https://drive.google.com/file/d/1OdnRK71RJBABK8n3jAITlh2YOH3P-7Bb/view?usp=sharing

IMDB Movie Case Study Question Draft saved

File Edit View Run Add-ons Help

Share Save Version 0

Run All Code Draft Session (21m)

IMDb Movie Assignment

Project Description: We have the data for the 100 top-rated movies from the past decade along with various pieces of information about the movie, its actors, and the voters who have rated these movies online.

Tech-Stack Used: kaggle.com

Result: In this assignment, we find some interesting insights into a few movies released between 1916 and 2016, using Python.

```
# Supress Warnings
import warnings
warnings.filterwarnings('ignore')
```

+ Code + Markdown

[100]:

```
# Import the numpy and pandas packages
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

Task 1: Reading and Inspection

Subtask 1.1: Import and read

Import and read the movie database. Store it in a variable called `movies`.

[101]:

```
import os
print(os.listdir("../input"))
```

```
['MovieAssignmentData.csv']
```

[102]:

```
movies = pd.DataFrame(pd.read_csv("../input/MovieAssignmentData.csv"))
movies.head()
```

[102…

	color	director_name	num_critic_for_reviews	duration	director_facebook_likes	actor_3_facebook_likes	actor_2_name	actor_1_facebook_likes	gross	genres	actor_1_name	movie_title	num_voted_users	cast_total_facebook_likes	actor_3
0	Color	James Cameron	723.0	178.0	0.0	855.0	Joel David Moore	1000.0	760505847.0	Action|Adventure|Fantasy|Sci-Fi	CCH Pounder	Avatar	886204	4834	We
1	Color	Gore Verbinski	302.0	169.0	563.0	1000.0	Orlando Bloom	40000.0	309404152.0	Action|Adventure|Fantasy	Johnny Depp	Pirates of the Caribbean: At World's End	471220	48350	Dav
2	Color	Sam Mendes	602.0	148.0	0.0	161.0	Rory Kinnear	11000.0	200074175.0	Action|Adventure|Thriller	Christoph Waltz	Spectre	275868	11700	Ste S
3	Color	Christopher Nolan	813.0	164.0	22000.0	23000.0	Christian Bale	27000.0	448130642.0	Action|Thriller	Tom Hardy	The Dark Knight Rises	1144337	106759	Gordor
4	NaN	Doug Walker	NaN	NaN	131.0	NaN	Rob Walker	131.0	NaN	Documentary	Doug Walker	Star Wars: Episode VII - The Force Awakens ...	8	143	

- Subtask 1.2: Inspect the dataframe

Inspect the dataframe's columns, shapes, variable types etc.

[103]:

```
movies.shape
```

[103… (5043, 28)

[104]:

```
# Concise summary of the Dataframe.
movies.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5043 entries, 0 to 5042
Data columns (total 28 columns):
color                        5024 non-null object
director_name                4939 non-null object
num_critic_for_reviews       4993 non-null float64
duration                     5028 non-null float64
director_facebook_likes      4939 non-null float64
actor_3_facebook_likes       5020 non-null float64
actor_2_name                 5030 non-null object
actor_1_facebook_likes       5036 non-null float64
gross                        4159 non-null float64
genres                       5043 non-null object
actor_1_name                 5036 non-null object
movie_title                  5043 non-null object
num_voted_users              5043 non-null int64
cast_total_facebook_likes    5043 non-null int64
actor_3_name                 5020 non-null object
facenumber_in_poster         5030 non-null float64
plot_keywords                4890 non-null object
movie_imdb_link              5043 non-null object
num_user_for_reviews         5022 non-null float64
language                     5031 non-null object
country                      5038 non-null object
content_rating               4740 non-null object
budget                       4551 non-null float64
title_year                   4935 non-null float64
actor_2_facebook_likes       5030 non-null float64
imdb_score                   5043 non-null float64
aspect_ratio                 4714 non-null float64
movie_facebook_likes         5043 non-null int64
dtypes: float64(13), int64(3), object(12)
```

```
memory usage: 1.1+ MB
```

[105]:

```python
# Generate descriptive statistics that summarize the central tendency, dispersion and shape of a dataset's distribution,
# excluding NaN values
movies.describe()
```

[105…

	num_critic_for_reviews	duration	director_facebook_likes	actor_3_facebook_likes	actor_1_facebook_likes	gross	num_voted_users	cast_total_facebook_likes	facenumber_in_poster	num_user_for_reviews	budget	title_year	actor_2_facebook_l
count	4993.000000	5028.000000	4939.000000	5020.000000	5036.000000	4.159000e+03	5.043000e+03	5043.000000	5030.000000	5022.000000	4.551000e+03	4935.000000	5030.00
mean	140.194272	107.201074	686.509212	645.009761	6560.047061	4.846841e+07	8.366816e+04	9699.063851	1.371173	272.770808	3.975262e+07	2002.470517	1651.75
std	121.601675	25.197441	2813.328607	1665.041728	15020.759120	6.845299e+07	1.384853e+05	18163.799124	2.013576	377.982886	2.061149e+08	12.474599	4042.43
min	1.000000	7.000000	0.000000	0.000000	0.000000	1.620000e+02	5.000000e+00	0.000000	0.000000	1.000000	2.180000e+02	1916.000000	0.00
25%	50.000000	93.000000	7.000000	133.000000	614.000000	5.340988e+06	8.593500e+03	1411.000000	0.000000	65.000000	6.000000e+06	1999.000000	281.00
50%	110.000000	103.000000	49.000000	371.500000	988.000000	2.551750e+07	3.435900e+04	3090.000000	1.000000	156.000000	2.000000e+07	2005.000000	595.00
75%	195.000000	118.000000	194.500000	636.000000	11000.000000	6.230944e+07	9.630900e+04	13756.500000	2.000000	326.000000	4.500000e+07	2011.000000	918.00
max	813.000000	511.000000	23000.000000	23000.000000	640000.000000	7.605058e+08	1.689764e+06	656730.000000	43.000000	5060.000000	1.221550e+10	2016.000000	137000.00

[106]:

```python
movies.color.describe()
```

[106…

```
count      5024
unique        2
top       Color
freq       4815
Name: color, dtype: object
```

[107]:

```python
movies.director_name.describe()
```

[107…

```
count                  4939
unique                 2398
top       Steven Spielberg
freq                     26
Name: director_name, dtype: object
```

[108]:

```python
movies.color.describe()
```

[108…

```
count      5024
unique        2
top       Color
freq       4815
Name: color, dtype: object
```

[109]:

```python
movies.genres.describe()
```

[109…

```
count      5043
unique      914
top       Drama
freq        236
Name: genres, dtype: object
```

[110]:

```python
movies.movie_title.describe()
```

[110…

```
count           5043
unique          4917
top       King Kong 
freq               3
Name: movie_title, dtype: object
```

[111]:

```python
movies.director_name.describe()
```

[111…

```
count                  4939
unique                 2398
top       Steven Spielberg
freq                     26
Name: director_name, dtype: object
```

[112]:

```python
movies.actor_3_name.describe()
```

[112…

```
count           5020
unique          3521
top       John Heard
freq               8
Name: actor_3_name, dtype: object
```

[113]:

```python
movies.plot_keywords.describe()
```

[113…

```
count               4890
unique              4760
top       based on novel
freq                   4
Name: plot_keywords, dtype: object
```

[114]:

```python
movies.movie_imdb_link.describe()
```

[114…

```
count                                                 5043
unique                                                4919
top       http://www.imdb.com/title/tt2638144/?ref_=fn_t...
freq                                                     3
Name: movie_imdb_link, dtype: object
```

[115]:

```python
movies.language.describe()
```

[115…

```
count        5031
unique         47
top       English
```

```
top         English
freq           4704
Name: language, dtype: object
```

[116]:

```
movies.country.describe()
```

[116…

```
count      5038
unique       65
top         USA
freq       3807
Name: country, dtype: object
```

[117]:

```
movies.content_rating.describe()
```

[117…

```
count      4740
unique       18
top           R
freq       2118
Name: content_rating, dtype: object
```

Task 2: Cleaning the Data

Now that we have loaded the dataset and inspected it, we see that most of the data is in place. As of now, no data cleaning is required, so let's start with some data manipulation, analysis, and visualisation to get various insights about the data.

- ### Subtask 2.1: Inspect Null values

Find out the number of Null values in all the columns and rows. Also, find the percentage of Null values in each column. Round off the percentages upto two decimal places.

[118]:

```
# Write your code for column-wise null count here
movies.isnull().sum()
```

[118…

```
color                          19
director_name                 104
num_critic_for_reviews         50
duration                       15
director_facebook_likes       104
actor_3_facebook_likes         23
actor_2_name                   13
actor_1_facebook_likes          7
gross                         884
genres                          0
actor_1_name                    7
movie_title                     0
num_voted_users                 0
cast_total_facebook_likes       0
actor_3_name                   23
facenumber_in_poster           13
plot_keywords                 153
movie_imdb_link                 0
num_user_for_reviews           21
language                       12
country                         5
content_rating                303
budget                        492
title_year                    108
actor_2_facebook_likes         13
imdb_score                      0
aspect_ratio                  329
movie_facebook_likes            0
dtype: int64
```

[119]:

```
# Write your code for row-wise null count here
movies.isnull().sum(axis = 1)
```

[119…

```
0         0
1         0
2         0
3         0
4        14
5         0
6         0
7         0
8         0
9         0
10        0
11        0
12        0
```

[120]:

```
# Write your code for column-wise null percentages here
round((movies.isnull().sum()/len(movies.index))*100,2)
```

[120…

```
color                         0.38
director_name                 2.06
num_critic_for_reviews        0.99
duration                      0.30
director_facebook_likes       2.06
actor_3_facebook_likes        0.46
actor_2_name                  0.26
actor_1_facebook_likes        0.14
gross                        17.53
genres                        0.00
actor_1_name                  0.14
movie_title                   0.00
num_voted_users               0.00
cast_total_facebook_likes     0.00
actor_3_name                  0.46
facenumber_in_poster          0.26
plot_keywords                 3.03
movie_imdb_link               0.00
num_user_for_reviews          0.42
language                      0.24
country                       0.10
content_rating                6.01
budget                        9.76
title_year                    2.14
actor_2_facebook_likes        0.26
imdb_score                    0.00
aspect_ratio                  6.52
movie_facebook_likes          0.00
dtype: float64
```

- ### Subtask 2.2: Drop unecessary columns

For this assignment, you will mostly be analyzing the movies with respect to the ratings, gross collection, popularity of movies, etc. So many of the columns in this dataframe are not required. So it is advised to drop the following columns.

- color
- director_facebook_likes
- actor_1_facebook_likes

- actor_1_facebook_likes
- actor_2_facebook_likes
- actor_3_facebook_likes
- actor_2_name
- cast_total_facebook_likes
- actor_3_name
- duration
- facenumber_in_poster
- content_rating
- country
- movie_imdb_link
- aspect_ratio
- plot_keywords

[121]:

```
# Write your code for dropping the columns here. It is advised to keep inspecting the dataframe after each set of operations
movies = movies.drop(["color","director_facebook_likes","actor_1_facebook_likes","actor_2_facebook_likes","actor_3_facebook_likes","actor_2_name","cast_total_facebook_likes","actor_3_nam
movies.head()
```

[121…

<table>
<tr><th></th><th>director_name</th><th>num_critic_for_reviews</th><th>gross</th><th>genres</th><th>actor_1_name</th><th>movie_title</th><th>num_voted_users</th><th>num_user_for_reviews</th><th>language</th><th>budget</th><th>title_year</th><th>imdb_score</th><th>movie_facebook_likes</th></tr>
<tr><td>0</td><td>James Cameron</td><td>723.0</td><td>760505847.0</td><td>Action|Adventure|Fantasy|Sci-Fi</td><td>CCH Pounder</td><td>Avatar</td><td>886204</td><td>3054.0</td><td>English</td><td>237000000.0</td><td>2009.0</td><td>7.9</td><td>33000</td></tr>
<tr><td>1</td><td>Gore Verbinski</td><td>302.0</td><td>309404152.0</td><td>Action|Adventure|Fantasy</td><td>Johnny Depp</td><td>Pirates of the Caribbean: At World's End</td><td>471220</td><td>1238.0</td><td>English</td><td>300000000.0</td><td>2007.0</td><td>7.1</td><td>0</td></tr>
<tr><td>2</td><td>Sam Mendes</td><td>602.0</td><td>200074175.0</td><td>Action|Adventure|Thriller</td><td>Christoph Waltz</td><td>Spectre</td><td>275868</td><td>994.0</td><td>English</td><td>245000000.0</td><td>2015.0</td><td>6.8</td><td>85000</td></tr>
<tr><td>3</td><td>Christopher Nolan</td><td>813.0</td><td>448130642.0</td><td>Action|Thriller</td><td>Tom Hardy</td><td>The Dark Knight Rises</td><td>1144337</td><td>2701.0</td><td>English</td><td>250000000.0</td><td>2012.0</td><td>8.5</td><td>164000</td></tr>
<tr><td>4</td><td>Doug Walker</td><td>NaN</td><td>NaN</td><td>Documentary</td><td>Doug Walker</td><td>Star Wars: Episode VII - The Force Awakens …</td><td>8</td><td>NaN</td><td>NaN</td><td>NaN</td><td>NaN</td><td>7.1</td><td>0</td></tr>
</table>

[122]:

```
movies.columns
```

[122…

```
Index(['director_name', 'num_critic_for_reviews', 'gross', 'genres',
       'actor_1_name', 'movie_title', 'num_voted_users',
       'num_user_for_reviews', 'language', 'budget', 'title_year',
       'imdb_score', 'movie_facebook_likes'],
      dtype='object')
```

- ### Subtask 2.3: Drop unecessary rows using columns with high Null percentages

Now, on inspection you might notice that some columns have large percentage (greater than 5%) of Null values. Drop all the rows which have Null values for such columns.

[123]:

```
# Write your code for dropping the rows here
round((movies.isnull().sum()/len(movies.index))*100,2)
```

[123…

```
director_name              2.06
num_critic_for_reviews     0.99
gross                     17.53
genres                     0.00
actor_1_name               0.14
movie_title                0.00
num_voted_users            0.00
num_user_for_reviews       0.42
language                   0.24
budget                     9.76
title_year                 2.14
imdb_score                 0.00
movie_facebook_likes       0.00
dtype: float64
```

[124]:

```
movies = movies[~np.isnan(movies['gross'])]
movies = movies[~np.isnan(movies['budget'])]
```

- ### Subtask 2.4: Fill NaN values

You might notice that the `language` column has some NaN values. Here, on inspection, you will see that it is safe to replace all the missing values with `'English'`.

[125]:

```
# Fill the NaN values in the 'language' column here
movies["language"] = movies.language.fillna("English")
movies.language.isnull().sum()
```

[125… 0

- ### Subtask 2.5: Check the number of retained rows

You might notice that two of the columns viz. `num_critic_for_reviews` and `actor_1_name` have small percentages of NaN values left. You can let these columns as it is for now. Check the number and percentage of the rows retained after completing all the tasks above.

[126]:

```
# Write your code for checking number of retained rows here
len(movies.index)/5043
```

[126… 0.7715645449137418

Checkpoint 1: You might have noticed that we still have around `77%` of the rows!

Task 3: Data Analysis

- ### Subtask 3.1: Change the unit of columns

Convert the unit of the `budget` and `gross` columns from `$` to `million $`.

[127]:

```python
# Write your code for unit conversion here
movies.budget = round(movies["budget"]/1000000,2)
movies.gross = round(movies["gross"]/1000000,2)
```

- ## Subtask 3.2: Find the movies with highest profit

1. Create a new column called `profit` which contains the difference of the two columns: `gross` and `budget`.
2. Sort the dataframe using the `profit` column as reference.
3. Extract the top ten profiting movies in descending order and store them in a new dataframe - `top10`

[128]:

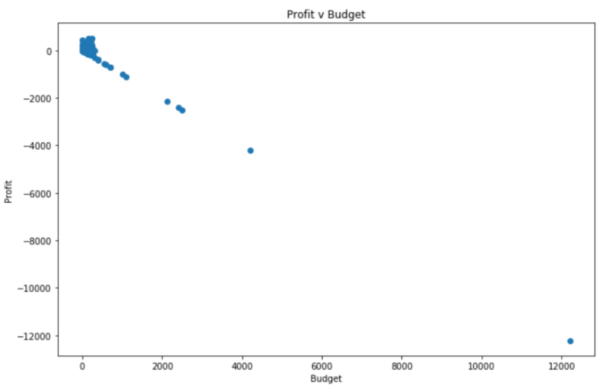
```python
# Write your code for creating the profit column here
movies["profit"] = movies.gross - movies.budget
```

[129]:

```python
# Write your code for sorting the dataframe here
movies_by_profit = movies.sort_values("profit",ascending = False)
```

[146]:

```python
# Write code for profit vs budget plot here
plt.figure(figsize=[11,7])
plt.scatter(movies.budget,movies.profit)
plt.title("Profit v Budget")
plt.xlabel("Budget")
plt.ylabel("Profit")
plt.show()
```



My Observation: Movies with higher budgets are not necessarily profitable

The dataset contains the 100 best performing movies from the year 2010 to 2016. However scatter plot tells a different story. You can notice that there are some movies with negative profit. Although good movies do incur losses, but there appear to be quite a few movie with losses. What can be the reason behind this? Lets have a closer look at this by finding the movies with negative profit.

[131]:

```python
# Write your code to get the top 10 profiting movies here
top10 = movies_by_profit.head(10)
top10
```

[131…

	director_name	num_critic_for_reviews	gross	genres	actor_1_name	movie_title	num_voted_users	num_user_for_reviews	language	budget	title_year	imdb_score	movie_facebook_likes	profit
0	James Cameron	723.0	760.51	Action|Adventure|Fantasy|Sci-Fi	CCH Pounder	Avatar	886204	3054.0	English	237.0	2009.0	7.9	33000	523.51
29	Colin Trevorrow	644.0	652.18	Action|Adventure|Sci-Fi|Thriller	Bryce Dallas Howard	Jurassic World	418214	1290.0	English	150.0	2015.0	7.0	150000	502.18
26	James Cameron	315.0	658.67	Drama|Romance	Leonardo DiCaprio	Titanic	793059	2528.0	English	200.0	1997.0	7.7	26000	458.67
3024	George Lucas	282.0	460.94	Action|Adventure|Fantasy|Sci-Fi	Harrison Ford	Star Wars: Episode IV - A New Hope	911097	1470.0	English	11.0	1977.0	8.7	33000	449.94
3080	Steven Spielberg	215.0	434.95	Family|Sci-Fi	Henry Thomas	E.T. the Extra-Terrestrial	281842	515.0	English	10.5	1982.0	7.9	34000	424.45
794	Joss Whedon	703.0	623.28	Action|Adventure|Sci-Fi	Chris Hemsworth	The Avengers	995415	1722.0	English	220.0	2012.0	8.1	123000	403.28
17	Joss Whedon	703.0	623.28	Action|Adventure|Sci-Fi	Chris Hemsworth	The Avengers	995415	1722.0	English	220.0	2012.0	8.1	123000	403.28
509	Roger Allers	186.0	422.78	Adventure|Animation|Drama|Family|Musical	Matthew Broderick	The Lion King	644348	656.0	English	45.0	1994.0	8.5	17000	377.78
240	George Lucas	320.0	474.54	Action|Adventure|Fantasy|Sci-Fi	Natalie Portman	Star Wars: Episode I - The Phantom Menace	534658	3597.0	English	115.0	1999.0	6.5	13000	359.54
66	Christopher Nolan	645.0	533.32	Action|Crime|Drama|Thriller	Christian Bale	The Dark Knight	1676169	4667.0	English	185.0	2008.0	9.0	37000	348.32

- ## Subtask 3.3: Drop duplicate values

After you found out the top 10 profiting movies, you might have notice a duplicate value. So, it seems like the dataframe has duplicate values as well. Drop the duplicate values from the dataframe and repeat `Subtask 3.2`.

[132]:

```python
# Write your code for dropping duplicate values here
movies = movies.drop_duplicates()
```

[133]:

```python
# Write code for repeating subtask 2 here
top10 = movies.head(10)
top10
```

[133…

	director_name	num_critic_for_reviews	gross	genres	actor_1_name	movie_title	num_voted_users	num_user_for_reviews	language	budget	title_year	imdb_score	movie_facebook_likes	profit
0	James Cameron	723.0	760.51	Action|Adventure|Fantasy|Sci-Fi	CCH Pounder	Avatar	886204	3054.0	English	237.0	2009.0	7.9	33000	523.51

1	Gore Verbinski	302.0	309.40	Action|Adventure|Fantasy	Johnny Depp	Pirates of the Caribbean: At World's End	471220	1238.0	English	300.0	2007.0	7.1	0	9.40
2	Sam Mendes	602.0	200.07	Action|Adventure|Thriller	Christoph Waltz	Spectre	275868	994.0	English	245.0	2015.0	6.8	85000	-44.93
3	Christopher Nolan	813.0	448.13	Action|Thriller	Tom Hardy	The Dark Knight Rises	1144337	2701.0	English	250.0	2012.0	8.5	164000	198.13
5	Andrew Stanton	462.0	73.06	Action|Adventure|Sci-Fi	Daryl Sabara	John Carter	212204	738.0	English	263.7	2012.0	6.6	24000	-190.64
6	Sam Raimi	392.0	336.53	Action|Adventure|Romance	J.K. Simmons	Spider-Man 3	383056	1902.0	English	258.0	2007.0	6.2	0	78.53
7	Nathan Greno	324.0	200.81	Adventure|Animation|Comedy|Family|Fantasy|Musi...	Brad Garrett	Tangled	294810	387.0	English	260.0	2010.0	7.8	29000	-59.19
8	Joss Whedon	635.0	458.99	Action|Adventure|Sci-Fi	Chris Hemsworth	Avengers: Age of Ultron	462669	1117.0	English	250.0	2015.0	7.5	118000	208.99
9	David Yates	375.0	301.96	Adventure|Family|Fantasy|Mystery	Alan Rickman	Harry Potter and the Half-Blood Prince	321795	973.0	English	250.0	2009.0	7.5	10000	51.96
10	Zack Snyder	673.0	330.25	Action|Adventure|Sci-Fi	Henry Cavill	Batman v Superman: Dawn of Justice	371639	3018.0	English	250.0	2016.0	6.9	197000	80.25

Checkpoint 2: You might spot two movies directed by `James Cameron` in the list.

- ## Subtask 3.4: Find IMDb Top 250

1. Create a new dataframe `IMDb_Top_250` and store the top 250 movies with the highest IMDb Rating (corresponding to the column: `imdb_score`). Also make sure that for all of these movies, the `num_voted_users` is greater than 25,000. Also add a `Rank` column containing the values 1 to 250 indicating the ranks of the corresponding films.
2. Extract all the movies in the `IMDb_Top_250` dataframe which are not in the English language and store them in a new dataframe named `Top_Foreign_Lang_Film`.

[134]:

```python
# Write your code for extracting the top 250 movies as per the IMDb score here. Make sure that you store it in a new dataframe
# and name that dataframe as 'IMDb_Top_250'
IMDb_Top_250 = movies[movies["num_voted_users"] > 25000]
IMDb_Top_250 = IMDb_Top_250.sort_values("imdb_score", ascending = False).head(250)
IMDb_Top_250["rank"] = list(range(1,251))
IMDb_Top_250
```

[134...

	director_name	num_critic_for_reviews	gross	genres	actor_1_name	movie_title	num_voted_users	num_user_for_reviews	language	budget	title_year	imdb_score	movie_facebook_likes	profit	rank
1937	Frank Darabont	199.0	28.34	Crime|Drama	Morgan Freeman	The Shawshank Redemption	1689764	4144.0	English	25.00	1994.0	9.3	108000	3.34	1
3466	Francis Ford Coppola	208.0	134.82	Crime|Drama	Al Pacino	The Godfather	1155770	2238.0	English	6.00	1972.0	9.2	43000	128.82	2
2837	Francis Ford Coppola	149.0	57.30	Crime|Drama	Robert De Niro	The Godfather: Part II	790926	650.0	English	13.00	1974.0	9.0	14000	44.30	3
66	Christopher Nolan	645.0	533.32	Action|Crime|Drama|Thriller	Christian Bale	The Dark Knight	1676169	4667.0	English	185.00	2008.0	9.0	37000	348.32	4
4498	Sergio Leone	181.0	6.10	Western	Clint Eastwood	The Good, the Bad and the Ugly	503509	780.0	Italian	1.20	1966.0	8.9	20000	4.90	5
3355	Quentin Tarantino	215.0	107.93	Crime|Drama	Bruce Willis	Pulp Fiction	1324680	2195.0	English	8.00	1994.0	8.9	45000	99.93	6
1874	Steven Spielberg	174.0	96.07	Biography|Drama|History	Liam Neeson	Schindler's List	865020	1273.0	English	22.00	1993.0	8.9	41000	74.07	7
339	Peter Jackson	328.0	377.02	Action|Adventure|Drama|Fantasy	Orlando Bloom	The Lord of the Rings: The Return of the King	1215718	3189.0	English	94.00	2003.0	8.9	16000	283.02	8
836	Robert Zemeckis	149.0	329.69	Comedy|Drama	Tom Hanks	Forrest Gump	1251222	1398.0	English	55.00	1994.0	8.8	59000	274.69	9
97	Christopher Nolan	642.0	292.57	Action|Adventure|Sci-Fi|Thriller	Leonardo DiCaprio	Inception	1468200	2803.0	English	160.00	2010.0	8.8	175000	132.57	10
683	David Fincher	315.0	37.02	Drama	Brad Pitt	Fight Club	1347461	2968.0	English	63.00	1999.0	8.8	48000	-25.98	11
2051	Irvin Kershner	223.0	290.16	Action|Adventure|Fantasy|Sci-Fi	Harrison Ford	Star Wars: Episode V - The Empire Strikes Back	837759	900.0	English	18.00	1980.0	8.8	17000	272.16	12
270	Peter Jackson	297.0	313.84	Action|Adventure|Drama|Fantasy	Christopher Lee	The Lord of the Rings: The Fellowship of the R...	1238746	5060.0	English	93.00	2001.0	8.8	21000	220.84	13
654	Lana Wachowski	313.0	171.38	Action|Sci-Fi	Keanu Reeves	The Matrix	1217752	3646.0	English	63.00	1999.0	8.7	25000	108.38	14
3867	Milos Forman	149.0	112.00	Drama	Scatman Crothers	One Flew Over the Cuckoo's Nest	680041	760.0	English	4.40	1975.0	8.7	32000	107.60	15
1903	Martin Scorsese	192.0	46.84	Biography|Crime|Drama	Robert De Niro	Goodfellas	728685	989.0	English	25.00	1990.0	8.7	25000	21.84	16
4747	Akira Kurosawa	153.0	0.27	Action|Adventure|Drama	Takashi Shimura	Seven Samurai	229012	596.0	Japanese	2.00	1954.0	8.7	11000	-1.73	17
3024	George Lucas	282.0	460.94	Action|Adventure|Fantasy|Sci-Fi	Harrison Ford	Star Wars: Episode IV - A New Hope	911097	1470.0	English	11.00	1977.0	8.7	33000	449.94	18
340	Peter Jackson	294.0	340.48	Action|Adventure|Drama|Fantasy	Christopher Lee	The Lord of the Rings: The Two Towers	1100446	2417.0	English	94.00	2002.0	8.7	10000	246.48	19
4029	Fernando Meirelles	214.0	7.56	Crime|Drama	Alice Braga	City of God	533200	749.0	Portuguese	3.30	2002.0	8.7	28000	4.26	20
1600	David Fincher	216.0	100.13	Crime|Drama|Mystery|Thriller	Morgan Freeman	Se7en	1023511	1080.0	English	33.00	1995.0	8.6	39000	67.13	21
96	Christopher Nolan	712.0	187.99	Adventure|Drama|Sci-Fi	Matthew McConaughey	Interstellar	928227	2725.0	English	165.00	2014.0	8.6	349000	22.99	22
2373	Hayao Miyazaki	246.0	10.05	Adventure|Animation|Family|Fantasy	Bunta Sugawara	Spirited Away	417971	902.0	Japanese	19.00	2001.0	8.6	28000	-8.95	23
648	Steven Spielberg	219.0	216.12	Action|Drama|War	Tom Hanks	Saving Private Ryan	881236	2277.0	English	70.00	1998.0	8.6	22000	146.12	24
4427	Charles Chaplin	120.0	0.16	Comedy|Drama|Family	Paulette Goddard	Modern Times	143086	211.0	English	1.50	1936.0	8.6	0	-1.34	25
2158	Jonathan Demme	185.0	130.73	Crime|Drama|Horror|Thriller	Anthony Hopkins	The Silence of the Lambs	887467	916.0	English	19.00	1991.0	8.6	40000	111.73	26
3175	Tony Kaye	162.0	6.71	Crime|Drama	Ethan Suplee	American History X	782437	1420.0	English	7.50	1998.0	8.6	35000	-0.79	27
3592	Bryan Singer	162.0	23.27	Crime|Drama|Mystery|Thriller	Kevin Spacey	The Usual Suspects	740918	1182.0	English	6.00	1995.0	8.6	28000	17.27	28
3716	Christopher Nolan	274.0	25.53	Mystery|Thriller	Callum Rennie	Memento	845580	2067.0	English	9.00	2000.0	8.5	40000	16.53	29
1233	Christopher Nolan	341.0	53.08	Drama|Mystery|Sci-Fi|Thriller	Christian Bale	The Prestige	844052	1100.0	English	40.00	2006.0	8.5	49000	13.08	30
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1871	Pierre Morel	309.0	145.00	Action|Thriller	Liam Neeson	Taken	483756	974.0	English	25.00	2008.0	7.9	29000	120.00	221
89	Don Hall	384.0	222.49	Action|Adventure|Animation|Comedy|Drama|Family...	Damon Wayans Jr.	Big Hero 6	279093	433.0	English	165.00	2014.0	7.9	41000	57.49	222
162	Dean DeBlois	292.0	177.00	Action|Adventure|Animation|Comedy|Family|Fantasy	Gerard Butler	How to Train Your Dragon 2	221128	343.0	English	145.00	2014.0	7.9	46000	32.00	223
2666	James Ivory	58.0	22.95	Drama|Romance	Anthony Hopkins	The Remains of the Day	45703	156.0	English	11.50	1993.0	7.9	0	11.45	224
2667	Paul Thomas Anderson	153.0	26.38	Drama	Don Cheadle	Boogie Nights	189032	560.0	English	15.00	1997.0	7.9	0	11.38	225
2863	Clint Eastwood	251.0	13.75	Drama|History|War	Yuki Matsuzaki	Letters from Iwo Jima	132149	316.0	Japanese	19.00	2006.0	7.9	5000	-5.25	226

3080	Steven Spielberg	215.0	434.95	Family|Sci-Fi	Henry Thomas	E.T. the Extra-Terrestrial	281842	515.0	English	10.50	1982.0	7.9	34000	424.45	227
3193	Paul Haggis	287.0	54.56	Crime|Drama|Thriller	Don Cheadle	Crash	361169	1624.0	English	6.50	2004.0	7.9	18000	48.06	228
3264	Michael Haneke	447.0	0.23	Drama|Romance	Isabelle Huppert	Amour	70382	190.0	French	8.90	2012.0	7.9	33000	-8.67	229
3357	Dan Gilroy	534.0	32.28	Crime|Drama|Thriller	Jake Gyllenhaal	Nightcrawler	293304	552.0	English	8.50	2014.0	7.9	65000	23.78	230
3361	Jonathan Dayton	270.0	59.89	Comedy|Drama	Steve Carell	Little Miss Sunshine	355810	889.0	English	8.00	2006.0	7.9	15000	51.89	231
99	Peter Jackson	645.0	303.00	Adventure|Fantasy	Aidan Turner	The Hobbit: An Unexpected Journey	637246	1367.0	English	180.00	2012.0	7.9	166000	123.00	232
3510	Yash Chopra	29.0	2.92	Drama|Musical|Romance	Shah Rukh Khan	Veer-Zaara	34449	119.0	Hindi	7.00	2004.0	7.9	2000	-4.08	233
3595	Darren Aronofsky	391.0	26.24	Drama|Sport	Mark Margolis	The Wrestler	251349	547.0	English	6.00	2008.0	7.9	10000	20.24	234
2558	Edgar Wright	285.0	23.62	Action|Comedy|Mystery	Bill Bailey	Hot Fuzz	352695	687.0	English	8.00	2007.0	7.9	17000	15.62	235
75	Doug Liman	585.0	100.19	Action|Adventure|Sci-Fi	Tom Cruise	Edge of Tomorrow	431620	741.0	English	178.00	2014.0	7.9	77000	-77.81	236
3677	Christophe Barratier	112.0	3.63	Drama|Music	Jean-Baptiste Maunier	The Chorus	44151	110.0	French	5.50	2004.0	7.9	0	-1.87	237
3680	Spike Lee	103.0	27.55	Drama	Ruby Dee	Do the Right Thing	59524	418.0	English	6.50	1989.0	7.9	0	21.05	238
69	Jon Favreau	486.0	318.30	Action|Adventure|Sci-Fi	Robert Downey Jr.	Iron Man	696338	1055.0	English	140.00	2008.0	7.9	10000	178.30	239
3768	Duncan Jones	415.0	5.01	Drama|Mystery|Sci-Fi	Kevin Spacey	Moon	260607	485.0	English	5.00	2009.0	7.9	47000	0.01	240
4082	Richard Linklater	405.0	8.11	Drama|Romance	Seamus Davey-Fitzpatrick	Before Midnight	95362	270.0	English	3.00	2013.0	7.9	62000	5.11	241
23	Peter Jackson	509.0	258.36	Adventure|Fantasy	Aidan Turner	The Hobbit: The Desolation of Smaug	483540	951.0	English	225.00	2013.0	7.9	83000	33.36	242
4415	Fabián Bielinsky	94.0	1.22	Crime|Drama|Thriller	Ricardo Darín	Nine Queens	38215	125.0	Spanish	1.50	2000.0	7.9	0	-0.28	243
4640	Cristian Mungiu	233.0	1.19	Drama	Anamaria Marinca	4 Months, 3 Weeks and 2 Days	44763	172.0	Romanian	0.59	2007.0	7.9	14000	0.60	244
4821	John Carpenter	318.0	47.00	Horror|Thriller	Jamie Lee Curtis	Halloween	157863	1191.0	English	0.30	1978.0	7.9	12000	46.70	245
4931	John Carney	232.0	9.44	Drama|Music|Romance	Glen Hansard	Once	90827	329.0	English	0.18	2007.0	7.9	26000	9.26	246
2605	Ang Lee	287.0	128.07	Action|Drama|Romance	Chen Chang	Crouching Tiger, Hidden Dragon	217740	1641.0	Mandarin	15.00	2000.0	7.9	0	113.07	247
3029	David O. Russell	410.0	93.57	Biography|Drama|Sport	Christian Bale	The Fighter	275869	389.0	English	25.00	2010.0	7.9	36000	68.57	248
2177	Tim Burton	111.0	56.36	Fantasy|Romance	Johnny Depp	Edward Scissorhands	357581	588.0	English	20.00	1990.0	7.9	16000	36.36	249
2487	George Cukor	82.0	72.00	Drama|Family|Musical|Romance	Jeremy Brett	My Fair Lady	66959	258.0	English	17.00	1964.0	7.9	0	55.00	250

250 rows × 15 columns

[135]:

```
Top_Foreign_Lang_Film = IMDb_Top_250[IMDb_Top_250["language"] != "English"]
Top_Foreign_Lang_Film.head()
```

[135…

	director_name	num_critic_for_reviews	gross	genres	actor_1_name	movie_title	num_voted_users	num_user_for_reviews	language	budget	title_year	imdb_score	movie_facebook_likes	profit	rank
4498	Sergio Leone	181.0	6.10	Western	Clint Eastwood	The Good, the Bad and the Ugly	503509	780.0	Italian	1.2	1966.0	8.9	20000	4.90	5
4747	Akira Kurosawa	153.0	0.27	Action|Adventure|Drama	Takashi Shimura	Seven Samurai	229012	596.0	Japanese	2.0	1954.0	8.7	11000	-1.73	17
4029	Fernando Meirelles	214.0	7.56	Crime|Drama	Alice Braga	City of God	533200	749.0	Portuguese	3.3	2002.0	8.7	28000	4.26	20
2373	Hayao Miyazaki	246.0	10.05	Adventure|Animation|Family|Fantasy	Bunta Sugawara	Spirited Away	417971	902.0	Japanese	19.0	2001.0	8.6	28000	-8.95	23
4259	Florian Henckel von Donnersmarck	215.0	11.28	Drama|Thriller	Sebastian Koch	The Lives of Others	259379	407.0	German	2.0	2006.0	8.5	39000	9.28	35

Checkpoint 3: Can you spot `Veer-Zaara` in the dataframe?

- ## Subtask 3.5: Find the best directors

1. Group the dataframe using the `director_name` column.
2. Find out the top 10 directors for whom the mean of `imdb_score` is the highest and store them in a new dataframe `top10director`.

[136]:

```
# Write your code for extracting the top 10 directors here
director = movies.pivot_table(values = 'imdb_score', index = 'director_name', aggfunc = 'mean')
director = director.sort_values(by = 'imdb_score', ascending = False)
director = director.head(10)
director
```

[136…

	imdb_score
director_name	
Charles Chaplin	8.600000
Tony Kaye	8.600000
Alfred Hitchcock	8.500000
Ron Fricke	8.500000
Damien Chazelle	8.500000
Majid Majidi	8.500000
Sergio Leone	8.433333
Christopher Nolan	8.425000
S.S. Rajamouli	8.400000
Marius A. Markevicius	8.400000

Checkpoint 4: No surprises that `Damien Chazelle` (director of Whiplash and La La Land) is in this list.

- ## Subtask 3.6: Find popular genres

You might have noticed the `genres` column in the dataframe with all the genres of the movies seperated by a pipe (`|`). Out of all the movie genres, the first two are most significant for any film.

1. Extract the first two genres from the `genres` column and store them in two new columns: `genre_1` and `genre_2`. Some of the movies might have only one genre. In such cases, extract the single genre into both the columns, i.e. for such movies the `genre_2` will be the same as `genre_1`.

2. Group the dataframe using `genre_1` as the primary column and `genre_2` as the secondary column.
3. Find out the 5 most popular combo of genres by finding the mean of the gross values using the `gross` column and store them in a new dataframe named `PopGenre`.

[137]:

```python
# Write your code for extracting the first two genres of each movie here

movies["genre_1"] = movies["genres"].str.split("|").str.get(0)
movies["genre_2"] = movies["genres"].str.split("|").str.get(1)
movies["genre_2"] = movies["genre_2"].fillna(movies["genre_1"])
movies.head()
```

[137…

<table>
<tr><th></th><th>director_name</th><th>num_critic_for_reviews</th><th>gross</th><th>genres</th><th>actor_1_name</th><th>movie_title</th><th>num_voted_users</th><th>num_user_for_reviews</th><th>language</th><th>budget</th><th>title_year</th><th>imdb_score</th><th>movie_facebook_likes</th><th>profit</th><th>genre_1</th><th>genre_2</th></tr>
<tr><td>0</td><td>James Cameron</td><td>723.0</td><td>760.51</td><td>Action|Adventure|Fantasy|Sci-Fi</td><td>CCH Pounder</td><td>Avatar</td><td>886204</td><td>3054.0</td><td>English</td><td>237.0</td><td>2009.0</td><td>7.9</td><td>33000</td><td>523.51</td><td>Action</td><td>Adventure</td></tr>
<tr><td>1</td><td>Gore Verbinski</td><td>302.0</td><td>309.40</td><td>Action|Adventure|Fantasy</td><td>Johnny Depp</td><td>Pirates of the Caribbean: At World's End</td><td>471220</td><td>1238.0</td><td>English</td><td>300.0</td><td>2007.0</td><td>7.1</td><td>0</td><td>9.40</td><td>Action</td><td>Adventure</td></tr>
<tr><td>2</td><td>Sam Mendes</td><td>602.0</td><td>200.07</td><td>Action|Adventure|Thriller</td><td>Christoph Waltz</td><td>Spectre</td><td>275868</td><td>994.0</td><td>English</td><td>245.0</td><td>2015.0</td><td>6.8</td><td>85000</td><td>-44.93</td><td>Action</td><td>Adventure</td></tr>
<tr><td>3</td><td>Christopher Nolan</td><td>813.0</td><td>448.13</td><td>Action|Thriller</td><td>Tom Hardy</td><td>The Dark Knight Rises</td><td>1144337</td><td>2701.0</td><td>English</td><td>250.0</td><td>2012.0</td><td>8.5</td><td>164000</td><td>198.13</td><td>Action</td><td>Thriller</td></tr>
<tr><td>5</td><td>Andrew Stanton</td><td>462.0</td><td>73.06</td><td>Action|Adventure|Sci-Fi</td><td>Daryl Sabara</td><td>John Carter</td><td>212204</td><td>738.0</td><td>English</td><td>263.7</td><td>2012.0</td><td>6.6</td><td>24000</td><td>-190.64</td><td>Action</td><td>Adventure</td></tr>
</table>

[138]:

```python
# Write your code for grouping the dataframe here
movies_by_segment = movies.groupby(['genre_1','genre_2'])
```

[139]:

```python
# Write your code for getting the 5 most popular combo of genres here
PopGenre = pd.DataFrame(movies_by_segment.gross.mean()).sort_values("gross", ascending = False).head()
PopGenre
```

[139…

genre_1	genre_2	gross
Family	Sci-Fi	434.950000
Adventure	Sci-Fi	228.628750
	Family	118.918824
	Animation	116.998462
Action	Adventure	109.595510

Checkpoint 5: Well, as it turns out. `Family + Sci-Fi` is the most popular combo of genres out there!

- ### Subtask 3.7: Find the critic-favorite and audience-favorite actors

1. Create three new dataframes namely, `Meryl_Streep`, `Leo_Caprio`, and `Brad_Pitt` which contain the movies in which the actors: 'Meryl Streep', 'Leonardo DiCaprio', and 'Brad Pitt' are the lead actors. Use only the `actor_1_name` column for extraction. Also, make sure that you use the names 'Meryl Streep', 'Leonardo DiCaprio', and 'Brad Pitt' for the said extraction.
2. Append the rows of all these dataframes and store them in a new dataframe named `Combined`.
3. Group the combined dataframe using the `actor_1_name` column.
4. Find the mean of the `num_critic_for_reviews` and `num_user_for_review` and identify the actors which have the highest mean.

[140]:

```python
# Write your code for creating three new dataframes here

Meryl_Streep = movies[movies["actor_1_name"] == "Meryl Streep"]
Meryl_Streep
```

[140…

<table>
<tr><th></th><th>director_name</th><th>num_critic_for_reviews</th><th>gross</th><th>genres</th><th>actor_1_name</th><th>movie_title</th><th>num_voted_users</th><th>num_user_for_reviews</th><th>language</th><th>budget</th><th>title_year</th><th>imdb_score</th><th>movie_facebook_likes</th><th>profit</th><th>genre_1</th><th>genre_2</th></tr>
<tr><td>410</td><td>Nancy Meyers</td><td>187.0</td><td>112.70</td><td>Comedy|Drama|Romance</td><td>Meryl Streep</td><td>It's Complicated</td><td>69860</td><td>214.0</td><td>English</td><td>85.0</td><td>2009.0</td><td>6.6</td><td>0</td><td>27.70</td><td>Comedy</td><td>Drama</td></tr>
<tr><td>1106</td><td>Curtis Hanson</td><td>42.0</td><td>46.82</td><td>Action|Adventure|Crime|Thriller</td><td>Meryl Streep</td><td>The River Wild</td><td>32544</td><td>69.0</td><td>English</td><td>45.0</td><td>1994.0</td><td>6.3</td><td>0</td><td>1.82</td><td>Action</td><td>Adventure</td></tr>
<tr><td>1204</td><td>Nora Ephron</td><td>252.0</td><td>94.13</td><td>Biography|Drama|Romance</td><td>Meryl Streep</td><td>Julie & Julia</td><td>79264</td><td>277.0</td><td>English</td><td>40.0</td><td>2009.0</td><td>7.0</td><td>13000</td><td>54.13</td><td>Biography</td><td>Drama</td></tr>
<tr><td>1408</td><td>David Frankel</td><td>208.0</td><td>124.73</td><td>Comedy|Drama|Romance</td><td>Meryl Streep</td><td>The Devil Wears Prada</td><td>286178</td><td>631.0</td><td>English</td><td>35.0</td><td>2006.0</td><td>6.8</td><td>0</td><td>89.73</td><td>Comedy</td><td>Drama</td></tr>
<tr><td>1483</td><td>Robert Redford</td><td>227.0</td><td>15.00</td><td>Drama|Thriller|War</td><td>Meryl Streep</td><td>Lions for Lambs</td><td>41170</td><td>298.0</td><td>English</td><td>35.0</td><td>2007.0</td><td>6.2</td><td>0</td><td>-20.00</td><td>Drama</td><td>Thriller</td></tr>
<tr><td>1575</td><td>Sydney Pollack</td><td>66.0</td><td>87.10</td><td>Biography|Drama|Romance</td><td>Meryl Streep</td><td>Out of Africa</td><td>52339</td><td>200.0</td><td>English</td><td>31.0</td><td>1985.0</td><td>7.2</td><td>0</td><td>56.10</td><td>Biography</td><td>Drama</td></tr>
<tr><td>1618</td><td>David Frankel</td><td>234.0</td><td>63.54</td><td>Comedy|Drama|Romance</td><td>Meryl Streep</td><td>Hope Springs</td><td>34258</td><td>178.0</td><td>English</td><td>30.0</td><td>2012.0</td><td>6.3</td><td>0</td><td>33.54</td><td>Comedy</td><td>Drama</td></tr>
<tr><td>1674</td><td>Carl Franklin</td><td>64.0</td><td>23.21</td><td>Drama</td><td>Meryl Streep</td><td>One True Thing</td><td>9283</td><td>112.0</td><td>English</td><td>30.0</td><td>1998.0</td><td>7.0</td><td>592</td><td>-6.79</td><td>Drama</td><td>Drama</td></tr>
<tr><td>1925</td><td>Stephen Daldry</td><td>174.0</td><td>41.60</td><td>Drama|Romance</td><td>Meryl Streep</td><td>The Hours</td><td>102123</td><td>660.0</td><td>English</td><td>25.0</td><td>2002.0</td><td>7.6</td><td>0</td><td>16.60</td><td>Drama</td><td>Romance</td></tr>
<tr><td>2781</td><td>Phyllida Lloyd</td><td>331.0</td><td>29.96</td><td>Biography|Drama|History</td><td>Meryl Streep</td><td>The Iron Lady</td><td>82327</td><td>350.0</td><td>English</td><td>13.0</td><td>2011.0</td><td>6.4</td><td>18000</td><td>16.96</td><td>Biography</td><td>Drama</td></tr>
<tr><td>3135</td><td>Robert Altman</td><td>211.0</td><td>20.34</td><td>Comedy|Drama|Music</td><td>Meryl Streep</td><td>A Prairie Home Companion</td><td>19655</td><td>280.0</td><td>English</td><td>10.0</td><td>2006.0</td><td>6.8</td><td>683</td><td>10.34</td><td>Comedy</td><td>Drama</td></tr>
</table>

[141]:

```python
Leo_Caprio = movies[movies["actor_1_name"] == "Leonardo DiCaprio"]
Leo_Caprio
```

[141…

<table>
<tr><th></th><th>director_name</th><th>num_critic_for_reviews</th><th>gross</th><th>genres</th><th>actor_1_name</th><th>movie_title</th><th>num_voted_users</th><th>num_user_for_reviews</th><th>language</th><th>budget</th><th>title_year</th><th>imdb_score</th><th>movie_facebook_likes</th><th>profit</th><th>genre_1</th><th>genre_2</th></tr>
<tr><td>26</td><td>James Cameron</td><td>315.0</td><td>658.67</td><td>Drama|Romance</td><td>Leonardo DiCaprio</td><td>Titanic</td><td>793059</td><td>2528.0</td><td>English</td><td>200.0</td><td>1997.0</td><td>7.7</td><td>26000</td><td>458.67</td><td>Drama</td><td>Romance</td></tr>
<tr><td>50</td><td>Baz Luhrmann</td><td>490.0</td><td>144.81</td><td>Drama|Romance</td><td>Leonardo DiCaprio</td><td>The Great Gatsby</td><td>362912</td><td>753.0</td><td>English</td><td>105.0</td><td>2013.0</td><td>7.3</td><td>115000</td><td>39.81</td><td>Drama</td><td>Romance</td></tr>
<tr><td>97</td><td>Christopher Nolan</td><td>642.0</td><td>292.57</td><td>Action|Adventure|Sci-Fi|Thriller</td><td>Leonardo DiCaprio</td><td>Inception</td><td>1468200</td><td>2803.0</td><td>English</td><td>160.0</td><td>2010.0</td><td>8.8</td><td>175000</td><td>132.57</td><td>Action</td><td>Adventure</td></tr>
<tr><td>179</td><td>Alejandro G. Iñárritu</td><td>556.0</td><td>183.64</td><td>Adventure|Drama|Thriller|Western</td><td>Leonardo DiCaprio</td><td>The Revenant</td><td>406020</td><td>1188.0</td><td>English</td><td>135.0</td><td>2015.0</td><td>8.1</td><td>190000</td><td>48.64</td><td>Adventure</td><td>Drama</td></tr>
<tr><td>257</td><td>Martin Scorsese</td><td>267.0</td><td>102.61</td><td>Biography|Drama</td><td>Leonardo DiCaprio</td><td>The Aviator</td><td>264318</td><td>799.0</td><td>English</td><td>110.0</td><td>2004.0</td><td>7.5</td><td>0</td><td>-7.39</td><td>Biography</td><td>Drama</td></tr>
<tr><td>296</td><td>Quentin Tarantino</td><td>765.0</td><td>162.80</td><td>Drama|Western</td><td>Leonardo DiCaprio</td><td>Django Unchained</td><td>955174</td><td>1193.0</td><td>English</td><td>100.0</td><td>2012.0</td><td>8.5</td><td>199000</td><td>62.80</td><td>Drama</td><td>Western</td></tr>
<tr><td>307</td><td>Edward Zwick</td><td>166.0</td><td>57.37</td><td>Adventure|Drama|Thriller</td><td>Leonardo DiCaprio</td><td>Blood Diamond</td><td>400292</td><td>657.0</td><td>English</td><td>100.0</td><td>2006.0</td><td>8.0</td><td>14000</td><td>-42.63</td><td>Adventure</td><td>Drama</td></tr>
<tr><td>308</td><td>Martin Scorsese</td><td>606.0</td><td>116.87</td><td>Biography|Comedy|Crime|Drama</td><td>Leonardo DiCaprio</td><td>The Wolf of Wall Street</td><td>780588</td><td>1138.0</td><td>English</td><td>100.0</td><td>2013.0</td><td>8.2</td><td>138000</td><td>16.87</td><td>Biography</td><td>Comedy</td></tr>
</table>

326	Martin Scorsese	233.0	77.68	Crime|Drama	Leonardo DiCaprio	Gangs of New York	314033	1166.0	English	100.0	2002.0	7.5	0	-22.32	Crime	Drama
361	Martin Scorsese	352.0	132.37	Crime|Drama|Thriller	Leonardo DiCaprio	The Departed	873649	2054.0	English	90.0	2006.0	8.5	29000	42.37	Crime	Drama
452	Martin Scorsese	490.0	127.97	Mystery|Thriller	Leonardo DiCaprio	Shutter Island	786092	964.0	English	80.0	2010.0	8.1	53000	47.97	Mystery	Thriller
641	Ridley Scott	238.0	39.38	Action|Drama|Thriller	Leonardo DiCaprio	Body of Lies	174248	263.0	English	70.0	2008.0	7.1	0	-30.62	Action	Drama
911	Steven Spielberg	194.0	164.44	Biography|Crime|Drama	Leonardo DiCaprio	Catch Me If You Can	525801	667.0	English	52.0	2002.0	8.0	15000	112.44	Biography	Crime
990	Danny Boyle	118.0	39.78	Adventure|Drama|Thriller	Leonardo DiCaprio	The Beach	176169	548.0	English	50.0	2000.0	6.6	0	-10.22	Adventure	Drama
1114	Sam Mendes	323.0	22.88	Drama|Romance	Leonardo DiCaprio	Revolutionary Road	152591	414.0	English	35.0	2008.0	7.3	0	-12.12	Drama	Romance
1422	Randall Wallace	83.0	56.88	Action|Adventure	Leonardo DiCaprio	The Man in the Iron Mask	125219	244.0	English	35.0	1998.0	6.4	0	21.88	Action	Adventure
1453	Clint Eastwood	392.0	37.30	Biography|Crime|Drama	Leonardo DiCaprio	J. Edgar	102728	279.0	English	35.0	2011.0	6.6	16000	2.30	Biography	Crime
1560	Sam Raimi	63.0	18.64	Action|Thriller|Western	Leonardo DiCaprio	The Quick and the Dead	69197	216.0	English	32.0	1995.0	6.4	0	-13.36	Action	Thriller
2067	Jerry Zaks	45.0	12.78	Drama	Leonardo DiCaprio	Marvin's Room	20163	71.0	English	23.0	1996.0	6.7	1000	-10.22	Drama	Drama
2757	Baz Luhrmann	106.0	46.34	Drama|Romance	Leonardo DiCaprio	Romeo + Juliet	167750	506.0	English	14.5	1996.0	6.8	10000	31.84	Drama	Romance
3476	Baz Luhrmann	490.0	144.81	Drama|Romance	Leonardo DiCaprio	The Great Gatsby	362933	753.0	English	105.0	2013.0	7.3	115000	39.81	Drama	Romance

[142]:

```
Brad_Pitt = movies[movies["actor_1_name"] == "Brad Pitt"]
Brad_Pitt.head()
```

[142...

	director_name	num_critic_for_reviews	gross	genres	actor_1_name	movie_title	num_voted_users	num_user_for_reviews	language	budget	title_year	imdb_score	movie_facebook_likes	profit	genre_1	genre_2
101	David Fincher	362.0	127.49	Drama|Fantasy|Romance	Brad Pitt	The Curious Case of Benjamin Button	459346	822.0	English	150.0	2008.0	7.8	23000	-22.51	Drama	Fantasy
147	Wolfgang Petersen	220.0	133.23	Adventure	Brad Pitt	Troy	381672	1694.0	English	175.0	2004.0	7.2	0	-41.77	Adventure	Adventure
254	Steven Soderbergh	198.0	125.53	Crime|Thriller	Brad Pitt	Ocean's Twelve	284852	627.0	English	110.0	2004.0	6.4	0	15.53	Crime	Thriller
255	Doug Liman	233.0	186.34	Action|Comedy|Crime|Romance|Thriller	Brad Pitt	Mr. & Mrs. Smith	348861	798.0	English	120.0	2005.0	6.5	0	66.34	Action	Comedy
382	Tony Scott	142.0	0.03	Action|Crime|Thriller	Brad Pitt	Spy Game	121259	361.0	English	92.0	2001.0	7.0	0	-91.97	Action	Crime

Checkpoint 6: `Leonardo` has aced both the lists!

[143]:

```
# Write your code for combining the three dataframes here
combined = pd.concat([Meryl_Streep,Leo_Caprio,Brad_Pitt ], axis = 0)
combined
```

[143...

	director_name	num_critic_for_reviews	gross	genres	actor_1_name	movie_title	num_voted_users	num_user_for_reviews	language	budget	title_year	imdb_score	movie_facebook_likes	profit	genre_1	genre_
410	Nancy Meyers	187.0	112.70	Comedy|Drama|Romance	Meryl Streep	It's Complicated	69860	214.0	English	85.0	2009.0	6.6	0	27.70	Comedy	Dram
1106	Curtis Hanson	42.0	46.82	Action|Adventure|Crime|Thriller	Meryl Streep	The River Wild	32544	69.0	English	45.0	1994.0	6.3	0	1.82	Action	Adventur
1204	Nora Ephron	252.0	94.13	Biography|Drama|Romance	Meryl Streep	Julie & Julia	79264	277.0	English	40.0	2009.0	7.0	13000	54.13	Biography	Dram
1408	David Frankel	208.0	124.73	Comedy|Drama|Romance	Meryl Streep	The Devil Wears Prada	286178	631.0	English	35.0	2006.0	6.8	0	89.73	Comedy	Dram
1483	Robert Redford	227.0	15.00	Drama|Thriller|War	Meryl Streep	Lions for Lambs	41170	298.0	English	35.0	2007.0	6.2	0	-20.00	Drama	Thrille
1575	Sydney Pollack	66.0	87.10	Biography|Drama|Romance	Meryl Streep	Out of Africa	52339	200.0	English	31.0	1985.0	7.2	0	56.10	Biography	Dram
1618	David Frankel	234.0	63.54	Comedy|Drama|Romance	Meryl Streep	Hope Springs	34258	178.0	English	30.0	2012.0	6.3	0	33.54	Comedy	Dram
1674	Carl Franklin	64.0	23.21	Drama	Meryl Streep	One True Thing	9283	112.0	English	30.0	1998.0	7.0	592	-6.79	Drama	Dram
1925	Stephen Daldry	174.0	41.60	Drama|Romance	Meryl Streep	The Hours	102123	660.0	English	25.0	2002.0	7.6	0	16.60	Drama	Romanc
2781	Phyllida Lloyd	331.0	29.96	Biography|Drama|History	Meryl Streep	The Iron Lady	82327	350.0	English	13.0	2011.0	6.4	18000	16.96	Biography	Dram
3135	Robert Altman	211.0	20.34	Comedy|Drama|Music	Meryl Streep	A Prairie Home Companion	19655	280.0	English	10.0	2006.0	6.8	683	10.34	Comedy	Dram
26	James Cameron	315.0	658.67	Drama|Romance	Leonardo DiCaprio	Titanic	793059	2528.0	English	200.0	1997.0	7.7	26000	458.67	Drama	Romanc
50	Baz Luhrmann	490.0	144.81	Drama|Romance	Leonardo DiCaprio	The Great Gatsby	362912	753.0	English	105.0	2013.0	7.3	115000	39.81	Drama	Romanc
97	Christopher Nolan	642.0	292.57	Action|Adventure|Sci-Fi|Thriller	Leonardo DiCaprio	Inception	1468200	2803.0	English	160.0	2010.0	8.8	175000	132.57	Action	Adventur
179	Alejandro G. Iñárritu	556.0	183.64	Adventure|Drama|Thriller|Western	Leonardo DiCaprio	The Revenant	406020	1188.0	English	135.0	2015.0	8.1	190000	48.64	Adventure	Dram
257	Martin Scorsese	267.0	102.61	Biography|Drama	Leonardo DiCaprio	The Aviator	264318	799.0	English	110.0	2004.0	7.5	0	-7.39	Biography	Dram
296	Quentin Tarantino	765.0	162.80	Drama|Western	Leonardo DiCaprio	Django Unchained	955174	1193.0	English	100.0	2012.0	8.5	199000	62.80	Drama	Wester
307	Edward Zwick	166.0	57.37	Adventure|Drama|Thriller	Leonardo DiCaprio	Blood Diamond	400292	657.0	English	100.0	2006.0	8.0	14000	-42.63	Adventure	Dram
308	Martin Scorsese	606.0	116.87	Biography|Comedy|Crime|Drama	Leonardo DiCaprio	The Wolf of Wall Street	780588	1138.0	English	100.0	2013.0	8.2	138000	16.87	Biography	Comed
326	Martin Scorsese	233.0	77.68	Crime|Drama	Leonardo DiCaprio	Gangs of New York	314033	1166.0	English	100.0	2002.0	7.5	0	-22.32	Crime	Dram
361	Martin Scorsese	352.0	132.37	Crime|Drama|Thriller	Leonardo DiCaprio	The Departed	873649	2054.0	English	90.0	2006.0	8.5	29000	42.37	Crime	Dram
452	Martin Scorsese	490.0	127.97	Mystery|Thriller	Leonardo DiCaprio	Shutter Island	786092	964.0	English	80.0	2010.0	8.1	53000	47.97	Mystery	Thrille
641	Ridley Scott	238.0	39.38	Action|Drama|Thriller	Leonardo DiCaprio	Body of Lies	174248	263.0	English	70.0	2008.0	7.1	0	-30.62	Action	Dram
911	Steven Spielberg	194.0	164.44	Biography|Crime|Drama	Leonardo DiCaprio	Catch Me If You Can	525801	667.0	English	52.0	2002.0	8.0	15000	112.44	Biography	Crim
990	Danny Boyle	118.0	39.78	Adventure|Drama|Thriller	Leonardo DiCaprio	The Beach	176169	548.0	English	50.0	2000.0	6.6	0	-10.22	Adventure	Dram
1114	Sam Mendes	323.0	22.88	Drama|Romance	Leonardo DiCaprio	Revolutionary Road	152591	414.0	English	35.0	2008.0	7.3	0	-12.12	Drama	Romanc
1422	Randall Wallace	83.0	56.88	Action|Adventure	Leonardo DiCaprio	The Man in the Iron	125219	244.0	English	35.0	1998.0	6.4	0	21.88	Action	Adventur

						Mask										
1453	Clint Eastwood	392.0	37.30	Biography|Crime|Drama	Leonardo DiCaprio	J. Edgar	102728	279.0	English	35.0	2011.0	6.6	16000	2.30	Biography	Crim
1560	Sam Raimi	63.0	18.64	Action|Thriller|Western	Leonardo DiCaprio	The Quick and the Dead	69197	216.0	English	32.0	1995.0	6.4	0	-13.36	Action	Thrille
2067	Jerry Zaks	45.0	12.78	Drama	Leonardo DiCaprio	Marvin's Room	20163	71.0	English	23.0	1996.0	6.7	1000	-10.22	Drama	Dram
2757	Baz Luhrmann	106.0	46.34	Drama|Romance	Leonardo DiCaprio	Romeo + Juliet	167750	506.0	English	14.5	1996.0	6.8	10000	31.84	Drama	Romanc
3476	Baz Luhrmann	490.0	144.81	Drama|Romance	Leonardo DiCaprio	The Great Gatsby	362933	753.0	English	105.0	2013.0	7.3	115000	39.81	Drama	Romanc
101	David Fincher	362.0	127.49	Drama|Fantasy|Romance	Brad Pitt	The Curious Case of Benjamin Button	459346	822.0	English	150.0	2008.0	7.8	23000	-22.51	Drama	Fantas
147	Wolfgang Petersen	220.0	133.23	Adventure	Brad Pitt	Troy	381672	1694.0	English	175.0	2004.0	7.2	0	-41.77	Adventure	Adventur
254	Steven Soderbergh	198.0	125.53	Crime|Thriller	Brad Pitt	Ocean's Twelve	284852	627.0	English	110.0	2004.0	6.4	0	15.53	Crime	Thrille
255	Doug Liman	233.0	186.34	Action|Comedy|Crime|Romance|Thriller	Brad Pitt	Mr. & Mrs. Smith	348861	798.0	English	120.0	2005.0	6.5	0	66.34	Action	Comed
382	Tony Scott	142.0	0.03	Action|Crime|Thriller	Brad Pitt	Spy Game	121259	361.0	English	92.0	2001.0	7.0	0	-91.97	Action	Crim
400	Steven Soderbergh	186.0	183.41	Crime|Thriller	Brad Pitt	Ocean's Eleven	402645	845.0	English	85.0	2001.0	7.8	0	98.41	Crime	Thrille
470	David Ayer	406.0	85.71	Action|Drama|War	Brad Pitt	Fury	303185	701.0	English	68.0	2014.0	7.6	82000	17.71	Action	Dram
611	Jean-Jacques Annaud	76.0	37.90	Adventure|Biography|Drama|History|War	Brad Pitt	Seven Years in Tibet	96385	119.0	English	70.0	1997.0	7.0	0	-32.10	Adventure	Biograph
683	David Fincher	315.0	37.02	Drama	Brad Pitt	Fight Club	1347461	2968.0	English	63.0	1999.0	8.8	48000	-25.98	Drama	Dram
792	Patrick Gilmore	98.0	26.29	Adventure|Animation|Comedy|Drama|Family|Fantas...	Brad Pitt	Sinbad: Legend of the Seven Seas	36144	91.0	English	60.0	2003.0	6.7	880	-33.71	Adventure	Animatio
940	Neil Jordan	120.0	105.26	Drama|Fantasy|Horror	Brad Pitt	Interview with the Vampire: The Vampire Chroni...	239752	406.0	English	60.0	1994.0	7.6	11000	45.26	Drama	Fantas
1490	Terrence Malick	584.0	13.30	Drama|Fantasy	Brad Pitt	The Tree of Life	136367	975.0	English	32.0	2011.0	6.7	39000	-18.70	Drama	Fantas
1722	Andrew Dominik	273.0	3.90	Biography|Crime|Drama|History|Western	Brad Pitt	The Assassination of Jesse James by the Coward...	136104	415.0	English	30.0	2007.0	7.5	0	-26.10	Biography	Crim
2204	Alejandro G. Iñárritu	285.0	34.30	Drama	Brad Pitt	Babel	243799	908.0	English	25.0	2006.0	7.5	0	9.30	Drama	Dram
2333	Angelina Jolie Pitt	131.0	0.53	Drama|Romance	Brad Pitt	By the Sea	7976	61.0	English	10.0	2015.0	5.3	0	-9.47	Drama	Romanc
2682	Andrew Dominik	414.0	14.94	Crime|Thriller	Brad Pitt	Killing Them Softly	111625	369.0	English	15.0	2012.0	6.2	20000	-0.06	Crime	Thrille
2898	Tony Scott	122.0	12.28	Action|Crime|Drama|Romance|Thriller	Brad Pitt	True Romance	163492	460.0	English	13.0	1993.0	8.0	15000	-0.72	Action	Crim

[144]:

```python
# Write your code for grouping the combined dataframe here
movies_by_lead = combined.groupby('actor_1_name')
```

[145]:

```python
# Write the code for finding the mean of critic reviews and audience reviews here
print(movies_by_lead["num_critic_for_reviews"].mean())
print(movies_by_lead["num_user_for_reviews"].mean())
```

```
actor_1_name
Brad Pitt            245.000000
Leonardo DiCaprio    330.190476
Meryl Streep         181.454545
Name: num_critic_for_reviews, dtype: float64
actor_1_name
Brad Pitt            742.352941
Leonardo DiCaprio    914.476190
Meryl Streep         297.181818
Name: num_user_for_reviews, dtype: float64
```

[ ]:

Console