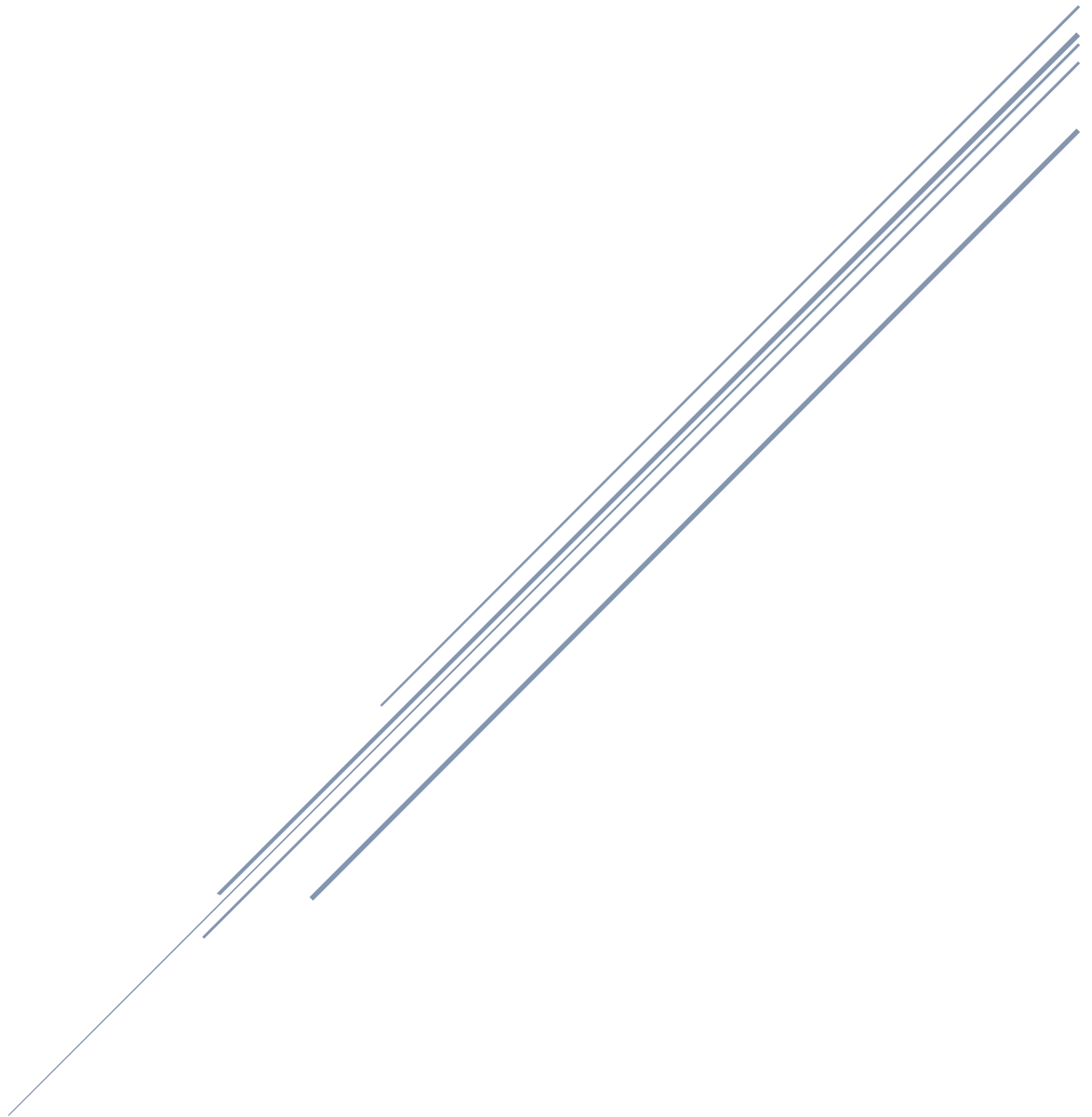


PROJECT REPORT

ANALYSIS IMDB TOP 1000 MOVIES



Kevin Kiding

Table of Content

Chapter 1.....	3
1.1 Background	3
1.2 Project Objective	3
1.3 Methods Applied.....	4
1.4 Benefits and Impacts.....	4
Chapter 2.....	6
2.1 Dataset Source	6
2.2 Data Overview.....	6
2.3 Data Cleaning and Transformation	8
2.4 Data Exploration.....	9
Chapter 3.....	11
3.1 Distribution of IMDB Ratings.....	11
3.2 Trend Analysis of Movie Released Year	12
3.3 Relationship between IMDB Rating and Movie Duration.....	13
3.4 Relationship between Movie Genre and IMDB Ratings.....	14
3.5 IMDB Rating Analysis of Movie Directors.....	15
3.6 Relationship between Average IMDB Rating and Film Duration	16
3.7 Analysis of Movie Revenue Distribution by Genre.....	17
3.8 Correlation Analysis between Features	18
Chapter 4.....	19
4.1 Summary of Findings.....	19
4.2 Recommendations.....	19
Chapter 5.....	21

Chapter 6.....	22
----------------	----

Chapter 1

Introduction

1.1 Background

A movie, also called a film, is a story told through moving images projected on a screen to create the illusion of motion. As a popular entertainment medium, movies immerse the audience in an imaginary world and a continuous narrative. Movies create stories or events using visuals that express emotions, relationships, and experiences that cannot usually be conveyed in any other way. Movies address social, political, environmental, and cultural issues as a record of society.

The IMDB Movie Dataset provides knowledge about industry impact and audience preferences. Analyzing this dataset of the top 1000 IMDB rated movies makes it possible to understand trends and influential factors in the world of cinema. To producers, filmmakers, and distributors, this dataset offers valuable information to inform decisions about production, marketing, and distribution. For audiences, it develops an appreciation of the world of cinema and helps them find movies that suit their interests.

Movies are a major form of entertainment and culture, bringing audiences into storylines and worlds through moving images. IMDB's dataset of high-rated movies enables comprehensive analysis of the industry, benefiting both filmmakers and audiences. This data provides vital information to guide production and marketing choices and enables a better understanding of audience preferences and cinematic trends. This dataset highlights the role of movies in addressing cultural issues and their influence on society.

1.2 Project Objective

The objectives of this analysis are as outlined below:

- To gain insights into audience preferences and critical acclaim, it is essential to analyze the IMDb ratings and running times of the top one thousand films.
- To gain insight into the development of the film industry, it is valuable to analyze trends in film release years over time.
- To determine the factors that influence success, it is important to examine the relationships between variables such as genre, director, rating, revenue, and runtime.

1.3 Methods Applied

In accordance with the following, the methods used in this analysis:

- Data Acquisition: The "IMDB Movies - Top 1000 Movies by IMDB Rating" dataset was obtained from Kaggle and other sources.
- Data Cleaning: Data cleaning procedures were performed, including managing missing values, removing duplicates, and converting data types.
- Analysis of Ratings Distribution: The distribution of IMDB ratings was analyzed to gain insights into audience preferences.
- Evaluation of Genres: The relationship between movie genres and ratings was investigated to identify commercially successful genres.
- Visualization Design: Visually compelling data visualizations were created to effectively communicate analysis findings to stakeholders.

1.4 Benefits and Impacts

Insight and perseverance are essential success factors.

- This study analyzes IMDB ratings and provides empirical insights into audience preferences and reception. This has the potential to improve filmmakers' and studios' decision-making.
- The purpose of this research is to analyze and identify trends and patterns in the release of films over a specific period. This helps us understand the evolution of the industry.
- This study investigates the relationships between rating, revenue, and duration. Improves understanding of factors that contribute to success.

The Strengthening of Audiences:

- This feature enables viewers to make more informed decisions by providing information about ratings, duration, genre, and other relevant data.
- Provides insight into subjective metrics, including ratings and financial performance.
- Audiences could be empowered to advocate for the content they find valuable.

Research and assets:

- This study presents a replicable framework and paradigm for analyzing film data.

- This study compiles and preprocesses a dataset containing the 1000 most popular films on IMDB to facilitate future research.
- Statistical analyses serve as benchmarks and reference factors for evaluating the performance of a film.

Chapter 2

Data Acquisition and Preparation

2.1 Dataset Source

In this project, the dataset used is "IMDB Movies - Top 1000 Movies by IMDB Rating" which is taken from a publicly available source, Kaggle. This dataset contains information about the top 1000 movies by IMDB rating. This dataset is a CSV file that can be accessed through the following link: [dataset link](https://raw.githubusercontent.com/kevinkevinn/data-analyst-portfolio/main/Top%201000%20Movie/imdb_top_1000.csv).

In order to start analyzing the data, the dataset was uploaded to the Google Collab environment using the following [code](#):

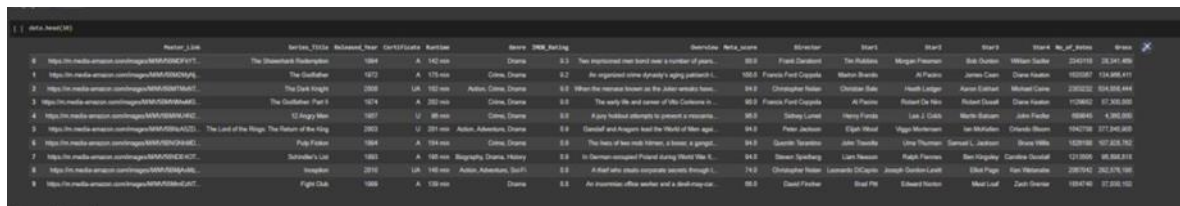
```
link = 'https://raw.githubusercontent.com/kevinkevinn/data-analyst-portfolio/main/Top%201000%20Movie/imdb_top_1000.csv'
data = pd.read_csv(link)
```

In this segment, we read the dataset in CSV format and store it in data variables using the pandas library. This dataset will serve as a base for additional analysis of preferences and trends in the movie industry.

2.2 Data Overview

This project uses the "IMDB Movies - Top 1000 Movies by IMDB Rating" dataset. This information comprises movie title, release year, classification certificate, genre, duration, director, lead actor, and IMDB rating. Pandas on Google Colab retrieved and loaded the dataset. This project will analyze IMDB rating and runtime, genre, and director.

The following is the code to display the top 10 data from the dataset:



Rank	Title	Release Year	Certificate	Runtime	Genre	IMDB Rating	Director	Actor
1	The Shawshank Redemption	1994	A	142 min	Drama	8.9	Frank Darabont	Morgan Freeman
2	The Godfather	1972	A	175 min	Crim, Drama	8.7	Francis Ford Coppola	Al Pacino
3	The Godfather Part II	1974	A	202 min	Crim, Drama	8.6	Francis Ford Coppola	Al Pacino
4	The Godfather Part I	1972	A	175 min	Crim, Drama	8.6	Francis Ford Coppola	Al Pacino
5	12 Angry Men	1957	U	96 min	Drama	8.9	Sidney Lumet	Henry Fonda
6	The Lord of the Rings: The Return of the King	2003	U	201 min	Action, Adventure, Drama	8.9	Peter Jackson	Elijah Wood
7	Pulp Fiction	1994	A	154 min	Crim, Drama	8.8	Quentin Tarantino	John Travolta
8	Schindler's List	1993	A	195 min	Bio-graphy, Drama, History	8.9	Steven Spielberg	Ralph Fiennes
9	Heat	1995	A	139 min	Action, Adventure, Thriller	8.3	Michael Bay	Al Pacino
10	Flight Club	1996	A	128 min	Drama	8.5	David Fincher	Brad Pitt

This analysis will examine notable aspects of the dataset "IMDB Movies - Top 1000 Movies by IMDB Rating." This dataset contains information about the 1,000 highest-rated films, as determined by IMDB ratings. This document's contents will be analyzed to identify noteworthy and valuable information.

```
data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 16 columns):
#   Column              Non-Null Count  Dtype
---  -
0   Poster_Link         1000 non-null   object
1   Series_Title        1000 non-null   object
2   Released_Year       1000 non-null   object
3   Certificate         899 non-null    object
4   Runtime             1000 non-null   object
5   Genre               1000 non-null   object
6   IMDb_Rating         1000 non-null   float64
7   Overview            1000 non-null   object
8   Meta_score          843 non-null    float64
9   Director            1000 non-null   object
10  Star1               1000 non-null   object
11  Star2               1000 non-null   object
12  Star3               1000 non-null   object
13  Star4               1000 non-null   object
14  No_of_Votes         1000 non-null   int64
15  Gross               831 non-null    object
dtypes: float64(2), int64(1), object(13)
memory usage: 125.1+ KB
```

The list titled "IMDB Movies - Top 1000 Movies by IMDB Rating" draws attention to several significant factors that merit investigation. Initially, it is important to note that certain columns, including "Certificate," "Meta_score," and "Gross," contain null values. As additional data analysis is conducted, including this lacking value may affect the veracity and dependability of the data. In addition, there is a disparity between the categories of data observed in certain domains and the expected data types. Instead of using an object data type, it would be preferable to designate a number or DateTime data type to the 'Released_Year' column. Likewise, retaining the 'Gross' column, which represents monetary quantities, is recommended as a numeric data type instead of an object data type. It is essential to address these concerns to ensure the veracity and suitability of the information for subsequent scholarly research.

```
data = pd.DataFrame(data)

def convert_runtime(time):
    minutes = time.split(' ')[0]
    if len(minutes) == 2:
        minutes = '0' + minutes
    return minutes + ' min'

# Using a function to change the 'Runtime' column
data['Runtime'] = data['Runtime'].apply(convert_runtime)

data
```

A single line of code is provided for addressing these concerns. The code uses the 'pd.DataFrame()' method to convert the 'data' DataFrame into a new DataFrame. 'convert_runtime' is a method that modifies the figures in the 'Runtime' column. If necessary, this method divides the time value, extracts the minutes, and adds a preceding zero to ensure that the result is always in minutes. The final step is to use the 'apply()' method to apply the 'convert_runtime' code to each value in the 'Runtime' column. This action will update the column values to reflect the new information. By executing this code, the 'Runtime' field is standardized, allowing accurate and consistent analysis of film running times.

2.3 Data Cleaning and Transformation

The provided code directed the multiple-step Dataset Cleaning and Transformation procedure. First, we calculated the total number of missing values in each column to evaluate the missing values in the dataset. The percentage of absent values relative to the total number of values in the dataset was then calculated. The information was then inserted into a DataFrame titled "missing_data."

In the subsequent phase, the dataset was cleansed by removing entries with missing values. This was accomplished with the `dropna()` method. A new, pristine dataset was generated. The cleansed data were then saved to a newly created file titled "[cleaned_data.csv](#)."

```
missing_values = data.isnull().sum()
total_values = len(data)
missing_percentage = (missing_values / total_values) * 100

missing_data = pd.DataFrame({'Jumlah Missing Value': missing_values, 'Persentase Missing Value': missing_percentage})
missing_data['Persentase Missing Value'] = missing_data['Persentase Missing Value'].map('{:.1f}%'.format)
print(missing_data)
```

	Jumlah Missing Value	Persentase Missing Value
Poster_Link	0	0.0%
Series_Title	0	0.0%
Released_Year	0	0.0%
Certificate	101	10.1%
Runtime	0	0.0%
Genre	0	0.0%
IMDB_Rating	0	0.0%
Overview	0	0.0%
Meta_score	107	10.7%
Director	0	0.0%
Star1	0	0.0%
Star2	0	0.0%
Star3	0	0.0%
Star4	0	0.0%
No_of_Votes	0	0.0%
Gross	169	16.9%

```
Drop missing value and create new dataset

[] # Drop rows with missing values
data_cleaned = data.dropna()

# Create and save the cleaned dataset to a new file
data_cleaned.to_csv('cleaned_data.csv', index=False)
```

Several columns' data types were modified to ensure each column's data type was appropriate. The 'Released_Year' column's data type has been changed to object (string), while the 'Certificate' column's data type has been changed to category. The columns 'Runtime', 'Gross', 'Meta_score', 'No_of_Votes', and 'IMDB_Rating' have been converted to a string, float, numeric (integer), and string, correspondingly. After removing commas and other characters, the 'Gross' column was converted to float data type.

```
# Convert 'Released_Year' to object (string) data type
data['Released_Year'] = data['Released_Year'].astype(str)

# Convert 'Certificate' to category data type
data['Certificate'] = data['Certificate'].astype('category')

# Convert 'Runtime' to string data type
data['Runtime'] = data['Runtime'].astype(str)

# Convert 'Gross' to string data type
data['Gross'] = data['Gross'].astype(str)

# Remove commas and other characters from the 'Gross' column values
data['Gross'] = data['Gross'].str.replace(',', '').str.replace('$', '')

# Convert 'Gross' to float data type
data['Gross'] = pd.to_numeric(data['Gross'], errors='coerce')

# Convert 'Meta_score' to float data type
data['Meta_score'] = data['Meta_score'].astype(float)

# Convert 'No_of_Votes' to numeric (integer) data type
data['No_of_Votes'] = pd.to_numeric(data['No_of_Votes'], errors='coerce')

# Convert 'IMDB_Rating' to string data type
data['IMDB_Rating'] = data['IMDB_Rating'].astype(str)
```

Finally, the modified dataset's head was displayed, providing a preview of the sanitized and transformed data. It also highlighted the modifications made to the structure and data types of the dataset.

2.4 Data Exploration

After cleansing and transformation, the dataset was investigated further. The sanitized dataset includes 714 rows and 16 columns. The columns in the dataset include 'Poster_Link', 'Series_Title', 'Released_Year', 'Certificate', 'Runtime', 'Genre', 'IMDB_Rating', 'Overview', 'Meta_score', 'Director', 'Star1', 'Star2', 'Star3', 'Star4', 'No_of_Votes', and 'Gross'. The columns were modified to ensure data type consistency. The column 'Certificate' now has the category data type, the column 'Meta_score' has the float64 data type, and the columns 'No_of_Votes' and 'Gross' have the int64 data type.

```
data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 714 entries, 0 to 713
Data columns (total 16 columns):
 #   Column              Non-Null Count  Dtype  
---  -
 0   Poster_Link         714 non-null   object 
 1   Series_Title        714 non-null   object 
 2   Released_Year       714 non-null   object 
 3   Certificate          714 non-null   category
 4   Runtime             714 non-null   object 
 5   Genre               714 non-null   object 
 6   IMDB_Rating         714 non-null   object 
 7   Overview            714 non-null   object 
 8   Meta_score          714 non-null   float64 
 9   Director            714 non-null   object 
10   Star1               714 non-null   object 
11   Star2               714 non-null   object 
12   Star3               714 non-null   object 
13   Star4               714 non-null   object 
14   No_of_Votes         714 non-null   int64  
15   Gross               714 non-null   int64  
dtypes: category(1), float64(1), int64(2), object(12)
memory usage: 84.9+ KB
```

After analyzing the 'Certificate' column, it was discovered that it contains a variety of movie classification values. The 'Certificate' column's frequency distribution reveals various classifications, including 'U' (Universal), 'A' (Adult), 'UA' (Parental Guidance), 'R' (Restricted), 'PG-13' (Parental Guidance-13), 'G' (General Audience), 'Passed', 'Approved', 'GP' (General Audience - Parental Guidance Suggested), 'TV-PG' (Television - Parental Guidance), and 'U/A' (Universal with Adult). Each category corresponds to a particular age restriction or recommendation for the intended audience.

Using the provided code, irrelevant columns were removed from the dataset. The columns 'Poster_Link', 'Overview', 'Star1', 'Star2', 'Star3', and 'Star4' have been removed from the dataset. The relevant columns were deemed superfluous for the dataset's intended analysis and insights. Eliminating these columns simplifies and focuses the dataset on the most influential attributes and factors on movie ratings and audience preferences. This makes the document more concise and efficient for further examination.

```
# Remove irrelevant columns
columns_to_drop = ['Poster_Link', 'Overview', 'Star1', 'Star2', 'Star3', 'Star4']
data.drop(columns_to_drop, axis=1, inplace=True)
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 714 entries, 0 to 713
Data columns (total 10 columns):
#   Column          Non-Null Count  Dtype
---  -
0   Series_Title     714 non-null    object
1   Released_Year   714 non-null    object
2   Certificate      714 non-null    category
3   Runtime         714 non-null    object
4   Genre           714 non-null    object
5   IMDb_Rating     714 non-null    object
6   Meta_score      714 non-null    float64
7   Director        714 non-null    object
8   No_of_Votes     714 non-null    int64
9   Gross           714 non-null    int64
dtypes: category(1), float64(1), int64(2), object(6)
memory usage: 51.4+ KB
```

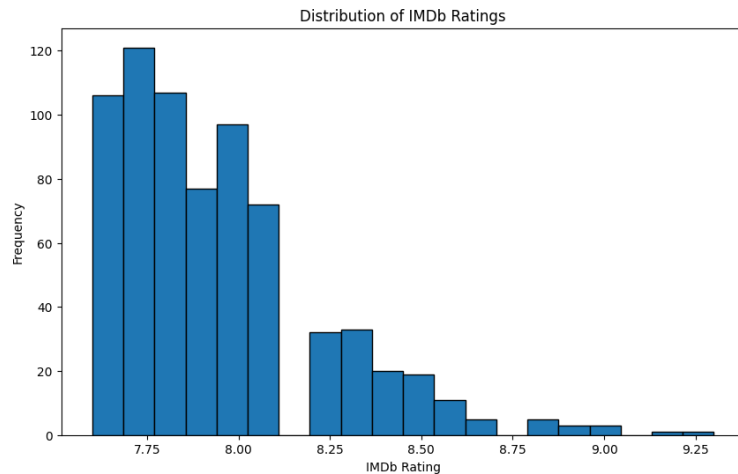
Analysis and Findings

3.1 Distribution of IMDB Ratings

The provided code converts the 'IMDB_Rating' column of the dataset to a floating-point value. Afterwards, it computes the minimum and maximum values for the IMDB ratings. To ensure that the highest rating falls within the amended range, a 0.01-point offset is incorporated. Consequently, the code uses a 20-bin histogram to depict the frequency distribution of IMDB ratings visually. The x-axis of the graph represents the IMDB ratings, while the y-axis shows the frequency of films within each rating category. The title of the chart is "IMDb Rating Distribution."

The distribution chart data reveals that a significant proportion of the films in the IMDB Top 1000 dataset have IMDb ratings between 7.6 and 8.5. The most prevalent ratings were 7.7, 7.8, and 8.0, with 121, 107, and 97 occurrences, respectively. The occurrence of higher ratings is inversely proportional to their respective frequency, indicating that as the rating increases, its prevalence decreases—a limited number of films with ratings higher than 8.5 out of 10. In addition, 9.0, 9.2, and 9.3 ratings are extremely uncommon, with only a single movie receiving each score.

The analysis of IMDb ratings within the IMDB Top 1000 Movies dataset indicates that most of the films within this dataset receive positive ratings. As indicated by the preponderance of ratings between 7.6 and 8.5, most films in the dataset were positively received by audiences. The observed distribution shows a progressive decrease in frequency as ratings increase, indicating that achieving higher ranks is more difficult and less common. In general, the distribution of films within the IMDB Top 1000 dataset indicates their acceptability and quality.

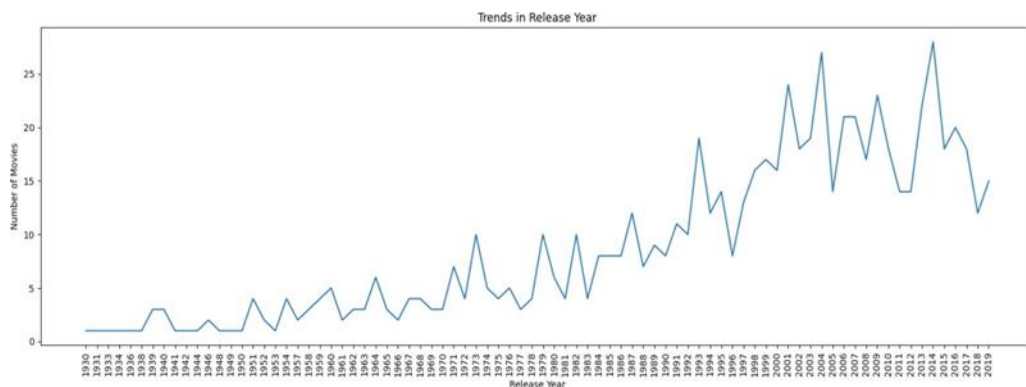


For a more clear view of the distribution chart, please click [here](#).

3.2 Trend Analysis of Movie Released Year

The distribution and evolution of film production throughout history can be better understood by analyzing the trend of movie release years. The dataset contains the number of films released annually from 1930 to 2019. The data have revealed fascinating patterns and fluctuations in the film industry.

After analyzing the data, early film production was relatively limited, with only a handful of films released each year. As the 20th century approaches its midpoint, the number of films produced increases gradually. With occasional fluctuations, the growth of film releases continues until the early 2000s. In 2004, the greatest number of films were issued in a single year, according to the dataset, with 27 films being released. After that, the caliber of filmmaking gradually declines, although the industry remains relatively stable.



For a more clear view of the distribution chart, please click [here](#).

A thorough understanding of the dynamics of the film industry can be attained through a thorough examination of the dominant trends associated with a film's release year. This study provides empirical evidence regarding fluctuations in the production of motion pictures during various time intervals.

Various factors, including but not limited to technological advancements, changing audience preferences, and economic considerations, can influence the observed patterns. The data presented here illustrates fluctuations in the film industry, which are characterized by alternating phases of growth and stability, thus demonstrating the inherently dynamic nature of this industry.

Analyzing trends in movie release years provides a broader understanding of the ever-changing nature of the entertainment industry. This statement emphasizes the ongoing efforts of filmmakers and production companies to meet audience expectations and adapt to changing market dynamics. The analysis provides additional evidence of cinema's profound historical and cultural significance as both an artistic medium and a widely accepted form of entertainment.

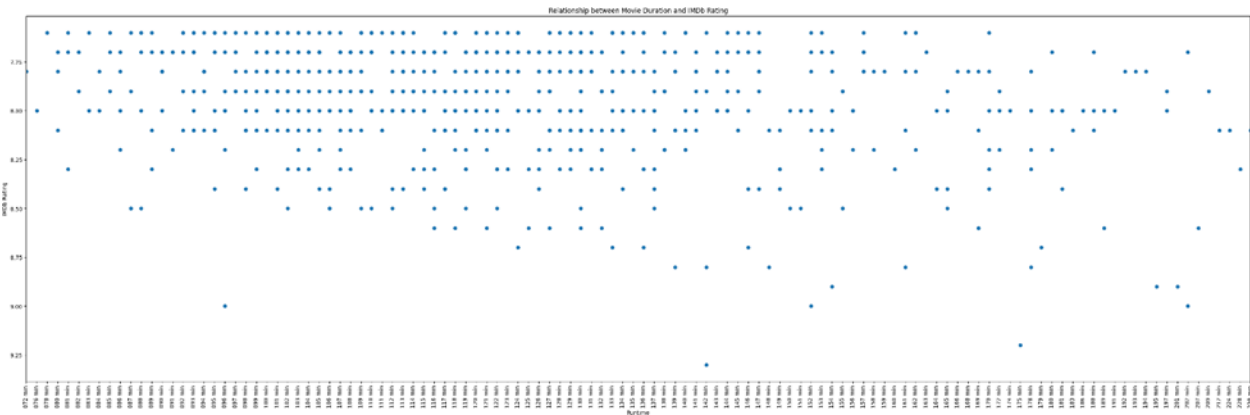
Analyzing trends in film release years provides a comprehensive insight into the development of the film industry. This study clarifies film production's temporal patterns and fluctuations, demonstrating the industry's adaptability and responsiveness to diverse influences. This observation emphasizes film's pervasive and enduring significance as a narrative communication and entertainment medium.

3.3 Relationship between IMDB Rating and Movie Duration

A scatterplot examined the relationship between an IMDB rating and a film's running time. The dataset consisted of films ranging in length from 72 minutes to 238 minutes. In addition, the movies' IMDB ratings ranged from 7.6 to 9.3 on a scale 10. The scatterplot did not, however, reveal a clear linear relationship between running time and IMDB rating. The lack of a distinct trend in the distribution of data points on the graph suggests that a film's rating is only partially determined by its running time. Various factors, including the narrative, the performance, and the overall production quality, probably influence the ratings.

Although a distinct linear pattern was not identified, it is still possible to identify clusters of data points that correspond to specific movie length intervals and have comparable IMDB ratings. This indicates that there may be a correlation between particular rating ranges and film running durations, although it is essential to note that this correlation does not apply to all movies. The scatterplot illustrates the significance of considering multiple factors when evaluating the performance and ratings of a movie. It demonstrates that duration alone is insufficient to measure a film's quality.

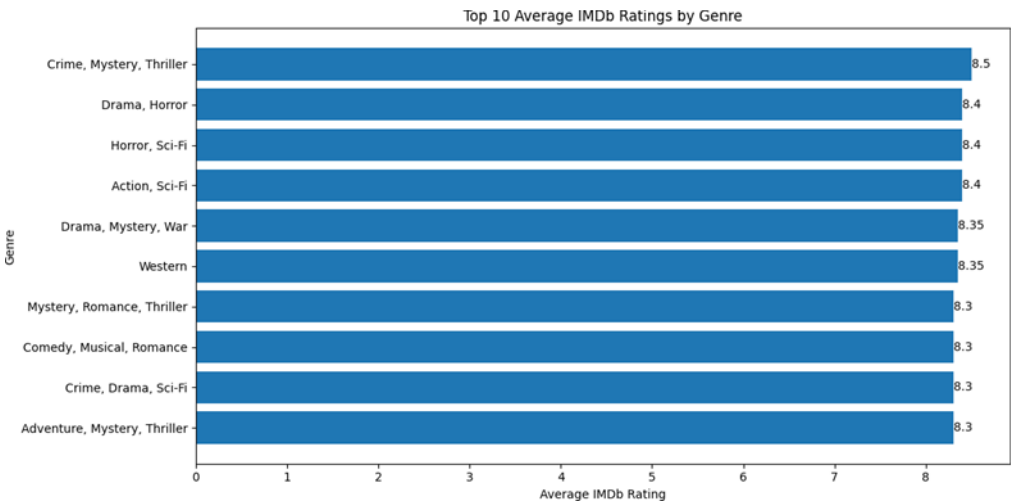
In conclusion, the scatterplot illustrates the factors affecting a movie's rating, specifically the correlation between the IMDB rating and the film's duration. The representation enables the identification of patterns and clusters within the data, despite the absence of a distinct linear relationship. This illustrates both the subjectivity of audience preferences and the complexity of filmmaking. It emphasizes that IMDB ratings are influenced by factors other than a film's running time.



To see the chart more clearly, click [here](#)

3.4 Relationship between Movie Genre and IMDB Ratings

The presented data illustrates the relationship between various film categories and their average IMDB ratings. According to IMDB, each category's average rating ranges from 7.60 to 8.40 out of 10. Each category is represented by a gauge that indicates the average rating. The graph enables the comparison of evaluations for diverse categories of content.



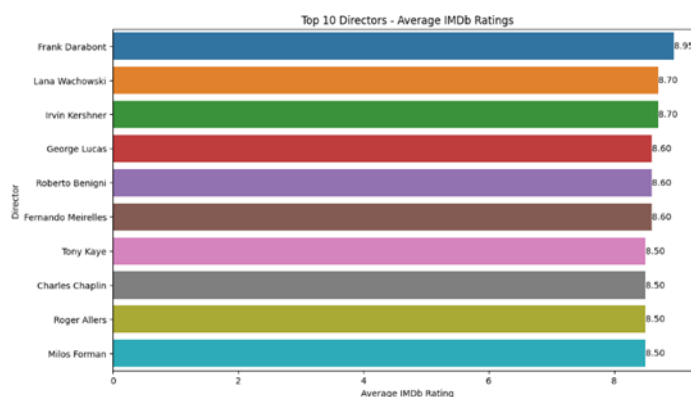
Top 10 values shown. Click [here](#) for full chart.

The bar graph displays the average IMDB ratings for prominent film genres. It appears that certain film genres receive higher average ratings than others. It is generally accepted that animation, drama, and biography films are the most popular genres. Typically, their products or services receive positive feedback. However, responses to horror and science fiction films tend to be more diverse.

It is essential to recognize that these evaluations are subjective. The predilection for them varies depending on the preferences of the individual. The evaluation of a film is influenced by several factors, including the plot, acting performances, direction, and production quality. The evaluations are not determined solely by the genre. Additionally, they serve as a reflection of everyone's opinion regarding the overall content of a film. This graph offers insightful information regarding the potential audience response to various film genres. However, individual perspectives can differ considerably. Ratings for films within the same genre can differ. This information can aid filmmakers and viewers in obtaining a deeper comprehension of general patterns. It is essential to remember, however, that ratings are subjective and that numerous factors contribute to a film's success.

3.5 IMDB Rating Analysis of Movie Directors

The results of the study on the IMDB ratings of film directors are displayed using a vertical bar graph. The graph illustrates the average IMDb rating for various filmmakers, emphasizing the wide variety of ratings they have received. The data comprises well-known filmmakers with consistently high average IMDb ratings. Frank Darabont, Lana Wachowski, Irvin Kershner, and George Lucas are notable examples. The ratings for these directors' range between 8.60 and 8.95 out of a possible 10. On the other hand, directors such as Ruben Fleischer, Joseph Kosinski, Jonathan Lynn, and Jonathan Levine have averaged 7.60 out of 10 points.



Top 10 displayed due to abundant data. Click [here](#) for full graph.

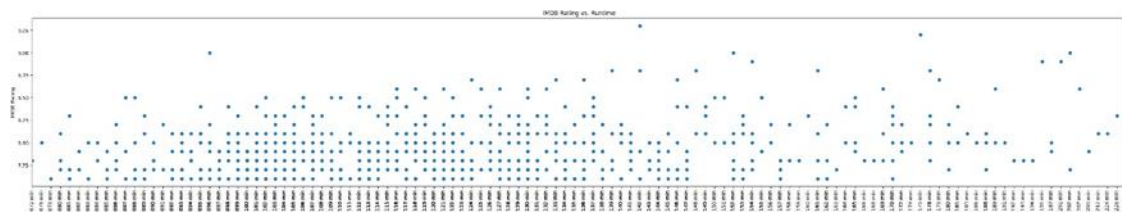
The depicted bar graph illustrates the fluctuations in average scores across various directors, emphasizing disparities in the responses of viewers and critics. Frank Darabont and Lana Wachowski consistently receive high average ratings, demonstrating their ability to create influential and widely praised films. In contrast, filmmakers such as Joseph Kosinski and Ruben Fleischer have comparatively lower average scores, indicating that their respective collections of work evoke a wide range of viewer responses. It is essential to remember that the graphed IMDB average ratings are subject to randomness and are influenced by the preferences of diverse viewers. By analyzing the data and scrutinizing the bar graph, one can gain insight into the average IMDB ratings of various directors and their overall perception. This information may be useful for spectators, directors, and producers evaluating prospective audience reactions to films created by multiple filmmakers.

Recognizing the subjective nature of ratings and the complex relationship between a film's commercial success and its overall quality is essential. The provided data provides an exhaustive overview of the average IMDB ratings assigned to film directors, as well as an illustration of the magnitude of variation within these ratings. The visual representation of the graph effectively highlights the disparities between filmmakers who consistently receive positive reviews and those who do not, highlighting the differences. This study provides valuable insights into how consumers perceive films based on filmmakers' involvement, thus providing practical benefits for film industry professionals and movie enthusiasts.

3.6 Relationship between Average IMDB Rating and Film Duration

The scatterplot graph illustrates the relationship between the average rating on IMDB and the film's running duration. The x-axis indicates the movie's length in minutes, while the y-axis shows the average rating given by IMDB. Each pixel on the graph corresponds to a unique film.

There is no apparent linear relationship between the duration of a film and its average rating on IMDB, as indicated by the scatterplot. A comparison between the two variables demonstrates this. Without a discernible pattern or trend, the points on the graph are dispersed. This indicates that the consensus rating of a film on IMDB is not solely based on its running time.



To see a more detailed chart, click [here](#)

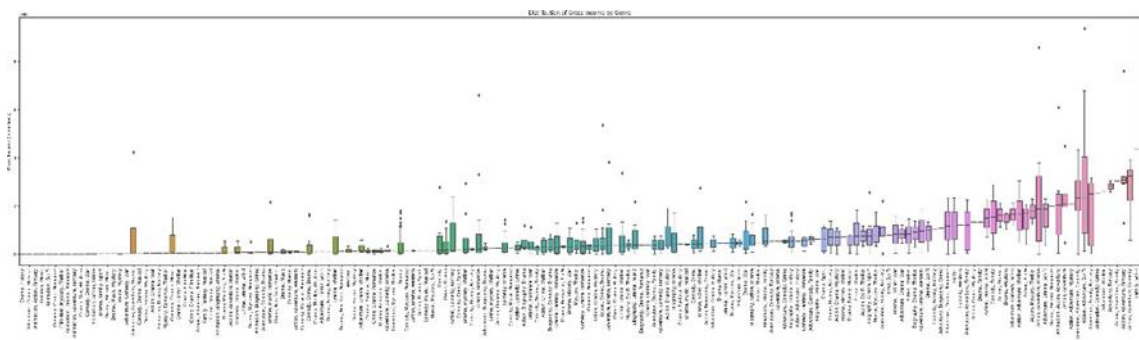
However, it is essential to remember that even movies with high average IMDB rankings might have wildly different playtime lengths depending on the film's specifics. Movies that obtain high ratings, often ranging from 8.0 to 8.5, are typically shorter, typically lasting between 80 and 90 minutes. This is because high ratings correlate to a faster viewing experience. Similarly, ratings tend to be higher for films with longer running periods, with the average being between 140 and 160 minutes.

On the other hand, it is important to remember that films with a range of running times typically have lower average ratings on IMDB. This suggests that the running time of a movie is one of many criteria considered when assigning a rating to it. The average rating on IMDB is decided by several elements, some of which include the quality of the plot, the actors' performances, the director of the movie, and the overall greatness of the picture.

3.7 Analysis of Movie Revenue Distribution by Genre

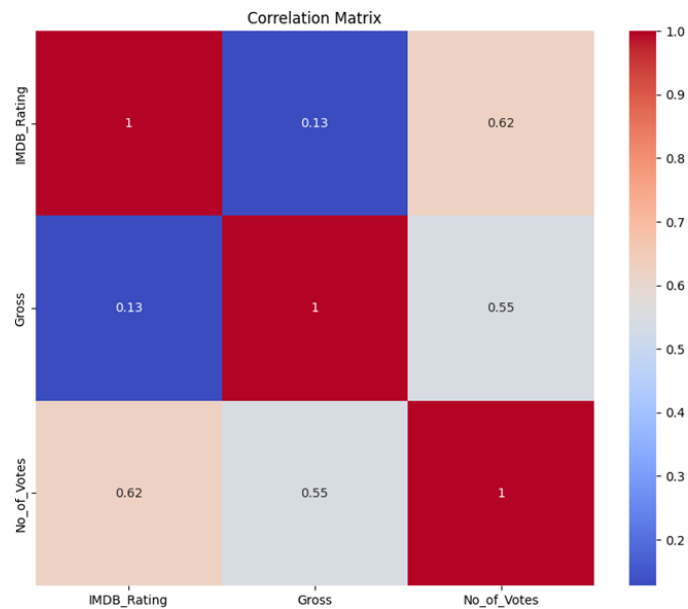
This study investigates the financial returns of various film genres. The presented data depicts the median and mean revenues for various film genres, such as drama, adventure, animation, comedy, and science fiction, among others. The median earnings for dramas are \$55,000, while the median earnings for animation and action films are \$151,086.

The dataset provides objective data regarding the financial performance of each genre, with a focus on the median revenues. This resource can help filmmakers, investors, and industry professionals make educated decisions regarding the production, distribution, and promotion of films. It is essential to monitor any profits that deviate significantly from the norm. The study does not provide guidance on how to manage these outliers effectively. Examining disparities in median earnings provides a conceptual framework for assessing the financial success of various film genres.



In order to see the boxplot graph more clearly, please click [here](#).

3.8 Correlation Analysis between Features



This study employed correlation analysis to investigate the relationships between IMDB rating, total revenue, and number of votes. The correlation coefficient of 0.13 between IMDB rating and cumulative income indicates a faint positive correlation, suggesting that films with higher ratings earn slightly more money. However, it should be noted that the intensity of this relationship needs to be increased to draw definitive conclusions.

There are stronger correlations between the IMDB rating, the number of votes (0.62), and the total income and the number of votes (0.55). The moderate correlations observed suggest a tendency for films with higher ratings and box office earnings to attract a larger number of votes, which indicates a higher level of popularity and audience engagement. The analysis reveals a moderate correlation between ratings and profits, with stronger correlations between ratings, earnings, and the number of ballots. This analysis illuminates the connections between film ratings, financial profitability, and audience participation.

Conclusion

4.1 Summary of Findings

The following is a summary of the results of the research that I have done on the Top 1000 IMDB Movie Dataset:

- The majority of the top 1000 films have IMDB ratings ranging from 7.6 to 8.5. Evaluations that surpass 8.5 stars are becoming less common.
- An analysis of historical trends shows that there have been fluctuations in the number of films released annually, reaching a peak in the early 2000s. This exemplifies the dynamic nature of the film industry.
- The IMDB rating of a film is not solely determined by its runtime. This suggests that there is no clear linear relationship between the two variables, indicating that ratings are influenced by various factors other than just the length of the film.
- Analyzing the median revenue by genre offers valuable insights into the financial performance. The animation, action, and fantasy subcategories are the highest-earning genres, with a median income of \$151,080.

4.2 Recommendations

Here, I offer recommendations for future researchers to enhance the quality of the study's results:

- It is advised that the dataset be expanded to cover a larger range of films than only the top 1000 films listed on IMDB in order to boost the data's representativeness.
- It is advised that other variables such as budgets, marketing costs, and awards be included to increase understanding of success aspects.
- Aside from quantitative ratings, sentiment analysis on reviews allows for a qualitative evaluation of audience responses.
- It is proposed that qualitative research approaches, such as surveys or interviews with both film experts and spectators, be used to supplement the quantitative data. In addition to numerical data, this technique will include contextual information.

- Examine longitudinal data spanning numerous years to determine trends and patterns within disciplines and the industry more closely.

Chapter 5

References

- Anggraeni, P., Mujiyanto, J., & Sofwan, A. (Year). The implementation of transposition translation procedures in English-Indonesian translation of epic movie subtitle. English Department, Faculty of Languages and Arts, Universitas Negeri Semarang, Indonesia.
- Coulson. (1978). Title of Coulson's work. (p. 622).
- Harshit Shankhdhar. (n.d.). IMDb Dataset of Top 1000 Movies and TV Shows. Retrieved from Kaggle: <https://www.kaggle.com/harshitshankhdhar/imdb-dataset-of-top-1000-movies-and-tv-shows>
- Hornby, A.S. (2010). Equivalence. In Oxford Advanced Learner's Dictionary (8th ed., p. 495). Oxford: Oxford University Press.
- Lorimer. (1995). Title of Lorimer's work. (p. 506). Microsoft Corporation. (2008). Microsoft Encarta.

Chapter 6

Appendices

- Import library package.

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

- Load dataset.

```
link = 'https://raw.githubusercontent.com/kevinkevinn/data-analyst-portfolio/main/Top%201000%20Movie/imdb_top_1000.csv'
data = pd.read_csv(link)
```

- Display the first 10 rows of dataset.

data.head(10)																	Python
	Poster_Link	Series_Title	Released_Year	Certificate	Runtime	Genre	IMDB_Rating	Overview	Meta_score	Director	Star1	Star2	Star3	Star4	No_of_Votes	Gross	
0	https://m.media-amazon.com/images/M/MV5BMDFkYT...	The Shawshank Redemption	1994	A	142 min	Drama	9.3	Two imprisoned men bond over a number of years...	80.0	Frank Darabont	Tim Robbins	Morgan Freeman	Bob Gunton	William Sadler	2143110	28,341,469	
1	https://m.media-amazon.com/images/M/MV5BM2MyNg...	The Godfather	1972	A	175 min	Crime, Drama	9.2	An organized crime dynasty's aging patriarch t...	100.0	Francis Ford Coppola	Marlon Brando	Al Pacino	James Caan	Diane Keaton	1620367	134,966,411	
2	https://m.media-amazon.com/images/M/MV5BM1MwNT...	The Dark Knight	2008	UA	152 min	Action, Crime, Drama	9.0	When the menace known as the Joker wreaks havoc...	84.0	Christopher Nolan	Christian Bale	Heath Ledger	Aaron Eckhart	Michael Caine	2303232	534,858,444	
3	https://m.media-amazon.com/images/M/MV5BMWwMG...	The Godfather: Part II	1974	A	202 min	Crime, Drama	9.0	The early life and career of Vito Corleone in ...	90.0	Francis Ford Coppola	Al Pacino	Robert De Niro	Robert Duvall	Diane Keaton	1129952	57,300,000	
4	https://m.media-amazon.com/images/M/MV5BMWU4N2...	12 Angry Men	1957	U	96 min	Crime, Drama	9.0	A jury holdout attempts to prevent a miscarria...	96.0	Sidney Lumet	Henry Fonda	Lee J. Cobb	Martin Balsam	John Fiedler	689845	4,360,000	
5	https://m.media-amazon.com/images/M/MV5BNzASZD...	The Lord of the Rings: The Return of the King	2003	U	201 min	Action, Adventure, Drama	8.9	Gandalf and Aragorn lead the World of Men agai...	94.0	Peter Jackson	Elijah Wood	Viggo Mortensen	Ian McKellen	Orlando Bloom	1642758	377,845,905	
6	https://m.media-amazon.com/images/M/MV5BNzNGNDMD...	Pulp Fiction	1994	A	154 min	Crime, Drama	8.9	The lives of two mob hitmen, a boxer, a gangst...	94.0	Quentin Tarantino	John Travolta	Uma Thurman	Samuel L. Jackson	Bruce Willis	1826188	107,928,762	
7	https://m.media-amazon.com/images/M/MV5BNDE4OT...	Schindler's List	1993	A	195 min	Biography, Drama, History	8.9	In German-occupied Poland during World War I...	94.0	Steven Spielberg	Liam Neeson	Ralph Fiennes	Ben Kingsley	Caroline Goodall	1213505	96,898,818	
8	https://m.media-amazon.com/images/M/MV5BMjMwMT...	Inception	2010	UA	148 min	Action, Adventure, Sci-Fi	8.8	A thief who steals corporate secrets through t...	74.0	Christopher Nolan	Leonardo DiCaprio	Joseph Gordon-Levitt	Elliot Page	Ken Watanabe	2067042	252,576,195	
9	https://m.media-amazon.com/images/M/MV5BMmEjNT...	Fight Club	1999	A	139 min	Drama	8.8	An insomniac office worker and a devil-may-ca...	66.0	David Fincher	Brad Pitt	Edward Norton	Meat Loaf	Zach Grenier	1854740	37,030,102	

- Explore dataset information.

```
data.info()
✓ 0.0s

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 16 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Poster_Link            1000 non-null   object
1   Series_Title           1000 non-null   object
2   Released_Year          1000 non-null   object
3   Certificate             899 non-null    object
4   Runtime                1000 non-null   object
5   Genre                  1000 non-null   object
6   IMDB_Rating            1000 non-null   float64
7   Overview               1000 non-null   object
8   Meta_score             843 non-null    float64
9   Director               1000 non-null   object
10  Star1                  1000 non-null   object
11  Star2                  1000 non-null   object
12  Star3                  1000 non-null   object
13  Star4                  1000 non-null   object
14  No_of_Votes            1000 non-null   int64
15  Gross                  831 non-null    object
dtypes: float64(2), int64(1), object(13)
memory usage: 125.1+ KB
```

- Convert the 'Runtime' column in the 'data' DataFrame by applying the 'convert_runtime' function to each value and assign the modified column back to 'data':

```
data = pd.DataFrame(data)

def convert_runtime(time):
    minutes = time.split(' ')[0]
    if len(minutes) == 2:
        minutes = '0' + minutes
    return minutes + ' min'

# Using a function to change the 'Runtime' column
data['Runtime'] = data['Runtime'].apply(convert_runtime)

data
```

- Count missing value:


```

missing_values = data.isnull().sum()
total_values = len(data)
missing_percentage = (missing_values / total_values) * 100

missing_data = pd.DataFrame({'Jumlah Missing Value': missing_values, 'Persentase Missing Value': missing_percentage})
missing_data['Persentase Missing Value'] = missing_data['Persentase Missing Value'].map("{:.1f}%".format)

print(missing_data)

```

✓ 0.0s

	Jumlah Missing Value	Persentase Missing Value
Poster_Link	0	0.0%
Series_Title	0	0.0%
Released_Year	0	0.0%
Certificate	101	10.1%
Runtime	0	0.0%
Genre	0	0.0%
IMDB_Rating	0	0.0%
Overview	0	0.0%
Meta_score	157	15.7%
Director	0	0.0%
Star1	0	0.0%
Star2	0	0.0%
Star3	0	0.0%
Star4	0	0.0%
No_of_Votes	0	0.0%
Gross	169	16.9%

- Drop missing value and create new dataset:

```

# Drop rows with missing values
data_cleaned = data.dropna()

# Create and save the cleaned dataset to a new file
data_cleaned.to_csv('cleaned_data.csv', index=False)

```

To view the dataset after cleaning, please [click](#)

- Read the new dataset:

```

data = pd.read_csv('cleaned_data.csv')
data.head()

```

✓ 0.0s

	Poster_Link	Series_Title	Released_Year	Certificate	Runtime	Genre	IMDB_Rating	Overview	Meta_score	Director	Star1	Star2	Star3	Star4	No_of_Votes	Gross
0	https://m.media-amazon.com/images/M/MV5BMjM0MTY1...	The Shawshank Redemption	1994	A	142 min	Drama	9.3	Two imprisoned men bond over a number of years...	80.0	Frank Darabont	Tim Robbins	Morgan Freeman	Bob Gunton	William Sadler	2343110	28,341,469
1	https://m.media-amazon.com/images/M/MV5BMjM0MTY1...	The Godfather	1972	A	175 min	Crime, Drama	9.2	An organized crime dynasty's aging patriarch L...	100.0	Francis Ford Coppola	Marlon Brando	Al Pacino	James Caan	Diane Keaton	1620367	134,965,411
2	https://m.media-amazon.com/images/M/MV5BMjM0MTY1...	The Dark Knight	2008	UA	152 min	Action, Crime, Drama	9.0	When the menace known as the Joker wreaks havo...	84.0	Christopher Nolan	Christian Bale	Heath Ledger	Aaron Eckhart	Michael Caine	2303232	534,858,444
3	https://m.media-amazon.com/images/M/MV5BMjM0MTY1...	The Godfather: Part II	1974	A	202 min	Crime, Drama	9.0	The early life and career of Vito Corleone in ...	90.0	Francis Ford Coppola	Al Pacino	Robert De Niro	Robert Duvall	Diane Keaton	1129952	57,300,000
4	https://m.media-amazon.com/images/M/MV5BMjM0MTY1...	12 Angry Men	1957	U	96 min	Crime, Drama	9.0	A jury holdout attempts to prevent a miscarria...	96.0	Sidney Lumet	Henry Fonda	Lee J. Cobb	Martin Boham	John Fiedler	689845	4,360,000

- Check for duplicate entries:

```

# Check for duplicate rows
duplicate_rows = data[data.duplicated()]

# Print the duplicate rows, if any
if len(duplicate_rows) > 0:
    print("Duplicate rows:")
    print(duplicate_rows)
else:
    print("No duplicate rows found.")

```

✓ 0.0s

No duplicate rows found.

- Customize the column data types:

```
# Convert 'Released_Year' to object (string) data type
data['Released_Year'] = data['Released_Year'].astype(str)

# Convert 'Certificate' to category data type
data['Certificate'] = data['Certificate'].astype('category')

# Convert 'Runtime' to string data type
data['Runtime'] = data['Runtime'].astype(str)

# Convert 'Gross' to string data type
data['Gross'] = data['Gross'].astype(str)

# Remove commas and other characters from the 'Gross' column values
data['Gross'] = data['Gross'].str.replace(',', '').str.replace('$', '')

# Convert 'Gross' to float data type
data['Gross'] = pd.to_numeric(data['Gross'], errors='coerce')

# Convert 'Meta_score' to float data type
data['Meta_score'] = data['Meta_score'].astype(float)

# Convert 'No_of_Votes' to numeric (integer) data type
data['No_of_Votes'] = pd.to_numeric(data['No_of_Votes'], errors='coerce')

# Convert 'IMDB_Rating' to string data type
data['IMDB_Rating'] = data['IMDB_Rating'].astype(str)
```

```
data.info()
✓ 0.0s

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 714 entries, 0 to 713
Data columns (total 16 columns):
#   Column          Non-Null Count  Dtype
---  ---
0   Poster_Link     714 non-null   object
1   Series_Title    714 non-null   object
2   Released_Year  714 non-null   object
3   Certificate      714 non-null   category
4   Runtime         714 non-null   object
5   Genre          714 non-null   object
6   IMDB_Rating     714 non-null   object
7   Overview        714 non-null   object
8   Meta_score      714 non-null   float64
9   Director        714 non-null   object
10  Star1           714 non-null   object
11  Star2           714 non-null   object
12  Star3           714 non-null   object
13  Star4           714 non-null   object
14  No_of_Votes     714 non-null   int64
15  Gross           714 non-null   int64
dtypes: category(1), float64(1), int64(2), object(12)
memory usage: 84.9+ KB
```

- Delete any unnecessary columns that are irrelevant to the analysis:

```
# Remove irrelevant columns
columns_to_drop = ['Poster_Link', 'Overview', 'Star1', 'Star2', 'Star3', 'Star4']
data.drop(columns_to_drop, axis=1, inplace=True)
data.info()

✓ 0.0s

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 714 entries, 0 to 713
Data columns (total 10 columns):
#   Column              Non-Null Count  Dtype
---  ---
0   Series_Title         714 non-null    object
1   Released_Year        714 non-null    object
2   Certificate          714 non-null    category
3   Runtime              714 non-null    object
4   Genre                714 non-null    object
5   IMDB_Rating          714 non-null    object
6   Meta_score           714 non-null    float64
7   Director             714 non-null    object
8   No_of_Votes          714 non-null    int64
9   Gross                714 non-null    int64
dtypes: category(1), float64(1), int64(2), object(6)
memory usage: 51.4+ KB
```

- Analysis data:
 - Distribution of IMDB Ratings:

```
# Convert 'IMDB_Rating' to float data type
data['IMDB_Rating'] = data['IMDB_Rating'].astype(float)

# Calculate the minimum and maximum IMDb ratings
min_rating = data['IMDB_Rating'].min()
max_rating = data['IMDB_Rating'].max()

# Add a small offset to the maximum rating
max_rating += 0.01

# Plot the distribution of IMDb ratings with adjusted range
plt.figure(figsize=(10, 6))
plt.hist(data['IMDB_Rating'], bins=20, edgecolor='black', range=(min_rating, max_rating))
plt.xlabel('IMDb Rating')
plt.ylabel('Frequency')
plt.title('Distribution of IMDb Ratings')

# Show the plot
plt.show()
```

- Trend Analysis of Movie based on released year:

```
# Filter out rows with 'PG' value in the 'Released_Year' column
data = data[data['Released_Year'] != 'PG']

# Analyze trends in release year after removing 'PG' rows
release_year_counts = data['Released_Year'].value_counts().sort_index()

plt.figure(figsize=(20, 6))
plt.plot(release_year_counts.index, release_year_counts.values)
plt.xlabel('Release Year')
plt.ylabel('Number of Movies')
plt.title('Trends in Release Year')
plt.xticks(rotation=90)
plt.show()
```

- Analysis of The Relationship between IMDB Rating and Movie Duration (minutes):

```
# Retrieve IMDb Rating and Movie Duration columns
imdb_ratings = data['IMDb_Rating']
durations = data['Runtime']

# Sort durations and imdb_ratings based on durations
sorted_indices = durations.argsort()
sorted_durations = durations.iloc[sorted_indices]
sorted_imdb_ratings = imdb_ratings.iloc[sorted_indices]

plt.figure(figsize=(40, 12))
plt.scatter(sorted_durations, sorted_imdb_ratings)
plt.xlabel('Runtime')
plt.ylabel('IMDb Rating')
plt.title('Relationship between Movie Duration and IMDb Rating')

# Memperbalik urutan nilai pada sumbu y
plt.gca().invert_yaxis()
plt.xticks(rotation=90)

# Sort the x-axis values in ascending order
plt.gca().set_xlim(sorted_durations.min(), sorted_durations.max())

plt.show()
```

- Analysis of Movie Revenue Distribution by Genre:

```
data['IMDb_Rating'] = pd.to_numeric(data['IMDb_Rating'], errors='coerce')
✓ 0.0s

# Calculate the average IMDb rating for each genre
genre_ratings = data.groupby('Genre')['IMDb_Rating'].mean().sort_values()

plt.figure(figsize=(12, len(genre_ratings) * 0.5)) # Resize images based on the number of genres
plt.barh(genre_ratings.index, genre_ratings.values) # Display all genres
plt.xlabel('Average IMDb Rating') # Change the x-axis label
plt.ylabel('Genre') # Change the y-axis label
plt.title('Average IMDb Ratings by Genre')

# Add value labels to each bar
for i, rating in enumerate(genre_ratings.values):
    plt.text(rating, i, str(round(rating, 2)), ha='left', va='center')

plt.tight_layout() # Adjust the layout to remove extra spacing
plt.show()
```

- Analysis of IMDb ratings of movie directors:

```
# Sort the director_ratings Series by values in descending order
director_ratings = data.groupby('Director')['IMDb_Rating'].mean().sort_values(ascending=False)

# Create a bar plot with the sorted data
plt.figure(figsize=(12, 70))
ax = sns.barplot(x=director_ratings.values, y=director_ratings.index, order=director_ratings.index)
plt.xlabel('Average IMDb Rating')
plt.ylabel('Director')
plt.title('Average IMDb Ratings by Director')

# Add value labels to each bar
for i, rating in enumerate(director_ratings.values):
    ax.annotate(f'{rating:.2f}', (rating, i), ha='left', va='center')

plt.show()
```

- Analysis of Movie revenue (in millions) by Genre:

```
# Group data by genre and calculate the median gross income for each genre
genre_gross_income = data.groupby('Genre')['Gross'].median().sort_values()

# Plot the distribution of gross income by genre with swapped x and y axes
plt.figure(figsize=(10, 45))
sns.boxplot(x=data['Gross'], y=data['Genre'], order=genre_gross_income.index)
plt.xlabel('Gross Income (in millions)')
plt.ylabel('Genre')
plt.title('Distribution of Gross Income by Genre')

# Rotate y-axis labels for better readability
plt.yticks(rotation=0)
plt.show()
```

- Correlation analysis between features:

```
# Select the relevant columns for correlation analysis
selected_columns = ['IMDB_Rating', 'Gross', 'Runtime', 'No_of_Votes']

# Convert 'Runtime' column to numeric by removing the 'min' string
data['Runtime'] = data['Runtime'].str.replace(' min', '').astype(float)

# Select the relevant columns for correlation analysis
selected_columns = ['IMDB_Rating', 'Gross', 'Runtime', 'No_of_Votes']

# Create a correlation matrix
correlation_matrix = data[selected_columns].corr()

# Plot the correlation matrix using a heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Matrix')
plt.show()
```