**DATA 230 Project Report**

**Title - IMDb Movie Data Visualisation**

Name: Ashmit Pareek

Student ID: 017401293

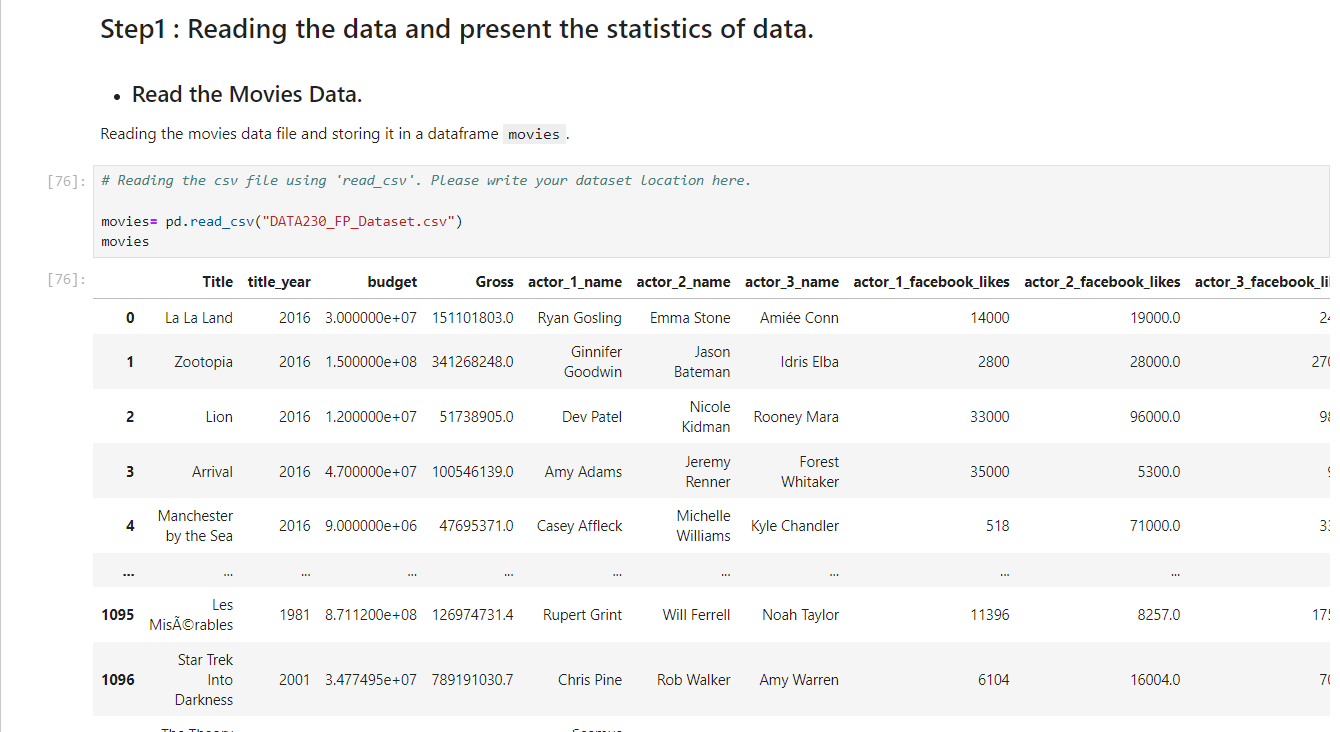
**https://github.com/ASHMITPAREEK/DATA-230-FINAL-PROJECT.git**

**Introduction**

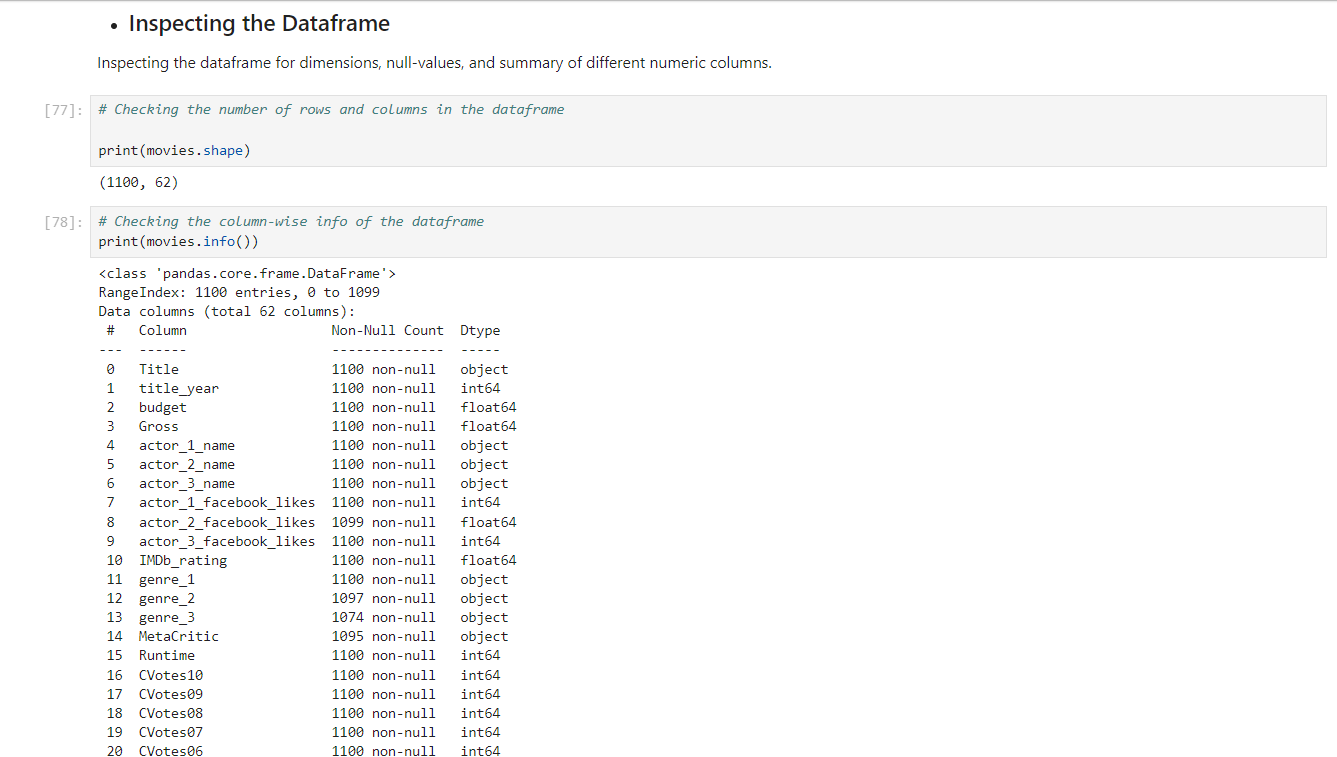
## This project undertakes an in-depth analysis of IMDb movie data, with a focus on uncovering trends within the film industry, understanding audience preferences, and evaluating critical reception. By utilizing a comprehensive dataset, the project employs various data visualization techniques to explore dimensions such as financial performance, viewer ratings, and demographic preferences. The ultimate objective is to derive insightful conclusions that enhance our understanding of the complexities within the film industry, shed light on the differential impact of genres, and highlight the interplay between demographic factors and the success of movies.

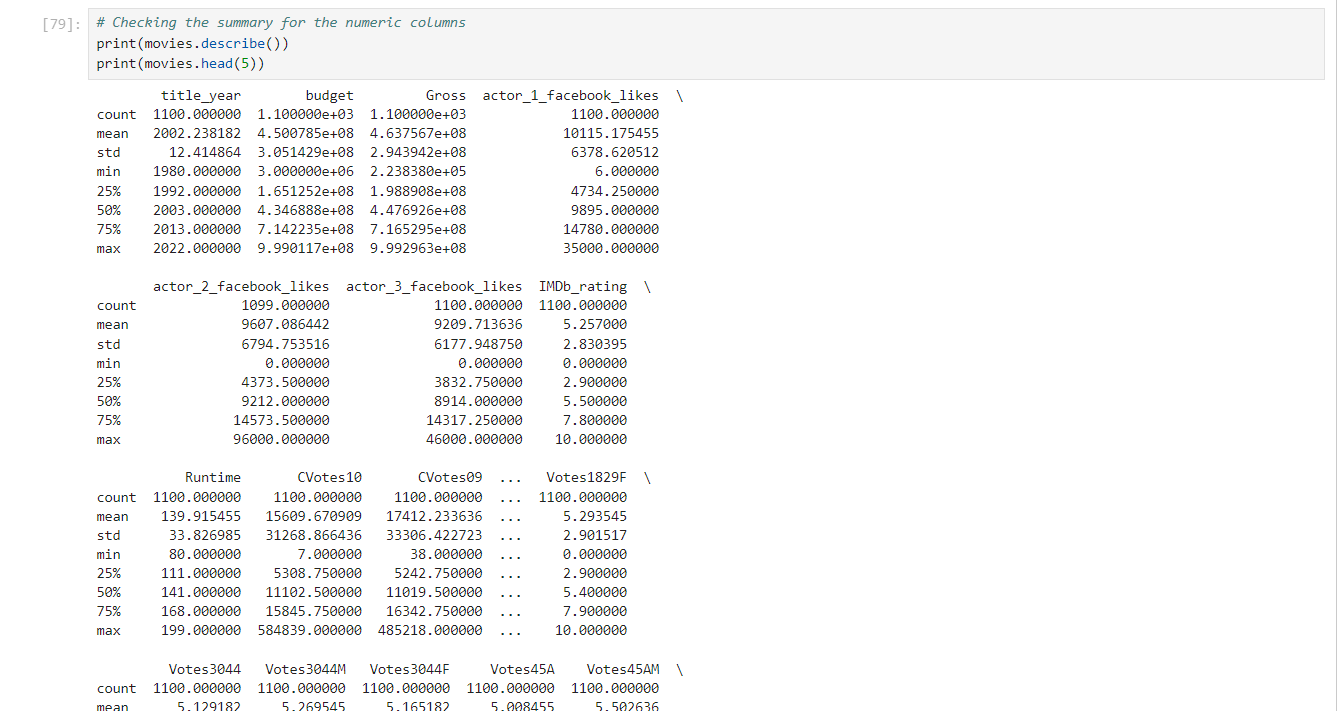
**Step1: Reading the dataset and present the statistics of data.**

* Reading the movies data file and storing it in a dataframe movies.

****

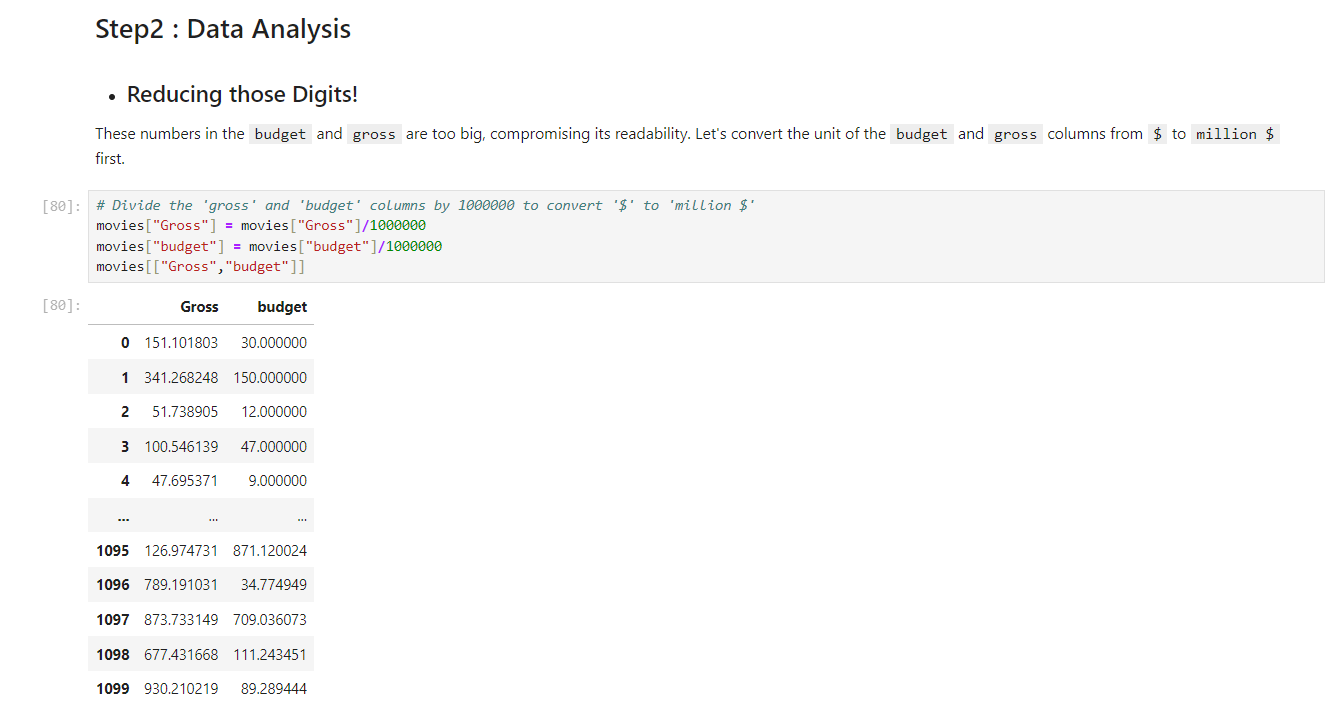
### Inspecting the Dataframe : Inspecting the dataframe for dimensions, null-values, and summary of different numeric columns.

****

****

**Step2 : Data Analysis**

### **Digit Normalization : These numbers in the budget and gross are too big, compromising its readability. Let's convert the unit of the budget and gross columns from $ to million $ first.**



### **Finding Profits : Let's Talk Profit!**

1. Creating a new column called `profit` which contains the difference of the two columns: `

gross` and `budget`.

2. Sorting the dataframe using the `profit` column as reference.

3. Extracting the top ten profiting movies in descending order and store them in a new

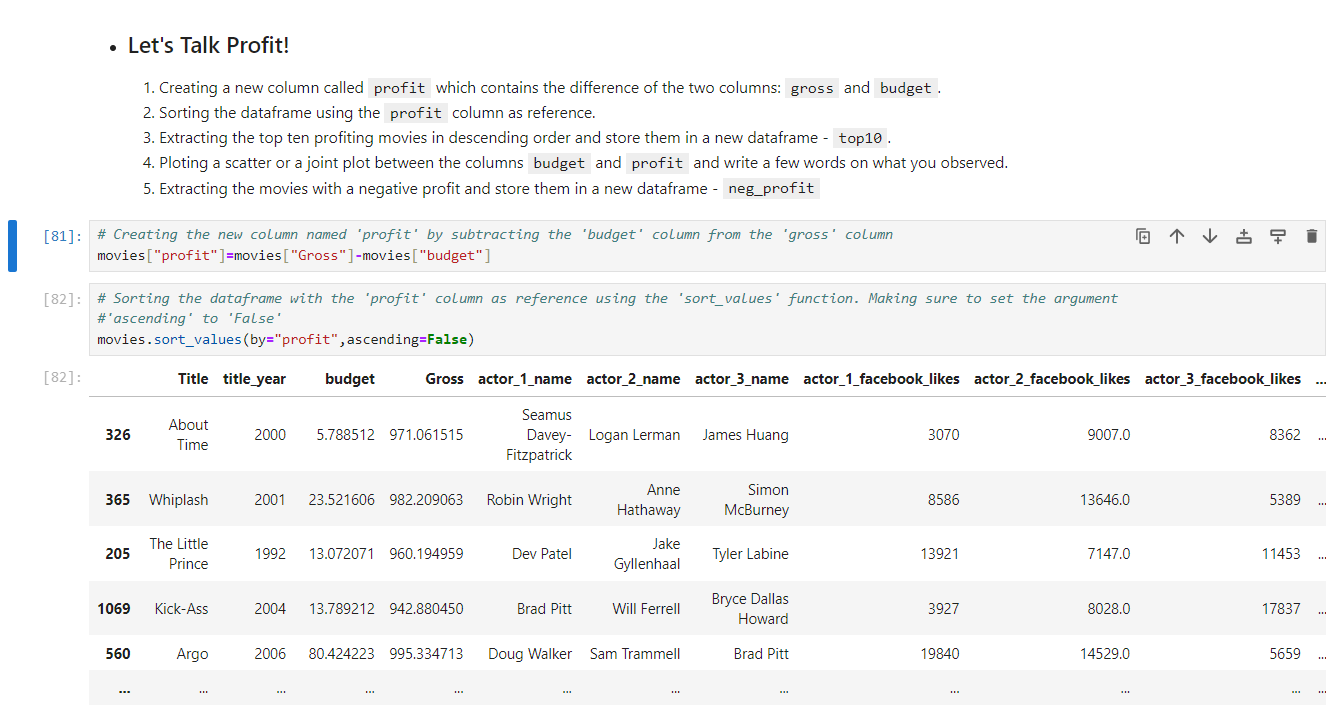
dataframe - `top10`.

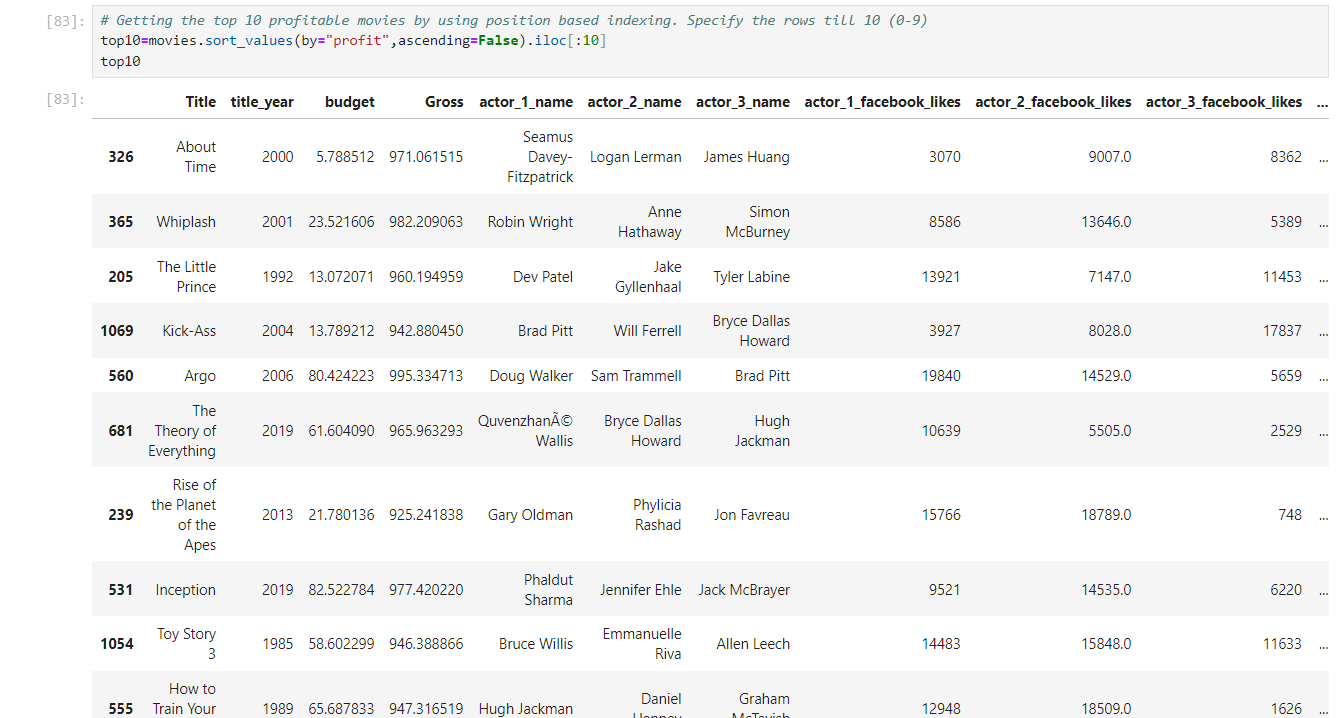
4. Ploting a scatter or a joint plot between the columns `budget` and `profit` and write a

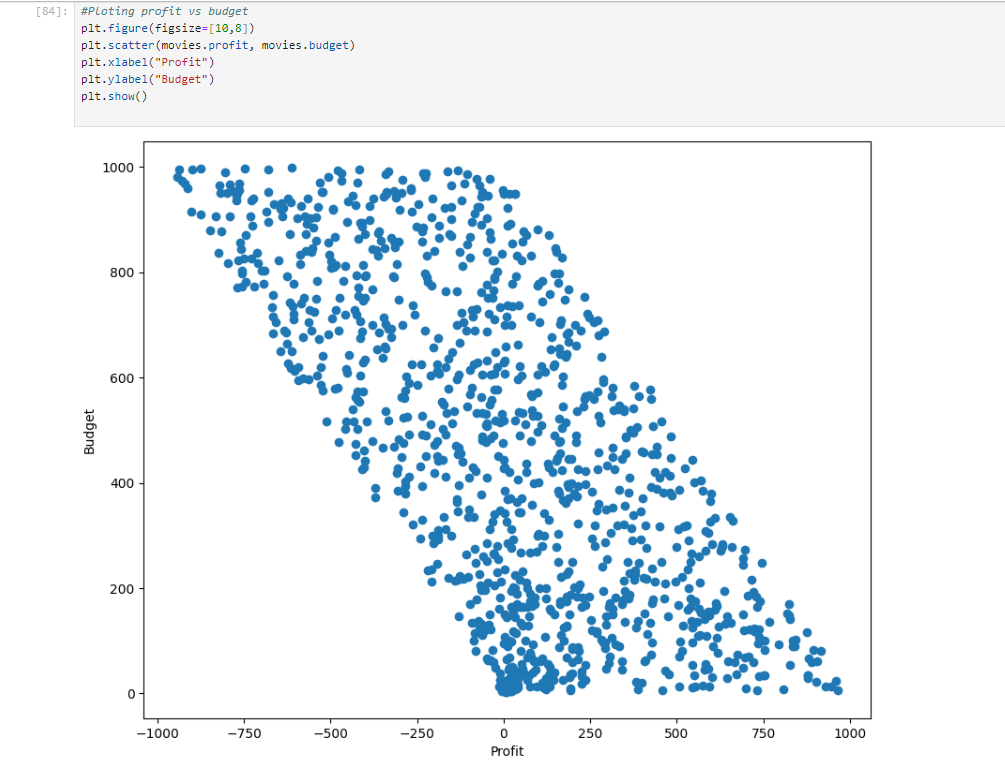
few words on what you observed.

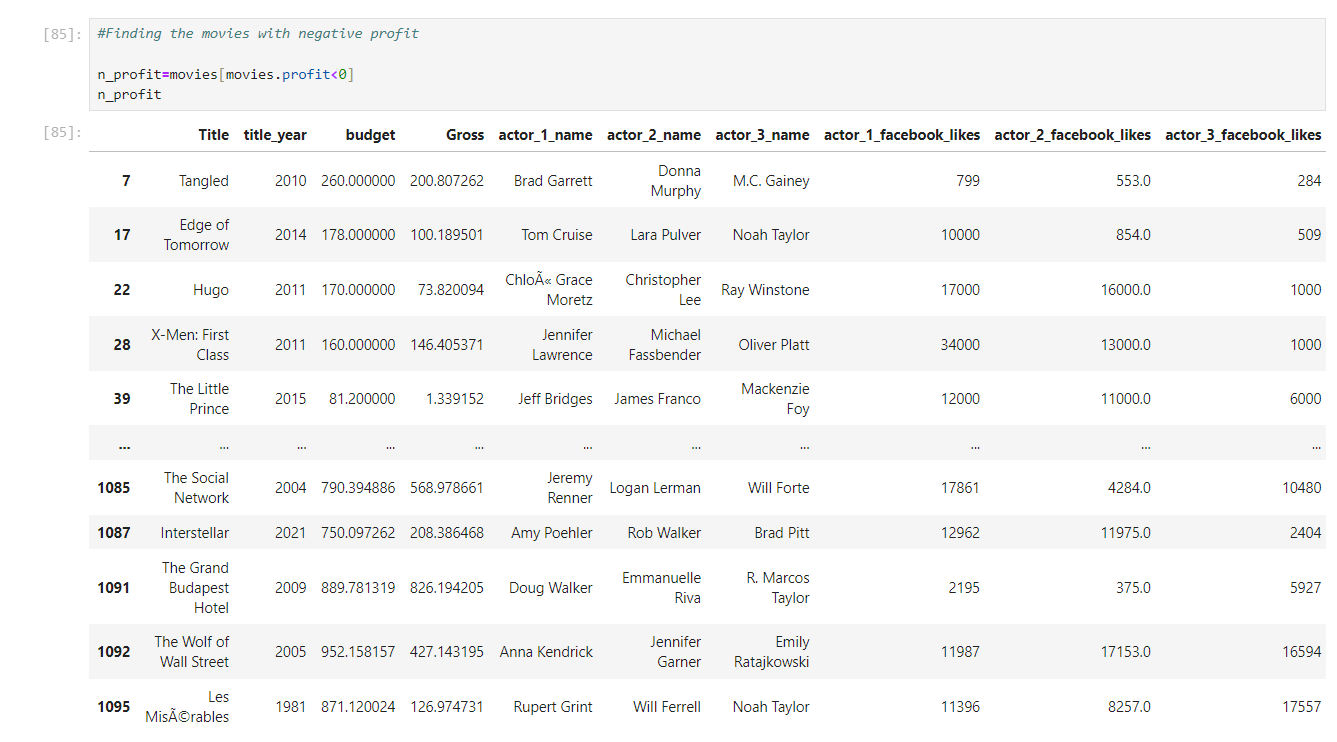
5. Extracting the movies with a negative profit and store them in a new dataframe -

neg\_profit`

****

****

****

****

**The movie 'Tangled'. Although its one of the highest grossing movies of all time, it has negative profit as per this result.**

### **The Relationships - General Audience and the Critics**

There is a column `MetaCritic` in this dataset. This is a very popular website where an average score is determined through the scores given by the top-rated critics. Secondly, we also have another column `IMDb\_rating` which tells you the IMDb rating of a movie. This rating is determined by taking the average of hundred-thousands of ratings from the general audience.

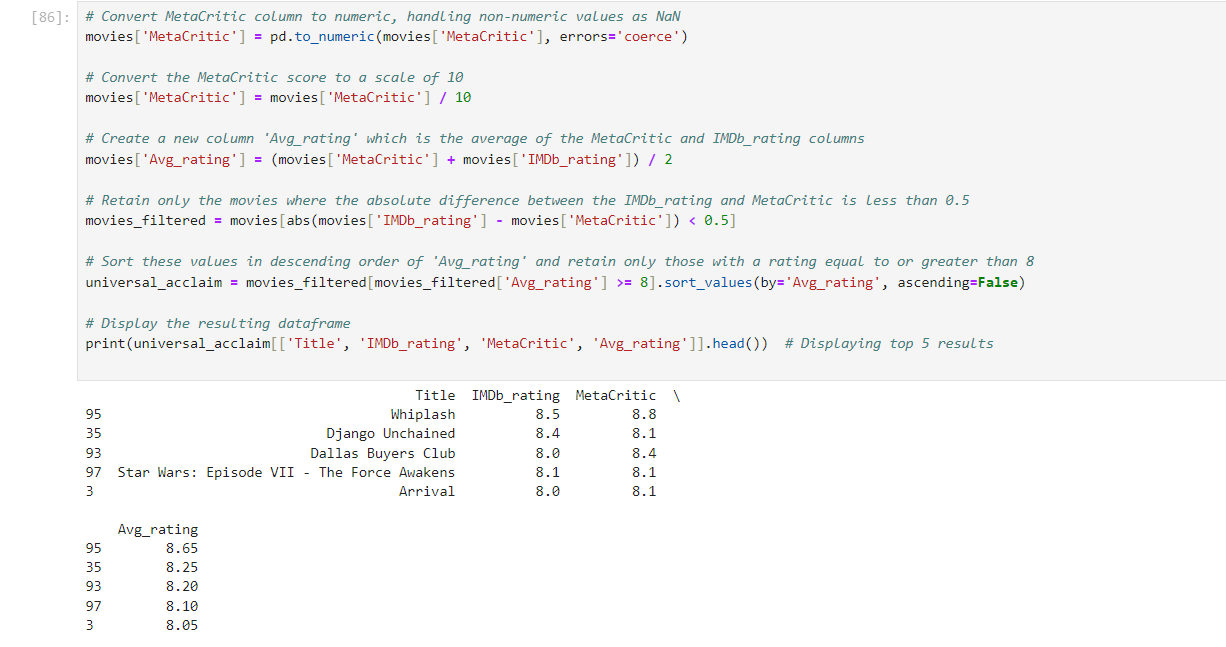
We need to find out the highest rated movies which have been liked by critics and audiences alike.

1. Firstly we will notice that the `MetaCritic` score is on a scale of `100` whereas the `IMDb\_rating` is on a scale of 10. First converting the `MetaCritic` column to a scale of 10.

2. Now, to find out the movies which have been liked by both critics and audiences alike and also have a high rating overall, we need to -

- Create a new column `Avg\_rating` which will have the average of the `MetaCritic` and `Rating` columns

- Retain only the movies in which the absolute difference(using abs() function) between the `IMDb\_rating` and `Metacritic` columns is less than 0.5.



The analysis of the dataset to find the highest rated movies that have been liked by both critics and audiences alike yielded the following results:

1.Whiplash - IMDb Rating: 8.5, MetaCritic Rating: 8.8, Average Rating: 8.65

2.Django Unchained - IMDb Rating: 8.4, MetaCritic Rating: 8.1, Average Rating: 8.25

3.Dallas Buyers Club - IMDb Rating: 8.0, MetaCritic Rating: 8.4, Average Rating: 8.20

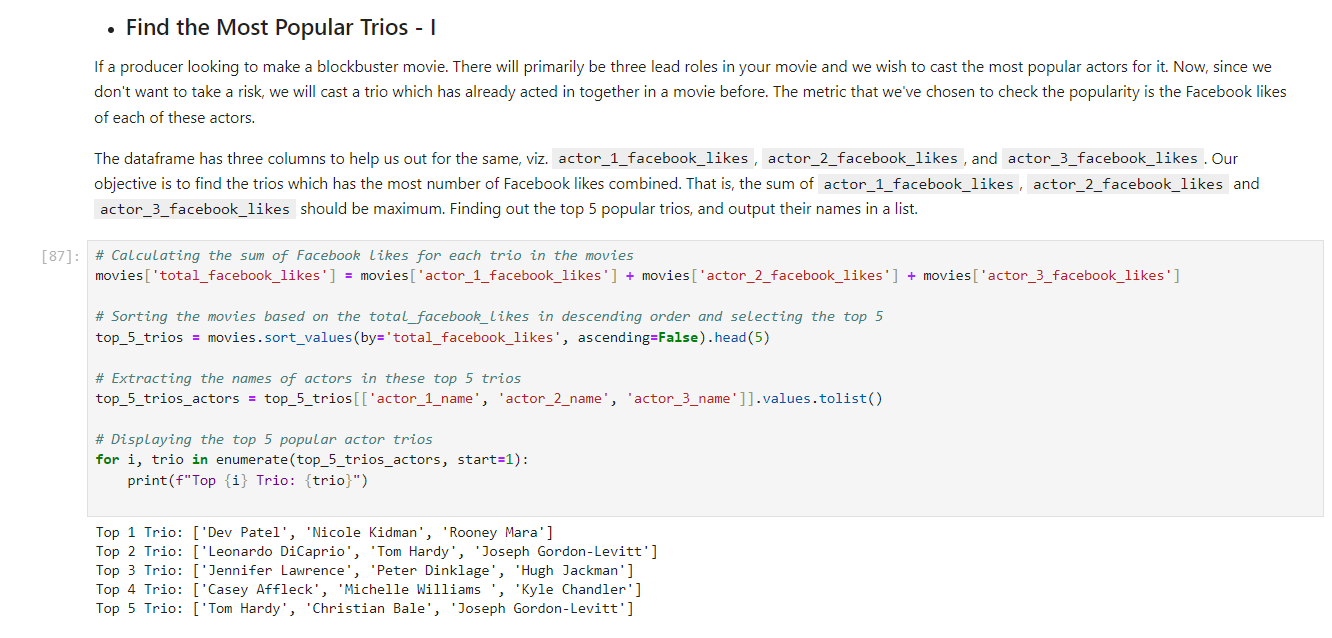
4.Star Wars: Episode VII - The Force Awakens - IMDb Rating: 8.1, MetaCritic Rating: 8.1, 5.Average Rating: 8.10

6.Arrival - IMDb Rating: 8.0, MetaCritic Rating: 8.1, Average Rating: 8.05

These movies have an average rating of 8 or higher and have a small difference (less than 0.5) between their IMDb and MetaCritic ratings, indicating a strong consensus between critics and general audiences. ​

### **Find the Most Popular Trios – I**

1. If a producer looking to make a blockbuster movie. There will primarily be three lead roles in your movie and we wish to cast the most popular actors for it. Now, since we don't want to take a risk, we will cast a trio which has already acted in together in a movie before. The metric that we've chosen to check the popularity is the Facebook likes of each of these actors.
2. The dataframe has three columns to help us out for the same, viz. actor\_1\_facebook\_likes, actor\_2\_facebook\_likes, and actor\_3\_facebook\_likes. Our objective is to find the trios which has the most number of Facebook likes combined. That is, the sum of actor\_1\_facebook\_likes, actor\_2\_facebook\_likes and actor\_3\_facebook\_likes should be maximum. Finding out the top 5 popular trios, and output their names in a list.



**Top 1 Trio: ['Dev Patel', 'Nicole Kidman', 'Rooney Mara']**

**Top 2 Trio: ['Leonardo DiCaprio', 'Tom Hardy', 'Joseph Gordon-Levitt']**

**Top 3 Trio: ['Jennifer Lawrence', 'Peter Dinklage', 'Hugh Jackman']**

**Top 4 Trio: ['Casey Affleck', 'Michelle Williams ', 'Kyle Chandler']**

**Top 5 Trio: ['Tom Hardy', 'Christian Bale', 'Joseph Gordon-Levitt']**

### **Find the Most Popular Trios – II**

In the previous task we found the popular trio based on the total number of facebook likes. Let's add a small condition to it and make sure that all three actors are popular. The condition is \*\*none of the three actors' Facebook likes should be less than half of the other two\*\*. For example, the following is a valid combo:

- actor\_1\_facebook\_likes: 70000

- actor\_2\_facebook\_likes: 40000

- actor\_3\_facebook\_likes: 50000

But the below one is not:

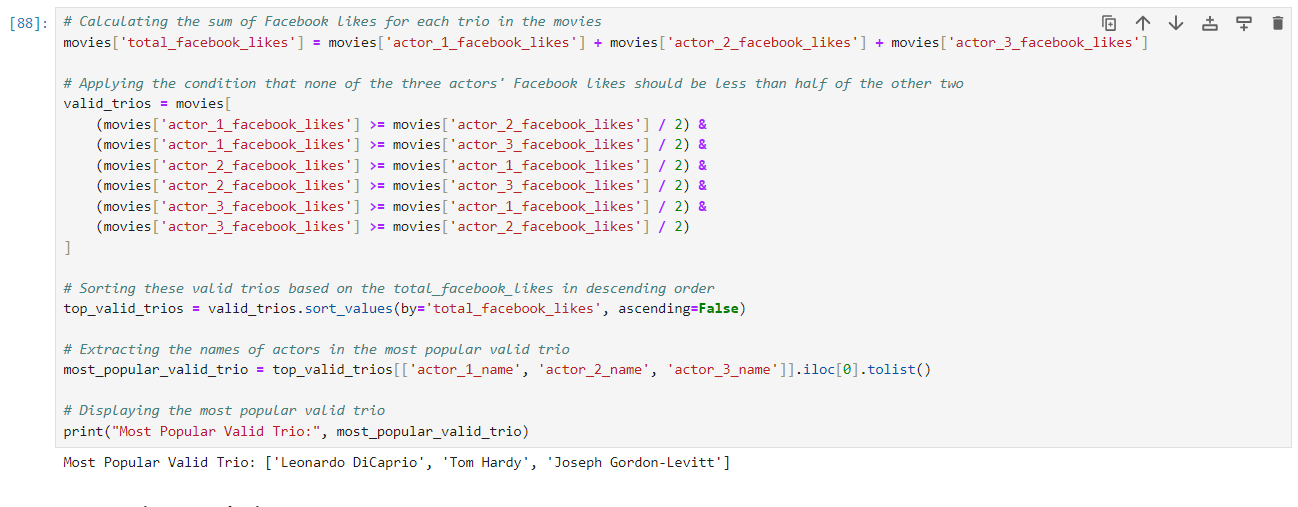
- actor\_1\_facebook\_likes: 70000

- actor\_2\_facebook\_likes: 40000

- actor\_3\_facebook\_likes: 30000

since in this case, `actor\_3\_facebook\_likes` is 30000, which is less than half of `actor\_1\_facebook\_likes`.

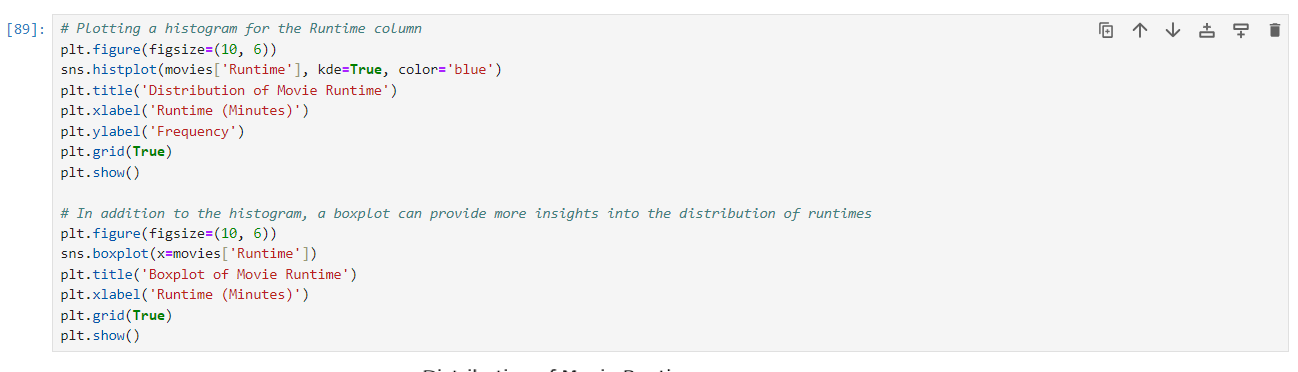
Having this condition ensures that we aren't getting any unpopular actor in your trio

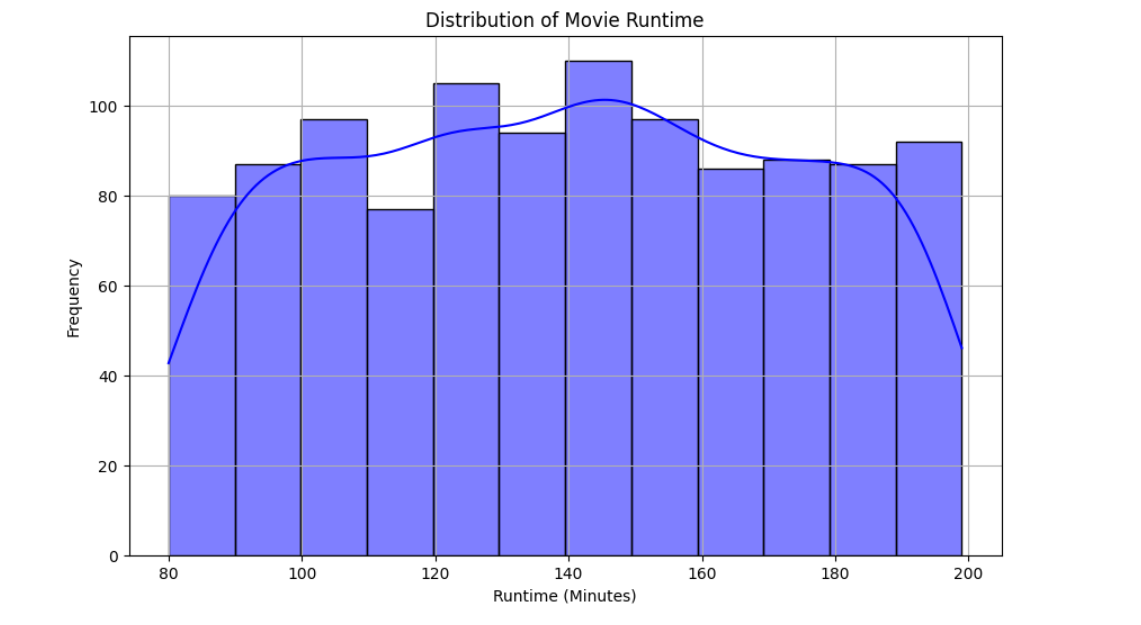


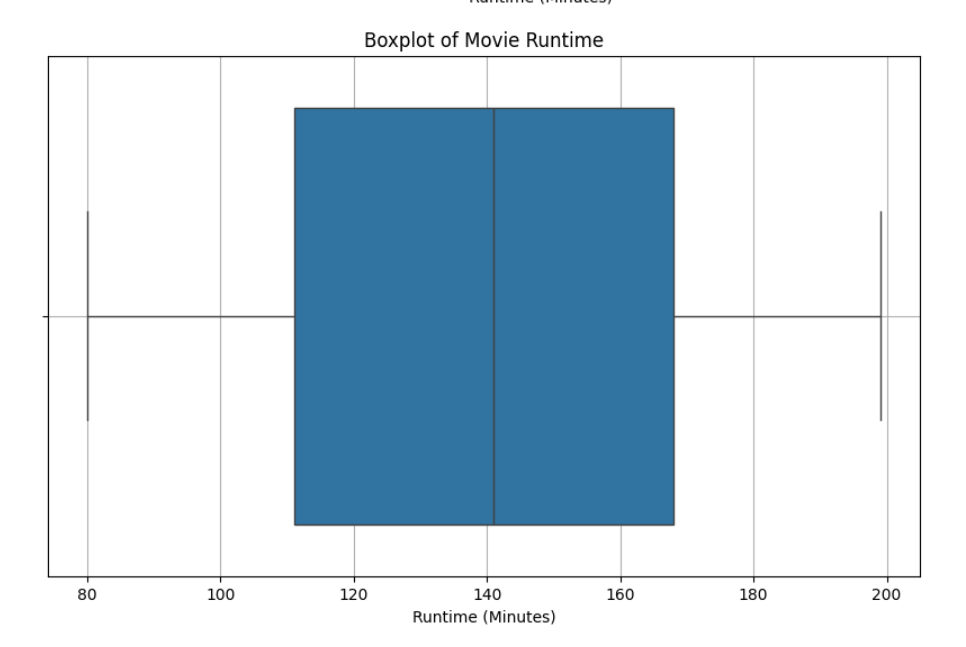
**Most Popular Valid Trio: ['Leonardo DiCaprio', 'Tom Hardy', 'Joseph Gordon-Levitt']**

* **Runtime Analysis**

There is a column named `Runtime` in the dataframe which primarily shows the length of the movie. It might be intersting to see how this variable this distributed. Ploting a `histogram` or `distplot` of seaborn to find the `Runtime` range most of the movies fall into.







**Most of the movies appear to be approximately 2.5 hour-long.**

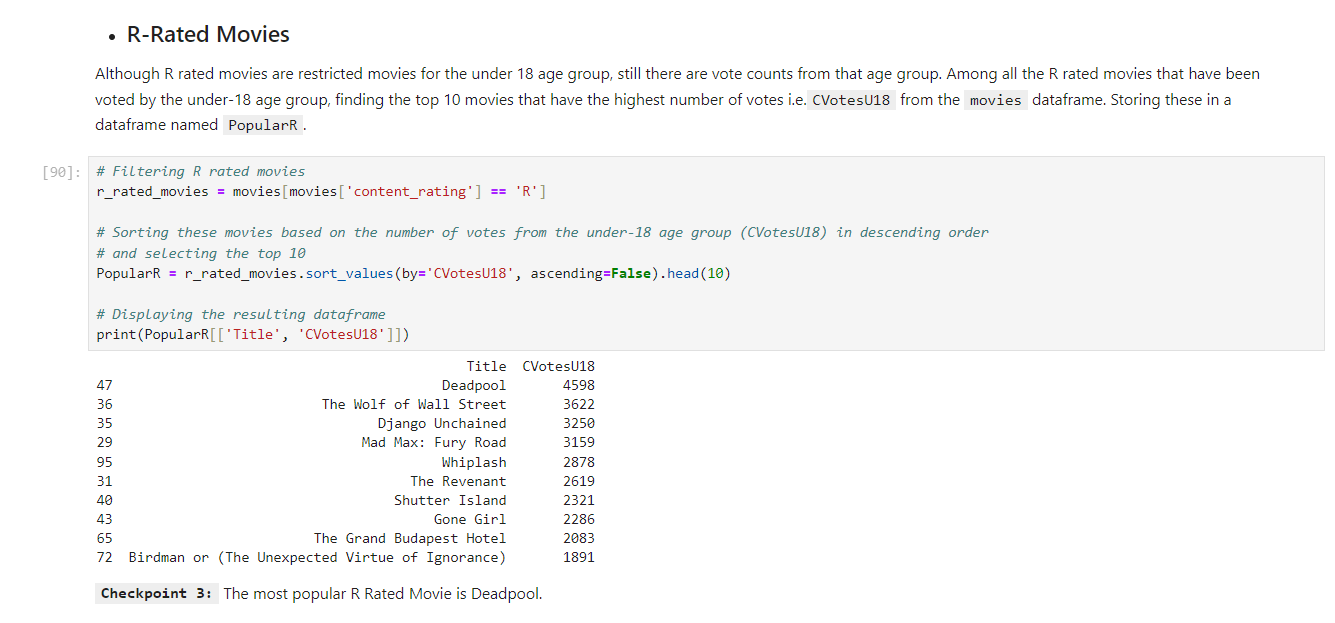
The first plot is a histogram with a Kernel Density Estimate (KDE) overlay, showing the distribution of movie runtimes. The histogram illustrates the frequency of movies across different runtime intervals, while the KDE provides a smooth estimation of the probability density function.

The second plot is a boxplot of movie runtimes. Boxplots are useful for visualizing the distribution of data through their quartiles. The box represents the interquartile range (IQR) with the median marked inside it. The whiskers extend to show the range of the data, and any points outside the whiskers are often considered outliers.

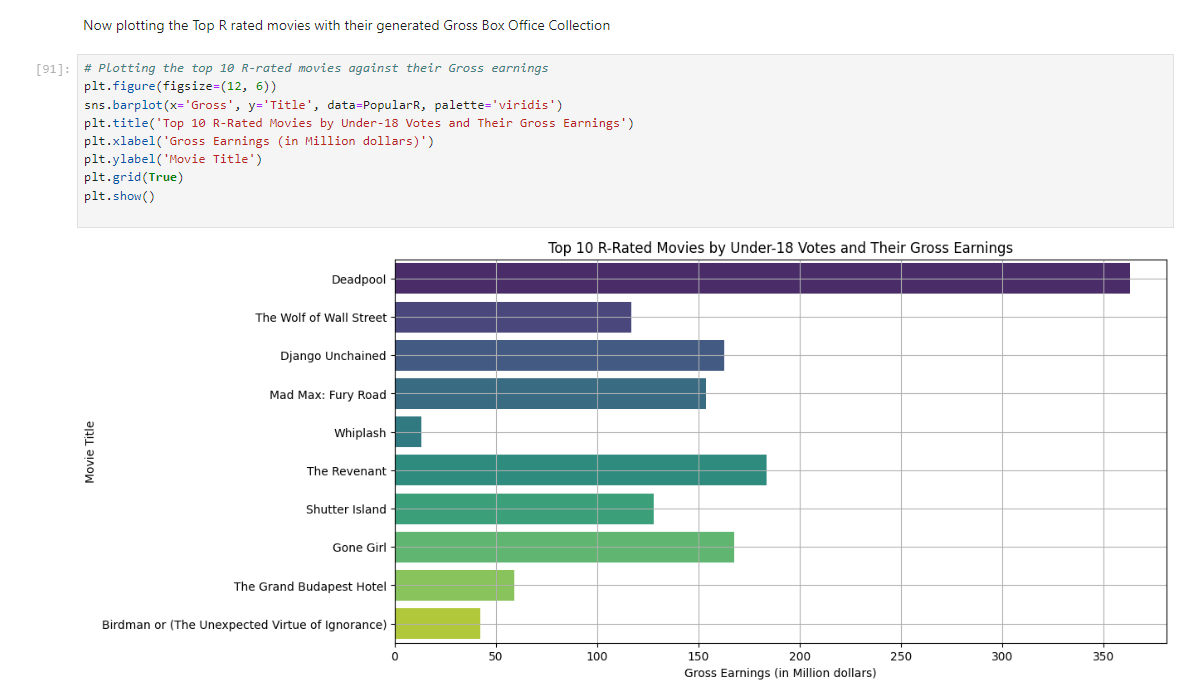
Together, these plots provide a comprehensive view of how the runtime of movies is distributed in your dataset. ​

### **R-Rated Movies**

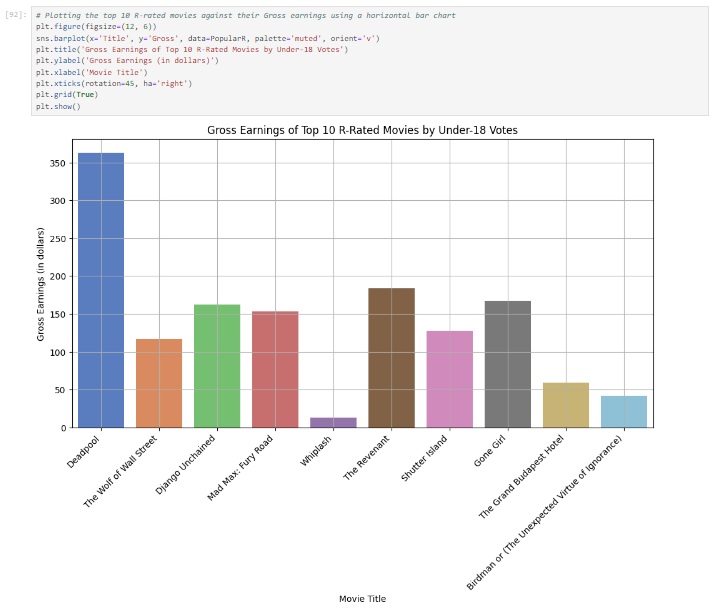
Although R rated movies are restricted movies for the under 18 age group, still there are vote counts from that age group. Among all the R rated movies that have been voted by the under-18 age group, finding the top 10 movies that have the highest number of votes i.e.CVotesU18 from the movies dataframe. Storing these in a dataframe named PopularR.



**Now plotting the Top R rated movies with their generated Gross Box Office Collection**



The bar plot above displays the top 10 R-rated movies, ranked by the number of votes they received from the under-18 age group, against their gross earnings. This visualization provides a clear comparison of the gross earnings of each movie, allowing you to see which of these popular R-rated movies among the younger audience also performed well in terms of box office revenue. ​



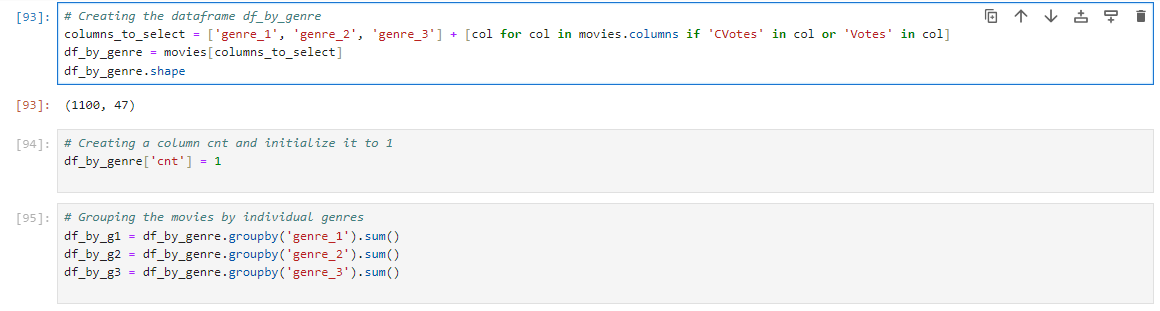
In this variation of the bar chart, each of the top 10 R-rated movies, popular among the under-18 age group, is plotted vertically against their gross earnings. This layout, with the movie titles on the x-axis and their earnings on the y-axis, provides a straightforward comparison of their financial performance. The rotation of the movie titles enhances readability, especially when dealing with longer names.

**Step3 : Demographic analysis**

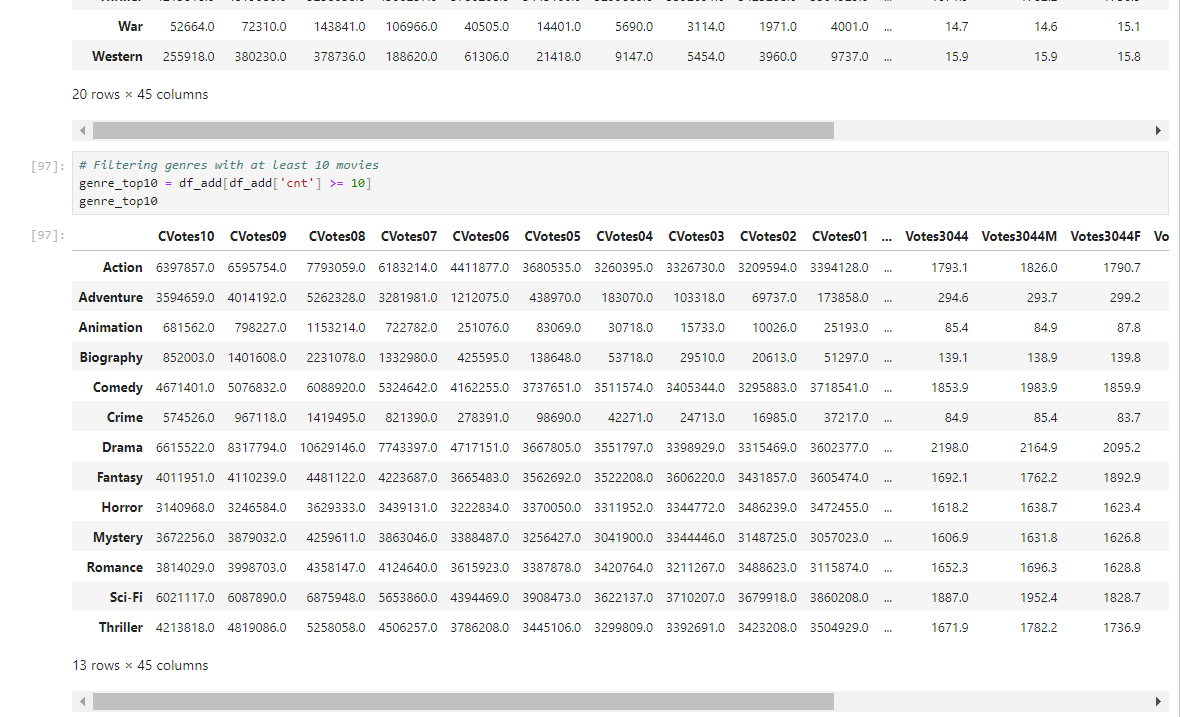
If we take a look at the last columns in the data frame, most of these are related to demographics of the voters. We also have three genre columns indicating the genres of a particular movie. We will analyse the voters across all demographics and also see how these vary across various genres.

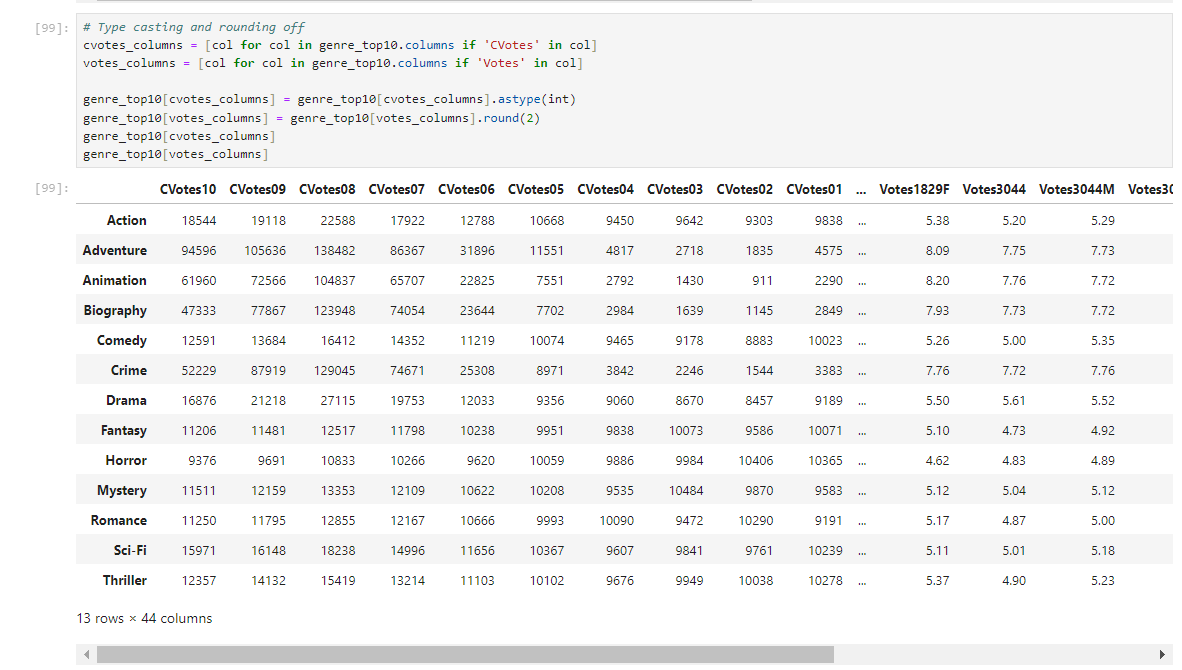
There are 3 columns in the data frame - genre\_1, genre\_2, and genre\_3. As a part of this task, we need to aggregate a few values over these 3 columns.

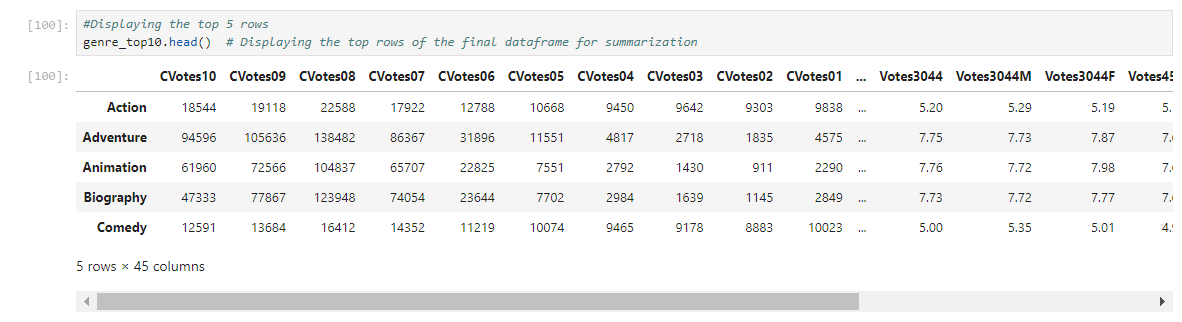
1. First creating a new dataframe df\_by\_genre that contains genre\_1, genre\_2, and genre\_3 and all the columns related to **CVotes/Votes** from the movies data frame. There are 47 columns to be extracted in total.
2. Now, Adding a column called cnt to the dataframe df\_by\_genre and initialize it to one. we will realise the use of this column by the end of this subtask.
3. First grouping the dataframe df\_by\_genre by genre\_1 and find the sum of all the numeric columns such as cnt, columns related to CVotes and Votes columns and store it in a dataframe df\_by\_g1.
4. Performing the same operation for genre\_2 and genre\_3 and store it dataframes df\_by\_g2 and df\_by\_g3 respectively.
5. Now that we have 3 dataframes performed by grouping over genre\_1, genre\_2, and genre\_3 separately, it's time to combine them. For this, adding the three dataframes and storing it in a new dataframe df\_add, so that the corresponding values of Votes/CVotes get added for each genre.There is a function called add() in pandas which lets you do this.
6. The column cnt on aggregation has basically kept the track of the number of occurences of each genre.Subset the genres that have atleast 10 movies into a new dataframe genre\_top10 based on the cnt column value.
7. Now, taking the mean of all the numeric columns by dividing them with the column value cnt and store it back to the same dataframe. We will be using this dataframe for further analysis in this task unless it is explicitly mentioned to use the dataframe movies.
8. Since the number of votes can't be a fraction, type cast all the CVotes related columns to integers. Also, rounding off all the Votes related columns upto two digits after the decimal point.









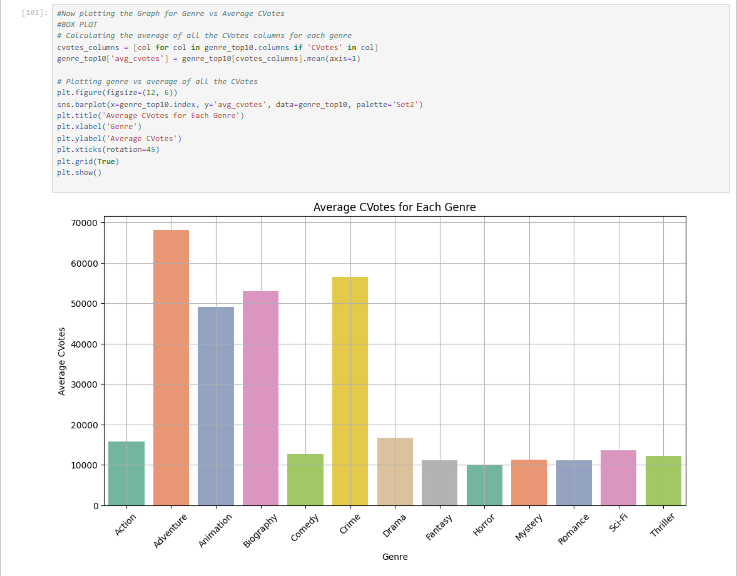


The final dataframe genre\_top10 provides a comprehensive view of the average voting patterns across different genres, taking into account the frequency of each genre in the dataset. The top rows of this dataframe are displayed above for reference. ​

**Step4: Data Visualization**

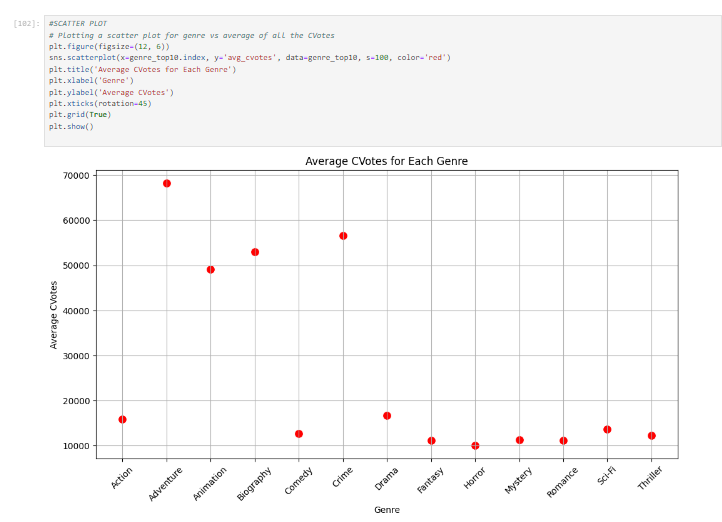
* **Genre vs Average CVotes**.

**1.Bar Chart**



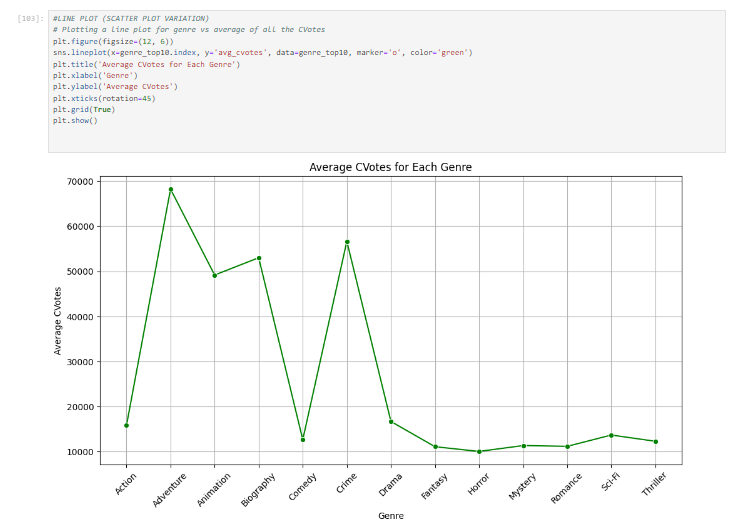
The bar plot above illustrates the average number of CVotes (cumulative votes across different voting categories) for each genre. This visualization helps in comparing the relative popularity of genres in terms of audience engagement and voting patterns on the platform from which this data was sourced. The genres are displayed along the x-axis, and the average CVotes are represented on the y-axis, providing a clear view of how genres rank in terms of viewer voting behavior.

**2.Scatter Plot**



The scatter plot above also depicts the average number of CVotes for each genre, similar to the bar plot but in a different visual format. Each dot represents a genre, with its position along the y-axis indicating the average CVotes. This type of plot is particularly useful for highlighting the distribution and individual data points, making it easier to identify patterns or outliers within genres based on audience voting behavior. ​

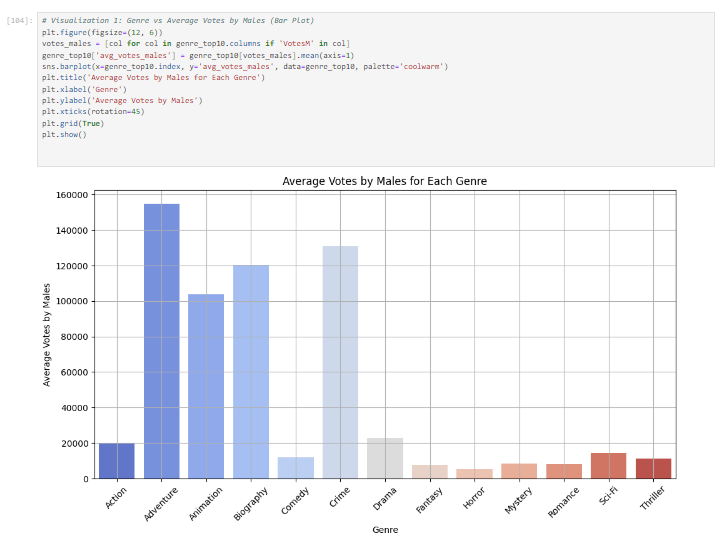
**3.Line Plot**



The line plot above provides another perspective on the average number of CVotes for each genre. In this visualization, each genre is connected by a line, with markers indicating the average CVotes for that genre. This type of plot is useful for observing trends and comparing genres more dynamically. The continuity of the line helps in visualizing the progression and relative standing of each genre in terms of audience engagement and voting patterns.

### **Different Visualizations from the Dataset**

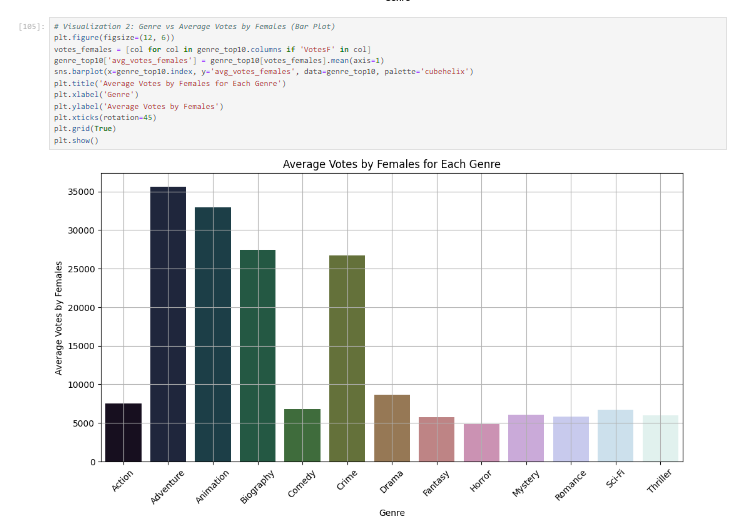
**# Visualization 1: Genre vs Average Votes by Males (Bar Plot)**

****

Average Votes by Males for Each Genre (Bar Plot): This bar plot shows how different genres are rated on average by male viewers. Some genres might have higher appeal or are rated more favorably by males, which is evident from the varying heights of the bars.

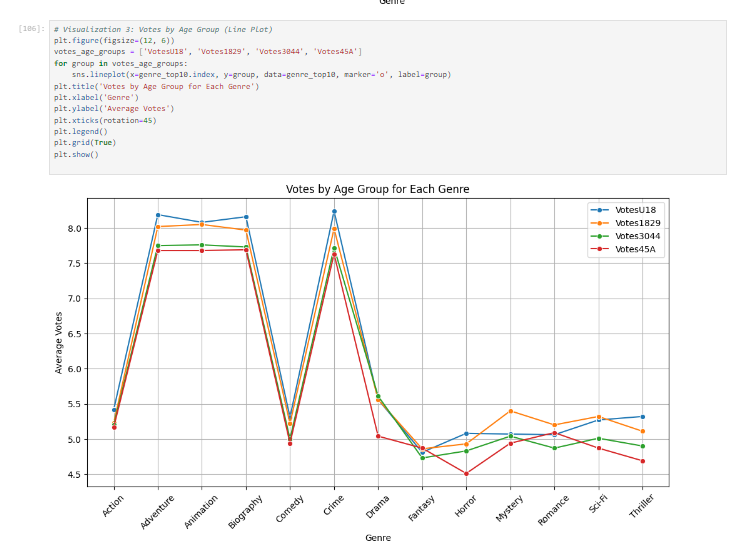
Here Adventure Genre has the most average votes by Men and Horror has the least average votes by men.

**# Visualization 2: Genre vs Average Votes by Females (Bar Plot)**

****

Average Votes by Females for Each Genre (Bar Plot): Similar to the previous plot, but focusing on female viewers. Comparing this with the male viewers' plot can reveal interesting differences in preferences or ratings between genders for different genres.

**# Visualization 3: Votes by Age Group (Line Plot)**



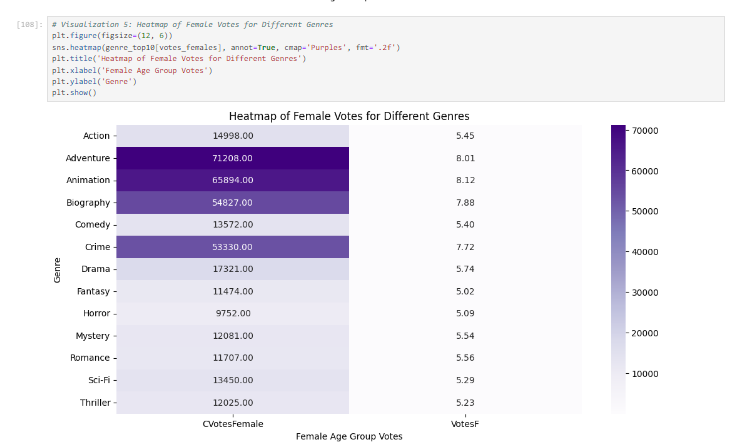
Votes by Age Group for Each Genre (Line Plot): This line plot displays how various age groups rate different genres. Each line represents an age group, providing a comparative view of their preferences. It's useful for understanding which genres are more popular or highly rated among different age groups

**# Visualization 4: Heatmap of Male Votes for Different Genres**



Heatmap of Male Votes for Different Genres: This heatmap provides a detailed view of how male viewers of different age groups rate each genre. The color intensity shows the level of rating, with warmer colors indicating higher ratings. It's a great way to visualize complex data and identify patterns or outliers.

**# Visualization 5: Heatmap of Female Votes for Different Genres**

****

Heatmap of Female Votes for Different Genres: Similar to the male votes heatmap, this one focuses on female viewers. It allows for a direct comparison of how female ratings vary across genres and age groups, which can be contrasted with the male ratings to spot gender-based trends.

Each of these visualizations offers unique insights into the dataset, highlighting how demographic factors like gender and age influence movie ratings across different genres. ​

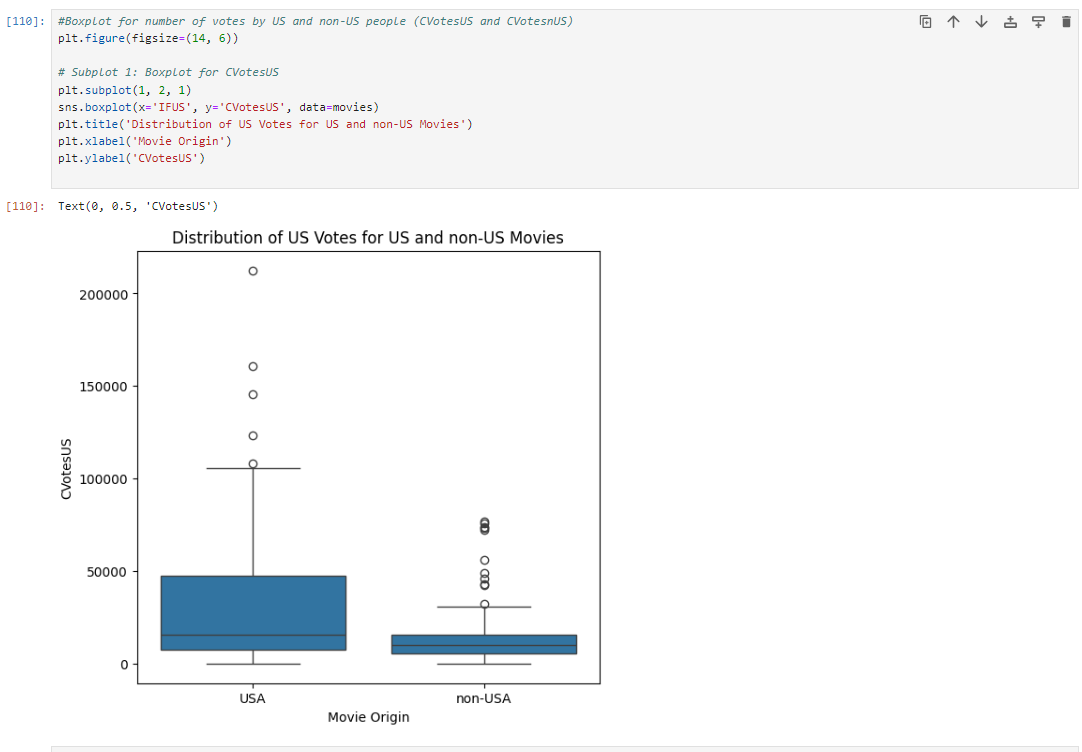
## **Step 5: US vs non-US Cross Analysis**

The dataset contains both the US and non-US movies.

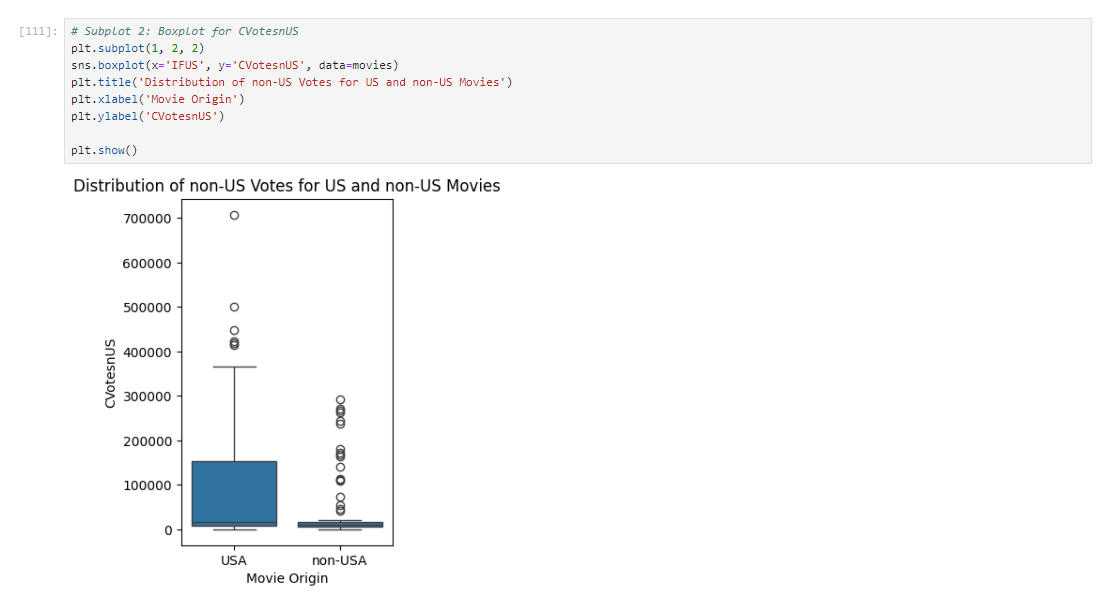
1. Creating a column IFUS in the dataframe movies. The column IFUS should contain the value "USA" if the Country of the movie is "USA". For all other countries other than the USA, IFUS should contain the value non-USA.
2. Now making a boxplot that shows how the number of votes from the US people i.e. CVotesUS is varying for the US and non-US movies. Making use of the column IFUS to make this plot. Similarly, making another subplot that shows how non US voters have voted for the US and non-US movies by plotting CVotesnUS for both the US and non-US movies. Writing any of two inferences/observations from these plots.
3. Again doing a similar analysis but with the ratings. Making a boxplot that shows how the ratings from the US people i.e. VotesUS is varying for the US and non-US movies. Similarly, making another subplot that shows how VotesnUS is varying for the US and non-US movies. Writing any of the two inferences/observations from these plots.



**#Boxplot for number of votes by US and non-US people (CVotesUS and CVotesnUS)**

****

**# Subplot 2: Boxplot for CvotesnUS**

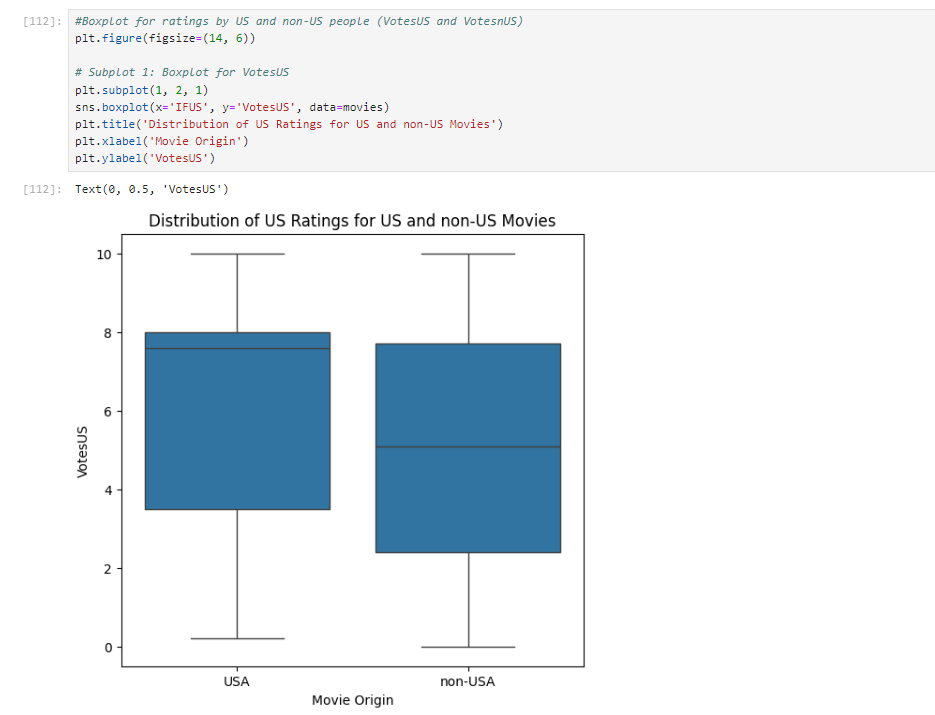
****

**Observations from Votes Count Boxplots (CVotesUS and CVotesnUS):**

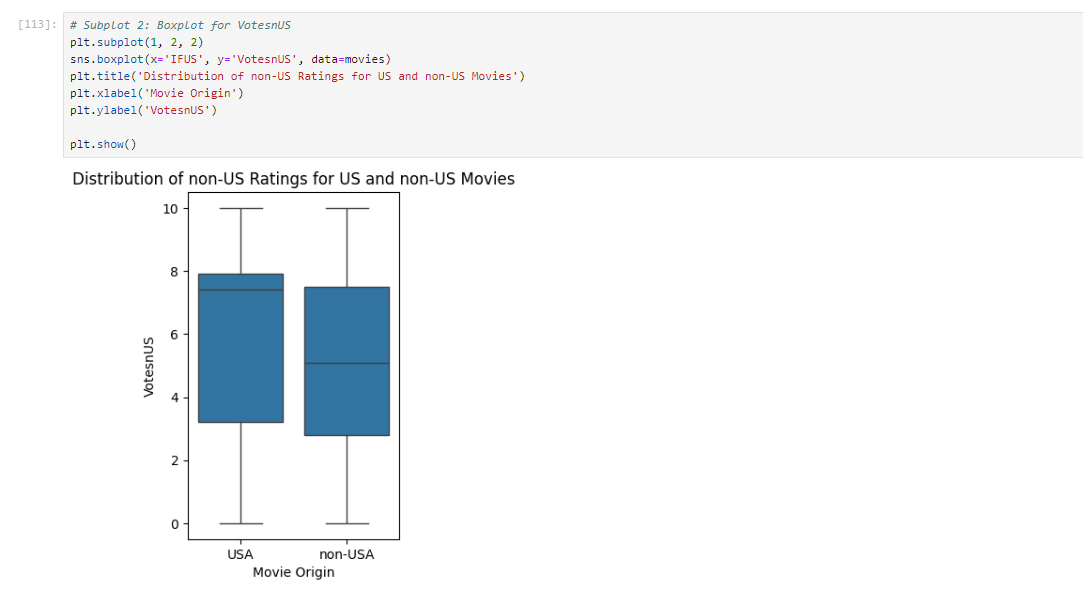
**US Votes Distribution:** The distribution of votes from US viewers (CVotesUS) shows a wider range and higher median for US movies compared to non-US movies. This suggests that US movies tend to receive more votes from US viewers, possibly indicating a higher level of engagement or popularity within the domestic market.

**Non-US Votes Distribution:** The boxplot for non-US votes (CVotesnUS) shows a similar trend, with US movies receiving a higher median number of votes from non-US viewers as well. However, the range of votes for non-US movies is more compact, indicating less variability in the number of votes non-US movies receive from international audiences.

**#Boxplot for ratings by US and non-US people (VotesUS and VotesnUS)**

****

**# Subplot 2: Boxplot for VotesnUS**

****

**Observations from Rating Boxplots (VotesUS and VotesnUS):**

**US Ratings Distribution:** The US ratings (VotesUS) boxplot demonstrates that US viewers rate US movies slightly higher on average compared to non-US movies. This could be attributed to cultural relevance, familiarity with the content, or patriotic bias.

**Non-US Ratings Distribution:** For non-US ratings (VotesnUS), both US and non-US movies seem to have similar median ratings, but US movies show a slightly wider interquartile range. This might imply that US movies have a more varied reception internationally, possibly due to differing tastes or cultural perceptions.

These observations suggest that there are noticeable differences in how US and non-US movies are received and rated by both domestic and international audiences.

**Conclusion**

The analysis reveals significant insights into the film industry. It highlights the correlation between profitability and audience ratings, underlines the popularity of specific actor trios, and elucidates demographic preferences across genres. The study also distinguishes between US and non-US movie reception, offering a global perspective. These findings are instrumental for filmmakers and marketers in strategizing productions and promotions. Future research could explore deeper into sub-genre analysis and changing trends in the digital streaming era.

**REFERENCES:** IMDB , KAGGLE, CHATGPT, [Mockaroo](https://www.mockaroo.com/), Upgrad