You 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. In this assignment, you will try to find some interesting insights into these movies and their voters, using Python.

### Subtask 1.1: Read the Movies Data.

Read the movies data file provided and store it in a data frame movies.

### Subtask 1.2: Inspect the Data frame

Inspect the data frame for dimensions, null-values, and summary of different numeric columns.

## Task 2: Data Analysis

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 visualization to get various insights about the data.

### Subtask 2.1: Reduce those Digits!

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.

### Subtask 2.2: Let's Talk Profit!

1. Create a new column called profit which contains the difference of the two columns: gross and budget.
2. Sort the data frame using the profit column as reference.
3. Extract the top ten profiting movies in descending order and store them in a new data frame - top10.
4. Plot a scatter or a joint plot between the columns budget and profit and write a few words on what you observed.
5. Extract the movies with a negative profit and store them in a new data frame - neg\_profit

### Subtask 2.3: The General Audience and the Critics

You might have noticed the 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. Second, you 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.

As a part of this subtask, you are required to find out the highest rated movies which have been liked by critics and audiences alike.

1. Firstly you will notice that the MetaCritic score is on a scale of 100 whereas the IMDb\_rating is on a scale of 10. First convert 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, you 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. Refer to this link to know how abs () function works - <https://www.geeksforgeeks.org/abs-in-python/> .
   * Sort these values in a descending order of Avg\_rating and retain only the movies with a rating equal to higher than 8 and store these movies in a new data frame Universal Acclaim.

### Subtask 2.4: Find the Most Popular Trios - I

You're a producer looking to make a blockbuster movie. There will primarily be three lead roles in your movie and you wish to cast the most popular actors for it. Now, since you don't want to take a risk, you will cast a trio which has already acted in together in a movie before. The metric that you've chosen to check the popularity is the Facebook likes of each of these actors.

The data frame has three columns to help you out for the same, viz. actor\_1\_facebook\_likes, actor\_2\_facebook\_likes, and actor\_3\_facebook\_likes. Your objective is to find the trios which have 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. Find out the top 5 popular trios, and output their names in a list.

### Subtask 2.5: Find the Most Popular Trios - II

In the previous subtask you 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 you aren't getting any unpopular actor in your trio (since the total likes calculated in the previous question doesn't tell anything about the individual popularities of each actor in the trio.).

You can do a manual inspection of the top 5 popular trios you have found in the previous subtask and check how many of those trios satisfy this condition. Also, which is the most popular trio after applying the condition above?

### Subtask 2.6: Runtime Analysis

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

### Subtask 2.7: 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, find the top 10 movies that have the highest number of votes i.e.CVotesU18 from the movies data frame. Store these in a data frame named Popular.

## Task 3: Demographic analysis

If you take a look at the last columns in the data frame, most of these are related to demographics of the voters (in the last subtask, i.e., 2.8, you made use one of these columns - CVotesU18). We also have three genre columns indicating the genres of a particular movie. We will extensively use these columns for the third and the final stage of our assignment wherein we will analyze the voters across all demographics and also see how these vary across various genres. So without further ado, let's get started with demographic analysis.

### Subtask 3.1 Combine the Data frame by Genres

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

1. First create a new data frame 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, add a column called cnt to the data frame df\_by\_genre and initialize it to one. You will realize the use of this column by the end of this subtask.
3. First group the data frame 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 data frame df\_by\_g1.
4. Perform the same operation for genre\_2 and genre\_3 and store it data frames df\_by\_g2 and df\_by\_g3 respectively.
5. Now that you have 3 data frames performed by grouping over genre\_1, genre\_2, and genre\_3 separately, it's time to combine them. For this, add the three data frames and store it in a new data frame 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. You can refer to this link to see how this function works. <https://pandas.pydata.org/pandas-docs/version/0.23.4/generated/pandas.DataFrame.add.html>
6. The column cnt on aggregation has basically kept the track of the number of occurrences of each genre. Subset the genres that have atleast 10 movies into a new data frame genre\_top10 based on the cnt column value.
7. Now, take the mean of all the numeric columns by dividing them with the column value cnt and store it back to the same data frame. We will be using this data frame for further analysis in this task unless it is explicitly mentioned to use the data frame movies.
8. Since the number of votes can't be a fraction, type cast all the CVotes related columns to integers. Also, round off all the Votes related columns up to two digits after the decimal point.

### Subtask 3.2: Genre Counts!

Now let's derive some insights from this data frame. Make a bar chart plotting different genres vs cnt using seaborn.

### Subtask 3.3: Gender and Genre

If you have closely looked at the Votes- and CVotes-related columns, you might have noticed the suffixes F and M indicating Female and Male. Since we have the vote counts for both males and females, across various age groups, let's now see how the popularity of genres varies between the two genders in the data frame.

1. Make the first heat map to see how the average number of votes of males is varying across the genres. Use seaborn heat map for this analysis. The X-axis should contain the four age-groups for males, i.e., CVotesU18M, CVotes1829M, CVotes3044M, and CVotes45AM. The Y-axis will have the genres and the annotation in the heat map tell the average number of votes for that age-male group.
2. Make the second heat map to see how the average number of votes of females is varying across the genres. Use seaborn heatmap for this analysis. The X-axis should contain the four age-groups for females, i.e., CVotesU18F, CVotes1829F, CVotes3044F, and CVotes45AF. The Y-axis will have the genres and the annotation in the heatmap tell the average number of votes for that age-female group.
3. Make sure that you plot these heatmaps side by side using subplots so that you can easily compare the two genders and derive insights.
4. Write your any three inferences from this plot. You can make use of the previous bar plot also here for better insights. Refer to this link- <https://seaborn.pydata.org/generated/seaborn.heatmap.html>. You might have to plot something similar to the fifth chart in this page (You have to plot two such heatmaps side by side).
5. Repeat subtasks 1 to 4, but now instead of taking the CVotes-related columns, you need to do the same process for the Votes-related columns. These heatmaps will show you how the two genders have rated movies across various genres.

You might need the below link for formatting your heatmap. <https://stackoverflow.com/questions/56942670/matplotlib-seaborn-first-and-last-row-cut-in-half-of-heatmap-plot>

* Note : Use genre\_top10 data frame for this subtask

### Subtask 3.4: US vs non-US Cross Analysis

### The dataset contains both the US and non-US movies. Let's analyse how both the US and the non-US voters have responded to the US and the non-US movies.

### Create 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.

### Now make 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. Make use of the column IFUS to make this plot. Similarly, make 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. Write any of your two inferences/observations from these plots.

### Again do a similar analysis but with the ratings. Make a boxplot that shows how the ratings from the US people i.e. VotesUS is varying for the US and non-US movies. Similarly, make another subplot that shows how VotesnUS is varying for the US and non-US movies. Write any of your two inferences/observations from these plots.

### Note : Use movies dataframe for this subtask. Make use of this documention to format your boxplot - <https://seaborn.pydata.org/generated/seaborn.boxplot.html>

### Subtask 3.5: Top 1000 Voters Vs Genres

### You might have also observed the column CVotes1000. This column represents the top 1000 voters on IMDb and gives the count for the number of these voters who have voted for a particular movie. Let's see how these top 1000 voters have voted across the genres.

### Sort the dataframe genre\_top10 based on the value of CVotes1000in a descending order.

### Make a seaborn barplot for genre vs CVotes1000.

### Write your inferences. You can also try to relate it with the heatmaps you did in the previous subtasks.