Project Report

Student Name: Ian Rowe

Email: ianrowe101@gmail.com

Student Number: E10610086

GitHub URL

https://github.com/ianrowe101/UCDPA_Ian_Rowe.git

Abstract

My aim for the project is to learn the fundamentals of Python and how it can be applicable in professional, educational, and personal environments. Expanding on these three areas and my goals for these:

- Professional:
    - I want to expand my skill set beyond Excel and hopefully use my new skills in Python to increase my value to my employer.
- Educational:
    - I completed Bachelor's in financial services in May 2021 during which I relied on Excel to pull data sources together to create data frames and visualisations however, Excel is limited compared to Python and as I start on the pathway to a Master's in financial services, I feel that learning some form of coding could be beneficial.
- Personal:
    - I have a keen interest in data analytics, especially in areas of sport, personal health/fitness and finance. I wanted to be able to develop tools that allow me to analyse these areas in more detail using code that may be adaptable across the different areas.

I created this project using tools I learned in the lectures and DataCamp. When choosing the topic, I wanted to choose a subject that was interesting to me. It is also something that I had tried to create in the past but was unable to either due to Excel limitations or inability to source data.

Introduction

The aim of the project was to test the impact of film production costs/budgets on how a film performs in financial terms, audience reception and critical reception. I choose the film industry as I am obsessed with films and wanted to delve deeper into the data to understand it more.

Dataset

In creating the dataset, I was important to have a large dataset. Unfortunately, Kaggle was unable provide just one up to date source. I needed a dataset with the following:

- Production Costs/Budgets
- Box Office Gross Amounts
- Release Dates/Years
- IMDB Scores
- Rotten Tomato Percentage

To capture all of the above, I downloaded 11 datasets from Kaggle carried out some reviews in Excel to make sure they were suitable.

I choose Kaggle to source these as the site had the best selection of film industry files and I am not aware of any live source or API that could provide me with live access to this range of information.

Implementation Process

I imported the python environments required to complete the project functions. I then outlined an overview of my project in text, adding some structure to the file.

The second step was to load and then cleanse the Kaggle .csv files. I created a loop that loaded each file into the kernel as data frames, considering there were 11 files, this saved me from repeating the same steps to reach the same position.

Once loaded, I created a dictionary containing the names of each file to create a loop that would allow me to look at the info and first rows of each data frame to give me an idea of what needed to be cleansed. I would use these loops and dictionary frequently to investigate the changes made by certain steps in the process. I added an analysis of the current condition of the data frames and the work I wanted to complete with them.

I noticed some columns in some of the data frames were strings/objects when I expected there to be numeric. This was due to the inclusion of special characters, such as “$” or “,” , so I replaced these characters with blanks and converted the columns to numeric formats. I then replaced the column headings in each file to ensure that file was uniformed.

Some of the release dates were also string/object, I converted these to a date format before extracting the release year to ensure each data frame contained a release year column. I then set the film_title columns in each file to upper case, removed any potential spaces in front/end of the title and removed any potential special cases that may cause missed joins later in the report. I then removed any duplicates which were present in each

individual data frame prior to stacking them with concat. I used (film_title + film_year) and (film_title + film_budget) to remove duplicates. I also did the same in removing N/As in each data frame.

I filtered the files to exclude films with runtimes less than an hour and greater than four hours as these were most likely shorts/miniseries and filtered out any movies which had a budget of zero as these would be no use to my analysis.

I then removed the columns I didn't need in the files I was going to use to build the production_budget files before I using concat to stack the files together. I added a new column which showed the budgets in millions ($0.00m format) to make it easier to read before removing duplicates based on the same criteria as before. I carried out some analysis and made some observations as text in the file.

The next step was to pull in US CPI data, this was pulled via web scraping and linked the file to a website with CPI data dating back to 1913 and then calculated what the percentage difference between each year and the CPI figure in 2021. This gave me a percentage to multiply the film budgets and grosses to adjust their values to today's value to ensure each film was playing on the same field. I used merge to pull this percentage into each data frame. I then added a new column segmenting the films based on production_budgets using an IF/ELIF function (see Results 1 & 2).

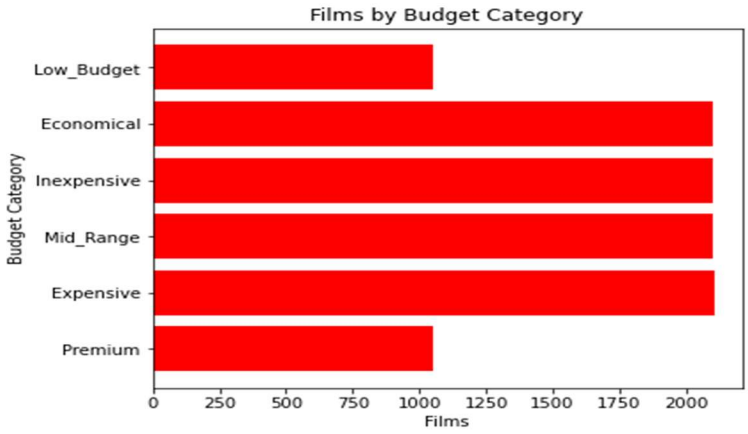
I repeated the steps for the film_gross figures before merging it with the production file to create a financial file. From this I carried out reviews into the correlation of budget and gross using functions and charts (see results 3 – 11). There is more analysis and details contained within the Python file. Once finished with the financial file, I returned to the production file to merge it with the IMDB file to create the audience reception file. The steps here were similar to the financial file but the results were different (see results 12 – 16). Then I repeated the steps once more for the Rotten Tomato file with results different from both the financial and IMDB files (see results 17 – 21).

My process concluded by adding a "pick a film" piece which allows me to enter the title of any film I can think of and run a loop which will pull me the production budget, gross amount, IMDB score and Rotten Tomato score when the film is present within the dataset.

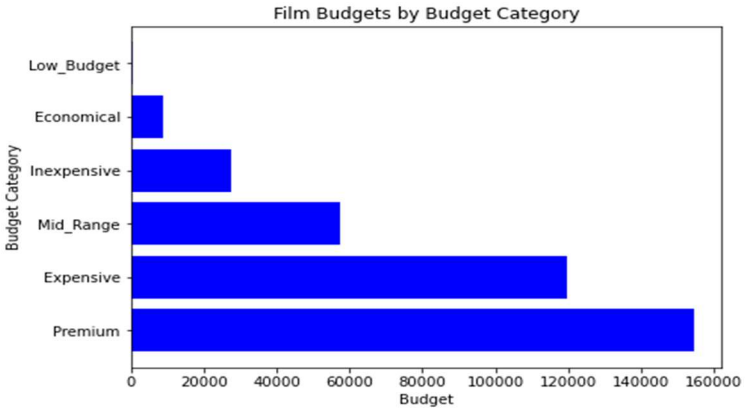
Results

Additional results found within the python file, below are the relevant images created by the python file.

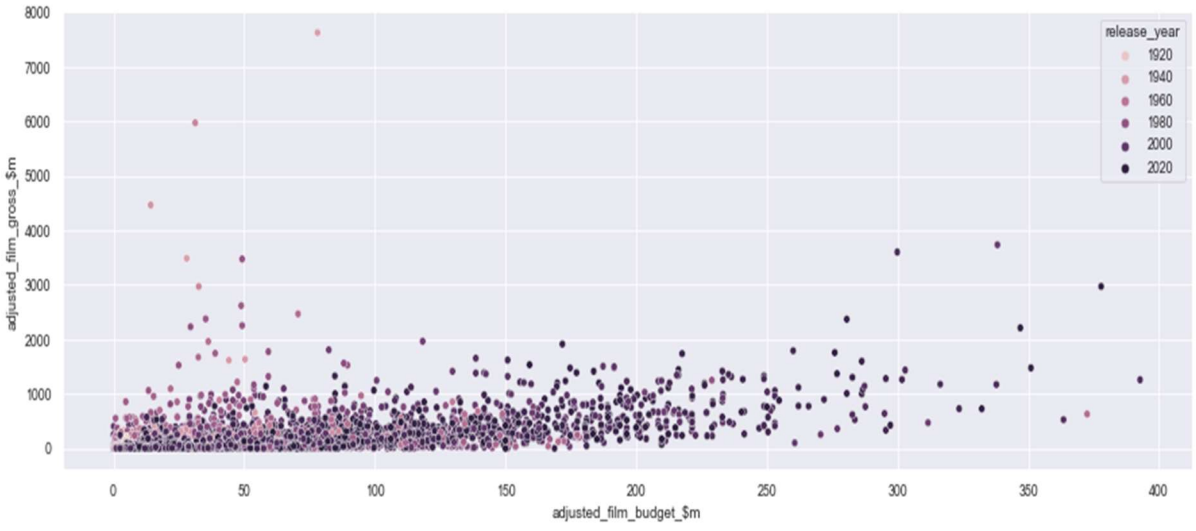
Result 1 – Split of films based on their budget categories.



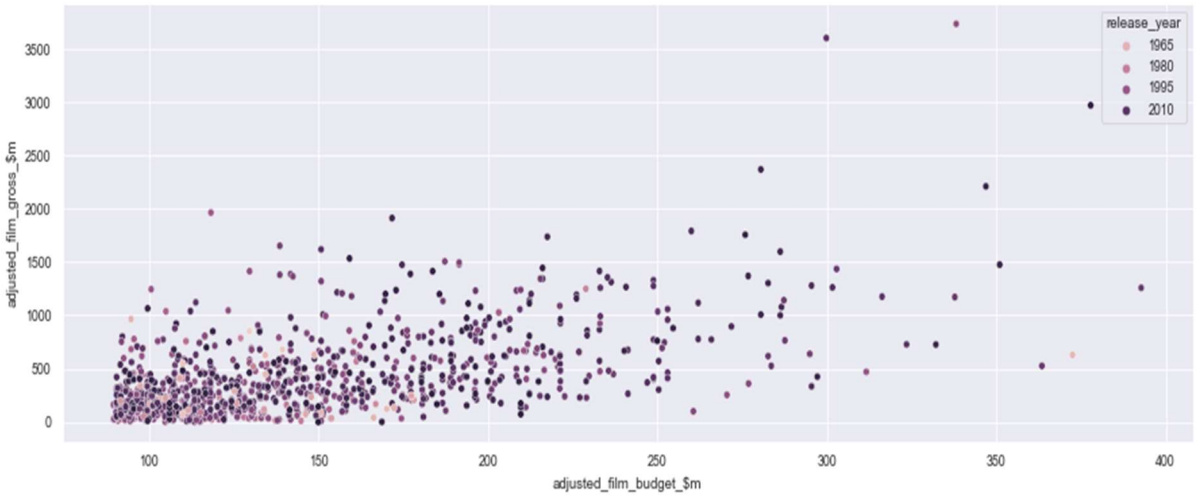
Result 2: Split of films based on the production budgets



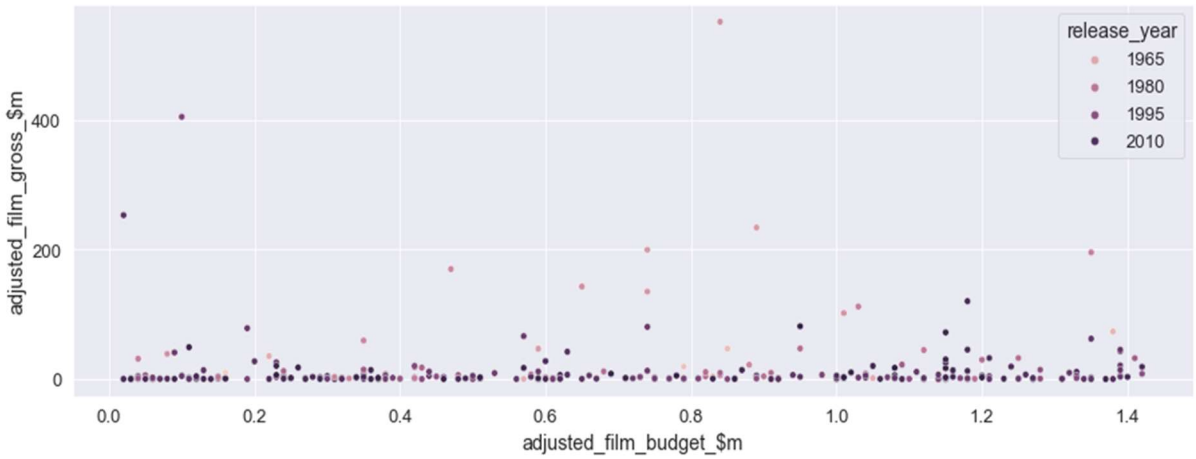
Result 3: Scatterplot of all films by budget (x) and gross (y):



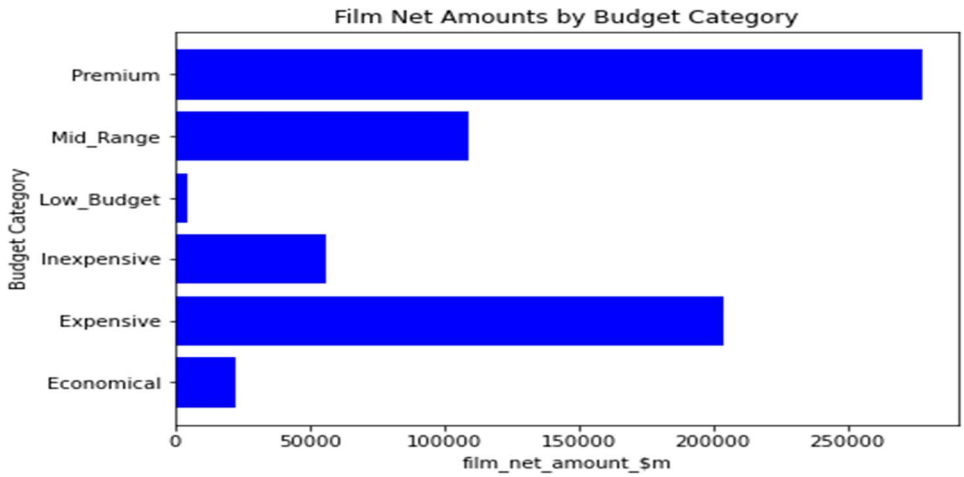
Result 4: Scatterplot of premium films by budget (x) and gross (y):



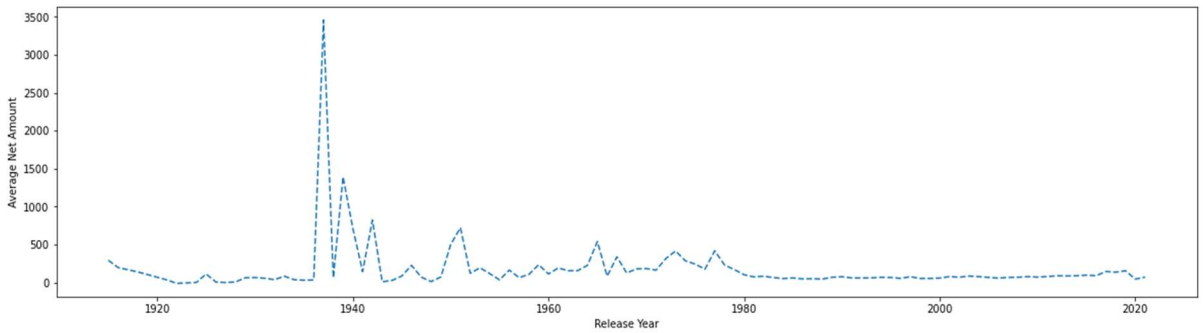
Result 5: Scatterplot of low_budget films by budget (x) and gross (y):



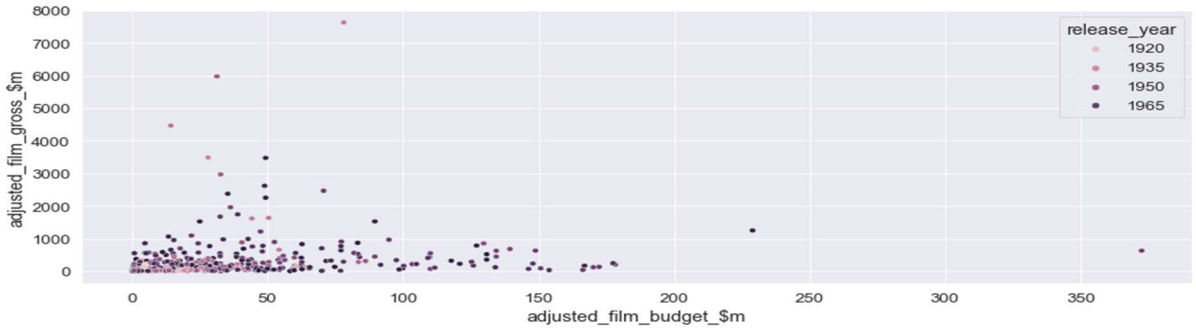
Result 6: Films by Net Amounts (Gross less Production)



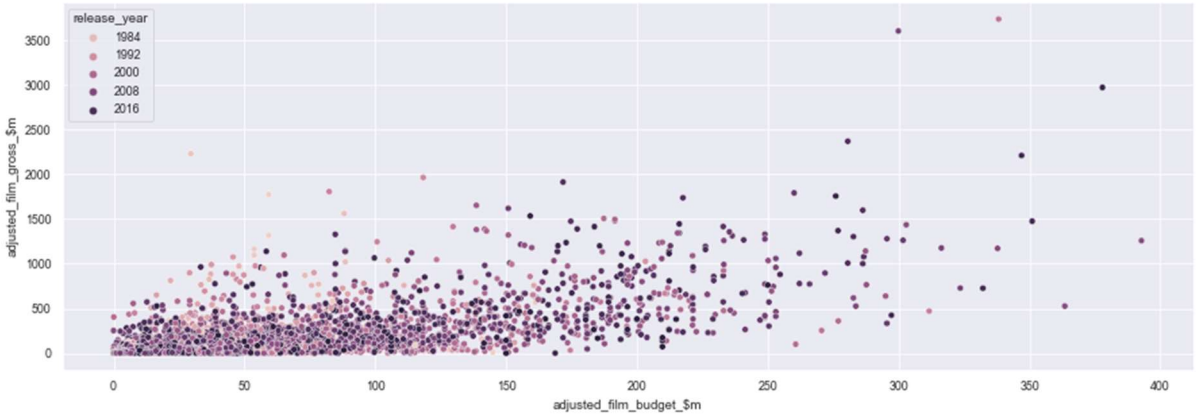
Result 7: Average Net Amount by year



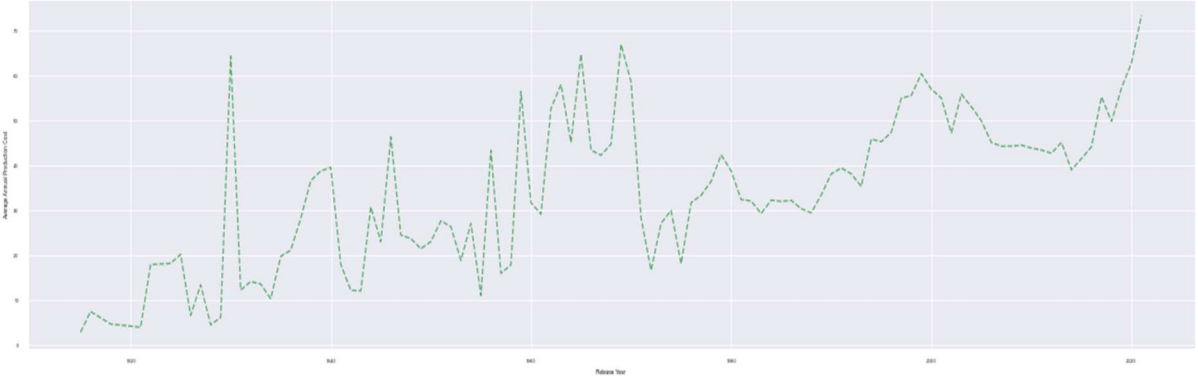
Result 8: Scatterplot of pre 1980 films by budget (x) and gross (y):



Result 9: Scatterplot of post 1980 films by budget (x) and gross (y):



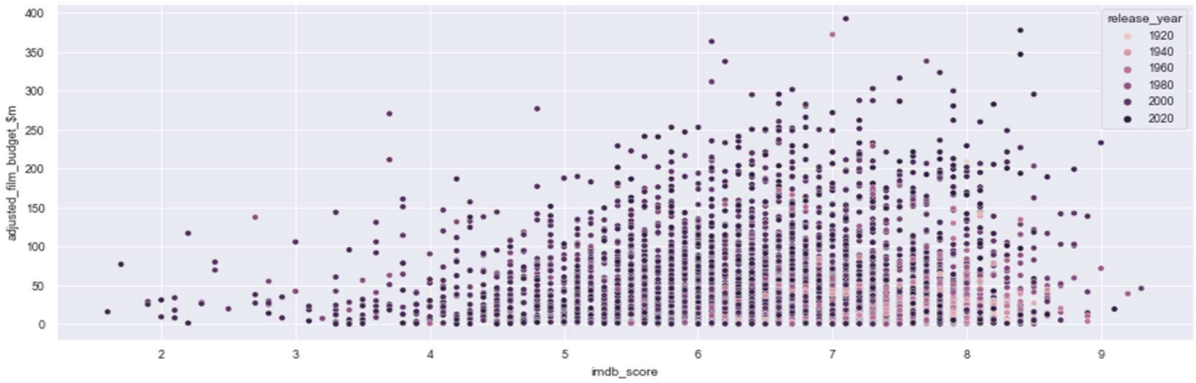
Result 10: Average Production Cost per Year



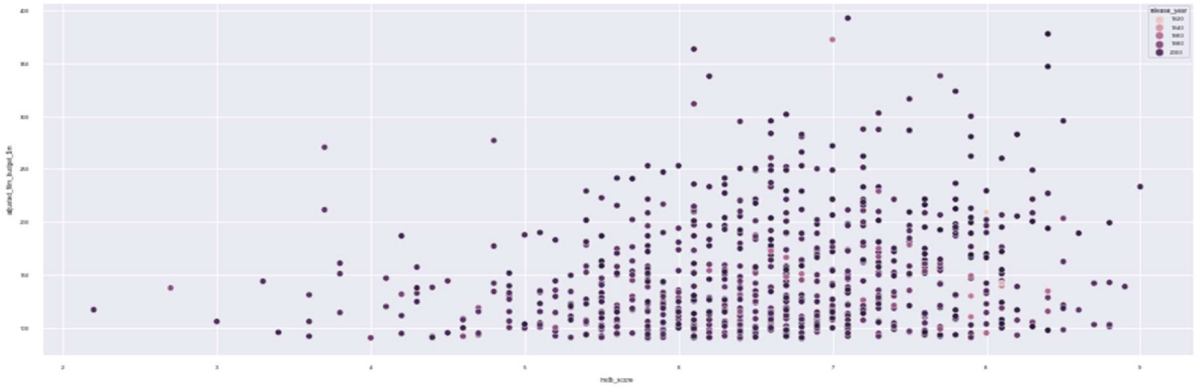
Result 11: Gross/Budget Category Splits by Film Profitability

film_profitability	budget_category	Profitable	Obscenely Profitable	Loss Making	Breakeven	loss_making_%	profitable_%
0	Premium	816	0	211	0	20.55	79.45
1	Expensive	1289	0	715	1	35.66	64.29
2	Mid_Range	1153	2	716	1	38.25	61.70
3	Economical	610	5	329	1	34.81	65.08
4	Inexpensive	931	1	621	0	39.99	60.01
5	Low_Budget	209	26	66	13	21.02	74.84

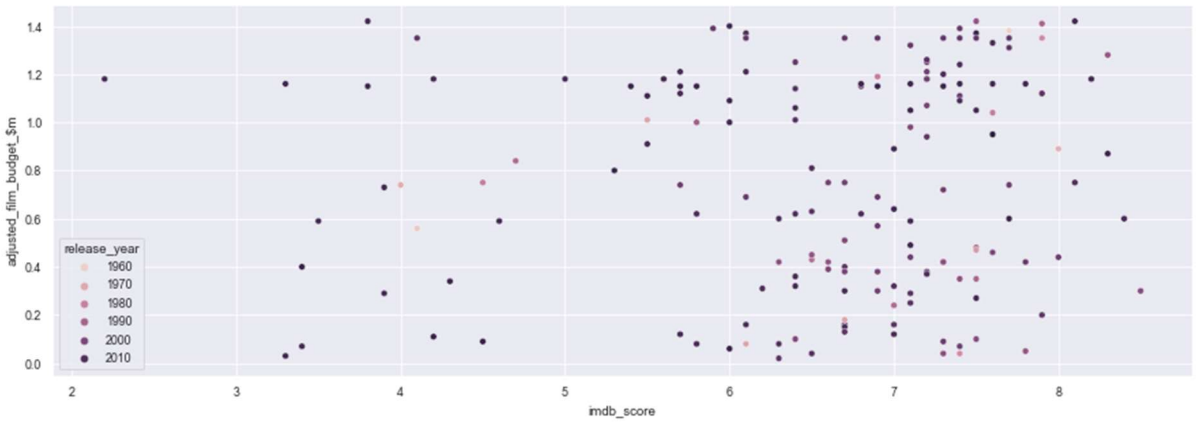
Result 12: Scatterplot of all films by IMDB score (x) and film budget (y):



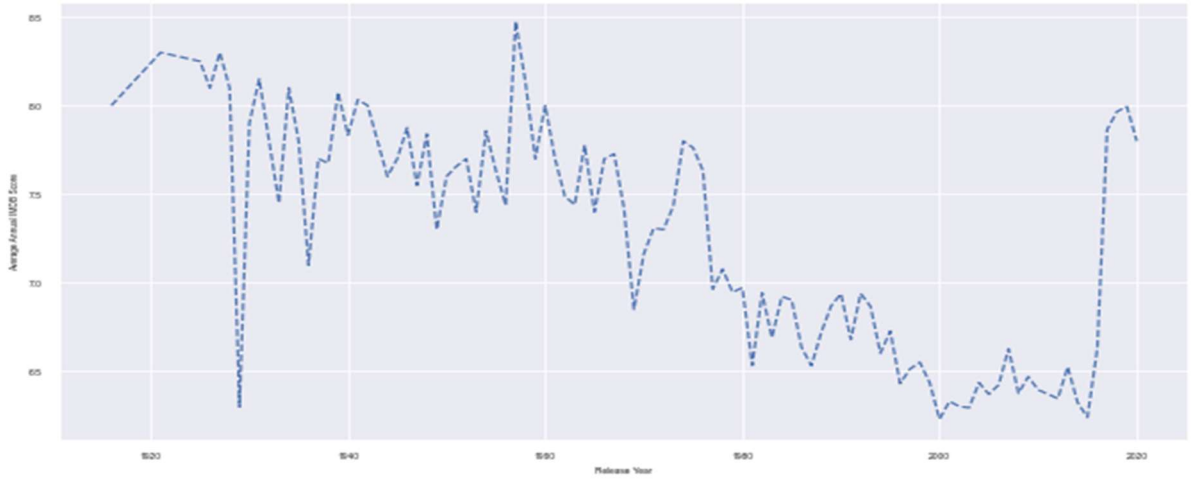
Result 13: Scatterplot of premium films by IMDB score (x) and film budget (y):



Result 14: Scatterplot of low_budget films by IMDB score (x) and film budget (y):



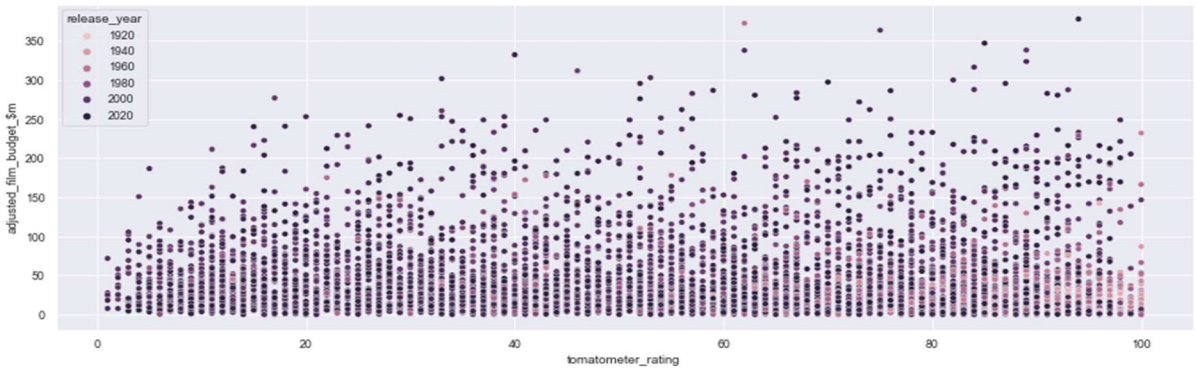
Result 15: Average IMDB Score by Year:



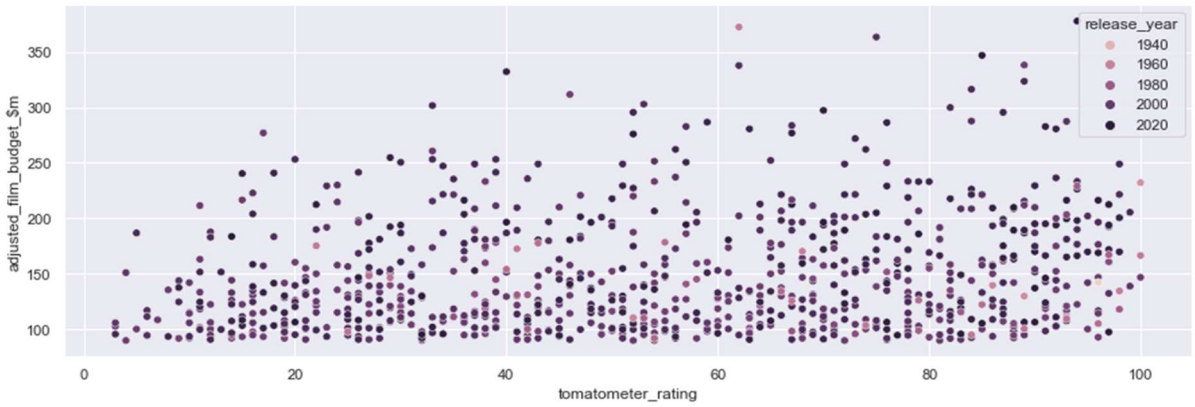
Result 16: Budget Category Splits by Film Audience Reception

audience_reception	budget_category	Adored	Above Average	Below Average	Poorly Received	below_average_%	above_average_%
0	Premium	87	328	311	53	46.73	53.27
1	Expensive	103	472	515	97	51.56	48.44
2	Economical	68	191	145	41	41.80	58.20
3	Mid_Range	84	396	346	97	48.00	52.00
4	Inexpensive	92	349	242	63	40.88	59.12
5	Low_Budget	13	86	44	21	39.63	60.37

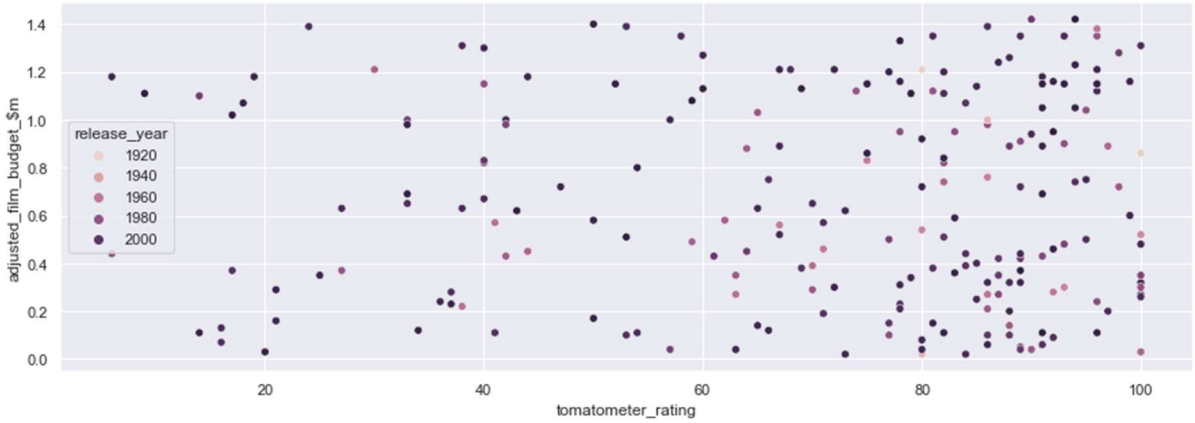
Result 17: Scatterplot of all films by Rotten Tomato percentage (x) and film budget (y):



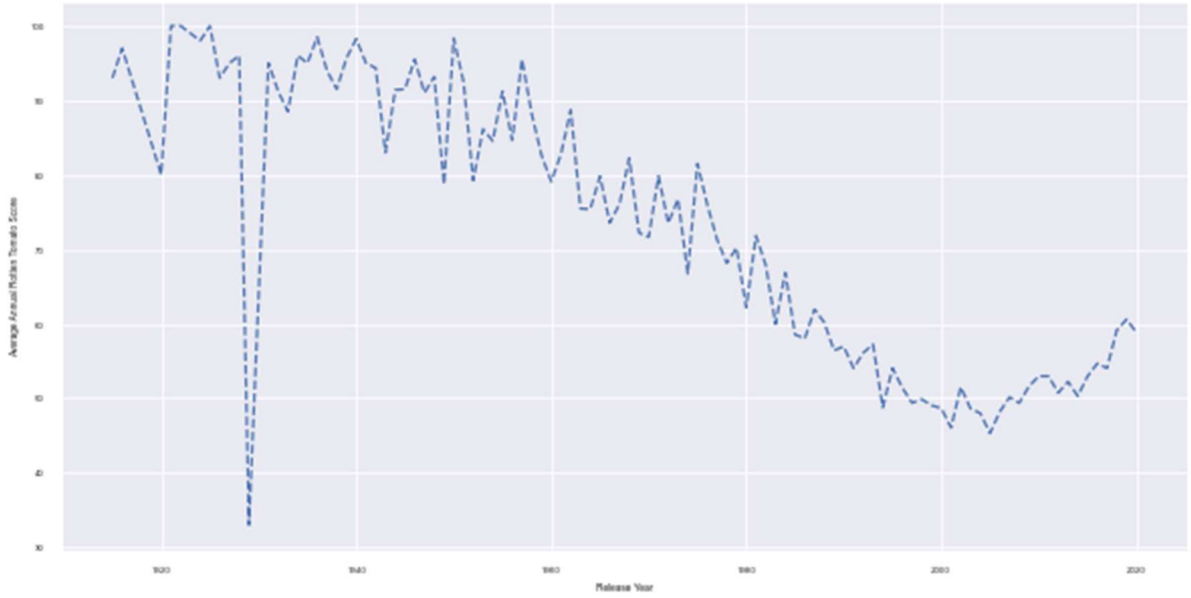
Result 18: Scatterplot of premium films by Rotten Tomato percentage (x) and film budget (y):



Result 19: Scatterplot of low_budget films by Rotten Tomato percentage (x) and film budget (y):



Result 20: Average Tomatometer score by year:



Result 21: Budget Category Splits by Film Critical Reception:

tomatometer_status	budget_category	Certified-Fresh	Rotten	Fresh	rotten_%	fresh_%
0	Premium	255	487	139	55.28	44.72
1	Expensive	348	955	310	59.21	40.79
2	Economical	183	273	191	42.19	57.81
3	Mid_Range	283	756	347	54.55	45.45
4	Inexpensive	325	514	257	46.90	53.10
5	Low_Budget	79	58	77	27.10	72.90

Insights

The first area that I will explore is the relationship between film production budgets and film gross amounts.

- With a correlation coefficient of 0.54 there is moderately positive correlation between the amount spent on film production and the amounts which the film grosses.
    - The coefficient increases to 0.57 when the film budget exceeds $89.58m suggesting that when production companies spend premium amounts on their films, they are more likely to return a higher gross.
    - The coefficient decreases to 0.03 when the film budget is less than $1.43m suggesting that low budget productions are not guaranteed to gross low amounts.
- The correlation coefficient links up well with the profitability data in the file as roughly 21% of films which spend either less than $1.43m or more than $89.58 make losses while almost 40% of the films in the data which spent between $1.43m and $7.80 are loss making. This suggests that films which manage to keep their budgets low or films which are provided with massive budgets tend to be safer bets against losses than films that fall in between those two budget ranges.
- In terms of the relationship between film production budgets and gross amounts over the years, the average net amount received per year tends to be volatile up until the beginning of the 1980s at which point the amount stabilised until some increases are seen in the late 2010s.
    - I ran the correlation coefficient on the pre-1980 films and clashed it against the post 1980 films and the difference was stark. The post 1980 films were over 4 times more correlated than the films which came before 1980. I provided my personal take on this in the file.

Now to look at the relationship between film budgets and IMDB scores:

- There is no real relationship here. Bigger budgets do not guarantee bigger scores from the audiences who provide ratings on IMDB.
    - In fact, even spending more on a film which produces a good score, like Cleopatra (Cost: $373m/7.0 IMDB) and Pirates of the Caribbean: At World's End (Cost: $393m/7.1 IMDB), reduces the spend per IMDB point compared to films with more modest spends which scored higher, like Clerks (Cost: $500k/7.8 IMDB).
- Lower budget movies tend to be better received; they are the only category where under 40% of the films are poorly received. However, this could be due to lower number of reviews on IMDB meaning the numbers are skewed higher than films with a greater number of reviews.

Finally, looking at the relationship between film budgets and critical reviews on Rotten Tomatoes

- This relationship is negatively correlated but only by a miniscule amount (-0.04) which means there is probably no correlation at all between the two.
- However, when I compare the percentage of films considered rotten by critics within each category of production budget, only 27% of the low budget (under $1.43m) are considered rotten while 59% of the expensive ($37.72m - $89.58m) and 55% of the premium category (>$89.58m) have been deemed rotten. This leans into a theory of mine that critics are harsher on films where big amounts have been spent and prefer the lower budgeted titles as they contain fewer special effects, CGI and higher paid casts.

In conclusion, I feel that if someone wanted to get into the film industry to make profits, then invest in or produce several low budget independent films, especially horror films (search the Saw, Blair Witch and Paranormal Activity films in the python file!). Failing that, invest in all the upcoming Marvel films as they're just printing money right now although the saturation point for these may be coming soon as the film quality declines from the previous phases.

References

CPI for data scraping. Available at: https://www.minneapolisfed.org/about-us/monetary-policy/inflation-calculator/consumer-price-index-1913- (Accessed 23 January 2022)

Worldwide Blockbusters. Available at: https://www.kaggle.com/narmelan/top-ten-blockbusters-20191977 (Accessed 22 December 2021)

The Movie Dataset: https://www.kaggle.com/rounakbanik/the-movies-dataset (Accessed 22 December 2021)

Top 4000 Movies Dataset: https://www.kaggle.com/axeltorbenson/top-4000-movies (Accessed 22 December 2021)

Collection of top grossing movies: https://www.kaggle.com/saivamshi/collections-of-top-grossing-movies (Accessed 22 December 2021)

All Time worldwide box office: https://www.kaggle.com/kkhandekar/all-time-worldwide-box-office (Accessed 22 December 2021)

Movie Industry Dataset: https://www.kaggle.com/danielgrijalvas/movies (Accessed 22 December 2021)

TMDB 5000: https://www.kaggle.com/tmdb/tmdb-movie-metadata (Accessed 22 December 2021)

Rotten Tomatoes: https://www.kaggle.com/stefanoleone992/rotten-tomatoes-movies-and-critic-reviews-dataset (Accessed 22 December 2021)

IMDB Top 1000: https://www.kaggle.com/harshitshankhdhar/imdb-dataset-of-top-1000-movies-and-tv-shows (Accessed 22 December 2021)

IMDB 5000: https://www.kaggle.com/carolzhangdc/imdb-5000-movie-dataset (Accessed 22 December 2021)