color='purple', figsize
plt.title('Movie Production Trend')
plt.ylabel('Number of Movies Produced')
plt.xlabel('Start year')
```

Out[9]:
```
Text(0.5, 0, 'Start_year')
```


Movie Production Trend
Number of Movies Produced
17500
15000
12500
10000
7500
5000
2500
0
2010
2012
2014
2016
2018
2020
2022
Start_year

Cleaning and exploring title_crew dataframe

In [10]:
```
title_crew.shape
```

Out[10]:
```
(146144, 3)
```

In [11]:
```
title_crew.head(10)
```

Out[11]:

	tconst	directors	writers
0	tt0285252	nm0899854	nm0899854
1	tt0438973	NaN	nm0175726,nm1802864
2	tt0462036	nm1940585	nm1940585
3	tt0835418	nm0151540	nm0310087,nm0841532
4	tt0878654	nm0089502,nm2291498,nm2292011	nm0284943
5	tt0879859	nm2416460	NaN
6	tt0996958	nm2286991	nm2286991,nm2651190
7	tt0999913	nm0527109	nm0527109,nm0329051,nm0001603,nm0930684
8	tt10003792	nm10539228	nm10539228
9	tt10005130	nm10540239	nm5482263,nm10540239

In [12]:
```
title_crew.duplicated(subset='tconst').value_counts()
```

Out[12]:
```
False    146144
dtype: int64
```

Merging title_basics with title_crew

After cleaning both dataframes, the two datasets are merged to create basics_crew dataframe

In [13]:
```
basics_crew = clean_title_basics.set_index('tconst').join(title_crew.set_index('tconst'), how ='left')
basics_crew.isna().sum()
```

Out[13]:
```
primary_title          0
original_title        21
start_year             0
runtime_minutes    31733
genres              5407
directors           5725
writers            35881
dtype: int64
```

Cleaning and exploring basics_crew dataframe

In [14]:
```
basics_crew.drop(columns=['writers','original_title','runtime_minutes'],axis =1, inplace=True)
```

In [15]:
```
basics_crew.rename(columns={'directors':'nconst'}, inplace=True)
```

In [16]:
```
basics_crew.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 146138 entries, tt10255736 to tt5475580
Data columns (total 4 columns):
 #   Column         Non-Null Count   Dtype 
---  ------         --------------   ----- 
 0   primary_title  146138 non-null  object
 1   start_year     146138 non-null  int64 
 2   genres         140731 non-null  object
 3   nconst         140413 non-null  object
dtypes: int64(1), object(3)
memory usage: 9.6+ MB
```

In [17]:
```
basics_crew.reset_index(inplace=True)
basics_crew.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 146138 entries, 0 to 146137
Data columns (total 5 columns):
 #   Column         Non-Null Count   Dtype 
---  ------         --------------   ----- 
 0   tconst         146138 non-null  object
 1   primary_title  146138 non-null  object
 2   start_year     146138 non-null  int64 
 3   genres         140731 non-null  object
 4   nconst         140413 non-null  object
dtypes: int64(1), object(4)
memory usage: 5.6+ MB
```

Cleaning and exploring name_basics dataframe

In [18]:
```
name_basics.shape
```

Out[18]: (606648, 6)

In [19]:
```
name_basics.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 606648 entries, 0 to 606647
Data columns (total 6 columns):
 #   Column              Non-Null Count   Dtype  
---  ------              --------------   -----  
 0   nconst              606648 non-null  object 
 1   primary_name        606648 non-null  object 
 2   birth_year          82736 non-null   float64
 3   death_year          6783 non-null    float64
 4   primary_profession  555308 non-null  object 
 5   known_for_titles    576444 non-null  object 
dtypes: float64(2), object(4)
memory usage: 27.8+ MB
```

In [20]:
```
name_basics.duplicated(subset='nconst').value_counts()
```

Out[20]:
```
False    606648
dtype: int64
```

In [21]:
```
name_basics.drop(columns=['primary_profession','known_for_titles','birth_year'],axis =1, inplace=True)
```

Merging basics_crew dataframe with name_basics dataframe

In [22]:
```
basics_crew_names = basics_crew.set_index('nconst').join(name_basics.set_index('nconst'), how='inner')
basics_crew_names.sample(20)
```

Out[22]:

nconst	tconst	primary_title	start_year	genres	primary_name	death_year
nm2117510	tt3486626	The Nut Job 2: Nutty by Nature	2017	Adventure,Animation,Comedy	Cal Brunker	NaN
nm5697340	tt3891162	My Amish World	2017	Drama	Sam Wickey	NaN
nm0895321	tt1706362	Et raadhus til alvor og fest	2010	Documentary	Nils Vest	NaN
nm4730513	tt4160706	Live Cargo	2016	Drama,Thriller	Logan Sandler	NaN
nm8481709	tt6112332	Awakening	2016	NaN	Yasufumi Minowa	NaN
nm6093409	tt3338526	I Don't Touch the Gold	2014	Adventure,Documentary	Paulina Pisarek	NaN
nm2821155	tt7488602	Ayaz	2017	Drama	Dersu Yavuz Altun	NaN
nm2385664	tt1671509	The Darkest Corner of Paradise	2010	Drama,Thriller	Henry Weintraub	NaN
nm8701955	tt6414526	Hanuman the Immortal 2	2011	Animation	Prakasan U.K.	NaN
nm0541391	tt5791442	Bikini Moon	2017	Drama	Milcho Manchevski	NaN
nm3499832	tt8176370	El secreto de los árboles	2016	History	Helena Sánchez	NaN
nm3142190	tt1296161	Jen	2010	Drama	Julian Allder	NaN
nm9524640	tt7817224	33 Yillik Direnis: Berfo Ana	2013	Documentary	Veysi Altay	NaN
nm0591472	tt2543826	The Long Island Railroad Massacre: 20 Years Later	2013	Documentary	Charlie Minn	NaN
nm6474835	tt3711604	Ángel Llorca: El último ensayo	2011	Documentary	Víctor M. Guerra	NaN
nm4065693	tt1836751	Bist roozi ké Tehran ra tékan dad	2010	Documentary,History,News	Ali Razi	NaN
nm4242247	tt4896002	Lost History	2013	Drama	Hussain Sewdin	NaN
nm0217219	tt4040888	Chorus	2015	Drama	François Delisle	NaN
nm4334978	tt6075948	Maria per Roma	2016	Comedy	Karen di Porto	NaN
nm0249928	tt3114378	Just Like Yesterday: Tony Christie	2013	Biography	David Innes Edwards	NaN

In [23]:
```
basics_crew_names.duplicated(subset='tconst').value_counts()
```

Out[23]:
```
False    124685
dtype: int64
```

In [24]:
```
#Identify which directors are still alive
alive_directors=basics_crew_names.loc[basics_crew_names['death_year'].isnull()]
```

In [25]:
```
#Filter out directors with less than 30 moies directed
alive_directors = alive_directors.groupby('primary_name').filter(lambda x : len(x)>30)
```

In [26]:
```
#Directors with the highest directed movies
alive_directors['primary_name'].value_counts()
```

Out[26]:
```
Omer Pasha             62
Stephan Düfel          48
Rajiv Chilaka          48
Larry Rosen            47
Gérard Courant         44
Claudio Costa          42
Nayato Fio Nuala       39
Graeme Duane           38
Eckhart Schmidt        37
Tetsuya Takehora       33
Michael Fredianelli    32
Name: primary_name, dtype: int64
```

In [27]:
```
title_ratings.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 73856 entries, 0 to 73855
Data columns (total 3 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   tconst         73856 non-null  object
 1   averagerating  73856 non-null  float64
 2   numvotes       73856 non-null  int64
dtypes: float64(1), int64(1), object(1)
memory usage: 1.7+ MB
```

Merging basics_crew_names dataframe with title_ratings dataframe

In [28]:
```
basics_crew_names.reset_index(inplace=True)
imdb_data = basics_crew_names.set_index('tconst').join(title_ratings.set_index('tconst'), how='left')
imdb_data.shape
```

Out[28]: (124685, 8)

In [29]:
```
imdb_data
```

Out[29]:

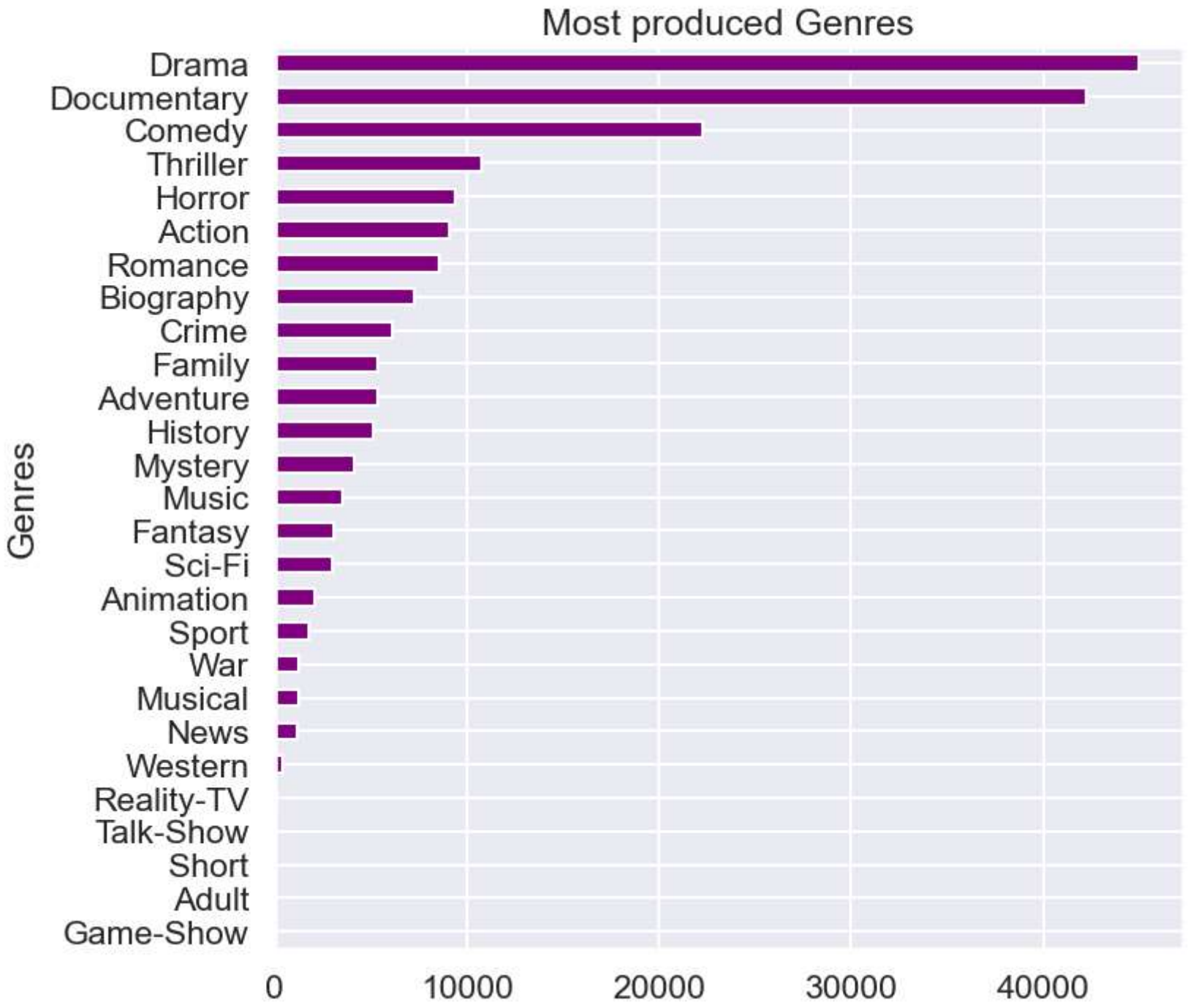
tconst	nconst	primary_title	start_year	genres	primary_name	death_year	averagerating	numvotes
tt0069049	nm0000080	The Other Side of the Wind	2018	Drama	Orson Welles	1985.0	6.9	4517.0
tt7139936	nm0000095	A Rainy Day in New York	2019	Comedy	Woody Allen	NaN	NaN	NaN
tt5825380	nm0000095	Wonder Wheel	2017	Drama	Woody Allen	NaN	6.2	19782.0
tt4513674	nm0000095	Café Society	2016	Comedy,Drama,Romance	Woody Allen	NaN	6.6	62466.0
tt3715320	nm0000095	Irrational Man	2015	Comedy,Drama	Woody Allen	NaN	6.6	50303.0
...	...	...	...	...	...	...	...	...
tt8742502	nm9992849	Jai Jagadeka Veera	2015	Drama	P.S. Ramu	NaN	NaN	NaN
tt8742566	nm9992900	White Spot	2018	Adventure,Documentary,Drama	Tin Brendel	NaN	NaN	NaN
tt8742574	nm9992905	Merata: How Mum Decolonised the Screen	2018	Biography,Documentary	Hepi Mita	NaN	7.3	36.0
tt8742576	nm9992906	A Room	2018	Crime,Drama,Fantasy	Dingli Diao	NaN	NaN	NaN
tt10299418	nm9993573	No Perfect Love	2019	Drama,Romance	Lakisha Louissaint	NaN	NaN	NaN

124685 rows × 8 columns

In [30]:
```
# Creating a dataframe from imbd_data dataframe without null genre values and counting
#the instances which a movie was categorized as a certain genre
imdb_genres = imdb_data[imdb_data["genres"].notnull()].copy()
imdb_genres["genres"] = imdb_genres["genres"].str.split(",")
split_genres = imdb_genres.explode("genres")["genres"].value_counts()
split_genres.sort_values(axis=0, ascending=True, inplace=True)
```

In [31]:
```
split_genres.plot(kind='barh',color='purple', figsize=(8,8))
plt.title('Most produced Genres')
plt.ylabel('Genres')
```

Out[31]:
```
Text(0, 0.5, 'Genres')
```


Most produced Genres
Genres
Drama
Documentary
Comedy
Thriller
Horror
Action
Romance
Biography
Crime
Family
Adventure
History
Mystery
Music
Fantasy
Sci-Fi
Animation
Sport
War
Musical
News
Western
Reality-TV
Talk-Show
Short
Adult
Game-Show
0
10000
20000
30000
40000

From the analysis of the IMDb datasets:Drama, Documentary and Comedy appear to be the genres frequently produced by IMDb.

Data Cleaning and exploration of movie_budgets dataframe

In [32]:
```
movie_budgets.drop(columns = ['id'], axis=1, inplace=True)
```

In [33]:
```
for column in ['production_budget', 'domestic_gross', 'worldwide_gross']:
    movie_budgets[column] = movie_budgets[column].str.replace('$','', regex=False)
    movie_budgets[column] = movie_budgets[column].str.replace(',','', regex=False)
    movie_budgets[column] = movie_budgets[column].astype('int64')
movie_budgets.head(10)
```

Out[33]:

	release_date	movie	production_budget	domestic_gross	worldwide_gross
0	Dec 18, 2009	Avatar	425000000	760507625	2776345279
1	May 20, 2011	Pirates of the Caribbean: On Stranger Tides	410600000	241063875	1045663875
2	Jun 7, 2019	Dark Phoenix	350000000	42762350	149762350
3	May 1, 2015	Avengers: Age of Ultron	330600000	459005868	1403013963
4	Dec 15, 2017	Star Wars Ep. VIII: The Last Jedi	317000000	620181382	1316721747
5	Dec 18, 2015	Star Wars Ep. VII: The Force Awakens	306000000	936662225	2053311220
6	Apr 27, 2018	Avengers: Infinity War	300000000	678815482	2048134200
7	May 24, 2007	Pirates of the Caribbean: At Worldâ□□s End	300000000	309420425	963420425
8	Nov 17, 2017	Justice League	300000000	229024295	655945209
9	Nov 6, 2015	Spectre	300000000	200074175	879620923

Feature Engineering

Creating profits column and Return on Investment (ROI) from the: production_budget and world_wide gross columns .

In [34]:
```
movie_budgets['profit'] = movie_budgets['worldwide_gross'] - movie_budgets['production_budget']
movie_budgets['ROI'] = movie_budgets['profit'] / movie_budgets['production_budget']
movie_budgets['profit_status'] = np.where(movie_budgets['ROI'] > 0, "Profit", "Loss")
movie_budgets['profitable_oversees'] = movie_budgets['worldwide_gross'] - movie_budgets['domestic_gross']
movie_budgets.sample(10)
```

Out[34]:

	release_date	movie	production_budget	domestic_gross	worldwide_gross	profit	ROI	profit_status	profitable_
5149	Aug 31, 1984	C.H.U.D.	1250000	4700000	4700000	3450000	2.760000	Profit	
2795	Dec 13, 1989	Glory	18000000	26593580	26593580	8593580	0.477421	Profit	
2445	Jan 1, 1970	Darling Lili	22000000	5000000	5000000	-17000000	-0.772727	Loss	
4698	May 24, 2013	Before Midnight	3000000	8110621	23251930	20251930	6.750643	Profit	
4332	Aug 14, 1998	Slums of Beverly Hills	5000000	5502773	5502773	502773	0.100555	Profit	
3307	Aug 14, 2015	Brothers	13000000	656688	17856688	4856688	0.373591	Profit	
1832	Jan 22, 2010	Extraordinary Measures	31000000	12482741	15826984	-15173016	-0.489452	Loss	
3521	Jun 4, 1982	Poltergeist	10700000	74706019	121706019	111006019	10.374394	Profit	
2279	Apr 15, 2011	The Conspirator	25000000	11538204	15907411	-9092589	-0.363704	Loss	
542	Jun 18, 1992	Batman Returns	80000000	162833635	266824291	186824291	2.335304	Profit	1

In [35]:
```
duplicates = movie_budgets[movie_budgets[['movie', 'profit']].duplicated()]
duplicates
```

Out[35]:

	release_date	movie	production_budget	domestic_gross	worldwide_gross	profit	ROI	profit_status	profitable_oversees

In [36]:
```
movie_budgets.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5782 entries, 0 to 5781
Data columns (total 9 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   release_date         5782 non-null   object 
 1   movie                5782 non-null   object 
 2   production_budget    5782 non-null   int64  
 3   domestic_gross       5782 non-null   int64  
 4   worldwide_gross      5782 non-null   int64  
 5   profit               5782 non-null   int64  
 6   ROI                  5782 non-null   float64
 7   profit_status        5782 non-null   object 
 8   profitable_oversees  5782 non-null   int64  
dtypes: float64(1), int64(5), object(3)
memory usage: 406.7+ KB
```

In [37]:
```
movie_budgets['release_date'] = pd.to_datetime(movie_budgets['release_date'])
```

In [38]:
```
movie_budgets['release_day_of_week'] = movie_budgets['release_date'].dt.day_name()
movie_budgets['release_month'] = movie_budgets['release_date'].dt.month_name()
movie_budgets.sample(30)
```

Out[38]:

	release_date	movie	production_budget	domestic_gross	worldwide_gross	profit	ROI	profit_status	profitable_
358	2009-07-01	Public Enemies	102500000	97104620	212282709	109782709	1.071051	Profit	1
173	2011-05-26	Kung Fu Panda 2	150000000	165249063	664837547	514837547	3.432250	Profit	4
2367	1980-05-21	Star Wars Ep. V: The Empire Strikes Back	23000000	290271960	534161334	511161334	22.224406	Profit	2
425	2003-12-17	The Lord of the Rings: The Return of the King	94000000	377845905	1141403341	1047403341	11.142589	Profit	7
4219	2018-09-28	Hell Fest	5500000	11107431	18119231	12619231	2.294406	Profit	
3715	2007-05-18	Severance	10000000	137221	5950002	-4049998	-0.405000	Loss	
4144	1989-10-13	Halloween 5: The Revenge of Michael Myers	6000000	11642254	11642254	5642254	0.940376	Profit	
249	2010-07-23	Salt	130000000	118311368	290650494	160650494	1.235773	Profit	1
4814	2004-07-28	Garden State	2500000	26782316	36028802	33528802	13.411521	Profit	
228	2011-03-04	Rango	135000000	123477607	245724600	110724600	0.820182	Profit	1
2827	2008-06-06	Mongol	18000000	5705761	27147349	9147349	0.508186	Profit	
5175	2012-06-01	Pink Ribbons, Inc.	1200000	26608	26608	-1173392	-0.977827	Loss	
891	1996-03-01	Up Close & Personal	60000000	51045801	100645801	40645801	0.677430	Profit	
939	2009-01-23	Inkheart	60000000	17303424	66655938	6655938	0.110932	Profit	
1763	2018-08-03	The Darkest Minds	34000000	12695691	38361428	4361428	0.128277	Profit	

	release_date	movie	production_budget	domestic_gross	worldwide_gross	profit	ROI	profit_status	profitable_
18	2012-12-14	The Hobbit: An Unexpected Journey	250000000	303003568	1017003568	767003568	3.068014	Profit	7
205	2003-07-09	Pirates of the Caribbean: The Curse of the Bla...	140000000	305411224	634954103	494954103	3.535386	Profit	3
2609	2002-07-26	The Country Bears	20000000	16988996	16988996	-3011004	-0.150550	Loss	
5689	2014-06-19	The Past is a Grotesque Animal	100000	20056	20056	-79944	-0.799440	Loss	
2303	2005-04-22	King's Ransom	25000000	4008527	4049527	-20950473	-0.838019	Loss	
913	2009-12-11	Invictus	60000000	37491364	124514011	64514011	1.075234	Profit	
4161	2006-12-21	Venus	6000000	3347411	7818479	1818479	0.303080	Profit	
4031	1992-05-01	Split Second	7000000	5430822	5430822	-1569178	-0.224168	Loss	
181	2004-05-14	Troy	150000000	133298577	484161265	334161265	2.227742	Profit	3
5022	2002-11-15	El crimen de padre Amaro	1800000	5719000	5719000	3919000	2.177222	Profit	
4848	2016-09-02	Antibirth	2500000	0	0	-2500000	-1.000000	Loss	
563	2010-01-15	The Book of Eli	80000000	94835059	158750817	78750817	0.984385	Profit	
1714	2007-12-21	Walk Hard: The Dewey Cox Story	35000000	18317151	20606053	-14393947	-0.411256	Loss	
1072	1998-12-23	The Thin Red Line	52000000	36400491	97709034	45709034	0.879020	Profit	

	release_date	movie	production_budget	domestic_gross	worldwide_gross	profit	ROI	profit_status	profitable_
1151	2005-06-10	The Adventures of Sharkboy and Lavagirl in 3-D	50000000	39177684	69425966	19425966	0.388519	Profit	

In [39]:
```
movie_budgets = movie_budgets[['movie', 'release_date','release_day_of_week','release_month','worldwide_gros
```

In [40]:
```
movie_budgets.sort_values(by="profit", axis=0, ascending=False, inplace=True)
```

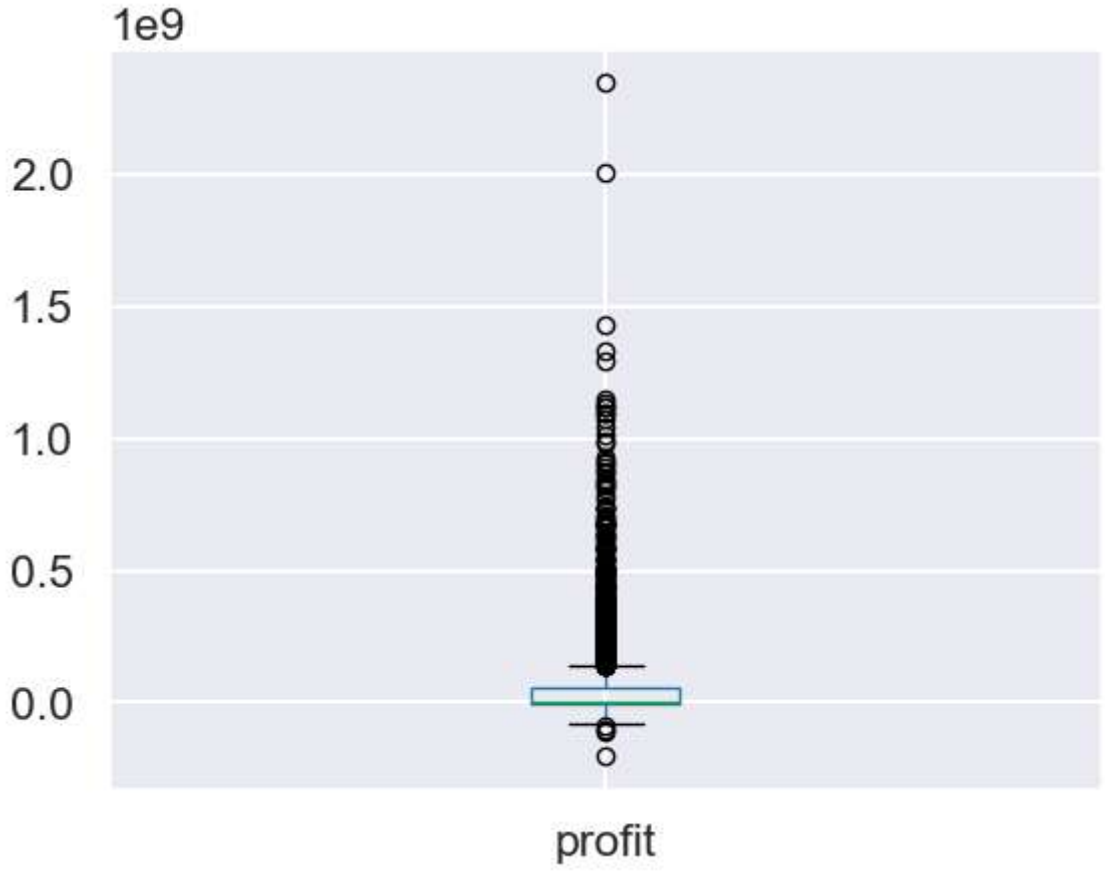
Merging imdb_data dataframe with movie_budgets

In [41]:
```
imdb_data.reset_index(inplace=True)
imdb_data.rename(columns={'primary_title':'movie'}, inplace=True)
```

In [42]:
```
combined_datasets = imdb_data.set_index("movie").join(movie_budgets.set_index("movie"), how="inner")
```

In [43]:
```
combined_datasets.sort_values(by='ROI',axis=0, ascending=False, inplace=True)
```

In [44]:
```
boxplot_ROI = combined_datasets.boxplot(column=['profit'])
```



In [45]:
```
Q1 = combined_datasets['ROI'].quantile(0.25)
Q3 = combined_datasets['ROI'].quantile(0.75)
IQR = Q3 - Q1

# identify outliers
threshold = 1.5
outliers = combined_datasets[(combined_datasets['ROI'] < Q1 - threshold * IQR) |
                             (combined_datasets['ROI'] > Q3 + threshold * IQR)]
```

In [46]:
```
clean_combined_datasets = combined_datasets.drop(outliers.index)
clean_combined_datasets
```

Out[46]:

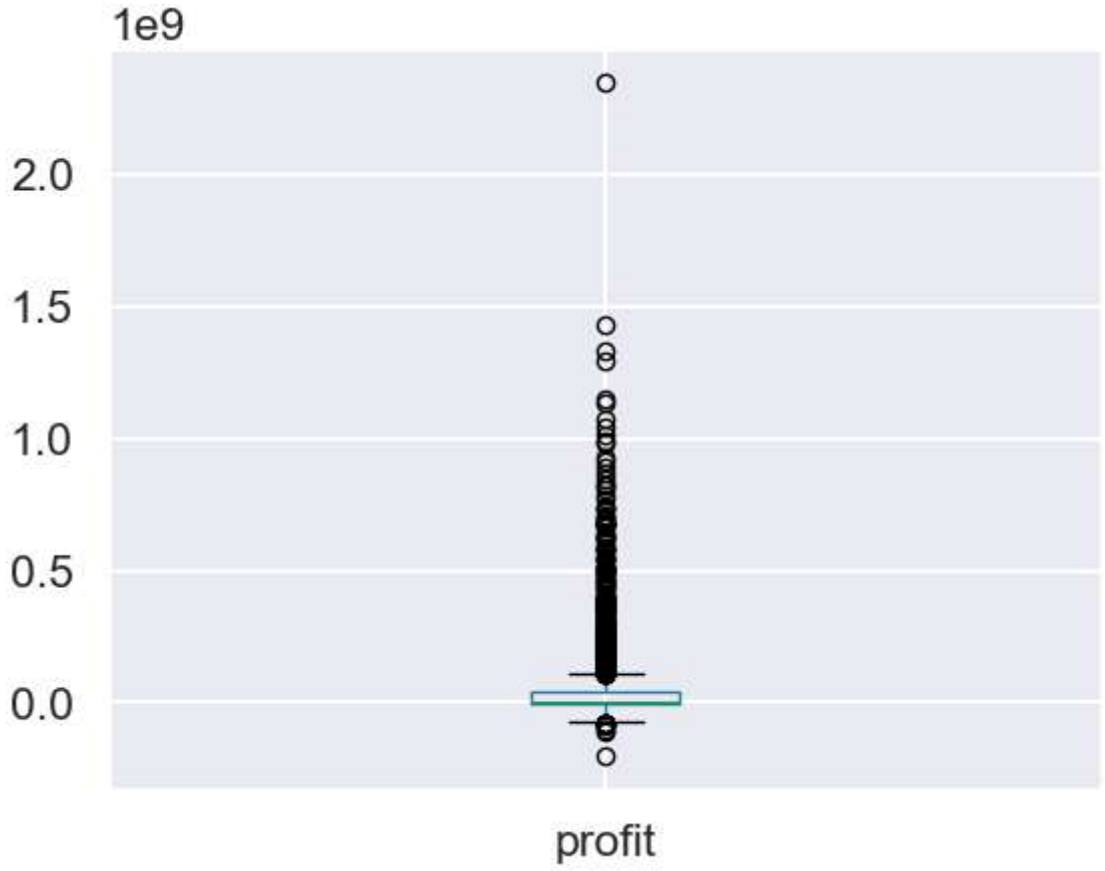
movie	tconst	nconst	start_year	genres	primary_name	death_year	averagerating	numvotes	releas
The Signal	tt2910814	nm1827931	2014	Drama,Mystery,Sci-Fi	William Eubank	NaN	6.1	58407.0	200
Hotel Transylvania 3: Summer Vacation	tt5220122	nm0850733	2018	Adventure,Animation,Comedy	Genndy Tartakovsky	NaN	6.3	42299.0	201
Winter's Bone	tt1399683	nm0335138	2010	Drama,Mystery	Debra Granik	NaN	7.2	127751.0	201
The Artist	tt1825978	nm3908851	2011	Thriller	Sunil Prem Vyas	NaN	6.8	6.0	201
The Artist	tt6684818	nm8608611	2015	Mystery	Avijit Sud	NaN	NaN	NaN	201
...	...	...	...	...	...	...	...	...	
Animals	tt1298530	nm2253409	2012	Drama,Fantasy	Marçal Forés	NaN	6.2	758.0	200
Animals	tt3297554	nm3317950	2014	Drama	Collin Schiffli	NaN	6.3	836.0	200
UnDivided	tt3564748	nm6321733	2013	Documentary	Sam Martin	NaN	8.4	8.0	201
Deadline	tt2623734	nm4240747	2016	Comedy,Romance	Charlie Lawton	NaN	8.5	15.0	200
#Horror	tt3526286	nm0836964	2015	Crime,Drama,Horror	Tara Subkoff	NaN	3.0	3092.0	201

2988 rows × 18 columns

In [47]:
```
clean_combined_datasets.shape
```

Out[47]:
```
(2988, 18)
```

In [48]:
```
boxplot_profit = clean_combined_datasets.boxplot(column=['profit'])
```



In [49]:
```
Q1 = clean_combined_datasets['profit'].quantile(0.25)
Q3 = clean_combined_datasets['profit'].quantile(0.75)
IQR = Q3 - Q1

# identify outliers
threshold = 1.5
outliers = clean_combined_datasets[(clean_combined_datasets['profit'] < Q1 - threshold * IQR) |
                                   (clean_combined_datasets['profit'] > Q3 + threshold * IQR)]
```

In [50]:
```
clean_combined_datasets = clean_combined_datasets.drop(outliers.index)
clean_combined_datasets
```

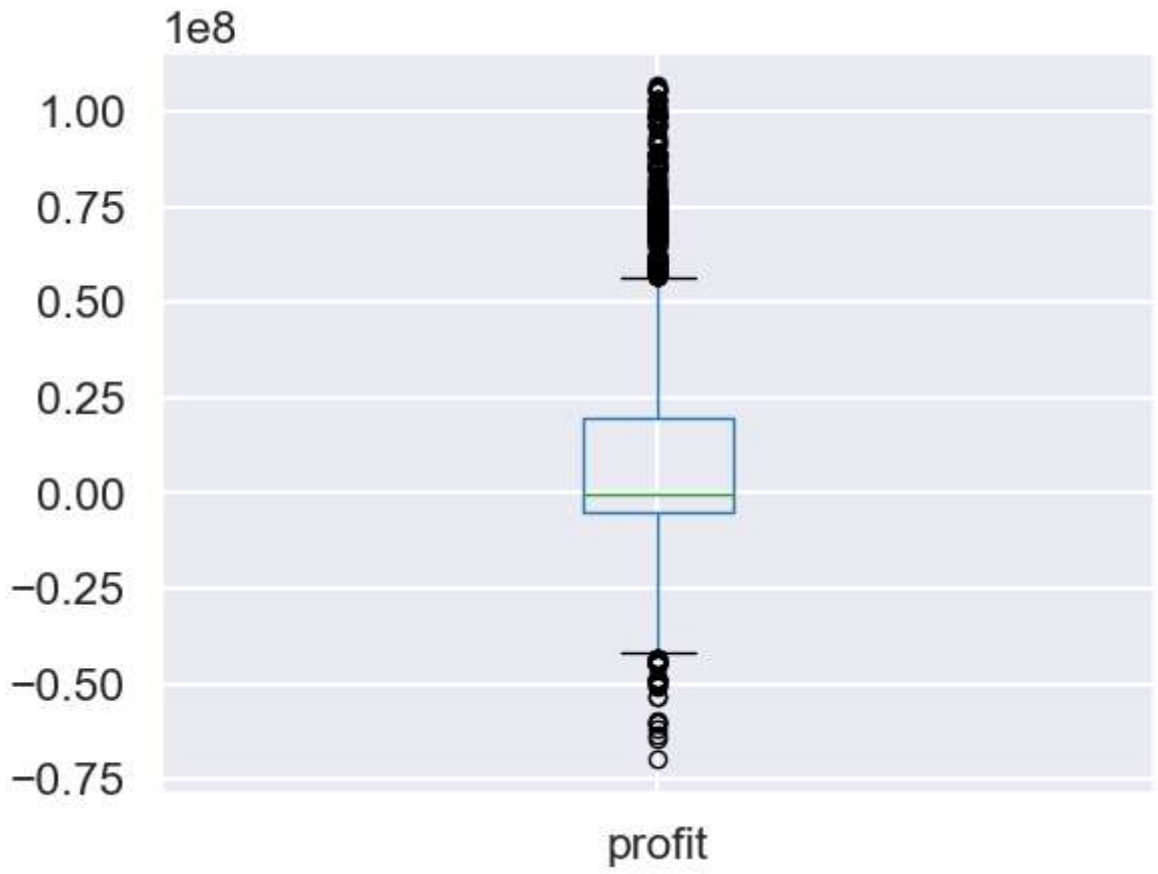
Out[50]:

movie	tconst	nconst	start_year	genres	primary_name	death_year	averagerating	numvotes	release_date	re
The Signal	tt2910814	nm1827931	2014	Drama,Mystery,Sci-Fi	William Eubank	NaN	6.1	58407.0	2008-02-22	
Winter's Bone	tt1399683	nm0335138	2010	Drama,Mystery	Debra Granik	NaN	7.2	127751.0	2010-06-11	
Barfi	tt3099638	nm5852684	2013	Comedy,Romance	Shekar	NaN	6.5	114.0	2012-09-14	
Friday	tt2344824	nm2118392	2012	Family	Lijin Jose	NaN	5.9	193.0	1995-04-26	
Dark Skies	tt2387433	nm0829820	2013	Horror,Sci-Fi,Thriller	Scott Stewart	NaN	6.3	65754.0	2013-02-22	
...	...	...	...	...	...	...	...	...	...	...
Animals	tt1298530	nm2253409	2012	Drama,Fantasy	Marçal Forés	NaN	6.2	758.0	2008-12-31	
Animals	tt3297554	nm3317950	2014	Drama	Collin Schiffli	NaN	6.3	836.0	2008-12-31	
UnDivided	tt3564748	nm6321733	2013	Documentary	Sam Martin	NaN	8.4	8.0	2015-02-03	
Deadline	tt2623734	nm4240747	2016	Comedy,Romance	Charlie Lawton	NaN	8.5	15.0	2009-12-31	
#Horror	tt3526286	nm0836964	2015	Crime,Drama,Horror	Tara Subkoff	NaN	3.0	3092.0	2015-11-20	

2595 rows × 18 columns

Analysis

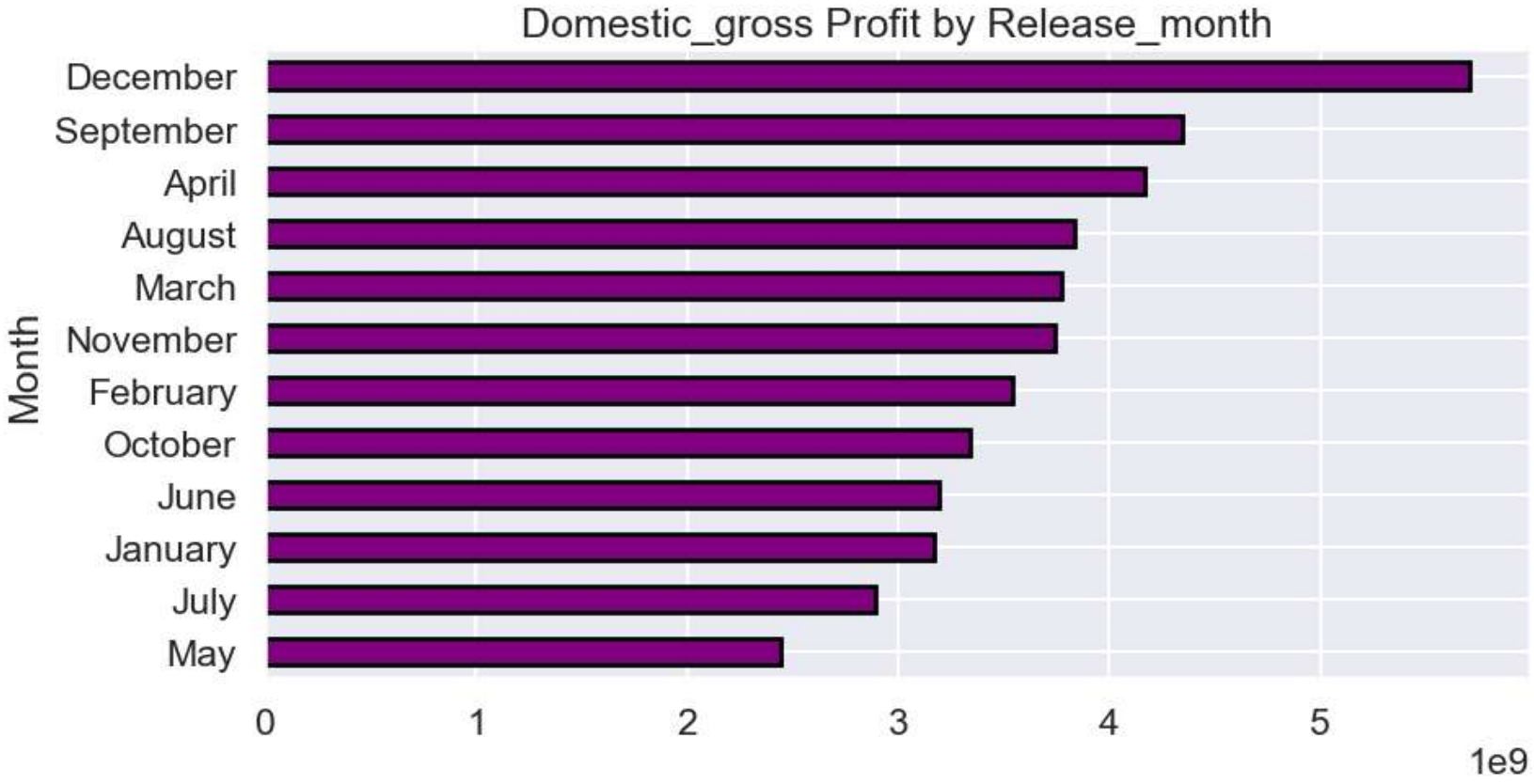
In [51]:
```
boxplot_profit = clean_combined_datasets.boxplot(column=['profit'])
```



What is the best month to realease a movie?

In [52]:
```
clean_combined_datasets.groupby(['release_month'])['domestic_gross'].sum().sort_values().plot(kind='barh', f
            color="purple",edgecolor="black",linewidth=2)
plt.title('Domestic_gross Profit by Release_month')
plt.ylabel('Month')
```

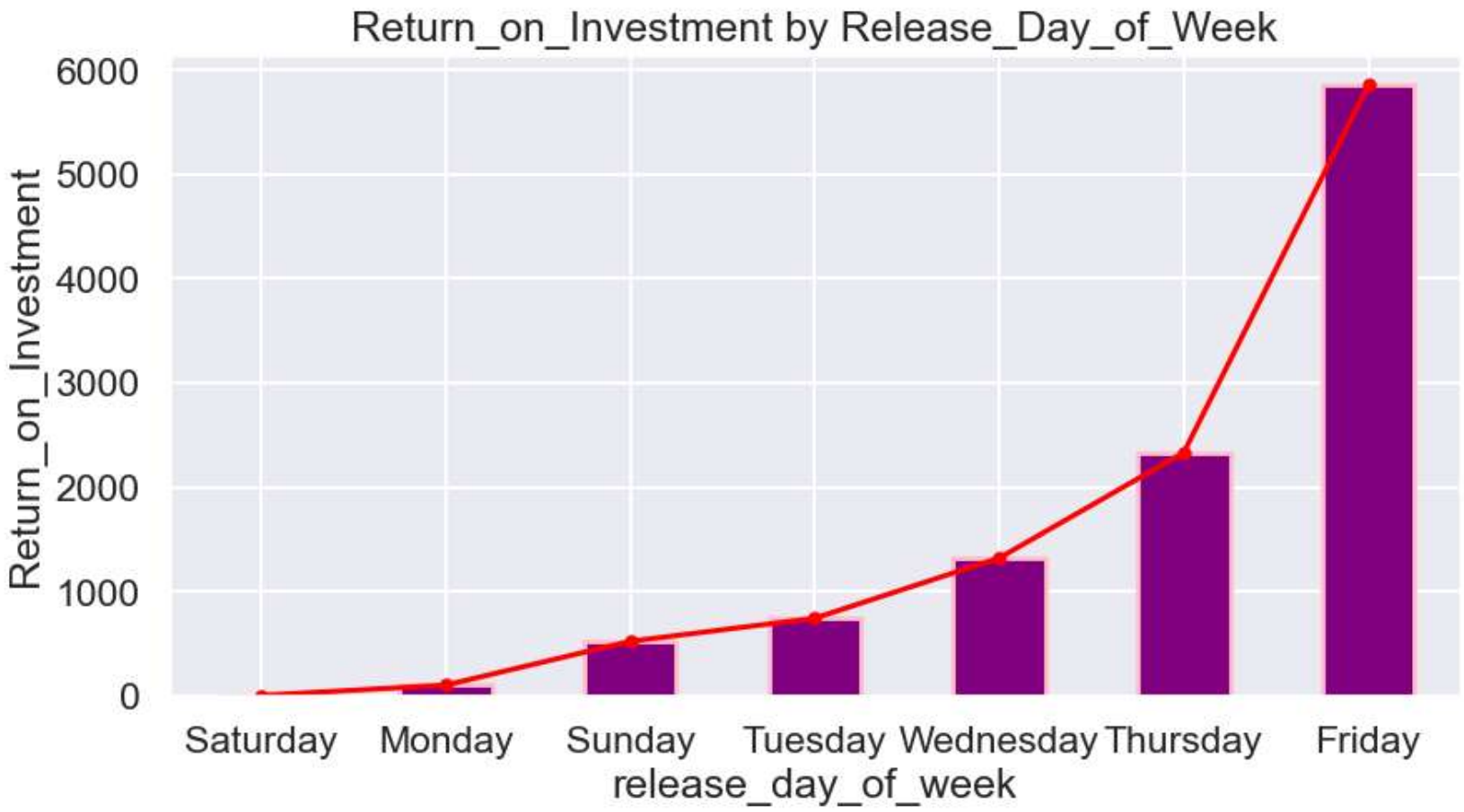
Out[52]:
```
Text(0, 0.5, 'Month')
```



What is the best day to realease a movie?

In [53]:
```
combined_datasets.groupby(['release_day_of_week'])['ROI'].sum().sort_values().plot(kind='bar', figsize=(10,5
            color="purple",edgecolor="pink",linewidth=2)
combined_datasets.groupby(['release_day_of_week'])['ROI'].sum().sort_values().plot(kind='line', marker="o",
            color="red",ms=5)
plt.title('Return_on_Investment by Release_Day_of_Week')
plt.ylabel('Return on Investment')
```

Out[53]:
```
Text(0, 0.5, 'Return_on_Investment')
```



In [54]:
```
clean_combined_datasets_copy = movie_budgets.loc[movie_budgets['profit_status'] =='Profit' ]
clean_combined_datasets_copy.sample(30)
```

Out[54]:

	movie	release_date	release_day_of_week	release_month	worldwide_gross	production_budget	profit	ROI	pro
5656	Once	2007-05-16	Wednesday	May	23323631	150000	23173631	154.490873	
4669	The Original Kings of Comedy	2000-08-18	Friday	August	38236338	3000000	35236338	11.745446	
409	The Mummy Returns	2001-05-04	Friday	May	435040395	98000000	337040395	3.439188	
3772	The Strangers	2008-05-30	Friday	May	83051676	9000000	74051676	8.227964	
1338	The Sweetest Thing	2002-04-12	Friday	April	63078756	43000000	20078756	0.466948	
2543	Conan the Barbarian	1982-05-14	Friday	May	79114085	20000000	59114085	2.955704	
2873	Friday the 13th	2009-02-13	Friday	February	92670237	17000000	75670237	4.451190	
4014	Flirting with Disaster	1996-03-22	Friday	March	16149180	7000000	9149180	1.307026	
358	Public Enemies	2009-07-01	Wednesday	July	212282709	102500000	109782709	1.071051	
2633	The Light Between Oceans	2016-09-02	Friday	September	22281732	20000000	2281732	0.114087	
5226	Beyond the Valley of the Dolls	1970-01-01	Thursday	January	9000000	1000000	8000000	8.000000	
3902	Woman on Top	2000-09-22	Friday	September	10192613	8000000	2192613	0.274077	
3754	Poltergeist III	1988-06-10	Friday	June	14114000	9500000	4614000	0.485684	
4001	Desperado	1995-08-25	Friday	August	25532388	7000000	18532388	2.647484	
5209	Casablanca	1943-01-23	Saturday	January	10496855	1039000	9457855	9.102844	
4908	Meet the Mormons	2014-10-10	Friday	October	6049171	2000000	4049171	2.024586	

	movie	release_date	release_day_of_week	release_month	worldwide_gross	production_budget	profit	ROI	pro
73	Iron Man	2008-05-02	Friday	May	585171547	186000000	399171547	2.146084	
2758	Adaptation	2002-12-06	Friday	December	32531759	18500000	14031759	0.758473	
1889	Blow	2001-04-06	Friday	April	83282296	30000000	53282296	1.776077	
328	The Green Hornet	2011-01-14	Friday	January	229155503	110000000	119155503	1.083232	
3201	Tea with Mussolini	1999-05-14	Friday	May	14395874	14000000	395874	0.028277	
1896	Atonement	2007-12-07	Friday	December	129779728	30000000	99779728	3.325991	
508	The Fast and the Furious: Tokyo Drift	2006-06-16	Friday	June	157794205	85000000	72794205	0.856402	
2792	The Fog	2005-10-14	Friday	October	37048526	18000000	19048526	1.058251	
719	The Lego Ninjago Movie	2017-09-22	Friday	September	122737201	70000000	52737201	0.753389	
379	The Emperor's New Groove	2000-12-15	Friday	December	169296573	100000000	69296573	0.692966	
158	Harry Potter and the Goblet of Fire	2005-11-18	Friday	November	897099794	150000000	747099794	4.980665	
447	Charlie's Angels	2000-11-03	Friday	November	259736090	90000000	169736090	1.885957	
2804	Young Sherlock Holmes	1985-12-04	Wednesday	December	19739000	18000000	1739000	0.096611	
2731	A Cinderella Story	2004-07-16	Friday	July	70164105	19000000	51164105	2.692848	

According to the analysis shown above, the best day to release a movie and receive the highest return on Investment is on Friday. The worst days would be saturday through Monday

In [55]:
```
#Filter to include only movies that made 50% above its production budget
combined_datasets_copy = combined_datasets[combined_datasets["ROI"]>20].copy()
```

In [56]:
```
combined_datasets_copy
```

Out[56]:

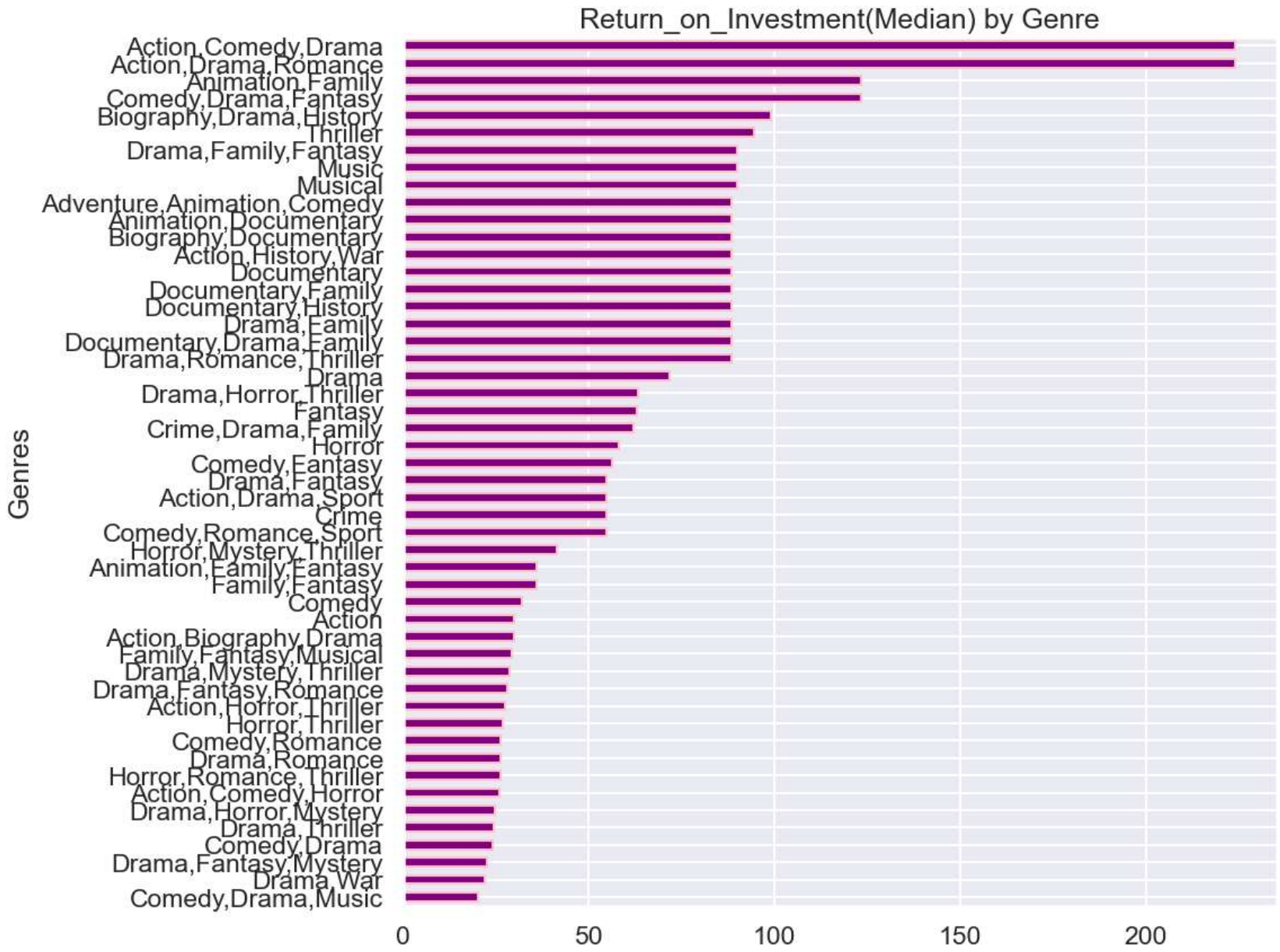
movie	tconst	nconst	start_year	genres	primary_name	death_year	averagerating	numvotes	release_date
Bambi	tt2668120	nm0509852	2013	Biography,Documentary	Sébastien Lifshitz	NaN	7.0	101.0	1942-08-13
Night of the Living Dead	tt3393372	nm5209991	2014	Horror	Chad Zuver	NaN	5.3	277.0	1968-10-01
Rocky	tt3080284	nm1126847	2013	Action,Comedy,Drama	Sujit Mondal	NaN	6.1	46.0	1976-11-21
Rocky	tt9430578	nm9645626	2019	Action,Drama,Romance	Adnan A. Shaikh	NaN	6.4	5.0	1976-11-21
Halloween	tt1502407	nm0337773	2018	Horror,Thriller	David Gordon Green	NaN	6.6	88395.0	1978-10-17
...	...	...	...	...	...	...	...	...	...
Ghost	tt1655400	nm2039774	2010	Drama,Fantasy,Mystery	Tarô Ohtani	NaN	5.8	229.0	1990-07-13
Ghost	tt2175671	nm1680452	2012	Horror	Puja Bedi	NaN	2.3	205.0	1990-07-13
Live and Let Die	tt6162924	nm8516338	2015	Drama,War	McCaslin Miles	NaN	NaN	NaN	1973-06-27
10 Cloverfield Lane	tt1179933	nm0870469	2016	Drama,Horror,Mystery	Dan Trachtenberg	NaN	7.2	260383.0	2016-03-11
La La Land	tt3783958	nm3227090	2016	Comedy,Drama,Music	Damien Chazelle	NaN	8.0	436070.0	2016-12-09

97 rows × 18 columns

What are the most sucessful movie genres?

In [57]:
```
combined_datasets_copy.groupby(['genres'])['ROI'].median().sort_values().plot(kind='barh', figsize=(10,10),
             color="purple",edgecolor="pink")
plt.title('Return_on_Investment(Median) by Genre')
plt.ylabel('Genres')
```

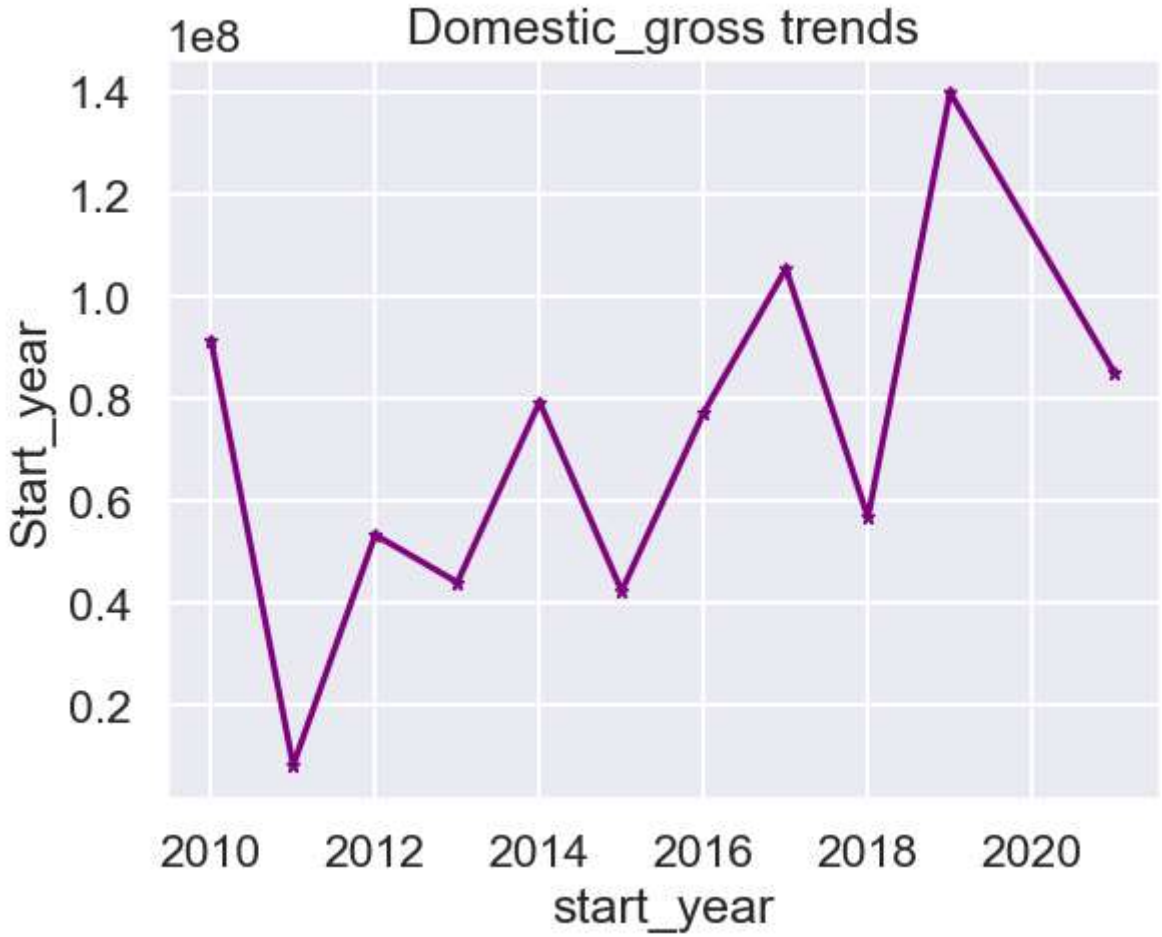
Out[57]:
```
Text(0, 0.5, 'Genres')
```


Return_on_Investment(Median) by Genre
Genres
Action,Comedy,Drama
Action,Drama,Romance
Animation,Family
Comedy,Drama,Fantasy
Biography,Drama,History
Thriller
Drama,Family,Fantasy
Music
Musical
Adventure,Animation,Comedy
Animation,Documentary
Biography,Documentary
Action,History,War
Documentary
Documentary,Family
Documentary,History
Drama,Family
Documentary,Drama,Family
Drama,Romance,Thriller
Drama
Drama,Horror,Thriller
Fantasy
Crime,Drama,Family
Horror
Comedy,Fantasy
Drama,Fantasy
Action,Drama,Sport
Crime
Comedy,Romance,Sport
Horror,Mystery,Thriller
Animation,Family,Fantasy
Family,Fantasy
Comedy
Action
Action,Biography,Drama
Family,Fantasy,Musical
Drama,Mystery,Thriller
Drama,Fantasy,Romance
Action,Horror,Thriller
Horror,Thriller
Comedy,Romance
Drama,Romance
Horror,Romance,Thriller
Action,Comedy,Horror
Drama,Horror,Mystery
Drama,Thriller
Comedy,Drama
Drama,Fantasy,Mystery
Drama,War
Comedy,Drama,Music
0
50
100
150
200

Domestic gross profit trend

In [58]:
```
combined_datasets_copy.groupby(['start_year'])['domestic_gross'].mean().plot(kind='line', marker="*",
            color="purple",ms=5)
plt.title('Domestic_gross trends')
plt.ylabel('Start_year')
```

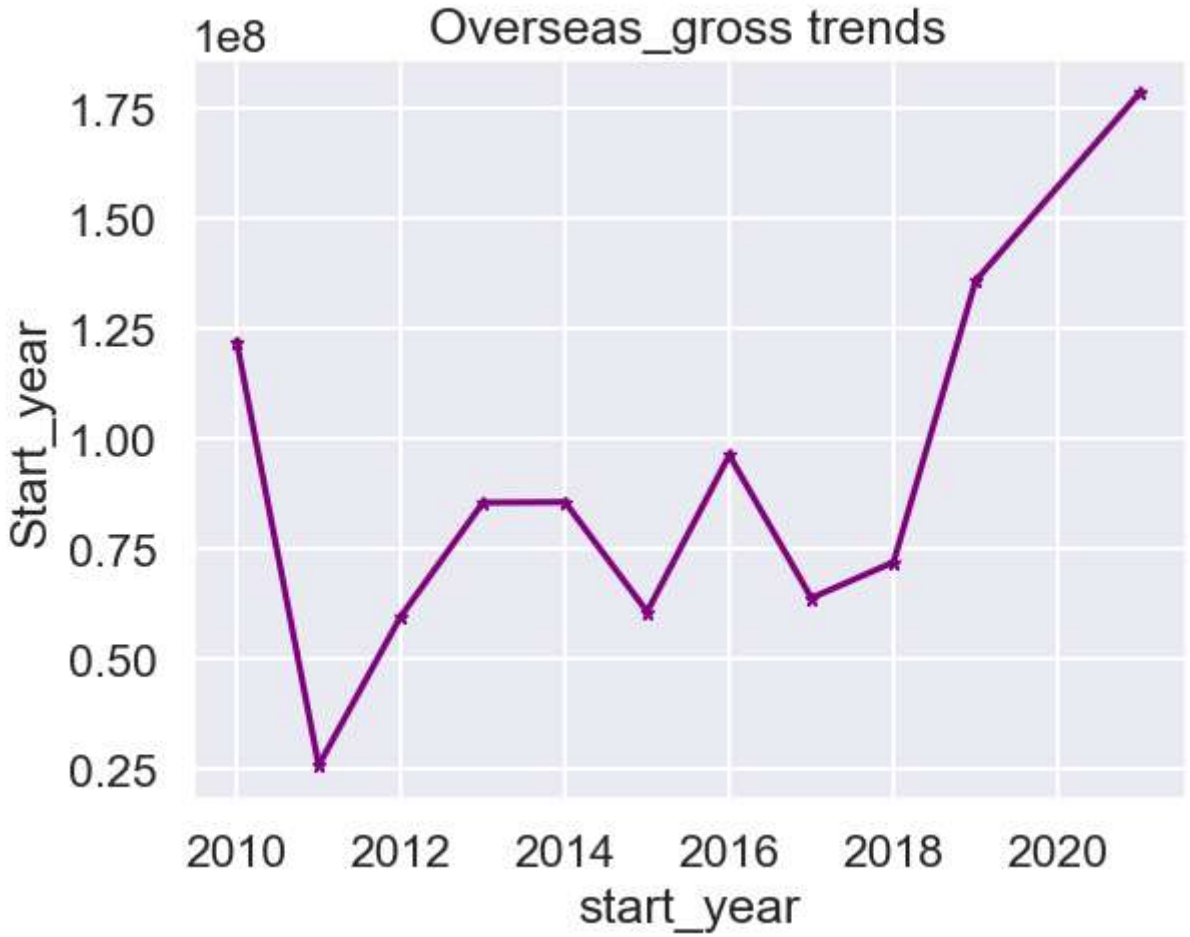
Out[58]:
```
Text(0, 0.5, 'Start_year')
```



Profit overseas gross profit trend

In [59]:
```
combined_datasets_copy.groupby(['start_year'])['profitable_oversees'].mean().plot(kind='line', marker="*",
            color="purple",ms=5)
plt.title('Overseas_gross trends')
plt.ylabel('Start_year')
```

Out[59]:
```
Text(0, 0.5, 'Start_year')
```



Investigating the correlation of data columns

In [60]:
```
combined_datasets.corr(numeric_only=True)
```

Out[60]:

	start_year	death_year	averagerating	numvotes	worldwide_gross	production_budget	profit	ROI	domes
start_year	1.000000	0.491465	0.053535	-0.099473	0.004165	0.004471	0.003699	0.025795	
death_year	0.491465	1.000000	-0.225766	-0.223234	0.098083	0.060751	0.100896	0.186618	
averagerating	0.053535	-0.225766	1.000000	0.312393	0.159434	0.116756	0.158664	0.032199	
numvotes	-0.099473	-0.223234	0.312393	1.000000	0.572600	0.492346	0.547742	0.047481	
worldwide_gross	0.004165	0.098083	0.159434	0.572600	1.000000	0.763032	0.981037	0.110012	
production_budget	0.004471	0.060751	0.116756	0.492346	0.763032	1.000000	0.623287	-0.074497	
profit	0.003699	0.100896	0.158664	0.547742	0.981037	0.623287	1.000000	0.155435	
ROI	0.025795	0.186618	0.032199	0.047481	0.110012	-0.074497	0.155435	1.000000	
domestic_gross	0.013245	0.038439	0.170402	0.570668	0.943683	0.698448	0.932270	0.128082	
profitable_oversees	-0.001187	0.156298	0.145251	0.544714	0.982268	0.761750	0.959970	0.094311	

Conclusions

This analysis leads to three recommendations which Microsoft Corporation can take into account before deciding on what movies to make:

- Focus on producing either of the top 2 genres: Drama and Documentary which also have the highest Return on Investment
- Release any movies it intends to produce on a Friday to realise the highest Return on Investment.
- Microsoft should also explore overseas markets as well as the profit sometimes is higher than the domestic profit.