Submission Info

- Student name: Kamile Yagci
- Student pace: self paced
- Scheduled project review date/time: 07/01/2021 at 12:00 PM CT
- Instructor name: Jeff Herman
- Blog post URL: https://kamileyagci.github.io/Movie-Industry-Study/ (https://kamileyagci.github.io/Movie-Industry-Study/)

Part 1 Content

- Import Libraries
- Explore Data Files
- Define Business Questions
- List Data Files to be used
- Merge Data
- Clean Data
- Save Data

Import Libraries

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

Explore Data Files

In [2]:
```
df_bomMovieGross = pd.read_csv('zippedData/bom.movie_gross.csv.gz')
df_bomMovieGross.info()
df_bomMovieGross.head()
```

```
<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
```

Out[2]:

	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

In [3]:
```
df_imdbNameBasics = pd.read_csv('zippedData/imdb.name.basics.csv.gz')
df_imdbNameBasics.info()
df_imdbNameBasics.head()
```

```
<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
```

Out[3]:

	nconst	primary_name	birth_year	death_year	primary_professio
0	nm0061671	Mary Ellen Bauder	NaN	NaN	miscellaneous,production_manager,produce
1	nm0061865	Joseph Bauer	NaN	NaN	composer,music_department,sound_departmen
2	nm0062070	Bruce Baum	NaN	NaN	miscellaneous,actor,write
3	nm0062195	Axel Baumann	NaN	NaN	camera_department,cinematographer,art_departmen
4	nm0062798	Pete Baxter	NaN	NaN	production_designer,art_department,set_decorato

In [4]:
```
df_imdbTitleAkas = pd.read_csv('zippedData/imdb.title.akas.csv.gz')
df_imdbTitleAkas.info()
df_imdbTitleAkas.head()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 331703 entries, 0 to 331702
Data columns (total 8 columns):
 #   Column             Non-Null Count   Dtype  
---  ------             --------------   -----  
 0   title_id           331703 non-null  object 
 1   ordering           331703 non-null  int64  
 2   title              331703 non-null  object 
 3   region             278410 non-null  object 
 4   language           41715 non-null   object 
 5   types              168447 non-null  object 
 6   attributes         14925 non-null   object 
 7   is_original_title  331678 non-null  float64
dtypes: float64(1), int64(1), object(6)
memory usage: 20.2+ MB
```

Out[4]:

	title_id	ordering	title	region	language	types	attributes	is_original_title
0	tt0369610	10	Джурасик свят	BG	bg	NaN	NaN	0.0
1	tt0369610	11	Jurashikku warudo	JP	NaN	imdbDisplay	NaN	0.0
2	tt0369610	12	Jurassic World: O Mundo dos Dinossauros	BR	NaN	imdbDisplay	NaN	0.0
3	tt0369610	13	O Mundo dos Dinossauros	BR	NaN	NaN	short title	0.0
4	tt0369610	14	Jurassic World	FR	NaN	imdbDisplay	NaN	0.0

In [5]:
```
df_imdbTitleBasics = pd.read_csv('zippedData/imdb.title.basics.csv.gz')
df_imdbTitleBasics.info()
df_imdbTitleBasics.head()
```

```
<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
```

Out[5]:

	tconst	primary_title	original_title	start_year	runtime_minutes	genres
0	tt0063540	Sunghursh	Sunghursh	2013	175.0	Action,Crime,Drama
1	tt0066787	One Day Before the Rainy Season	Ashad Ka Ek Din	2019	114.0	Biography,Drama
2	tt0069049	The Other Side of the Wind	The Other Side of the Wind	2018	122.0	Drama
3	tt0069204	Sabse Bada Sukh	Sabse Bada Sukh	2018	NaN	Comedy,Drama
4	tt0100275	The Wandering Soap Opera	La Telenovela Errante	2017	80.0	Comedy,Drama,Fantasy

In [6]:
```
df_imdbTitleCrew = pd.read_csv('zippedData/imdb.title.crew.csv.gz')
df_imdbTitleCrew.info()
df_imdbTitleCrew.head()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 146144 entries, 0 to 146143
Data columns (total 3 columns):
 #   Column     Non-Null Count   Dtype 
---  ------     --------------   ----- 
 0   tconst     146144 non-null  object
 1   directors  140417 non-null  object
 2   writers    110261 non-null  object
dtypes: object(3)
memory usage: 3.3+ MB
```

Out[6]:

	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

In [7]:
```
df_imdbTitlePrincipals = pd.read_csv('zippedData/imdb.title.principals.csv.
df_imdbTitlePrincipals.info()
df_imdbTitlePrincipals.head()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1028186 entries, 0 to 1028185
Data columns (total 6 columns):
 #   Column      Non-Null Count    Dtype 
---  ------      --------------    ----- 
 0   tconst      1028186 non-null  object
 1   ordering    1028186 non-null  int64 
 2   nconst      1028186 non-null  object
 3   category    1028186 non-null  object
 4   job         177684 non-null   object
 5   characters  393360 non-null   object
dtypes: int64(1), object(5)
memory usage: 47.1+ MB
```

Out[7]:

	tconst	ordering	nconst	category	job	characters
0	tt0111414	1	nm0246005	actor	NaN	["The Man"]
1	tt0111414	2	nm0398271	director	NaN	NaN
2	tt0111414	3	nm3739909	producer	producer	NaN
3	tt0323808	10	nm0059247	editor	NaN	NaN
4	tt0323808	1	nm3579312	actress	NaN	["Beth Boothby"]

In [8]:
```
df_imdbTitleRatings = pd.read_csv('zippedData/imdb.title.ratings.csv.gz')
df_imdbTitleRatings.info()
df_imdbTitleRatings.head()
```

```
<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
```

Out[8]:

	tconst	averagerating	numvotes
0	tt10356526	8.3	31
1	tt10384606	8.9	559
2	tt1042974	6.4	20
3	tt1043726	4.2	50352
4	tt1060240	6.5	21

In [9]:
```
df_rtMovieInfo = pd.read_csv('zippedData/rt.movie_info.tsv.gz', sep='\t', e
df_rtMovieInfo.info()
df_rtMovieInfo.head()
```

```
<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
```

Out[9]:

	id	synopsis	rating	genre	director	writer	theater_date	dvd_
0	1	This gritty, fast-paced, and innovative police...	R	Action and Adventure|Classics|Drama	William Friedkin	Ernest Tidyman	Oct 9, 1971	Se
1	3	New York City, not-too-distant-future: Eric Pa...	R	Drama|Science Fiction and Fantasy	David Cronenberg	David Cronenberg|Don DeLillo	Aug 17, 2012	J
2	5	Illeana Douglas delivers a superb performance ...	R	Drama|Musical and Performing Arts	Allison Anders	Allison Anders	Sep 13, 1996	Ap
3	6	Michael Douglas runs afoul of a treacherous su...	R	Drama|Mystery and Suspense	Barry Levinson	Paul Attanasio|Michael Crichton	Dec 9, 1994	Au
4	7	NaN	NR	Drama|Romance	Rodney Bennett	Giles Cooper	NaN	

In [10]:
```
df_rtReviews = pd.read_csv('zippedData/rt.reviews.tsv.gz', sep='\t', encodi
df_rtReviews.info()
df_rtReviews.head()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 54432 entries, 0 to 54431
Data columns (total 8 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   id          54432 non-null  int64 
 1   review      48869 non-null  object
 2   rating      40915 non-null  object
 3   fresh       54432 non-null  object
 4   critic      51710 non-null  object
 5   top_critic  54432 non-null  int64 
 6   publisher   54123 non-null  object
 7   date        54432 non-null  object
dtypes: int64(2), object(6)
memory usage: 3.3+ MB
```

Out[10]:

	id	review	rating	fresh	critic	top_critic	publisher	date
0	3	A distinctly gallows take on contemporary fina...	3/5	fresh	PJ Nabarro	0	Patrick Nabarro	November 10, 2018
1	3	It's an allegory in search of a meaning that n...	NaN	rotten	Annalee Newitz	0	io9.com	May 23, 2018
2	3	... life lived in a bubble in financial dealin...	NaN	fresh	Sean Axmaker	0	Stream on Demand	January 4, 2018
3	3	Continuing along a line introduced in last yea...	NaN	fresh	Daniel Kasman	0	MUBI	November 16, 2017
4	3	... a perverse twist on neorealism...	NaN	fresh	NaN	0	Cinema Scope	October 12, 2017

In [11]:
```
df_tmdbMovies = pd.read_csv('zippedData/tmdb.movies.csv.gz')
df_tmdbMovies.info()
df_tmdbMovies.head()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 26517 entries, 0 to 26516
Data columns (total 10 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Unnamed: 0         26517 non-null  int64  
 1   genre_ids          26517 non-null  object 
 2   id                 26517 non-null  int64  
 3   original_language  26517 non-null  object 
 4   original_title     26517 non-null  object 
 5   popularity         26517 non-null  float64
 6   release_date       26517 non-null  object 
 7   title              26517 non-null  object 
 8   vote_average       26517 non-null  float64
 9   vote_count         26517 non-null  int64  
dtypes: float64(2), int64(3), object(5)
memory usage: 2.0+ MB
```

Out[11]:

	Unnamed: 0	genre_ids	id	original_language	original_title	popularity	release_date	title
0	0	[12, 14, 10751]	12444	en	Harry Potter and the Deathly Hallows: Part 1	33.533	2010-11-19	Harry Potter and the Deathly Hallows: Part 1
1	1	[14, 12, 16, 10751]	10191	en	How to Train Your Dragon	28.734	2010-03-26	How to Train Your Dragon
2	2	[12, 28, 878]	10138	en	Iron Man 2	28.515	2010-05-07	Iron Man 2
3	3	[16, 35, 10751]	862	en	Toy Story	28.005	1995-11-22	Toy Story
4	4	[28, 878, 12]	27205	en	Inception	27.920	2010-07-16	Inception

In [6]:
```
df_tnMovieBudgets = pd.read_csv('zippedData/tn.movie_budgets.csv.gz')
df_tnMovieBudgets.info()
df_tnMovieBudgets.head()
```

```
<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   object
 4   domestic_gross     5782 non-null   object
 5   worldwide_gross    5782 non-null   object
dtypes: int64(1), object(5)
memory usage: 271.2+ KB
```

Out[6]:

	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

Define Business Questions

1. Is there a correlation between the production budget and the profit?
2. Is there a correlation between the ratings and the profit?
3. Which directors, writers, actors and actresses make the most profit?
4. Which genres make the most profit?

List Data Files to be used

- tn.movie_budgets.csv.gz
- imdb.title.basics.csv.gz
- imdb.title.ratings.csv.gz
- imdb.title.principals.csv.gz
- imdb.name.basics.csv.gz

Merge Data

In [13]:
```
# Merge the Data files to be used
df_Merged = pd.merge(df_tnMovieBudgets, df_imdbTitleBasics, left_on="movie"
df_Merged = pd.merge(df_Merged, df_imdbTitleRatings, on="tconst")
df_Merged = pd.merge(df_Merged, df_imdbTitlePrincipals, on="tconst")
df_Merged = pd.merge(df_Merged, df_imdbNameBasics, on="nconst")
df_Merged.info()
df_Merged.tail()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 24983 entries, 0 to 24982
Data columns (total 24 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   id                  24983 non-null  int64
 1   release_date        24983 non-null  object
 2   movie               24983 non-null  object
 3   production_budget   24983 non-null  object
 4   domestic_gross      24983 non-null  object
 5   worldwide_gross     24983 non-null  object
 6   tconst              24983 non-null  object
 7   primary_title       24983 non-null  object
 8   original_title      24983 non-null  object
 9   start_year          24983 non-null  int64
 10  runtime_minutes     24117 non-null  float64
 11  genres              24943 non-null  object
 12  averagerating       24983 non-null  float64
 13  numvotes            24983 non-null  int64
 14  ordering            24983 non-null  int64
 15  nconst              24983 non-null  object
 16  category            24983 non-null  object
 17  job                 8374 non-null   object
 18  characters          10118 non-null  object
 19  primary_name        24983 non-null  object
 20  birth_year          13108 non-null  float64
 21  death_year          520 non-null    float64
 22  primary_profession  24721 non-null  object
 23  known_for_titles    24906 non-null  object
dtypes: float64(4), int64(4), object(16)
memory usage: 4.8+ MB
```

Out[13]:

	id	release_date	movie	production_budget	domestic_gross	worldwide_gross	tcons
24978	81	Sep 29, 2015	A Plague So Pleasant	$1,400	$0	$0	tt210764
24979	81	Sep 29, 2015	A Plague So Pleasant	$1,400	$0	$0	tt210764
24980	81	Sep 29, 2015	A Plague So Pleasant	$1,400	$0	$0	tt210764

	id	release_date	movie	production_budget	domestic_gross	worldwide_gross	tcons
24981	81	Sep 29, 2015	A Plague So Pleasant	$1,400	$0	$0	tt210764
24982	81	Sep 29, 2015	A Plague So Pleasant	$1,400	$0	$0	tt210764

5 rows × 24 columns

Clean Data

In [14]:
```
# Remove the unnecessary columns
drop_Columns = ['id', 'primary_title', 'original_title', 'runtime_minutes',
                'job', 'start_year', 'birth_year', 'primary_profession', 'k
df_Merged.drop(columns = drop_Columns, inplace=True)
df_Merged.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 24983 entries, 0 to 24982
Data columns (total 14 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   release_date       24983 non-null  object
 1   movie              24983 non-null  object
 2   production_budget  24983 non-null  object
 3   domestic_gross     24983 non-null  object
 4   worldwide_gross    24983 non-null  object
 5   tconst             24983 non-null  object
 6   genres             24943 non-null  object
 7   averagerating      24983 non-null  float64
 8   numvotes           24983 non-null  int64
 9   nconst             24983 non-null  object
 10  category           24983 non-null  object
 11  characters         10118 non-null  object
 12  primary_name       24983 non-null  object
 13  death_year         520 non-null    float64
dtypes: float64(2), int64(1), object(11)
memory usage: 2.9+ MB
```

In [15]:
```
# Remove rows with $0 worldwide_gross
df_Merged.drop(df_Merged[df_Merged.worldwide_gross == '$0'].index, inplace=
df_Merged.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 22676 entries, 0 to 24971
Data columns (total 14 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   release_date       22676 non-null  object 
 1   movie              22676 non-null  object 
 2   production_budget  22676 non-null  object 
 3   domestic_gross     22676 non-null  object 
 4   worldwide_gross    22676 non-null  object 
 5   tconst             22676 non-null  object 
 6   genres             22636 non-null  object 
 7   averagerating      22676 non-null  float64
 8   numvotes           22676 non-null  int64  
 9   nconst             22676 non-null  object 
 10  category           22676 non-null  object 
 11  characters         9158 non-null   object 
 12  primary_name       22676 non-null  object 
 13  death_year         492 non-null    float64
dtypes: float64(2), int64(1), object(11)
memory usage: 2.6+ MB
```

In [16]:
```
df_Merged[df_Merged.production_budget == '$0']
```

Out[16]:

release_date	movie	production_budget	domestic_gross	worldwide_gross	tconst	genres	averag

In [17]:
```
# Change the currency columns to float
df_Merged['production_budget'] = df_Merged['production_budget'].str.replace
df_Merged['production_budget'] = df_Merged['production_budget'].str.replace
df_Merged['production_budget'] = df_Merged['production_budget'].astype(floa

df_Merged['domestic_gross'] = df_Merged['domestic_gross'].str.replace(",",
df_Merged['domestic_gross'] = df_Merged['domestic_gross'].str.replace("$",
df_Merged['domestic_gross'] = df_Merged['domestic_gross'].astype(float)

df_Merged['worldwide_gross'] = df_Merged['worldwide_gross'].str.replace(","
df_Merged['worldwide_gross'] = df_Merged['worldwide_gross'].str.replace("$"
df_Merged['worldwide_gross'] = df_Merged['worldwide_gross'].astype(float)
```

In [18]:
```
df_Merged.info()
df_Merged.head()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 22676 entries, 0 to 24971
Data columns (total 14 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   release_date       22676 non-null  object 
 1   movie              22676 non-null  object 
 2   production_budget  22676 non-null  float64
 3   domestic_gross     22676 non-null  float64
 4   worldwide_gross    22676 non-null  float64
 5   tconst             22676 non-null  object 
 6   genres             22636 non-null  object 
 7   averagerating      22676 non-null  float64
 8   numvotes           22676 non-null  int64  
 9   nconst             22676 non-null  object 
 10  category           22676 non-null  object 
 11  characters         9158 non-null   object 
 12  primary_name       22676 non-null  object 
 13  death_year         492 non-null    float64
dtypes: float64(5), int64(1), object(8)
memory usage: 2.6+ MB
```

Out[18]:

	release_date	movie	production_budget	domestic_gross	worldwide_gross	tconst	
0	May 20, 2011	Pirates of the Caribbean: On Stranger Tides	410600000.0	241063875.0	1.045664e+09	tt1298650	Actior
1	May 20, 2011	Pirates of the Caribbean: On Stranger Tides	410600000.0	241063875.0	1.045664e+09	tt1298650	Actior
2	Jul 2, 2013	The Lone Ranger	275000000.0	89302115.0	2.600021e+08	tt1210819	Action
3	May 26, 2017	Pirates of the Caribbean: Dead Men Tell No Tales	230000000.0	172558876.0	7.882411e+08	tt1790809	Actior
4	Mar 5, 2010	Alice in Wonderland	200000000.0	334191110.0	1.025491e+09	tt1014759	Adver

In [19]:
```
# Seperate genres
df_Merged['genres'] = df_Merged['genres'].str.split(',')
df_Merged = df_Merged.explode('genres')
```

In [25]:
```
# Slice the release year from the release date
df_Merged['release_year'] = df_Merged['release_date'].str.slice(start=-4).a
```

In [26]:
```
df_Merged.info()
df_Merged.head()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 53305 entries, 0 to 24971
Data columns (total 15 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   release_date       53305 non-null  object 
 1   movie              53305 non-null  object 
 2   production_budget  53305 non-null  float64
 3   domestic_gross     53305 non-null  float64
 4   worldwide_gross    53305 non-null  float64
 5   tconst             53305 non-null  object 
 6   genres             53265 non-null  object 
 7   averagerating      53305 non-null  float64
 8   numvotes           53305 non-null  int64  
 9   nconst             53305 non-null  object 
 10  category           53305 non-null  object 
 11  characters         21488 non-null  object 
 12  primary_name       53305 non-null  object 
 13  death_year         1243 non-null   float64
 14  release_year       53305 non-null  int64  
dtypes: float64(5), int64(2), object(8)
memory usage: 7.8+ MB
```

Out[26]:

	release_date	movie	production_budget	domestic_gross	worldwide_gross	tconst	genr
0	May 20, 2011	Pirates of the Caribbean: On Stranger Tides	410600000.0	241063875.0	1.045664e+09	tt1298650	Acti
0	May 20, 2011	Pirates of the Caribbean: On Stranger Tides	410600000.0	241063875.0	1.045664e+09	tt1298650	Adventu
0	May 20, 2011	Pirates of the Caribbean: On Stranger Tides	410600000.0	241063875.0	1.045664e+09	tt1298650	Fanta
1	May 20, 2011	Pirates of the Caribbean: On Stranger Tides	410600000.0	241063875.0	1.045664e+09	tt1298650	Acti
1	May 20, 2011	Pirates of the Caribbean: On Stranger Tides	410600000.0	241063875.0	1.045664e+09	tt1298650	Adventu

Create Profit columns

In [27]:
```
#Two types of profits
df_Merged['profit_gross'] = df_Merged['worldwide_gross'] - df_Merged['produ
df_Merged['profit_rate'] = df_Merged['worldwide_gross'] / df_Merged['produc
df_Merged.info()
df_Merged.head()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 53305 entries, 0 to 24971
Data columns (total 17 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   release_date       53305 non-null  object 
 1   movie              53305 non-null  object 
 2   production_budget  53305 non-null  float64
 3   domestic_gross     53305 non-null  float64
 4   worldwide_gross    53305 non-null  float64
 5   tconst             53305 non-null  object 
 6   genres             53265 non-null  object 
 7   averagerating      53305 non-null  float64
 8   numvotes           53305 non-null  int64  
 9   nconst             53305 non-null  object 
 10  category           53305 non-null  object 
 11  characters         21488 non-null  object 
 12  primary_name       53305 non-null  object 
 13  death_year         1243 non-null   float64
 14  release_year       53305 non-null  int64  
 15  profit_gross       53305 non-null  float64
 16  profit_rate        53305 non-null  float64
dtypes: float64(7), int64(2), object(8)
memory usage: 8.6+ MB
```

Out[27]:

	release_date	movie	production_budget	domestic_gross	worldwide_gross	tconst	genr
0	May 20, 2011	Pirates of the Caribbean: On Stranger Tides	410600000.0	241063875.0	1.045664e+09	tt1298650	Acti
0	May 20, 2011	Pirates of the Caribbean: On Stranger Tides	410600000.0	241063875.0	1.045664e+09	tt1298650	Adventu
0	May 20, 2011	Pirates of the Caribbean: On Stranger Tides	410600000.0	241063875.0	1.045664e+09	tt1298650	Fanta
1	May 20, 2011	Pirates of the Caribbean: On Stranger Tides	410600000.0	241063875.0	1.045664e+09	tt1298650	Acti

	release_date	movie	production_budget	domestic_gross	worldwide_gross	tconst	genı
1	May 20, 2011	Pirates of the Caribbean: On Stranger Tides	410600000.0	241063875.0	1.045664e+09	tt1298650	Adventı

In [29]:
```
#Sort values based on production_budget
df_Merged.sort_values(by='production_budget', ascending=False, inplace=True
```

Save Data

In [30]:
```
df_Merged.to_csv('zippedData/myData.csv')
```

In [ ]: