In [1]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
%matplotlib inline
df1 = pd.read_csv('bom.movie_gross.csv')
df1
```

Out[1]:

	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

In [2]:
```
# Checking the data information
df1.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
```

In [3]:
```
df2= pd.read_csv('tn.movie_budgets.csv')
df2
```

Out[3]:

	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 [4]:
```
df3 = pd.read_csv('title.basics.csv')
df3
```

Out[4]:

	tconst	primary_title	original_title	start_year	runtime_minutes	
0	tt0063540	Sunghursh	Sunghursh	2013	175.0	Action,Crim
1	tt0066787	One Day Before the Rainy Season	Ashad Ka Ek Din	2019	114.0	Biograpl
2	tt0069049	The Other Side of the Wind	The Other Side of the Wind	2018	122.0	
3	tt0069204	Sabse Bada Sukh	Sabse Bada Sukh	2018	NaN	Comec
4	tt0100275	The Wandering Soap Opera	La Telenovela Errante	2017	80.0	Comedy,Drama
...	...	...	...	...	...	
146139	tt9916538	Kuambil Lagi Hatiku	Kuambil Lagi Hatiku	2019	123.0	
146140	tt9916622	Rodolpho Teóphilo - O Legado de um Pioneiro	Rodolpho Teóphilo - O Legado de um Pioneiro	2015	NaN	Doc
146141	tt9916706	Dankyavar Danka	Dankyavar Danka	2013	NaN	
146142	tt9916730	6 Gunn	6 Gunn	2017	116.0	
146143	tt9916754	Chico Albuquerque - Revelações	Chico Albuquerque - Revelações	2013	NaN	Doc

146144 rows × 6 columns

In [5]:
```
# Merging datasets

df4 = pd.concat([df1, df2], axis=0)
df5 = df4.merge(df3, left_index=True, right_index=True, how='outer')
df5
```

Out[5]:

	title	studio	domestic_gross	foreign_gross	year	id	release_date
0	Toy Story 3	BV	415000000.0	652000000	2010.0	NaN	NaN
0	NaN	NaN	$760,507,625	NaN	NaN	1.0	Dec 18, 2009
1	Alice in Wonderland (2010)	BV	334200000.0	691300000	2010.0	NaN	NaN
1	NaN	NaN	$241,063,875	NaN	NaN	2.0	May 20, 2011
2	Harry Potter and the Deathly Hallows Part 1	WB	296000000.0	664300000	2010.0	NaN	NaN
...	...	...	...	...	...	...	...
146139	NaN	NaN	NaN	NaN	NaN	NaN	NaN
146140	NaN	NaN	NaN	NaN	NaN	NaN	NaN
146141	NaN	NaN	NaN	NaN	NaN	NaN	NaN
146142	NaN	NaN	NaN	NaN	NaN	NaN	NaN
146143	NaN	NaN	NaN	NaN	NaN	NaN	NaN

149531 rows × 16 columns

Data Cleaning the Merged Datasets

In [6]:
```
# Data preview of the merged datasets
df5.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 149531 entries, 0 to 146143
Data columns (total 16 columns):
 #   Column             Non-Null Count   Dtype  
---  ------             --------------   -----  
 0   title              3387 non-null    object 
 1   studio             3382 non-null    object 
 2   domestic_gross     9141 non-null    object 
 3   foreign_gross      2037 non-null    object 
 4   year               3387 non-null    float64
 5   id                 5782 non-null    float64
 6   release_date       5782 non-null    object 
 7   movie              5782 non-null    object 
 8   production_budget  5782 non-null    object 
 9   worldwide_gross    5782 non-null    object 
 10  tconst             149531 non-null  object 
 11  primary_title      149530 non-null  object 
 12  original_title     149509 non-null  object 
 13  start_year         149531 non-null  int64  
 14  runtime_minutes    116373 non-null  float64
 15  genres             143946 non-null  object 
dtypes: float64(3), int64(1), object(12)
memory usage: 19.4+ MB
```

In [7]:
```
# Determining the shape of the merged datasets
df5.shape
```

Out[7]:
```
(149531, 16)
```

In [9]:
```
# Finding out the columns of the DataFrame
df5.columns
```

Out[9]:
```
Index(['title', 'studio', 'domestic_gross', 'foreign_gross', 'year',
'id',
       'release_date', 'movie', 'production_budget', 'worldwide_gros
s',
       'tconst', 'primary_title', 'original_title', 'start_year',
       'runtime_minutes', 'genres'],
      dtype='object')
```

In [10]:
```
# Dropping irrelevant columns
df5.drop(['id', 'year', 'tconst', 'domestic_gross', 'foreign_gross', 'p
```

In [11]:
```
# Checking the remaining columns after scraping irrelevant columns
df5.columns
```

Out[11]:
```
Index(['studio', 'release_date', 'movie', 'production_budget',
       'worldwide_gross', 'original_title', 'start_year', 'runtime_min
utes',
       'genres'],
      dtype='object')
```

In [12]:
```
df5.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 149531 entries, 0 to 146143
Data columns (total 9 columns):
 #   Column             Non-Null Count   Dtype  
---  ------             --------------   -----  
 0   studio             3382 non-null    object 
 1   release_date       5782 non-null    object 
 2   movie              5782 non-null    object 
 3   production_budget  5782 non-null    object 
 4   worldwide_gross    5782 non-null    object 
 5   original_title     149509 non-null  object 
 6   start_year         149531 non-null  int64  
 7   runtime_minutes    116373 non-null  float64
 8   genres             143946 non-null  object 
dtypes: float64(1), int64(1), object(7)
memory usage: 11.4+ MB
```

In [13]:
```
# Using .str() and .replace() to take out the $ and , from prodution bu
df5['production_budget'] = df5['production_budget'].str.replace('$', ''
df5['production_budget'] = df5['production_budget'].str.replace(',', ''
df5['worldwide_gross'] = df5['worldwide_gross'].str.replace('$', '')
df5['worldwide_gross'] = df5['worldwide_gross'].str.replace(',', '')
df5
```

Out[13]:

	studio	release_date	movie	production_budget	worldwide_gross	original_ti
0	BV	NaN	NaN	NaN	NaN	Sunghu
0	NaN	Dec 18, 2009	Avatar	425000000	2776345279	Sunghu
1	BV	NaN	NaN	NaN	NaN	Ashad Ka
1	NaN	May 20, 2011	Pirates of the Caribbean: On Stranger Tides	410600000	1045663875	Ashad Ka
2	WB	NaN	NaN	NaN	NaN	The Otl Side of W
...	...	...	...	...	...	
146139	NaN	NaN	NaN	NaN	NaN	Kuambil L Hat
146140	NaN	NaN	NaN	NaN	NaN	Rodolp Teóphilo - Legado um Pione
146141	NaN	NaN	NaN	NaN	NaN	Dankya Dar
146142	NaN	NaN	NaN	NaN	NaN	6 Gu
146143	NaN	NaN	NaN	NaN	NaN	Ch Albuquerc - Revelaçõ

149531 rows × 9 columns

In [14]:
```
# Replacing the NaN values with 0 to create a leeway to integer convers
df5['production_budget'] = df5['production_budget'].fillna(0)
df5['worldwide_gross'] = df5['worldwide_gross'].fillna(0)
df5
```

Out[14]:

	studio	release_date	movie	production_budget	worldwide_gross	original_ti
0	BV	NaN	NaN	0	0	Sunghu
0	NaN	Dec 18, 2009	Avatar	425000000	2776345279	Sunghu
1	BV	NaN	NaN	0	0	Ashad Ka
1	NaN	May 20, 2011	Pirates of the Caribbean: On Stranger Tides	410600000	1045663875	Ashad Ka
2	WB	NaN	NaN	0	0	The Otl Side of W
...	...	...	...	...	...	
146139	NaN	NaN	NaN	0	0	Kuambil L Hat
146140	NaN	NaN	NaN	0	0	Rodolp Teóphilo - Legado um Pione
146141	NaN	NaN	NaN	0	0	Dankya Dar
146142	NaN	NaN	NaN	0	0	6 Gu
146143	NaN	NaN	NaN	0	0	Ch Albuquerc - Revelaçõ

149531 rows × 9 columns

In [15]:
```
# Converting dtypes of production budget and worldwide gross from objec
df5['production_budget'] = df5['production_budget'].astype(float)
df5['worldwide_gross'] = df5['worldwide_gross'].astype(float)
df5.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 149531 entries, 0 to 146143
Data columns (total 9 columns):
 #   Column             Non-Null Count   Dtype  
---  ------             --------------   -----  
 0   studio             3382 non-null    object 
 1   release_date       5782 non-null    object 
 2   movie              5782 non-null    object 
 3   production_budget  149531 non-null  float64
 4   worldwide_gross    149531 non-null  float64
 5   original_title     149509 non-null  object 
 6   start_year         149531 non-null  int64  
 7   runtime_minutes    116373 non-null  float64
 8   genres             143946 non-null  object 
dtypes: float64(3), int64(1), object(5)
memory usage: 11.4+ MB
```

In [16]:
```
# Checking the sum of missing data
df5.isna().sum()
```

Out[16]:
```
studio               146149
release_date         143749
movie                143749
production_budget         0
worldwide_gross           0
original_title           22
start_year                0
runtime_minutes       33158
genres                 5585
dtype: int64
```

In [17]:
```
# Checking the percentage of missing data to determine which need to be
df5.isna().mean()
```

Out[17]:
```
studio               0.977383
release_date         0.961332
movie                0.961332
production_budget    0.000000
worldwide_gross      0.000000
original_title       0.000147
start_year           0.000000
runtime_minutes      0.221747
genres               0.037350
dtype: float64
```

In [18]:
```
df5.drop(['studio', 'release_date', 'movie'], axis = 1, inplace = True)
```

In [19]:
```
# Checking our data
df5.columns
```

Out[19]:
```
Index(['production_budget', 'worldwide_gross', 'original_title', 'star
t_year',
       'runtime_minutes', 'genres'],
      dtype='object')
```

In [20]:
```
df5.describe()
```

Out[20]:

	production_budget	worldwide_gross	start_year	runtime_minutes
count	1.495310e+05	1.495310e+05	149531.000000	116373.000000
mean	1.221422e+06	3.537598e+06	2014.675907	86.224915
std	1.023131e+07	3.861793e+07	2.766890	164.993271
min	0.000000e+00	0.000000e+00	2010.000000	1.000000
25%	0.000000e+00	0.000000e+00	2012.000000	70.000000
50%	0.000000e+00	0.000000e+00	2015.000000	87.000000
75%	0.000000e+00	0.000000e+00	2017.000000	99.000000
max	4.250000e+08	2.776345e+09	2115.000000	51420.000000

In [23]:
```
df5.to_csv('dataset1.csv', index=False)
```