Final Project Submission

Please fill out:

- Student name: WINNIE ONDURU
- Student pace: PART TIME
- Scheduled project review date/time:
- Instructor name: MS.D.MONGINA
- Blog post URL:

MICROSOFTS NEW STUDIO PROJECT ANALYSIS

Overview

```
Microsoft has decided to get in on the fun that the big companies creating
original video content are in.
This they are doing this by creating a new movie studio.They however, lack
knowledge and expertise in creating films and It’s is for this reason that
they need to determine which types of films are currently successful in the
market.
```

Business Understanding

```
I need to conduct an analysis to identify the current top-grossing film
genres that are driving box office success. I also need to examine the
budget and competitive landscape of successful films and consider any
emerging trends in the industry. The findings from this analysis will help
Microsoft's new movie studio to make informed decisions on what type of
films to create.
The questions ill have to answer in order to objectively get clear
recommendations that will be beneficial to Microsoft are
```

```
   1.How does the release month of a film affect its gross values i.e
Worldwide and Domestic?

   2.Which is the most popular film genre ?

   3.Which Studios are making the highest profit?
```

Data Understanding

These data were provided by Flatiron.I had several datasets in zipped files, namely:
    * Box Office Mojo
    * IMDB
    * Rotten Tomatoes
    * TheMovieDB
    * The Numbers

For this analysis though i used :
    * bom.movie_gross.csv has five columns comprising of movie titles, studios, financial incomes both domestic
      and foreign and the release year.
    * tn.movie_budgets.csv contains information on released films, including their names, release dates,
      and financial data such as production budget and worldwide gross.
    * rt.movie_info.tsv tells us more about each movie.It has twelve columns
namelyid,synopsis,genre,rating,director,writer,theater_date,dvd_date, currency,box_office,runtime and studio.

Data Analysis

Lets begin by importing all the necessary Packages and Libraries

In [38]:
```
# importing necessary packages
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import sqlite3
```

I checked my directory so as to access all the folders with the datasets that i am going to use for this project.

Reading the Datasets:

The First Dataset is 'bom.movie_gross.csv'

In [39]:
```
df_bom_movies_gross = pd.read_csv('bom.movie_gross.csv')
df_bom_movies_gross
```

Out[39]:

	title	studio	domestic_gross	foreign_gross	year
0	Toy Story 3	BV	415000000.0	652000000	2010
1	Alice in Wonderland (2010)	BV	334200000.0	691300000	2010
2	Harry Potter and the Deathly Hallows Part 1	WB	296000000.0	664300000	2010
3	Inception	WB	292600000.0	535700000	2010
4	Shrek Forever After	P/DW	238700000.0	513900000	2010
...	...	...	...	...	...
3382	The Quake	Magn.	6200.0	NaN	2018
3383	Edward II (2018 re-release)	FM	4800.0	NaN	2018
3384	El Pacto	Sony	2500.0	NaN	2018
3385	The Swan	Synergetic	2400.0	NaN	2018
3386	An Actor Prepares	Grav.	1700.0	NaN	2018

3387 rows × 5 columns

```
Finding out more about the Dataset,Checking to see if there are any
Duplicate Values,Any Null Values e.t.c
```

In [40]:
```
df_bom_movies_gross.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3387 entries, 0 to 3386
Data columns (total 5 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   title           3387 non-null   object 
 1   studio          3382 non-null   object 
 2   domestic_gross  3359 non-null   float64
 3   foreign_gross   2037 non-null   object 
 4   year            3387 non-null   int64  
dtypes: float64(1), int64(1), object(3)
memory usage: 132.4+ KB
```

```
Lets first check for duplicates
```

In [41]:
```
df_bom_movies_gross.duplicated()
```

Out[41]:
```
0       False
1       False
2       False
3       False
4       False
        ...
3382    False
3383    False
3384    False
3385    False
3386    False
Length: 3387, dtype: bool
```

Next,Because i'll need the column foreign_gross,i am going to change its Datatype to float instead of object

In [42]:
```
#I converted the 'foreign_gross' column to float by removing commas from the 
df_bom_movies_gross['foreign_gross'] = [float(str(x).replace(',', '')) for x 
df_bom_movies_gross
```

Out[42]:

	title	studio	domestic_gross	foreign_gross	year
0	Toy Story 3	BV	415000000.0	652000000.0	2010
1	Alice in Wonderland (2010)	BV	334200000.0	691300000.0	2010
2	Harry Potter and the Deathly Hallows Part 1	WB	296000000.0	664300000.0	2010
3	Inception	WB	292600000.0	535700000.0	2010
4	Shrek Forever After	P/DW	238700000.0	513900000.0	2010
...	...	...	...	...	...
3382	The Quake	Magn.	6200.0	NaN	2018
3383	Edward II (2018 re-release)	FM	4800.0	NaN	2018
3384	El Pacto	Sony	2500.0	NaN	2018
3385	The Swan	Synergetic	2400.0	NaN	2018
3386	An Actor Prepares	Grav.	1700.0	NaN	2018

3387 rows × 5 columns

Having changed the foreign_gross column and confirmed its changed,Now lets check for Null values
I'll also use the sum() method to sum up the True values in each column then return the total number of missing values
in each column.

After which i find the % of each null value in each column to get an idea of how much missing data we have for each column.

In [43]:
```
df_bom_movies_gross.isnull().sum()*100/len(df_bom_movies_gross)
```

Out[43]:
```
title              0.000000
studio             0.147623
domestic_gross     0.826690
foreign_gross     39.858282
year               0.000000
dtype: float64
```

At this point i'll just drop the Null values because i really dont think it will make much difference.
After which i'll find out to confirm the Null values were dropped

In [44]:
```
df_bom_movies_gross.dropna(axis = 0,inplace = True)
df_bom_movies_gross.isnull().sum()
```

Out[44]:
```
title             0
studio            0
domestic_gross    0
foreign_gross     0
year              0
dtype: int64
```

In [ ]:

The Second Dataset is 'tn.movie_budgets.csv'

In [45]:
```
df_movies_budgets = pd.read_csv('tn.movie_budgets.csv')
df_movies_budgets
```

Out[45]:

	id	release_date	movie	production_budget	domestic_gross	worldwide_gross
0	1	Dec 18, 2009	Avatar	$425,000,000	$760,507,625	$2,776,345,279
1	2	May 20, 2011	Pirates of the Caribbean: On Stranger Tides	$410,600,000	$241,063,875	$1,045,663,875
2	3	Jun 7, 2019	Dark Phoenix	$350,000,000	$42,762,350	$149,762,350
3	4	May 1, 2015	Avengers: Age of Ultron	$330,600,000	$459,005,868	$1,403,013,963
4	5	Dec 15, 2017	Star Wars Ep. VIII: The Last Jedi	$317,000,000	$620,181,382	$1,316,721,747
...	...	...	...	...	...	...
5777	78	Dec 31, 2018	Red 11	$7,000	$0	$0
5778	79	Apr 2, 1999	Following	$6,000	$48,482	$240,495
5779	80	Jul 13, 2005	Return to the Land of Wonders	$5,000	$1,338	$1,338
5780	81	Sep 29, 2015	A Plague So Pleasant	$1,400	$0	$0
5781	82	Aug 5, 2005	My Date With Drew	$1,100	$181,041	$181,041

5782 rows × 6 columns

In [46]:
```
#Lets begin by checking for null values
df_movies_budgets.isnull().sum()
```

Out[46]:
```
id                 0
release_date       0
movie              0
production_budget  0
domestic_gross     0
worldwide_gross    0
dtype: int64
```

In [47]:
```
#Now that there are no Null Values,how about any duplicates?
df_movies_budgets.duplicated()
```

Out[47]:
```
0       False
1       False
2       False
3       False
4       False
        ...
5777    False
5778    False
5779    False
5780    False
5781    False
Length: 5782, dtype: bool
```

Now,lets work on our main columns, production_budget, domestic_gross and worlwide_gross by changing the dataset from Object to Integers inorder to make it easy to work with in this analysis.

This will be done by removing $ signs and the , commas.

In [51]:
```
df_movies_budgets['production_budget'] = df_movies_budgets['production_budget
df_movies_budgets['domestic_gross'] = df_movies_budgets['domestic_gross'].asty
df_movies_budgets['worldwide_gross'] = df_movies_budgets['worldwide_gross'].a

for r in ['production_budget', 'domestic_gross', 'worldwide_gross']:
    # Removes $ symbol and ,
    df_movies_budgets[r] = df_movies_budgets[r].str.replace('$', '').str.repl
    df_movies_budgets[r] = df_movies_budgets[r].astype(float) # Converts from
    # Divides by 100,000,000 to make it easier for the visualization
    df_movies_budgets[r] = (df_movies_budgets[r]).astype('int64') # Converts
df_movies_budgets
```

```
C:\Users\USER\AppData\Local\Temp\ipykernel_5568\3403968726.py:7: FutureWa
rning: The default value of regex will change from True to False in a fut
ure version. In addition, single character regular expressions will *not*
be treated as literal strings when regex=True.
  df_movies_budgets[r] = df_movies_budgets[r].str.replace('$', '').str.re
place(',', '')
```

Out[51]:

	id	release_date	movie	production_budget	domestic_gross	worldwide_gross
0	1	Dec 18, 2009	Avatar	425000000	760507625	2776345279
1	2	May 20, 2011	Pirates of the Caribbean: On Stranger Tides	410600000	241063875	1045663875
2	3	Jun 7, 2019	Dark Phoenix	350000000	42762350	149762350
3	4	May 1, 2015	Avengers: Age of Ultron	330600000	459005868	1403013963

In [14]:
```
df_movies_budgets.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5782 entries, 0 to 5781
Data columns (total 6 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   id                 5782 non-null   int64
 1   release_date       5782 non-null   object
 2   movie              5782 non-null   object
 3   production_budget  5782 non-null   int64
 4   domestic_gross     5782 non-null   int64
 5   worldwide_gross    5782 non-null   int64
dtypes: int64(4), object(2)
memory usage: 271.2+ KB
```

In [ ]:

The Third Dataset is 'rt.movie_info.tsv'

In [52]:
```
df_movies_info = pd.read_csv('rt.movie_info.tsv',sep='\t')
df_movies_info
```

Out[52]:

	id	synopsis	rating	genre	director	writer	the
0	1	This gritty, fast-paced, and innovative police...	R	Action and Adventure|Classics|Drama	William Friedkin	Ernest Tidyman	O
1	3	New York City, not-too-distant-future: Eric Pa...	R	Drama|Science Fiction and Fantasy	David Cronenberg	David Cronenberg|Don DeLillo	
2	5	Illeana Douglas delivers a superb performance ...	R	Drama|Musical and Performing Arts	Allison Anders	Allison Anders	
3	6	Michael Douglas runs afoul of a treacherous su...	R	Drama|Mystery and Suspense	Barry Levinson	Paul Attanasio|Michael Crichton	D
4	7	NaN	NR	Drama|Romance	Rodney Bennett	Giles Cooper	
...	...	...	...	...	...	...	
1555	1996	Forget terrorists or hijackers -- there's a ha...	R	Action and Adventure|Horror|Mystery and Suspense	NaN	NaN	
1556	1997	The popular Saturday Night Live sketch was exp...	PG	Comedy|Science Fiction and Fantasy	Steve Barron	Terry Turner|Tom Davis|Dan Aykroyd|Bonnie Turner	Ju
1557	1998	Based on a novel by Richard Powell, when the l...	G	Classics|Comedy|Drama|Musical and Performing Arts	Gordon Douglas	NaN	Ja
1558	1999	The Sandlot is a coming-of-age story about a g...	PG	Comedy|Drama|Kids and Family|Sports and Fitness	David Mickey Evans	David Mickey Evans|Robert Gunter	A
1559	2000	Suspended from the force, Paris cop Hubert is ...	R	Action and Adventure|Art House and Internation...	NaN	Luc Besson	

1560 rows × 12 columns

Finding out more about the Dataset,Checking to see if there are any Duplicate Values,Any Null Values e.t.c

In [53]:
```
df_movies_info.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1560 entries, 0 to 1559
Data columns (total 12 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   id            1560 non-null   int64 
 1   synopsis      1498 non-null   object
 2   rating        1557 non-null   object
 3   genre         1552 non-null   object
 4   director      1361 non-null   object
 5   writer        1111 non-null   object
 6   theater_date  1201 non-null   object
 7   dvd_date      1201 non-null   object
 8   currency      340 non-null    object
 9   box_office    340 non-null    object
 10  runtime       1530 non-null   object
 11  studio        494 non-null    object
dtypes: int64(1), object(11)
memory usage: 146.4+ KB
```

The most important columns are Id,Ratings,Genre,Runtime and Studio

So lets go ahead and remove/drop the other unrequired columns and confirm its done

In [54]:
```
df_movies_info.drop(['synopsis','director','writer','theater_date','dvd_date'
df_movies_info.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1560 entries, 0 to 1559
Data columns (total 5 columns):
 #   Column   Non-Null Count  Dtype 
---  ------   --------------  ----- 
 0   id       1560 non-null   int64 
 1   rating   1557 non-null   object
 2   genre    1552 non-null   object
 3   runtime  1530 non-null   object
 4   studio   494 non-null    object
dtypes: int64(1), object(4)
memory usage: 61.1+ KB
```

In [55]:
```
#Then check for duplicates
df_movies_info.duplicated()
```

Out[55]:
```
0       False
1       False
2       False
3       False
4       False
        ...
1555    False
1556    False
1557    False
1558    False
1559    False
Length: 1560, dtype: bool
```

In [56]:
```
#Also check for Null Values in %
df_movies_info.isnull().sum()*100/len(df_movies_info)
```

Out[56]:
```
id          0.000000
rating      0.192308
genre       0.512821
runtime     1.923077
studio     68.333333
dtype: float64
```

```
Since the remaining most important columns have missing values and i cant
drop them,am going to replace the Null values
in each and every of these columns as required.
```

In [57]:
```
#First,lets work on Ratings Column
df_movies_info['rating'].fillna('NA', inplace = True)

#Then Genre Column
df_movies_info['genre'].fillna('NA', inplace = True)

#Finally for Runtime Column
# Remove units of measurement from runtime column
df_movies_info['runtime'] = [float(str(x).replace('minutes', '')) for x in df

# Convert runtime column to float
df_movies_info['runtime'] = pd.to_numeric(df_movies_info['runtime'], errors='

# Inpute missing values in runtime column with median value
median_runtime = df_movies_info['runtime'].median()
df_movies_info['runtime'].fillna(median_runtime, inplace=True)

# Lets Change runtime column back to string representation of minutes
df_movies_info['runtime'] = df_movies_info['runtime'].astype(int).astype(str)


df_movies_info.isnull().sum()*100/len(df_movies_info)
```

Out[57]:
```
id          0.000000
rating      0.000000
genre       0.000000
runtime     0.000000
studio     68.333333
dtype: float64
```

In [58]:
```
df_movies_info
```

Out[58]:

	id	rating	genre	runtime	studio
0	1	R	Action and Adventure|Classics|Drama	104 minutes	NaN
1	3	R	Drama|Science Fiction and Fantasy	108 minutes	Entertainment One
2	5	R	Drama|Musical and Performing Arts	116 minutes	NaN
3	6	R	Drama|Mystery and Suspense	128 minutes	NaN
4	7	NR	Drama|Romance	200 minutes	NaN
...	...	...	...	...	...
1555	1996	R	Action and Adventure|Horror|Mystery and Suspense	106 minutes	New Line Cinema
1556	1997	PG	Comedy|Science Fiction and Fantasy	88 minutes	Paramount Vantage
1557	1998	G	Classics|Comedy|Drama|Musical and Performing Arts	111 minutes	NaN
1558	1999	PG	Comedy|Drama|Kids and Family|Sports and Fitness	101 minutes	NaN
1559	2000	R	Action and Adventure|Art House and Internation...	94 minutes	Columbia Pictures

1560 rows × 5 columns

MERGING ALL DATASETS

```
Lets now merge our datasets to come up with the final dataset that i am
going to use for this analysis.
```

```
The dataset df_movies_info and df_bom_movies_gross are joined first to form
df_dataset_one
```

In [59]:
```
df_dataset_one = pd.merge(df_bom_movies_gross, df_movies_info, on='studio')
df_dataset_one
```

Out[59]:

	title	studio	domestic_gross	foreign_gross	year	id	rating	genre	runti
0	Harry Potter and the Deathly Hallows Part 1	WB	296000000.0	664300000.0	2010	611	R	Drama|Mystery and Suspense	1 minut
1	Inception	WB	292600000.0	535700000.0	2010	611	R	Drama|Mystery and Suspense	1 minut
2	Clash of the Titans (2010)	WB	163200000.0	330000000.0	2010	611	R	Drama|Mystery and Suspense	1 minut
3	Due Date	WB	100500000.0	111200000.0	2010	611	R	Drama|Mystery and Suspense	1 minut
4	Yogi Bear	WB	100200000.0	101300000.0	2010	611	R	Drama|Mystery and Suspense	1 minut
...	...	...	...	...	...	...	...	...	
360	Lady Bird	A24	49000000.0	30000000.0	2017	1399	R	Drama|Horror	minut
361	The Disaster Artist	A24	21100000.0	8700000.0	2017	1399	R	Drama|Horror	minut
362	It Comes At Night	A24	14000000.0	5300000.0	2017	1399	R	Drama|Horror	minut
363	Hereditary	A24	44100000.0	35300000.0	2018	1399	R	Drama|Horror	minut
364	The Children Act	A24	548000.0	17000000.0	2018	1399	R	Drama|Horror	minut

365 rows × 9 columns

Secondly we merge the third df_movies_budgets dataset into df_dataset_one to form my final datset for analysis called df_final_dataset

But first,which columns does df_movies_budgets have?

In [60]:
```
df_movies_budgets
```

Out[60]:

	id	release_date	movie	production_budget	domestic_gross	worldwide_gross
0	1	Dec 18, 2009	Avatar	425000000	760507625	2776345279
1	2	May 20, 2011	Pirates of the Caribbean: On Stranger Tides	410600000	241063875	1045663875
2	3	Jun 7, 2019	Dark Phoenix	350000000	42762350	149762350
3	4	May 1, 2015	Avengers: Age of Ultron	330600000	459005868	1403013963
4	5	Dec 15, 2017	Star Wars Ep. VIII: The Last Jedi	317000000	620181382	1316721747
...	...	...	...	...	...	...
5777	78	Dec 31, 2018	Red 11	7000	0	0
5778	79	Apr 2, 1999	Following	6000	48482	240495
5779	80	Jul 13, 2005	Return to the Land of Wonders	5000	1338	1338
5780	81	Sep 29, 2015	A Plague So Pleasant	1400	0	0
5781	82	Aug 5, 2005	My Date With Drew	1100	181041	181041

5782 rows × 6 columns

```
Since Title and Movie columns are the same,i am going to change the column
Movie in the table df_movies_budgets
to Title then join df_dataset_one to  my final dataset called
df_dataset_final
```

In [61]:
```
df_movies_budgets.rename(columns={'movie': 'title'},inplace = 'True')
df_movies_budgets
```

Out[61]:

	id	release_date	title	production_budget	domestic_gross	worldwide_gross
0	1	Dec 18, 2009	Avatar	425000000	760507625	2776345279
1	2	May 20, 2011	Pirates of the Caribbean: On Stranger Tides	410600000	241063875	1045663875
2	3	Jun 7, 2019	Dark Phoenix	350000000	42762350	149762350
3	4	May 1, 2015	Avengers: Age of Ultron	330600000	459005868	1403013963
4	5	Dec 15, 2017	Star Wars Ep. VIII: The Last Jedi	317000000	620181382	1316721747
...	...	...	...	...	...	...
5777	78	Dec 31, 2018	Red 11	7000	0	0
5778	79	Apr 2, 1999	Following	6000	48482	240495
5779	80	Jul 13, 2005	Return to the Land of Wonders	5000	1338	1338
5780	81	Sep 29, 2015	A Plague So Pleasant	1400	0	0
5781	82	Aug 5, 2005	My Date With Drew	1100	181041	181041

5782 rows × 6 columns

```
Here it is!Lets do the final last merge.
```

In [62]:
```
df_dataset_final = pd.merge(df_dataset_one,df_movies_budgets, on = 'title')
df_dataset_final
```

Out[62]:

	title	studio	domestic_gross_x	foreign_gross	year	id_x	rating	genre	rur
0	Inception	WB	292600000.0	535700000.0	2010	611	R	Drama|Mystery and Suspense	mil
1	Due Date	WB	100500000.0	111200000.0	2010	611	R	Drama|Mystery and Suspense	mil
2	Yogi Bear	WB	100200000.0	101300000.0	2010	611	R	Drama|Mystery and Suspense	mil
3	The Book of Eli	WB	94800000.0	62300000.0	2010	611	R	Drama|Mystery and Suspense	mil
4	The Town	WB	92200000.0	61800000.0	2010	611	R	Drama|Mystery and Suspense	mil
...	...	...	...	...	...	...	...	...	
245	The Witch	A24	25100000.0	15300000.0	2016	1399	R	Drama|Horror	mil
246	American Honey	A24	663000.0	1200000.0	2016	1399	R	Drama|Horror	mil
247	Lady Bird	A24	49000000.0	30000000.0	2017	1399	R	Drama|Horror	mil
248	The Disaster Artist	A24	21100000.0	8700000.0	2017	1399	R	Drama|Horror	mil
249	Hereditary	A24	44100000.0	35300000.0	2018	1399	R	Drama|Horror	mil

250 rows × 14 columns

In [63]:
```
#Now lets drop one column for domestic_gross since they are two
df_dataset_final.drop(['domestic_gross_y'],axis = 1,inplace = True)
df_dataset_final
```

Out[63]:

	title	studio	domestic_gross_x	foreign_gross	year	id_x	rating	genre	rur
0	Inception	WB	292600000.0	535700000.0	2010	611	R	Drama|Mystery and Suspense	mi
1	Due Date	WB	100500000.0	111200000.0	2010	611	R	Drama|Mystery and Suspense	mi
2	Yogi Bear	WB	100200000.0	101300000.0	2010	611	R	Drama|Mystery and Suspense	mi
3	The Book of Eli	WB	94800000.0	62300000.0	2010	611	R	Drama|Mystery and Suspense	mi
4	The Town	WB	92200000.0	61800000.0	2010	611	R	Drama|Mystery and Suspense	mi
...	...	...	...	...	...	...	...	...	
245	The Witch	A24	25100000.0	15300000.0	2016	1399	R	Drama|Horror	mi
246	American Honey	A24	663000.0	1200000.0	2016	1399	R	Drama|Horror	mi
247	Lady Bird	A24	49000000.0	30000000.0	2017	1399	R	Drama|Horror	mi
248	The Disaster Artist	A24	21100000.0	8700000.0	2017	1399	R	Drama|Horror	mi
249	Hereditary	A24	44100000.0	35300000.0	2018	1399	R	Drama|Horror	mi

250 rows × 13 columns

In [64]:
```
df_dataset_final.duplicated()
```

Out[64]:
```
0      False
1      False
2      False
3      False
4      False
       ...
245    False
246    False
247    False
248    False
249    False
Length: 250, dtype: bool
```

In [67]:
```
df_dataset_final.isnull().sum()*100/len(df_dataset_final)
```

Out[67]:
```
title                0.0
studio               0.0
domestic_gross_x     0.0
foreign_gross        0.0
year                 0.0
id_x                 0.0
rating               0.0
genre                0.0
runtime              0.0
id_y                 0.0
release_date         0.0
production_budget    0.0
worldwide_gross      0.0
dtype: float64
```

In [68]:
```
#Lets drop the null values in column foreign gross because a small % and i do
df_dataset_final.dropna(axis = 0, inplace = True)
df_dataset_final.isnull().sum()*100/len(df_dataset_final)
```

Out[68]:
```
title                0.0
studio               0.0
domestic_gross_x     0.0
foreign_gross        0.0
year                 0.0
id_x                 0.0
rating               0.0
genre                0.0
runtime              0.0
id_y                 0.0
release_date         0.0
production_budget    0.0
worldwide_gross      0.0
dtype: float64
```

Now,this is how the final dataset looking like:

In [69]:
```
df_dataset_final
```

Out[69]:

	title	studio	domestic_gross_x	foreign_gross	year	id_x	rating	genre	rur
0	Inception	WB	292600000.0	535700000.0	2010	611	R	Drama|Mystery and Suspense	mi
1	Due Date	WB	100500000.0	111200000.0	2010	611	R	Drama|Mystery and Suspense	mi
2	Yogi Bear	WB	100200000.0	101300000.0	2010	611	R	Drama|Mystery and Suspense	mi
3	The Book of Eli	WB	94800000.0	62300000.0	2010	611	R	Drama|Mystery and Suspense	mi
4	The Town	WB	92200000.0	61800000.0	2010	611	R	Drama|Mystery and Suspense	mi
...	...	...	...	...	...	...	...	...	
245	The Witch	A24	25100000.0	15300000.0	2016	1399	R	Drama|Horror	mi
246	American Honey	A24	663000.0	1200000.0	2016	1399	R	Drama|Horror	mi
247	Lady Bird	A24	49000000.0	30000000.0	2017	1399	R	Drama|Horror	mi
248	The Disaster Artist	A24	21100000.0	8700000.0	2017	1399	R	Drama|Horror	mi
249	Hereditary	A24	44100000.0	35300000.0	2018	1399	R	Drama|Horror	mi

250 rows × 13 columns

In [70]:
```
df_dataset_final.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 250 entries, 0 to 249
Data columns (total 13 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   title              250 non-null    object 
 1   studio             250 non-null    object 
 2   domestic_gross_x   250 non-null    float64
 3   foreign_gross      250 non-null    float64
 4   year               250 non-null    int64  
 5   id_x               250 non-null    int64  
 6   rating             250 non-null    object 
 7   genre              250 non-null    object 
 8   runtime            250 non-null    object 
 9   id_y               250 non-null    int64  
 10  release_date       250 non-null    object 
 11  production_budget  250 non-null    int64  
 12  worldwide_gross    250 non-null    int64  
dtypes: float64(2), int64(5), object(6)
memory usage: 27.3+ KB
```

In [ ]:
```

```

In [ ]:
```

```

QUESTION ONE

1. How does the release month of a movie affect its gross values i.e Worldwide and Domestic?

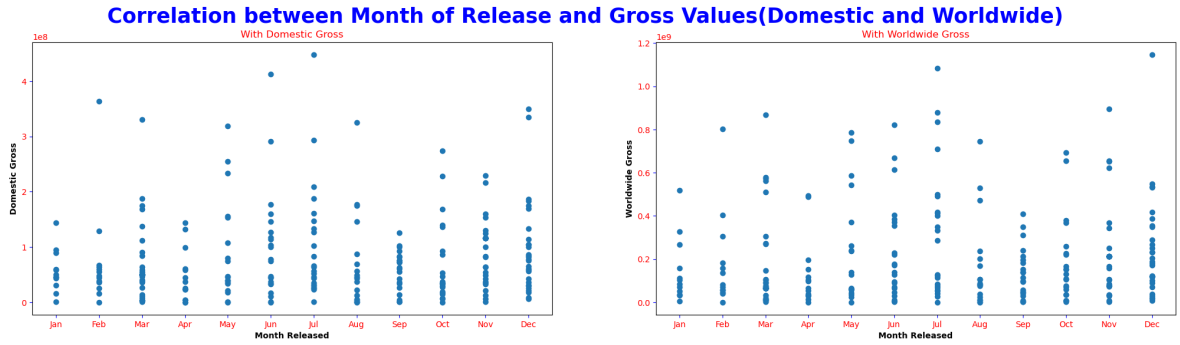
In [72]:
```
#Lets first change the release_date column to datetime format
df_dataset_final['release_date'] = pd.to_datetime(df_dataset_final['release_d
#create new variables month that contain the month information from the releas
#in integer and abbreviated string formats
month = df_dataset_final['release_date'].dt.month
month_abbr = df_dataset_final['release_date'].dt.strftime('%b')

fig, (ax1,ax2) = plt.subplots(ncols=2, figsize=(25,6))
fig.suptitle('Correlation between Month of Release and Gross Values(Domestic a

#Plotting ax1
ax1.scatter(month, df_dataset_final['domestic_gross_x'])
ax1.tick_params(color='blue', labelcolor='red')
ax1.set_title('With Domestic Gross', color ='red')
ax1.set_xlabel('Month Released',fontweight='bold')
ax1.set_ylabel('Domestic Gross',fontweight='bold')
ax1.set_xticks(month.unique())
ax1.set_xticklabels(month_abbr.unique())

#Plotting ax2
ax2.scatter(month, df_dataset_final['worldwide_gross'])
ax2.tick_params(color='blue', labelcolor='red')
ax2.set_title('With Worldwide Gross', color='red')
ax2.set_xlabel('Month Released',fontweight='bold')
ax2.set_ylabel('Worldwide Gross',fontweight='bold')
ax2.set_xticks(month.unique())
ax2.set_xticklabels(month_abbr.unique())
```

Out[72]:
```
[Text(7, 0, 'Jul'),
 Text(11, 0, 'Nov'),
 Text(12, 0, 'Dec'),
 Text(1, 0, 'Jan'),
 Text(9, 0, 'Sep'),
 Text(10, 0, 'Oct'),
 Text(2, 0, 'Feb'),
 Text(4, 0, 'Apr'),
 Text(6, 0, 'Jun'),
 Text(5, 0, 'May'),
 Text(3, 0, 'Mar'),
 Text(8, 0, 'Aug')]
```



December and July looks like good months to release the films.

In [ ]:

In [ ]:

QUESTION TWO

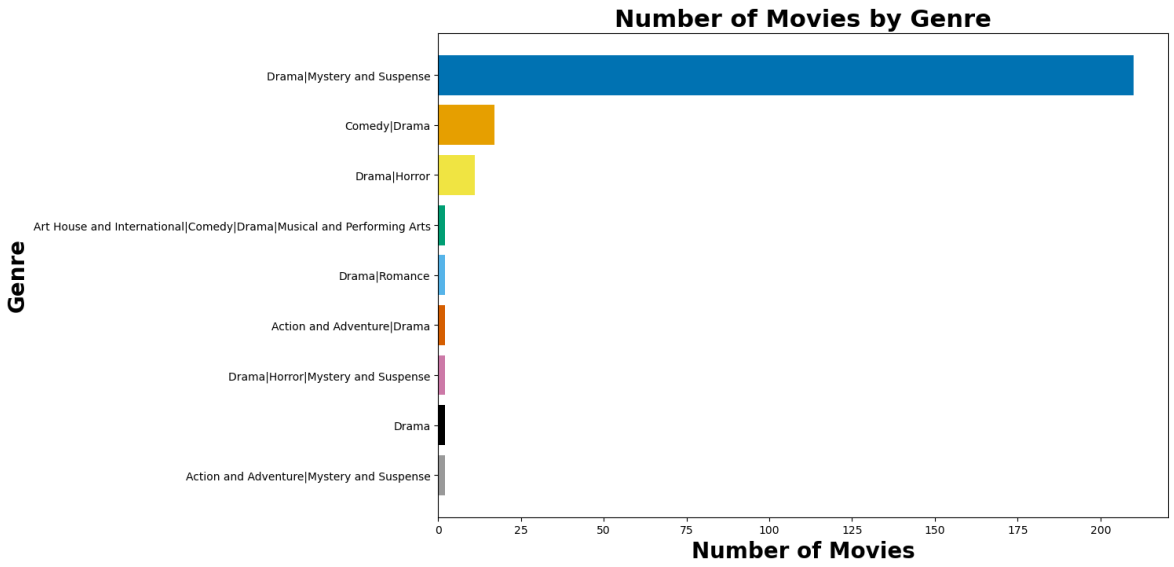
2.Which is the most popular genre ?

In [73]:
```
#Obtaining value counts
genre_counts = df_dataset_final['genre'].value_counts()
genre_counts
```

Out[73]:
```
Drama|Mystery and Suspense                                                  210
Comedy|Drama                                                                17
Drama|Horror                                                                11
Art House and International|Comedy|Drama|Musical and Performing Arts         2
Drama|Romance                                                               2
Action and Adventure|Drama                                                  2
Drama|Horror|Mystery and Suspense                                           2
Drama                                                                       2
Action and Adventure|Mystery and Suspense                                   2
Name: genre, dtype: int64
```

In [74]:
```
# Create a bar plot of the data
# Set the axis labels and title
genre_counts = dict(df_dataset_final['genre'].value_counts())
fig, ax = plt.subplots(figsize=(12,8))
ax.barh(list(genre_counts.keys()), genre_counts.values(),color=['#0072b2', '#
ax.invert_yaxis()
plt.xlabel('Number of Movies',fontweight='bold',size = '20')
plt.ylabel('Genre',fontweight='bold',size = '20')
plt.title('Number of Movies by Genre',fontweight='bold',size = '22')
```

Out[74]: Text(0.5, 1.0, 'Number of Movies by Genre')



The most popular genre is Drama|Mystery and Suspence

In [ ]:

QUESTION THREE

3.Which Studios are making the highest profit?

In [75]:
```
# First, i calculate the profit for each movie and storing the values in a ne
df_dataset_final['profit'] = df_dataset_final['worldwide_gross'] - df_dataset

# Then, group the movies by their respective studios and sum up the profits
studio_profit = df_dataset_final.groupby('studio')['profit'].sum()

# Finally, i sort the data by profit in descending order
studio_profit = studio_profit.sort_values(ascending=False)

# Print the resulting DataFrame
print(studio_profit)
```

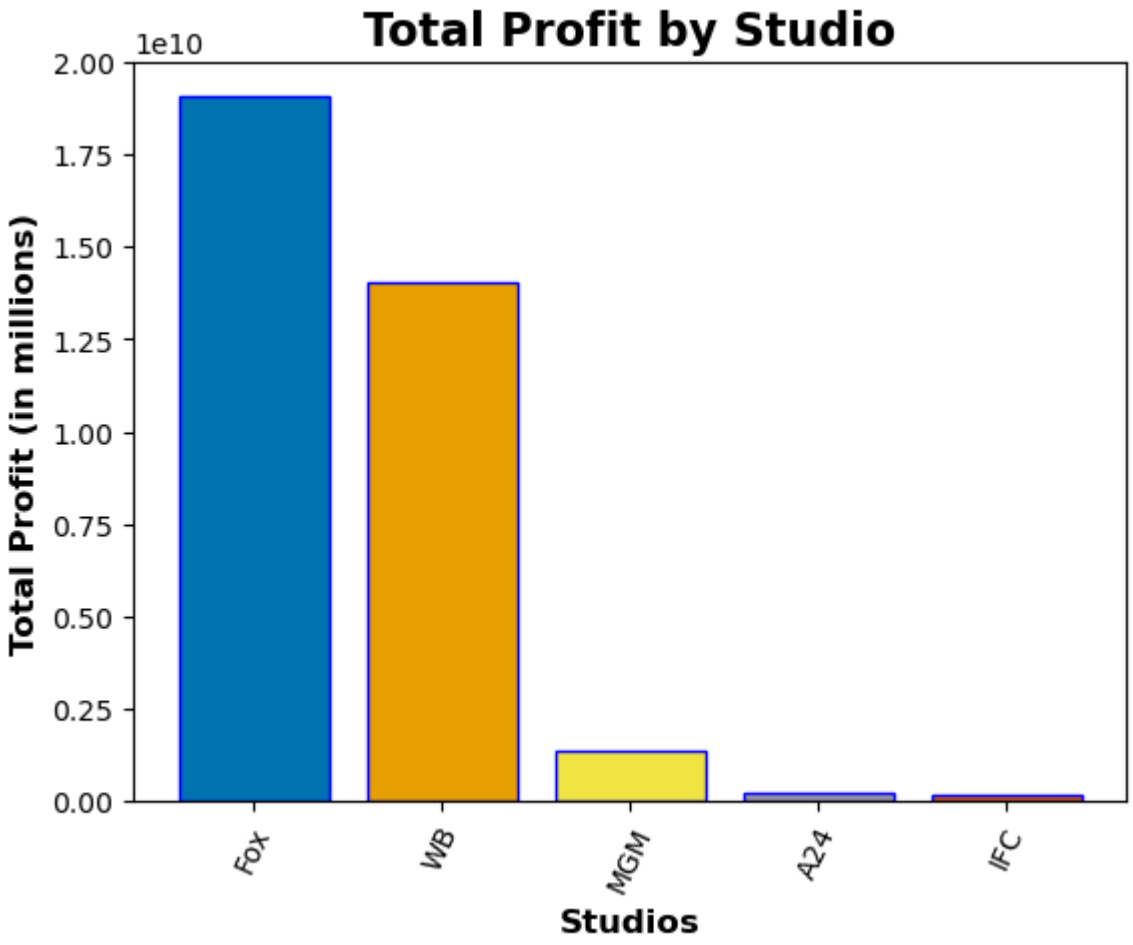
```
studio
Fox    19054087780
WB     14038224007
MGM     1354914904
A24      249329992
IFC      156825818
Name: profit, dtype: int64
```

In [76]:
```
# Plot a bar chart of the studio profits
plt.bar(studio_profit.index, studio_profit.values,color=['#0072b2', '#e69f00'

# Add labels and title to the chart, and customize their appearance
plt.xlabel('Studios', fontsize=12, fontweight='bold')
plt.ylabel('Total Profit (in millions)', fontsize=12, fontweight='bold')
plt.title('Total Profit by Studio', fontsize=16, fontweight='bold')

# Rotate the x-axis labels for better visibility
plt.xticks(rotation=65)

# Display the chart
plt.show()
```



The Studios that will be major competitors of Microsoft are FOX,WB,MGM,A24 and IFC.
This is because they are the leading Studios in terms of profit making in the Film Industry.

In [ ]:

RECOMENDATIONS

All said and done,After my analysis, these would be my data driven recommendations to Microsoft:

1.Know your audience,how so you may ask,the top film genres overall were Drama|Mystery and Suspence.
Microsoft should then consider creating movies within these genre parameters if they want to generate interest and increase their likelihood of achieving high popularity since these kind of films draw the crowds.

2.The highest profiting studios have the best business models that should inform Microsofts practice for their own studio.
The business strategies of these studios should be carefully scrutinized to determine and emulate their methods for success.
These studios include FOX,WB,MGM,A24 and IFC.

3.Finally the findings of the study presented above lead one to the conclusion that there is, in fact,
a positive link between the release month/period and the gross values i.e Domestic and Worldwide.
The films released in the month of July and December generally grossed higher compared to other months.

In [ ]:

NEXT STEPS

Its said that the core advantage of data is that it tell us something about a world we did not know of before.
More analysis can still be done on issues like(but not limited to) the Film directors ,the Runtime of each Film and even the Ratings.

With the above analysis and recomendations,Microsoft should be now ready to enter into the Film industry head high with data driven decisions.

END.
THANK YOU.

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: