Final Submission:

Shengnan Ke 05/11/2022

i. Motivation surrounding project topic

Speaking for myself, I am an extreme movie and TV show lover. The TV at my house always shows the most trendy dramas, comedies, thriller movies, etc. So lots of my friends like to come to my place and watch movies with me. And I enjoy the feeling to have company and being a host. But sometimes I found it hard to decide on what to watch as a “Director of Cinema”. I am thinking about getting some datasets from some rating platforms. And find out what patterns can make a movie or TV show more popular than others? Is there one? If so, based on these patterns, I can host more attractive movie nights for my friends!

The major task for this project is:

- Which ratings(certificates) of movies are more popular?
- Which genres of movies are more popular?
- Does the director influence the popularity of movies?
- Is there an inner relationship between the Metascore, IMDb rating, and the Gross Box office of the movies?
- Based on the gathered information, create my own 2022 movie playlist.

ii. Brief description of data sources

Two major data resources:

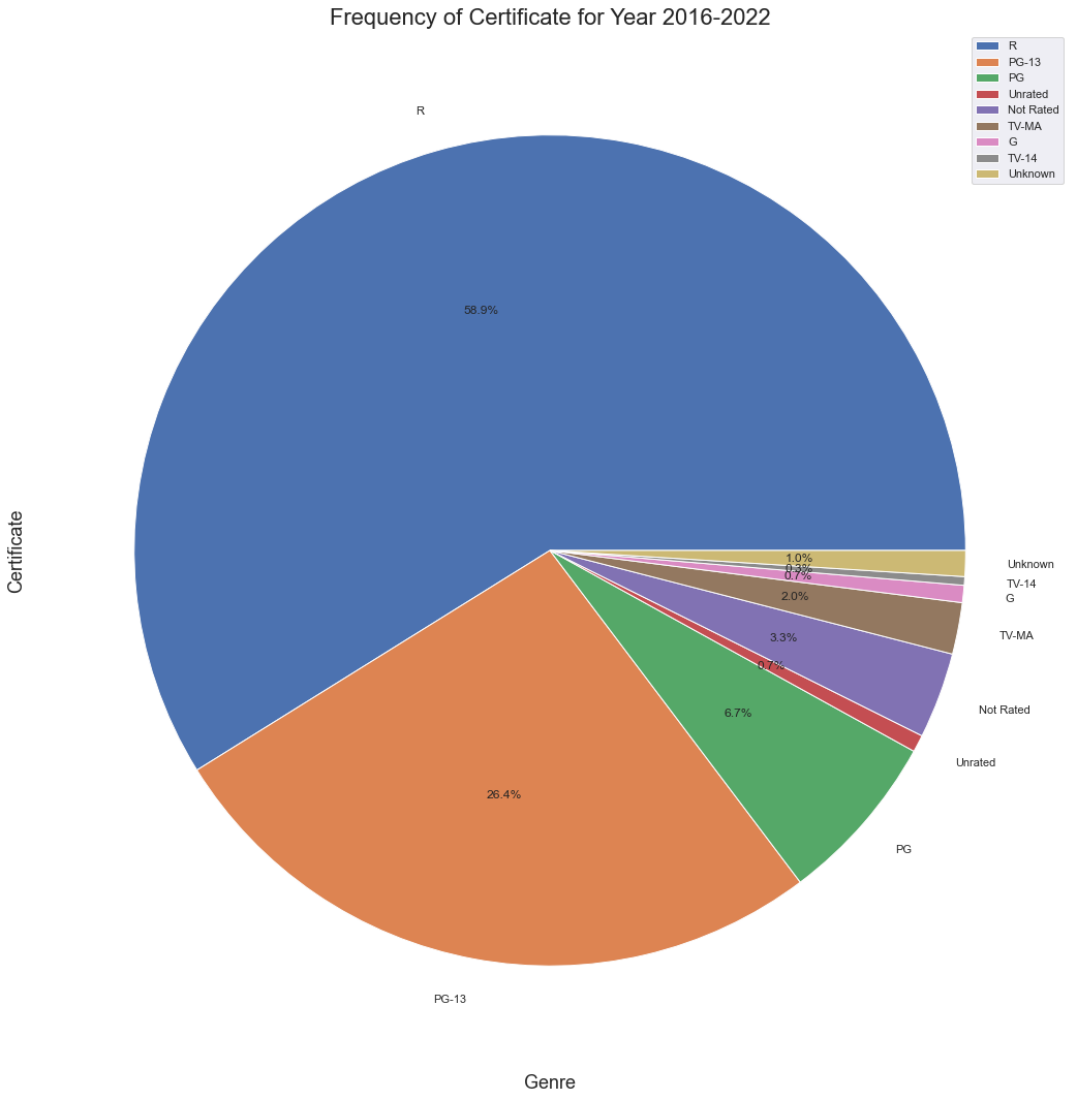
IMDb & The Movie Database (TMDB)

1. Here are the links which I scrape the information for the Top 50 movies from 2016 to 2021:
    a. https://www.imdb.com/list/ls506716903/ 2021
    b. https://www.imdb.com/list/ls081993179/?sort=list_order,asc&st_dt=&mode=detail&page=1 2020
    c. https://www.imdb.com/list/ls043151343/ 2019
    d. https://www.imdb.com/list/ls023345789/ 2018
    e. https://www.imdb.com/list/ls062905646/ 2017
    f. https://www.imdb.com/list/ls060610711/ 2016

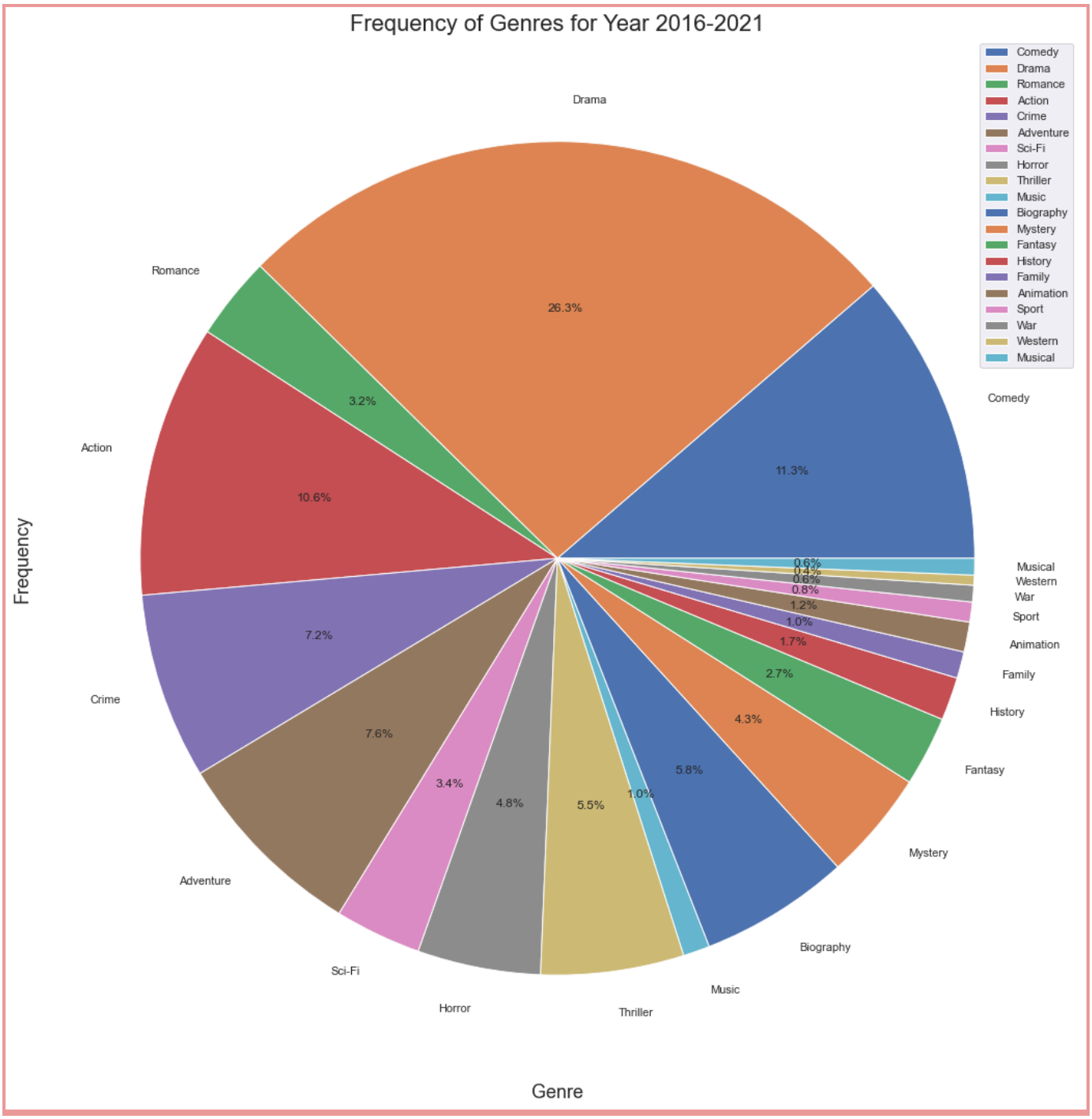
2. Here is the link used to get a list of movie titles that came out in 2022: https://www.imdb.com/list/ls090466457/
3. Since we are requested to use API for gathering data. I used The Movie Database (TMDB) API for gathering information about movies that come out in 2022. Here is the link: https://www.themoviedb.org/settings/api

iii. Analysis performed

1. This graph is to show the percentage of movies with different ratings. It is very clear to see that more than half of the Top movies from 2016 to 2021 are rated R - 58.9%. Around 26.4% of movies are PG - 13, and 6.7% of movies are PG. Perhaps because, in general cases, movies rated R are less restricted by the “system”. This allows the thematic core or content of these movies to spread out in all directions so that the content can be more interesting/attractive to teenagers (R-rated movies can be viewed by people over 18 years old with a 21+ escort) or adults. And since teens and adults are the main force in rating movies on those online platforms(IMDb, Rotten Tomatoes, etc), these lists of movies are summary. So, I'm not surprised by the results.



2. Among the top movies from 2016 to 2022. This graph is to show the percentage of movies with different genres. Apparently, Drama is the most popular genre which takes 26.3%. Comedy and Action movies are ranked tight, they are 11.3% and 10.6%. Crime and Advature follow closely behind, 7.6% and 7.2%. These are the top 5 genres: Drama, Comedy, Action, Crime, and Advature.



3. IMDb Rating v.s. Metascore

This is to find out if there is an inner relationship between the IMDb Rating and the Metascore of the movie. According to the distribution of the graph, I would say there is some sort of connection between the IMDb Rating and the Metascore. There are two different individual rating systems. Since it's rating on the same content, they must take into account similar parameters. So it makes sense.

I also did a t-test. However, according to the p-value. They are not related to each other. But we also need to take into account there are many 0 values after I modified the data for creating the scattered plot.

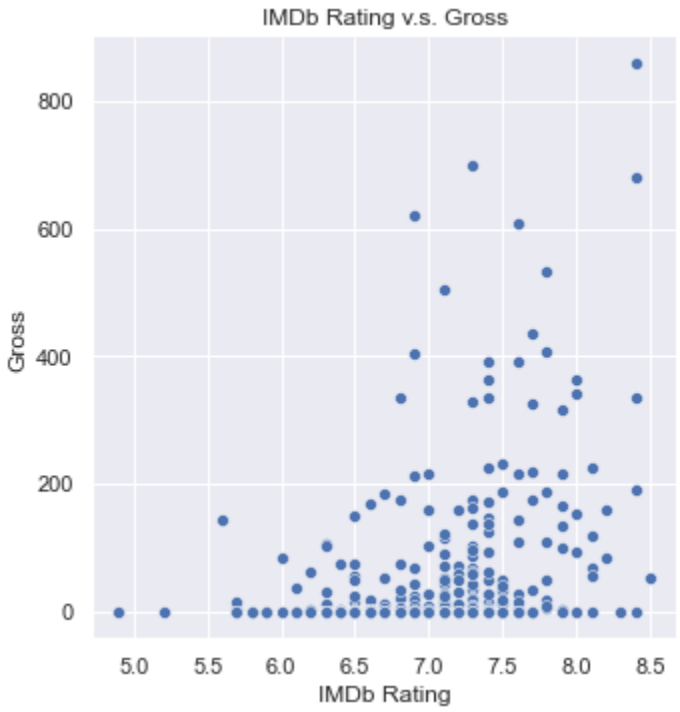
	Variable	N	Mean	SD	SE	95% Conf.	Interval
0	Metascore	299.0	68.548495	17.323225	1.001828	66.576941	70.520049
1	Gross	299.0	65.062441	127.588349	7.378625	50.541629	79.583254
2	combined	598.0	66.805468	90.986804	3.720727	59.498163	74.112774

Independent t-test results

0	Difference (Metascore - Gross) =	3.4861
1	Degrees of freedom =	596.0000
2	t =	0.4682
3	Two side test p value =	0.6398
4	Difference < 0 p value =	0.6801
5	Difference > 0 p value =	0.3199
6	Cohen's d =	0.0383
7	Hedge's g =	0.0382
8	Glass's delta =	0.2012
9	Pearson's r =	0.0192

4. IMDb Rating v.s. Gross

This is to find out if there is an inner relationship between the IMDb Rating and the Gross of the movie. If we ignore these 0 values and take a close look at some cases, we can tell a little pattern that there is a relation between the rating and the gross Box office. For example, the point on the right top corner. It created a high(around 8.5), and the gross Box office is above 800 Million dollars. And by looking at the p-value, it's 0 which is less than 0.05, then that result is said to be statistically significant.



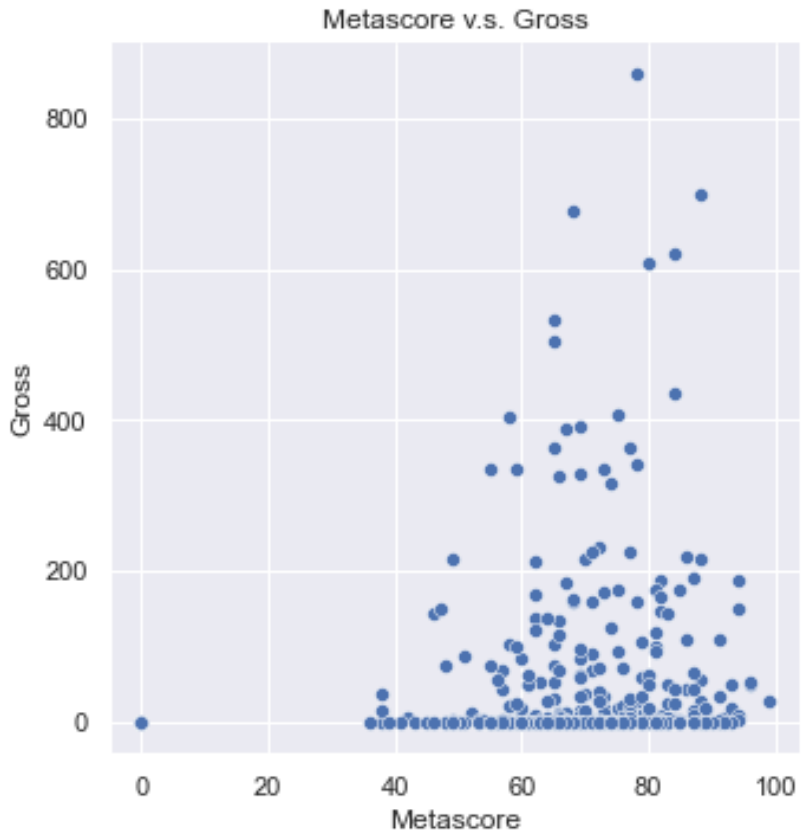
	Variable	N	Mean	SD	SE	95% Conf.	Interval
0	IMDb Rating	299.0	7.024415	0.632009	0.036550	6.952486	7.096344
1	Gross	299.0	65.062441	127.588349	7.378625	50.541629	79.583254
2	combined	598.0	36.043428	94.707300	3.872870	28.437323	43.649533

Independent t-test results

0	Difference (IMDb Rating - Gross) =	-58.0380
1	Degrees of freedom =	596.0000
2	t =	-7.8656
3	Two side test p value =	0.0000
4	Difference < 0 p value =	0.0000
5	Difference > 0 p value =	1.0000
6	Cohen's d =	-0.6433
7	Hedge's g =	-0.6425
8	Glass's delta =	-91.8310
9	Pearson's r =	0.3067

5. Metascore v.s. Gross

Similar to the graph from above. This is to find out if there is an inner relationship between the Metascore and the Gross of the movie. Maybe because a lot of the movies don't have data on the Metascore, both the graph and the t-test results show there is no relationship between them.



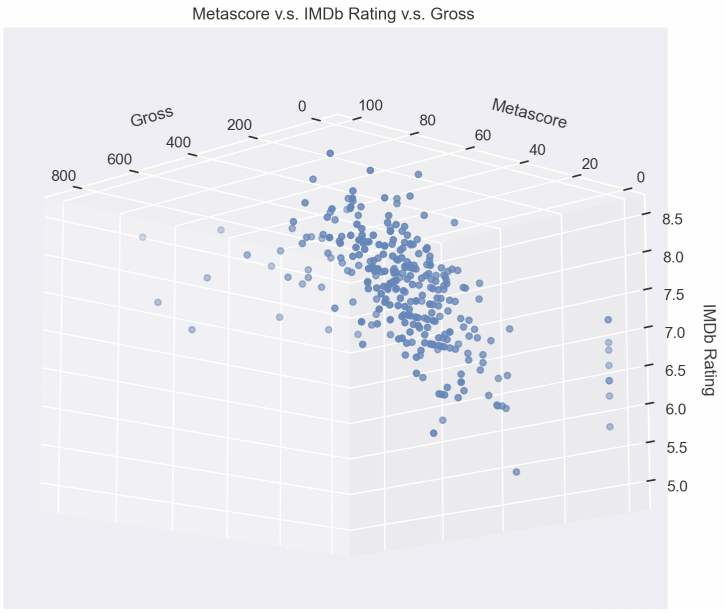
	Variable	N	Mean	SD	SE	95% Conf.	Interval
0	IMDb Rating	299.0	70.244147	6.320091	0.365500	69.524858	70.963436
1	Metascore	299.0	68.548495	17.323225	1.001828	66.576941	70.520049
2	combined	598.0	69.396321	13.055805	0.533892	68.347787	70.444855

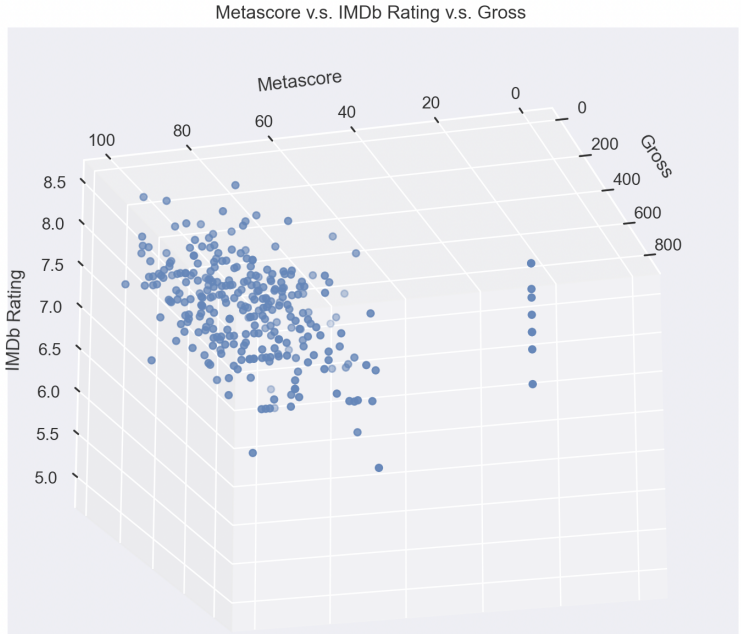
Independent t-test results

0	Difference (IMDb Rating - Metascore) =	1.6957
1	Degrees of freedom =	596.0000
2	t =	1.5900
3	Two side test p value =	0.1124
4	Difference < 0 p value =	0.9438
5	Difference > 0 p value =	0.0562
6	Cohen's d =	0.1300
7	Hedge's g =	0.1299
8	Glass's delta =	0.2683
9	Pearson's r =	0.0650

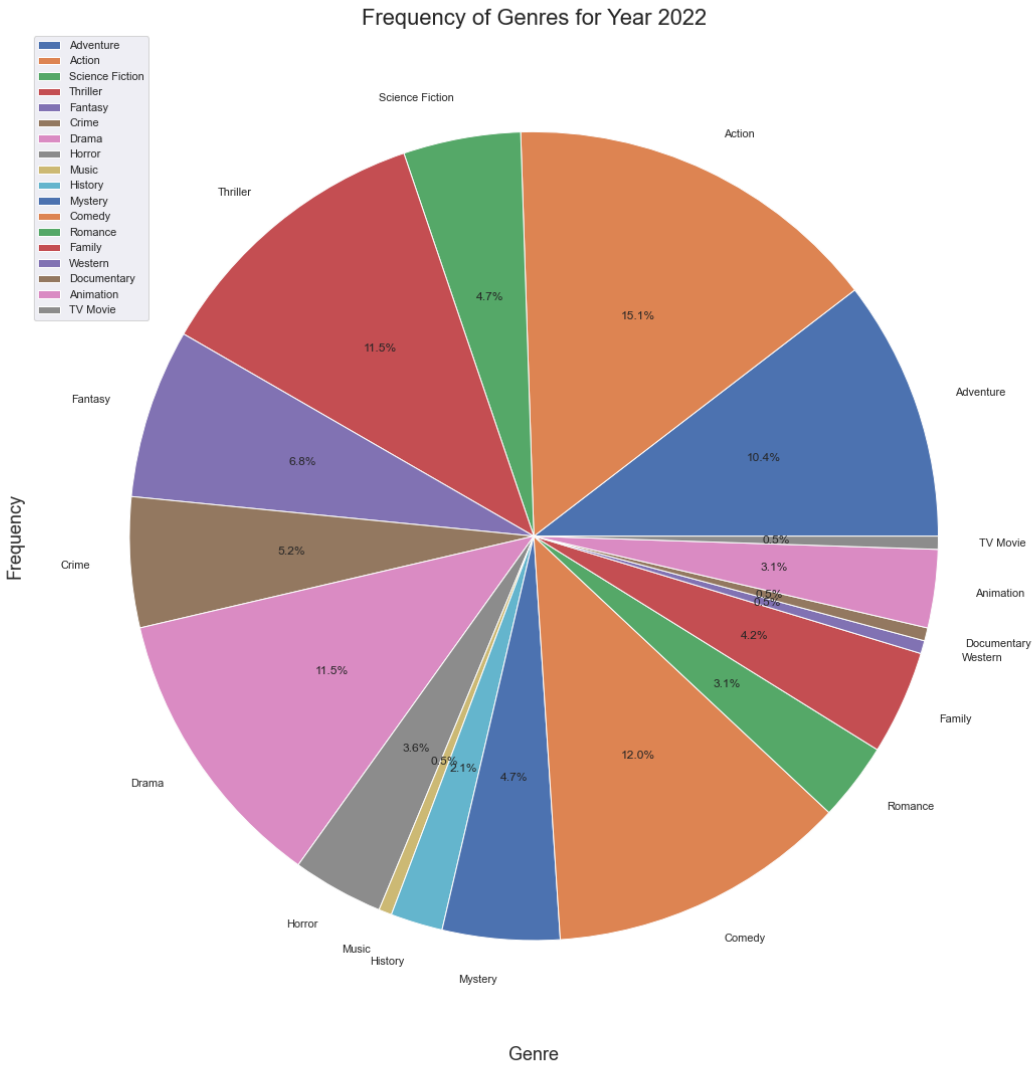
6. Metascore v.s. IMDb Rating v.s. Gross

This graph supposes to be a 3D active graph which is easier for us to see the relationship between these three datasets. However, due to the report's format. I cannot show this pattern to you. You can move it around when you run the code. However even with this single angle, we can tell there is a very dense area, all the dots are clustered together.





7. The purpose of this pie chart is to get a simple idea of the distribution of the types of movies in 2022. From the chart, it seems that the various film genres are distributed somewhat evenly. But action, comedy, thrillers, drama, and adventure are significantly more than other ones.



8. Among the top movies from 2016 to 2021, we find out if there are directors has more than one movie on the list. Surprisingly, there are many directors who have more than one movie on the list.

   Here is the list of directors who has more than one movie on the list:

   'Yorgos Lanthimos', 'Anthony Russo', 'Denis Villeneuve', 'Barry Jenkins', 'Todd Phillips', 'Jeff Nichols', 'Robert Eggers', 'James Wan', 'Peter Berg', 'Martin

Scorsese', 'Stephen Frears', 'Luca Guadagnino', 'Taika Waititi', 'Clint Eastwood', 'Jordan Peele', 'Chad Stahelski', 'Greta Gerwig', 'Guillermo del Toro', 'Benny Safdie', 'M. Night Shyamalan', 'James Mangold', 'S. Craig Zahler', 'Rian Johnson', 'Edgar Wright', 'Doug Liman', 'Christopher Nolan', 'Jon Watts', 'Michael Showalter', 'Stephen Chbosky', 'Noah Baumbach', 'David Michôd', 'James Gunn', 'Chris McKay', 'David Leitch', 'Ridley Scott', 'George Clooney', 'Ari Aster', 'John Krasinski', 'Adam McKay', 'Paul Schrader', 'Leigh Whannell', 'David Lowery', 'Marielle Heller', 'Ron Howard', 'Aaron Sorkin', 'J Blakeson'

Here is another list, that the director has a movie in the top list in the past 6 years and also has a movie in 2022:

'Martin Scorsese', 'Ric Roman Waugh', 'Guillermo del Toro', 'Edgar Wright', 'Robert Eggers', 'Simon Curtis', 'Matt Reeves', 'Scott Derrickson', 'David Gordon Green', 'Ryan Coogler', 'Jeff Fowler', 'Yorgos Lanthimos', 'Ari Aster', 'Shawn Levy', 'Olivia Wilde', 'David Leitch', 'Rian Johnson', 'Taika Waititi', 'Sam Hargrave'

Here is the list of directors, who not only have more than one movie in the past 6 years but also have a movie in 2022:

'David Leitch', 'Martin Scorsese', 'Robert Eggers', 'Guillermo del Toro', 'Yorgos Lanthimos', 'Rian Johnson', 'Taika Waititi', 'Ari Aster', 'Edgar Wright'

iv. Conclusions drawn

- Drama, Comedy, Action, Crime, and Adventure are the most popular genres of movies in the past 6 years. I believe this has somewhat influenced the variety of movies this year. Because seeing the variety of movies this year, the most is these six genres: action, comedy, thrillers, drama, and adventure which have a lot of overlap with the top 5 from the past 6 years.
- Rated R might be the one with the highest expectation from everyone.
- Movies from the following directors might be a new trend across the world. They have more than one movie in the past 6 years and are on the Top 50 of the year list. And they all have a movie in 2022.

  'David Leitch', 'Martin Scorsese', 'Robert Eggers', 'Guillermo del Toro', 'Yorgos Lanthimos', 'Rian Johnson', 'Taika Waititi', 'Ari Aster', 'Edgar Wright'

- Therefore, the movies directed by the directors above are going to be on my playlist which is listed below.

'Thor: Love and Thunder', 'Killers of the Flower Moon', 'Knives Out 2', 'Disappointment Blvd.', 'Baby Driver 2', 'Bullet Train', 'The Northman', 'Poor Things', 'Cat Person'

README.txt

DOWNLOAD BEFORE EXECUTING:

Please download the “chromedriver_mac64.zip” from
https://chromedriver.chromium.org/downloads this website.
Please download it according to your own operating system.
And check which version of the chrome browser you are using.
After you successfully downloaded the package from the above website.
Next, open that zip file and place it in the folder where the python script is.
Then, install the library accordingly in your terminal or command line.

```
# pip install bs4
# pip install html5lib
# pip install csv
# pip install pandas
# pip install requests
# pip install selenium
# pip install json
# pip install seaborn
# pip install tmdbsimple
# pip install researchpy
```

Next just import libraries accordingly. You should be able to successfully run it on your computer.

```
# import requests
# from bs4 import BeautifulSoup
# import csv
# import pandas as pd
# import selenium as se
# from selenium import webdriver
# from selenium.webdriver.common.by import By
# import tmdbsimple as tmdb
# import json
# import matplotlib.pyplot as plt
# from collections import Counter
# import seaborn as sns
# import researchpy as rp
# import sys
```

GET API KEY:

Please make sure that you get your own API_key. Else, it might cause errors. That you might not able to load the data from The movie database.

5 Major Parts of the final_project.py File:

Part_1: WEB Data Scraping

This part of the python script is for gathering data about “Top-rated” movies from the past years(2016-2021).

def get_xxxx_info_from_web():

This function is being used 6 times to scrape information about the Top 50 movies from 2016 to 2021. For each year, I save the data into one single CSV file. And I named it 'Top_movies_xxxx.csv' (xxxx stands for the year from 2016 to 2021)

def combined_movie_info():

This function is used to combine all the CSV files we got from the above function. And it saves all the data in a single excel file as ‘2016_2021_combined_movie_info.xlsx’

Part_2: WEB Data Scraping

Next, I would like to gather some data about movies that just came out in 2022. This part of the python file is to help me get prepared to use TMDB(The Movie Database) API. Because in TMDB, if we want to request detailed information about certain movies, we must provide the TMDB **ID** of the movies.

def get_2022_Title_from_web():

This function is being used to get a list of movies that came out in 2022 but only included the movie title. The resource list is from https://www.imdb.com/list/ls090466457/
I save this list of movie titles as 'Draft1_2022_movie_titles.csv'.

def get_2022_movie_id_from_web():

After we get the '2022movielist.csv'. Accordingly, we need to find out the TMDB ID of this list of movies. In this function, I used a library function called **selenium.** This library function helped me to automatically search movies on that list and get the TMDB ID of each movie. And this data is saved in 'Draft2_2022_Movie_Title_ID.csv'.

def combine_2022_movie_info():

This function is to remove certain movies from the list. Some of the movies do not exist in TMDB. So by merging these two CSV files, we can get a list of movies that do exist in TMDB. And this updated list is saved as 'Modified_movie_TI.csv'.

def filtered_movie_list():
By checking on the 'Modified_movie_TI.csv'. I realized that some of the old movies have the same title as the movie in 2022 and it also exists in the TMDB. But what we looking for is just the list of 2022 movies. I realized that the ID of each movie relates to the time that the movie came out. So I filtered out the old movies and save them as a new dataset, named 'Final_ver_2022_Movie_Title_ID.csv'.

Part_3: API_scraping
After we got the 2022 movie list with the titles and IDs. we can use their TMDB ID to find detailed information about these 2022 movies.
Firstly, I need to require an API_KEY from this website: https://www.themoviedb.org/settings/api
You would have to register an account to get your own API Key (v3 auth).
Mine is 5c7a1c5e6cdd8d3132826246ee7c7761
Next, I use my api_key and the movie's TMDB ID to request information about the movies.
Here is an example API request:
https://api.themoviedb.org/3/movie/ "movie ID"?api_key=" the API key"
By reading this 'Final_ver_2022_Movie_Title_ID.csv' file, we save the movie IDs as a list.

def get_data(API_key, Movie_ID):
This function works with the API, we can get these ['Movie_name', 'popularity', 'vote', 'vote_count', 'budget', 'Collection_name"] information.

def get_2022_info()
This function for iterating through the list of movie IDs. By calling the get_data function. We can get these ['Movie_name', 'popularity', 'vote', 'vote_count', 'budget', 'Collection_name"] for each movie on the list. And we save these data as '2022_movie_data.csv'.

def combine_2022_movie_info()
This function is for merging the Certificate & ID columns from 'Final_ver_2022_Movie_Title_ID.csv' to the '2022_movie_data.csv'. Now we have the combination. We save this combined information as 'Motified_2022_movie_data.csv', Which contains ['Index', 'Title', 'popularity', 'vote_average', 'vote_count', 'budget', 'Collection_name', 'ID', 'Certificate']. And by adding one index column, we get the final outcome of this part of the python script: '2022_basic_movie_data.csv'.

Part_4: API_scraping

- API scraping with the library function

pip install tmdbsimple

import tmdbsimple as tmdb

Since I'm using a support library function tmdbsimple to scrape more detailed information of movies. Please make sure to install it first, then, run this python script.

Similar to the last python script - API_scrape.py. You would have to register an account to get your own API Key (v3 auth).

Mine is 5c7a1c5e6cdd8d3132826246ee7c7761

Next, I use my api_key and the movie's TMDB ID to request information about the movies.

Here is an example API request:

https://api.themoviedb.org/3/movie/ "movie ID"?api_key=" the API key"

By reading this 'Final_ver_2022_Movie_Title_ID.csv' file, we save the movie IDs as a list.

def get_movie_genre_info():

This function is to identify which genre each movie belongs to. And I save this information as "2022_movie_genre.csv". And I added an index column to the previous file and save it as 'motified_2022_movie_genre.csv'.

def get_movie_director_info():

This function is to identify who is/are the director(s) of each movie belongs. And I save this information as "2022_movie_directors.csv". And I added an index column to the previous file and save it as 'motified_2022_movie_directors.csv'.

def combine_genre_director():

This function is for combining 'motified_2022_movie_genre.csv' and 'motified_2022_movie_directors.csv' based on the index column. The final outcome for this function is '2022_genre_director.csv'.

At the end of this function, we combine the info we get from the last function '2022_genre_director.csv'. and the final outcome from API_scrape.py '2022_basic_movie_data.csv'.

The final outcome for this python script is '2022_movie_all_info.csv'. This csv file contains these information: [Index, Title, popularity, vote_average, vote_count, budget, Collection_name, ID, Certificate, Genre(s), Director(s)].

Part_5: Start Analyze!

def count_certificate():
Using info from '2016_2021_combined_movie_info.xlsx', among the top movies from 2016 to 2022. This function is to find out the percentage of movies with different ratings. The output graph is saved as 'past_movies_count_Certificates.png'.

def count_genre():
Using info from '2016_2021_combined_movie_info.xlsx', among the top movies from 2016 to 2022. This function is to find out the percentage of movies with different genres. The output graph is saved as 'past_movies_count_genres.png'

def IMDbr_Metascore():
Using info from '2016_2021_combined_movie_info.xlsx', among the top movies from 2016 to 2022. This function is to find out if there is an inner relationship between the IMDb Rating and the Metascore of the movie. The output graph is saved as 'IMDb_Rating_Metascore.png'

def IMDbr_Gross():
Using info from '2016_2021_combined_movie_info.xlsx', among the top movies from 2016 to 2022. This function is to find out if there is an inner relationship between the IMDb Rating and the Gross of the movie. The output graph is saved as 'IMDb_Rating_Gross.png'

def Metascore_Gross():
Using info from '2016_2021_combined_movie_info.xlsx', among the top movies from 2016 to 2022. This function is to find out if there is an inner relationship between the Metascore and the Gross of the movie. The output graph is saved as 'Metascore_Gross.png'

def Gross_Metascore_IMDbr():
Using info from '2016_2021_combined_movie_info.xlsx', among the top movies from 2016 to 2022. This function is to find out if there is an inner relationship between the Metascore, IMDb_Rating, and the Gross of the movie. The output graph is saved as 'Metascore_IMDb_Rating_Gross.png'

def directors_analysis():
Using info from '2016_2021_combined_movie_info.xlsx', and '2022_movie_all_info.csv'. Among the top movies from 2016 to 2021, find out if there are directors has more than one movie on the list. Compare to 2022, find directors who have more than one movie in

the past 6 years also have a movie in 2022. Finally, find out the movie title of these directors.

def analyze_2022_directors():

Using info from '2016_2021_combined_movie_info.xlsx', and '2022_movie_all_info.csv'. Find out movie directors who have at least one movie in the past 6 years and have a movie in 2022.

def analyze_2022_genres():

This function is to get a simple idea of the distribution of the types of movies in 2022. The graph is saved as "2022_count_genres.png"

Running on the command line:

Cd to the folder where you saved this "**Final_Project.py" file.**

I'm using a Mac, and I saved it in the Final_Project_SK. So I was executing this line in the terminal. And this worked perfectly on my laptop.

python -u Final_Project.py

Example output:

```
(base) kknanxx@KKNANXXs-MacBook-Pro Final_Project_SK % python -u Final_Project.py
/Users/kknanxx/Desktop/510_PY/Final_Project_SK/Final_Project.py:568: DeprecationWarning: exe
cutable_path has been deprecated, please pass in a Service object
  browser = webdriver.Chrome(executable_path='/Users/kknanxx/Desktop/510_PY/Final_Project_SK
/chromedriver')
It MIGHT TAKE SOME TIME TO GETTHER ALL THE DATA
Data Done! Analyze Start !
    Variable      N       Mean          SD        SE  95% Conf.   Interval
0  Metascore  299.0  68.548495   17.323225  1.001828  66.576941  70.520049
1      Gross  299.0  65.062441  127.588349  7.378625  50.541629  79.583254
2   combined  598.0  66.805468   90.986804  3.720727  59.498163  74.112774
 Independent t-test   results
0  Difference (Metascore - Gross) =     3.4861
1              Degrees of freedom =   596.0000
2                               t =     0.4682
3           Two side test p value =     0.6398
4          Difference < 0 p value =     0.6801
5          Difference > 0 p value =     0.3199
6                      Cohen's d =     0.0383
7                      Hedge's g =     0.0382
8                  Glass's delta =     0.2012
9                    Pearson's r =     0.0192
      Variable      N       Mean          SD        SE  95% Conf.   Interval
0  IMDb Rating  299.0   7.024415    0.632009  0.036550   6.952486   7.096344
1        Gross  299.0  65.062441  127.588349  7.378625  50.541629  79.583254
```