Problem statement:

In this assignment, you will try to find some interesting insights into a few movies released between 1916 and 2016, using Python. This is an individual assignment wherein you will download a movie dataset, write Python code to explore the data, gain insights into the movies, actors, directors, and collections, and submit the code.

* Change the unit of columns (from $ to million $)
* Find top 10 movies of highest gross profit
* Drop duplicate values
* Find Top 250 IMDB rated movies along with rank
* Find the top 10 best directors
* Find top 10 popular genres
* Find top 10 the critic-favorite and audience-favorite actors

Task 1: Reading and Inspection

Subtask 1.1: Import and read

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

Subtask 1.2: Inspect the dataframe

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

Task 2: Cleaning the Data[**¶**](http://localhost:8888/notebooks/Downloads/DS%2Bin%2BPython%2BMovie%2BAssignment.ipynb#Task-2:-Cleaning-the-Data)

Subtask 2.1: Inspect Null values[¶](http://localhost:8888/notebooks/Downloads/DS%2Bin%2BPython%2BMovie%2BAssignment.ipynb#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.

Subtask 2.2: Drop unecessary columns[¶](http://localhost:8888/notebooks/Downloads/DS%2Bin%2BPython%2BMovie%2BAssignment.ipynb#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.

Subtask 2.3: Drop unecessary rows using columns with high Null percentages[¶](http://localhost:8888/notebooks/Downloads/DS%2Bin%2BPython%2BMovie%2BAssignment.ipynb#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.

Subtask 2.4: Drop unecessary rows[¶](http://localhost:8888/notebooks/Downloads/DS%2Bin%2BPython%2BMovie%2BAssignment.ipynb#Subtask-2.4:-Drop-unecessary-rows)

Some of the rows might have greater than five NaN values. Such rows aren't of much use for the analysis and hence, should be removed.

Subtask 2.5: Fill NaN values[¶](http://localhost:8888/notebooks/Downloads/DS%2Bin%2BPython%2BMovie%2BAssignment.ipynb#Subtask-2.5:-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'

Subtask 2.6: Check the number of retained rows[¶](http://localhost:8888/notebooks/Downloads/DS%2Bin%2BPython%2BMovie%2BAssignment.ipynb#Subtask-2.6:-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.

Task 3: Data Analysis[¶](http://localhost:8888/notebooks/Downloads/DS%2Bin%2BPython%2BMovie%2BAssignment.ipynb#Task-3:-Data-Analysis)

Subtask 3.1: Change the unit of columns[¶](http://localhost:8888/notebooks/Downloads/DS%2Bin%2BPython%2BMovie%2BAssignment.ipynb#Subtask-3.1:-Change-the-unit-of-columns)

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

Subtask 3.2: Find the movies with highest profit[¶](http://localhost:8888/notebooks/Downloads/DS%2Bin%2BPython%2BMovie%2BAssignment.ipynb#Subtask-3.2:-Find-the-movies-with-highest-profit)

Create a new column called profit which contains the difference of the two columns: gross and budget.

Sort the dataframe using the profit column as reference.

Extract the top ten profiting movies in descending order and store them in a new dataframe - top10

Subtask 3.3: Drop duplicate values[¶](http://localhost:8888/notebooks/Downloads/DS%2Bin%2BPython%2BMovie%2BAssignment.ipynb#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.

Subtask 3.4: Find IMDb Top 250[¶](http://localhost:8888/notebooks/Downloads/DS%2Bin%2BPython%2BMovie%2BAssignment.ipynb#Subtask-3.4:-Find-IMDb-Top-250)

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.

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.

Subtask 3.5: Find the best directors[¶](http://localhost:8888/notebooks/Downloads/DS%2Bin%2BPython%2BMovie%2BAssignment.ipynb#Subtask-3.5:-Find-the-best-directors)

Group the dataframe using the director\_name column.

Find out the top 10 directors for whom the mean of imdb\_score is the highest and store them in a new dataframe top10 director.

Subtask 3.6: Find popular genres[¶](http://localhost:8888/notebooks/Downloads/DS%2Bin%2BPython%2BMovie%2BAssignment.ipynb#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.

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.

Group the dataframe using genre\_1 as the primary column and genre\_2 as the secondary column.

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.

Subtask 3.7: Find the critic-favorite and audience-favorite actors[¶](http://localhost:8888/notebooks/Downloads/DS%2Bin%2BPython%2BMovie%2BAssignment.ipynb#Subtask-3.7:-Find-the-critic-favorite-and-audience-favorite-actors)

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.

Append the rows of all these dataframes and store them in a new dataframe named Combined.

Group the combined dataframe using the actor\_1\_name column.

Find the mean of the num\_critic\_for\_reviews and num\_user\_for\_review and identify the actors which have the highest mean.