Project Report

# GitHub URL

<https://github.com/ianrowe101/assignment_files.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 am currently working in an Analyst role which is heavily reliant on Excel and while I would consider myself at an advanced level in Excel, I feel that I need to expand my skillset so that I may add further value to my employer either now or sometime in the future.
* Educational:
  + I completed Bachelor’s in financial services in May 2021 during which I used Excel to pull data from online sources and put them 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 felt that learning some form of coding could only benefit me in the future.
* Personal:
  + Aside from working in an analytical role, I am also interested in the use of data in my day-to-day life, be that from using the analytical tools to manage my finances, my workouts, my diet, my fantasy football team, my football manager team on the PC or any other interests where marginal gains can be made using data analysis. 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.

In terms of this project, I created it using the tools I learned via the lectures and in DataCamp and in choosing the topic, I wanted to choose a subject that was interesting to me while also choosing 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

My aim for the project was to test the impact on production costs/budgets of the film industry and if there is any link/correlation in how much money is spent on a film against how well that film performs in financial terms, audience reception and critical reception. Essentially, I wanted to learn if more equalled more.

I choose the film industry as I am obsessed with film and spend chunks of each day either watching films, reading film trivia, or listening to film podcasts/audiobooks which dig deeper or provide greater insight into the industry. It is the only interest of mine that rivals sport and because it had some financial aspects to it, production costs and box office gross amounts, I felt I could use my knowledge of financial measurements/analytics and marry it with my knowledge of the film industry to provide an insight into an area of the industry that has always been a curiosity to me.

# Dataset

In creating the dataset, I wanted to create as large a population as I could possibly find but in searching Kaggle, I was unable to find one up to date source which provided me with everything I needed. I needed the following:

* Production Costs/Budgets
* Box Office Gross Amounts
* Release Dates/Years
* IMDB Scores
  + For Audience Ratings
* Rotten Tomato Percentage
  + For Critics’ Ratings

In addition to these, I wanted to make sure I was pulling in the most recent data available as so much has happened in the film industry in the last five years. To achieve all of this I downloaded 11 datasets from Kaggle carried out some reviews in Excel to make sure they were suitable and then named them as:

* all\_time\_worldwide\_bo
* blockbuster
* imdb\_movie\_metadata
* imdb\_top\_1000
* movie\_industry\_dataset
* rotten\_tomatoes\_movie
* tgm\_bo\_summary
* tmdb\_movie
* tmds\_movies\_metadata
* top\_grossing\_film
* top\_movies\_data

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

# Implementation Process

I began by importing the python environments required to complete the functions within the project. 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 datasets that I had downloaded from Kaggle. All files were in .csv formats so I created a loop that loaded each file into the kernel as data frames. Doing this removed the need to load them individually, which considering there were 11 files, this saved me from repeating several of the same steps to reach the same position.

Once they were 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 which would give me an idea of what needed to be cleansed. I would use these loops and the dictionary of data frame names to investigate the changes made by certain steps in the process. I then gave an analysis of the current condition of the data frames and the work I wanted to complete with them.

The first areas that needed to be addressed came from columns in some data frames where I expected there to be numeric entries but due to the inclusion of special characters such as “$” or “,” they loaded in as strings. 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 each file had the below column headings:

* release\_year
* release\_date (where date was present)
* film\_title
* film\_budget
* film\_runtime
* film\_gross

Where the release dates were a 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 when it got to that stage of the report. I knew that I would come across duplicates when it came to stacking the data frames on top of each other via concat but before I get to that point, I removed any duplicates which were present in each individual data frame prior to stacking. 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 then filtered the files to exclude films with runtimes less than an hour and greater than four hours as these would not be the types of films which would fall into my interests. I then filtered out any movies which had a budget of zero as these would be impossible to fit into 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 used 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 1914 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

# Results

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

Chart, bar chart

Description automatically generated

Result 2: Split of films based on the production budgets

Chart, bar chart

Description automatically generated

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

Chart, scatter chart

Description automatically generated

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

Chart, scatter chart

Description automatically generated

Result 5: Scatterplot of low\_budget films by budget (x) and gross (y):

Chart, scatter chart

Description automatically generated

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

Chart, bar chart

Description automatically generated

Result 7: Average Net Amount by year

Chart

Description automatically generated

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

Chart, scatter chart

Description automatically generated

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

Chart, scatter chart

Description automatically generated

Result 10: Average Production Cost per Year

Graphical user interface, chart

Description automatically generated

Result 11: Gross/Budget Category Splits by Film Profitability

A screenshot of a computer

Description automatically generated with medium confidence

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

Chart, scatter chart

Description automatically generated

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

Chart, scatter chart

Description automatically generated

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

Chart, scatter chart

Description automatically generated

Result 15: Average IMDB Score by Year:

Graphical user interface, chart, line chart

Description automatically generated

Result 16: Budget Category Splits by Film Audience Reception

A screenshot of a computer

Description automatically generated with medium confidence

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

A picture containing text, people, day

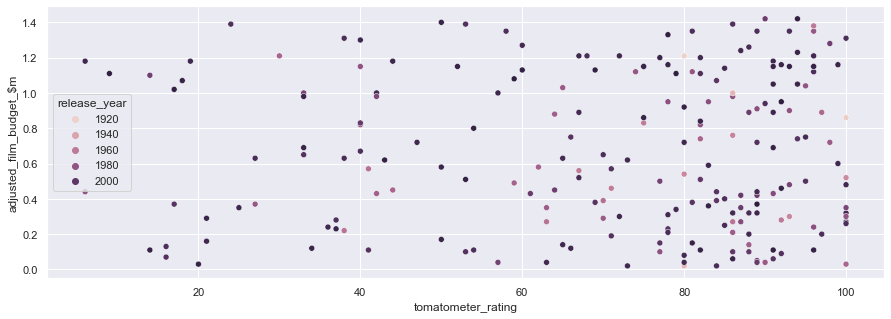
Description automatically generated

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

Chart, scatter chart

Description automatically generated

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



Result 20: Average Tomatometer score by year:

Graphical user interface, chart, scatter chart

Description automatically generated

Result 21: Budget Category Splits by Film Critical Reception:

A screenshot of a computer screen

Description automatically generated with medium confidence

# Insights

(Point out at least 5 insights in bullet points)

# References

(Include any references if required)