VISUALIZATION ANALYSIS REPORT

ANALYSIS OF MOVIE METADATA

SUBMITTED BY:

NAME: SHREYA S PATIL

SEMESTER: VI

ORGANIZATION: M S RAMAIAH INSTITUTE OF TECHNOLOGY

INTRODUCTION:

The visualization analysis report provides a comprehensive overview and exploration of the dataset through various visualizations. Visualization is a powerful tool for understanding data, revealing patterns, trends, and relationships that may not be immediately apparent from raw data alone. In this report, we leverage visualization techniques to gain insights into the dataset, identify correlations, distributions, and outliers, and explore the relationships between different features.

The report begins with an introduction to the dataset, highlighting its context, source, and main objectives. It then proceeds to present a series of visualizations, each designed to uncover different aspects of the data. These visualizations include histograms, scatter plots, box plots, pair plots, and correlation heatmaps, among others. Each visualization is accompanied by an analysis that interprets the insights gleaned from the visual representation of the data.

By employing a variety of visualization techniques, this report aims to provide a comprehensive understanding of the dataset, identify patterns and trends, and draw actionable insights that can inform decision-making and further analysis. Through clear and informative visualizations, readers can gain valuable insights into the dataset and its underlying characteristics, contributing to a deeper understanding of the data and its implications.

OBJECTIVE:

The objective of this analysis is to visually explore the relationships between features in the provided movie metadata dataset and gain insights into their distributions and correlations.

INTRODUCTION TO THE DATASET:

The dataset under consideration contains information about various movies, including both numerical and categorical attributes. Each row in the dataset represents a unique movie, and each column provides specific details about that movie. Below is a brief overview of the attributes present in the dataset:

1. Color: Indicates whether the movie is coloured or black and white.
2. Director Name: Name of the director of the movie.
3. Num Critic for Reviews: Number of critic reviews for the movie.
4. Duration: Duration of the movie in minutes.
5. Director Facebook Likes: Number of Facebook likes for the director.
6. Actor 3 Facebook Likes: Number of Facebook likes for the third actor.
7. Actor 2 Name: Name of the second actor in the movie.
8. Actor 1 Facebook Likes: Number of Facebook likes for the first actor.
9. Gross: Gross earnings of the movie.
10. Genres: Genres of the movie, separated by '|' if multiple genres.
11. Actor 1 Name: Name of the first actor in the movie.
12. Movie Title: Title of the movie.
13. Num Voted Users: Number of users who voted for the movie.
14. Cast Total Facebook Likes: Total Facebook likes for the cast of the movie.
15. Actor 3 Name: Name of the third actor in the movie.
16. Facenumber in Poster: Number of faces present in the movie poster.
17. Plot Keywords: Keywords describing the plot of the movie.
18. Movie IMDb Link: IMDb link for the movie.
19. Num User for Reviews: Number of user reviews for the movie.
20. Language: Language of the movie.
21. Country: Country where the movie was produced.
22. Content Rating: Content rating of the movie.
23. Budget: Budget of the movie.
24. Title Year: Year of release of the movie.
25. Actor 2 Facebook Likes: Number of Facebook likes for the second actor.
26. IMDb Score: IMDb score rating of the movie.
27. Aspect Ratio: Aspect ratio of the movie.
28. Movie Facebook Likes: Number of Facebook likes for the movie.

This dataset provides a comprehensive set of attributes that can be used for various analyses, including exploring trends in movie preferences, analyzing the impact of directors and actors on movie success, and understanding the factors influencing IMDb scores and gross earnings of movies. In the following sections, we will perform visualizations and analyses to gain insights into the dataset.

VISUALIZATION METHODS:

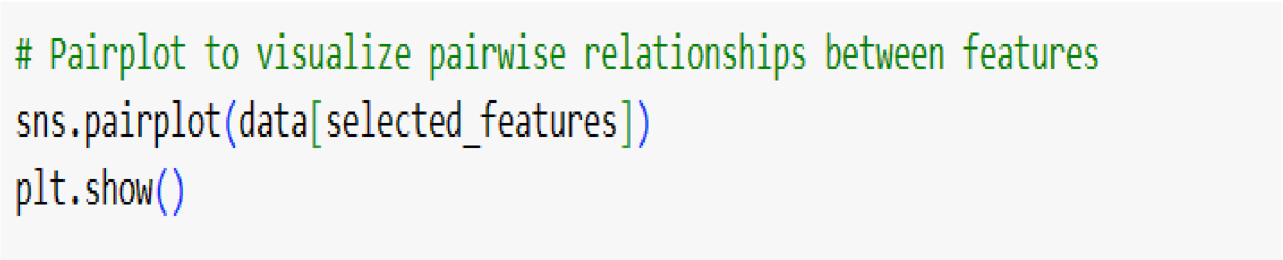
Pairplot:

A pairplot is a type of visualization provided by the seaborn library in Python. It displays pairwise relationships between variables in a dataset. For every pair of features, a scatterplot is drawn to represent the relationship between them. Along the diagonal, histograms are displayed for each individual feature.

Pairplot can provide:

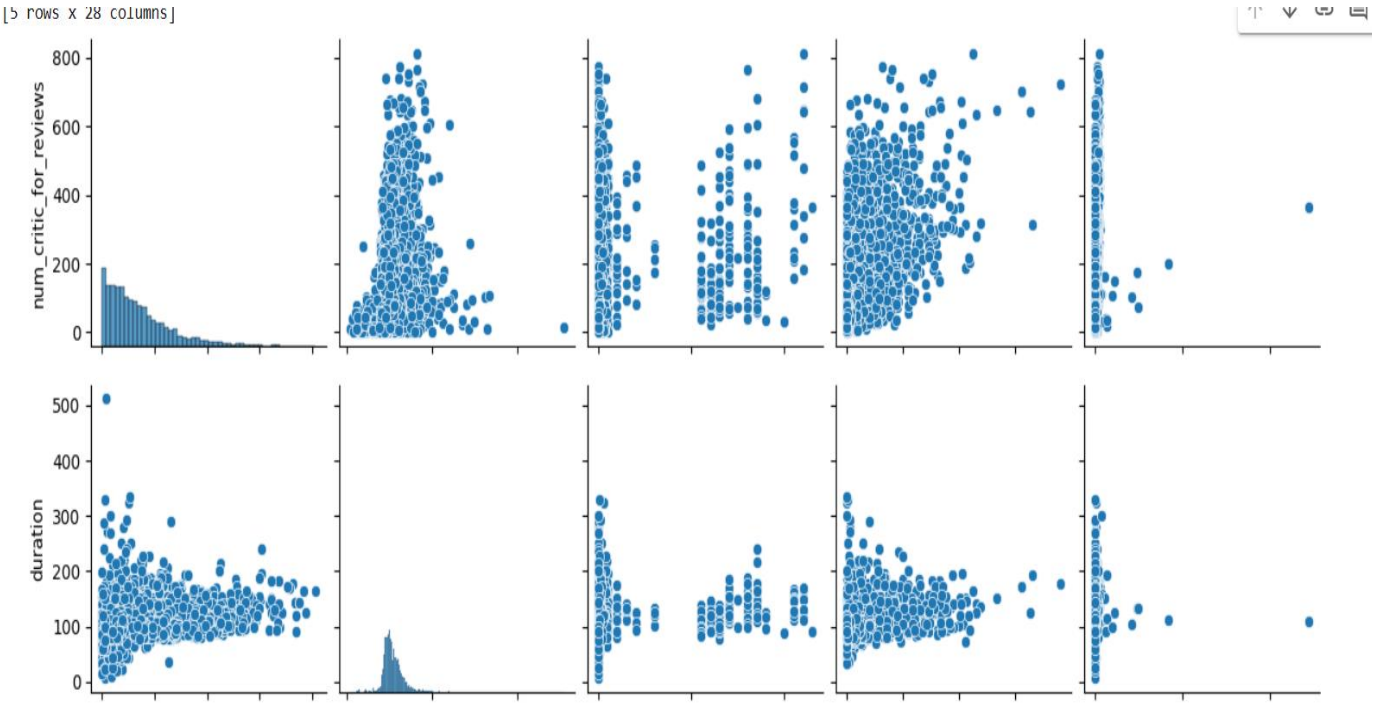
- Scatterplots: Pairplots display scatterplots for each pair of numerical features. These scatterplots help visualize the relationship between two variables. They can reveal patterns such as linear relationships, non-linear relationships, clusters, outliers, and correlations.
- Histograms: Along the diagonal of the pairplot, histograms are displayed for each individual feature. Histograms show the distribution of values for each feature. They help understand the spread, central tendency, and skewness of the data. By examining these histograms, you can gain insights into the distribution of each variable and identify any potential outliers or unusual patterns.
- Kernel Density Estimates (KDE): KDE plots are overlaid on the histograms to provide a smooth estimate of the probability density function of each feature. KDE plots offer additional insights into the distribution of data and can reveal more subtle patterns compared to histograms.
- Overall, pairplots are useful for gaining a comprehensive understanding of the relationships between multiple variables in a dataset. They allow for quick visual exploration and can help identify interesting patterns and relationships that may warrant further investigation.

CODE:



```
# Pairplot to visualize pairwise relationships between features
sns.pairplot(data[selected_features])
plt.show()
```

RESULT:



```
[5 rows x 28 columns]
```

INSIGHTS:

The pairplot visualization provides scatterplots for each pair of numerical features along with histograms along the diagonal. Let's analyze the histograms:

Histograms on the Diagonal:

- In the histogram plots along the diagonal, each individual feature is visualized.
- For instance, in the first row and first column, the histogram plot indicates that the feature is not normally distributed and is left-skewed.
- Similarly, in the graph present in the second row and second column, the distribution is slightly left-skewed, indicating that it is closer to a normal distribution.
- These observations provide insights into the distribution of individual numerical features, which can be useful for understanding their underlying characteristics and potential transformations needed for further analysis.

CORRELATION MATRIX HEATMAP:

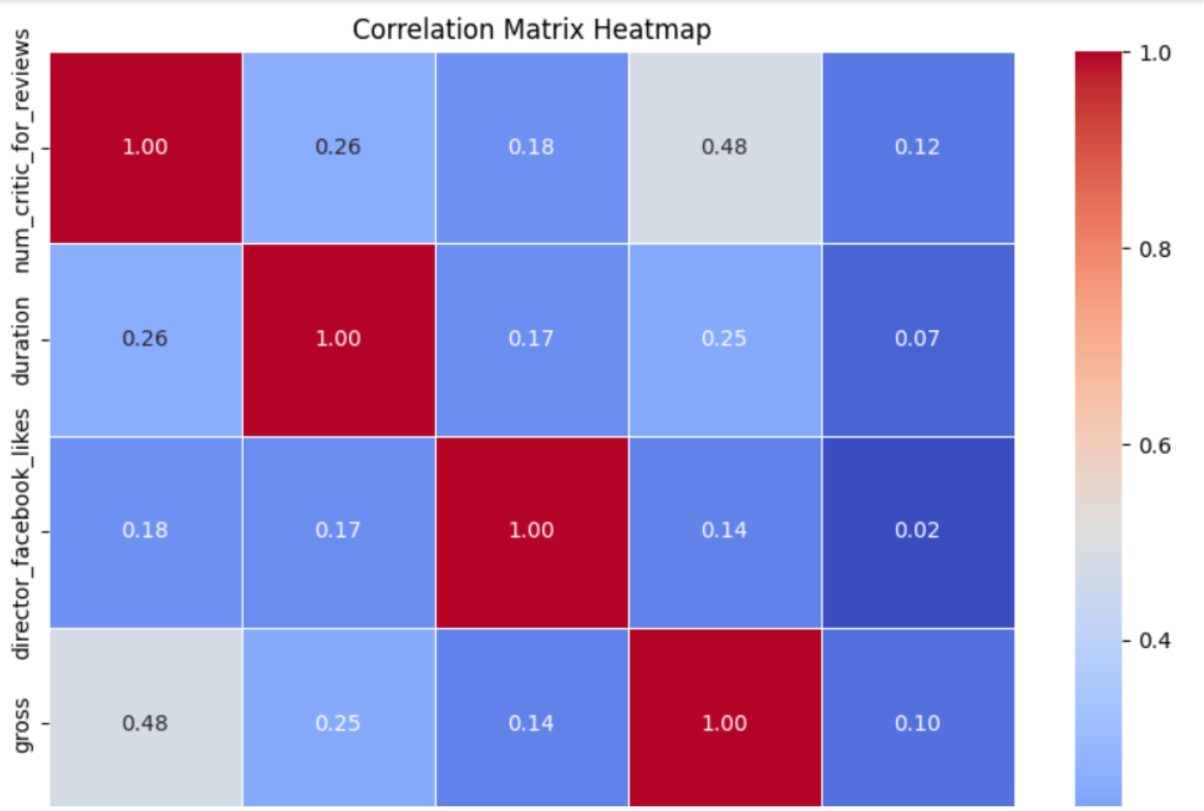
The correlation matrix heatmap provides insights into the relationships between pairs of variables in our dataset. In this visualization, each cell represents the correlation coefficient between two variables. Here's what we can observe:

- Strong Positive Correlation:
  - On the diagonal of the heatmap, we observe strong positive correlation coefficients, indicated by a value close to 1.
  - This is expected since the diagonal elements represent the correlation of each variable with itself, which is always perfect positive correlation.
- Interpreting Colors:
  - White Color: Indicates no correlation between the two features, as seen in the cells away from the diagonal.
  - Light Blue to Blue Color: Represents positive correlation, with varying strengths. Lighter shades denote weaker correlations, while darker shades indicate stronger correlations.
  - Other Colors: Different shades represent varying degrees of correlation, with warmer colors (e.g., red) indicating positive correlation and cooler colors (e.g., green) indicating negative correlation.

CODE:

```python
# Correlation matrix heatmap
correlation_matrix = data[selected_features].corr()
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f", linewidths=0.5)
plt.title('Correlation Matrix Heatmap')
plt.show()
```

RESULT:



INSIGHTS:

Positive correlation implies that as one feature increases, the other tends to increase as well. Conversely, negative correlation suggests that as one feature increases, the other tends to decrease.

Diagonal Elements:

- The diagonal elements exhibit a correlation coefficient of 1, reflecting the perfect positive correlation of each variable with itself.

By examining the correlation matrix heatmap, we can identify potential patterns and relationships between features, aiding in further analysis and model development.

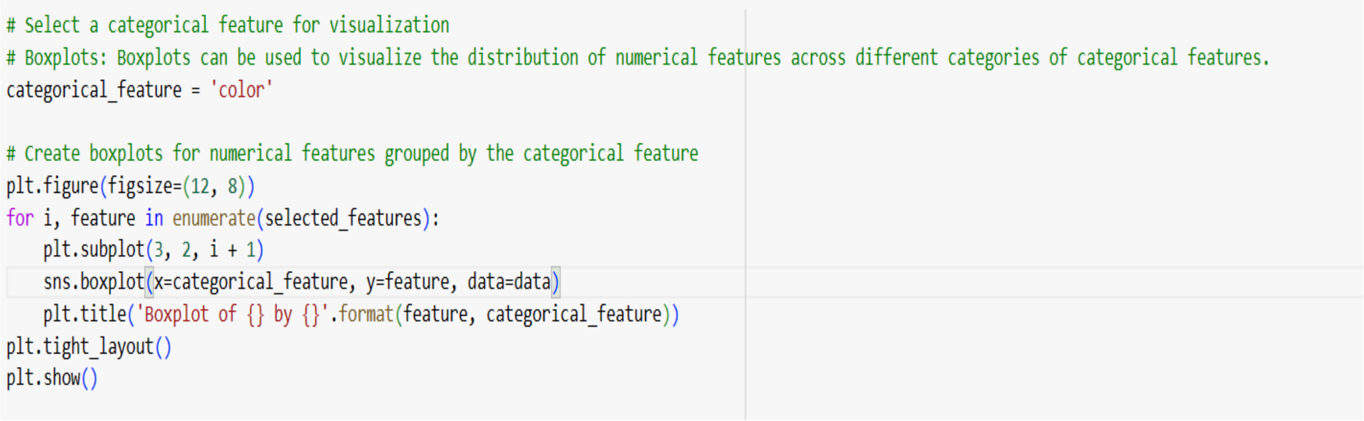
BOXPLOT:

A boxplot is a standardized way of visualizing the distribution of numerical data through quartiles. It displays the distribution of data based on five summary statistics: the minimum, first quartile (Q1), median (Q2), third quartile (Q3), and maximum.

Here's how a boxplot works and what each part represents:

- Minimum and Maximum: The whiskers (lines extending from the box) represent the minimum and maximum values of the data. They indicate the range of the data distribution.
- Quartiles (Q1, Q2, Q3): The box is divided into three parts by two horizontal lines. The bottom edge of the box represents the first quartile (Q1), which is the 25th percentile of the data. The line inside the box represents the median (Q2), which is the 50th percentile. The top edge of the box represents the third quartile (Q3), which is the 75th percentile.
- Interquartile Range (IQR): The length of the box represents the interquartile range (IQR), which is the range covered by the middle 50% of the data (from Q1 to Q3). It provides a measure of the spread of the data.
- Outliers: Data points that fall below the lower whisker (minimum) or above the upper whisker (maximum) are considered outliers and are plotted individually as points.

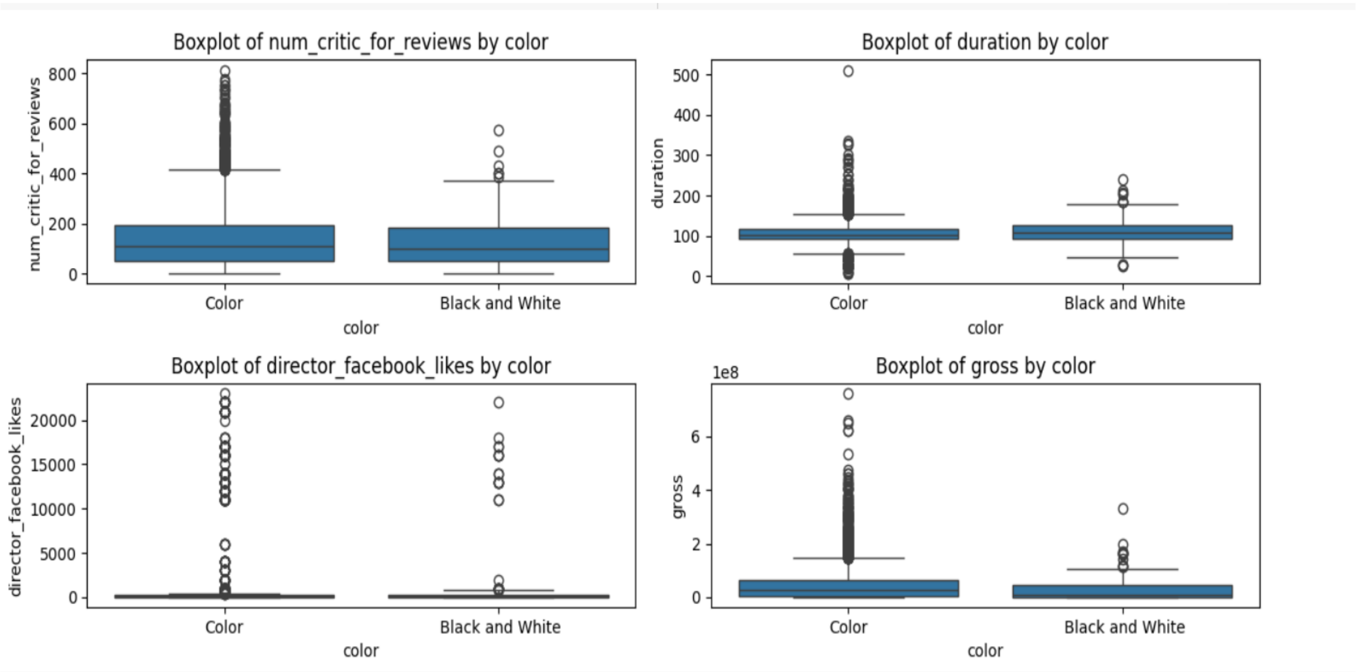
CODE:



```
# Select a categorical feature for visualization
# Boxplots: Boxplots can be used to visualize the distribution of numerical features across different categories of categorical features.
categorical_feature = 'color'

# Create boxplots for numerical features grouped by the categorical feature
plt.figure(figsize=(12, 8))
for i, feature in enumerate(selected_features):
    plt.subplot(3, 2, i + 1)
    sns.boxplot(x=categorical_feature, y=feature, data=data)
    plt.title('Boxplot of {} by {}'.format(feature, categorical_feature))
plt.tight_layout()
plt.show()
```

RESULT:



INSIGHTS:

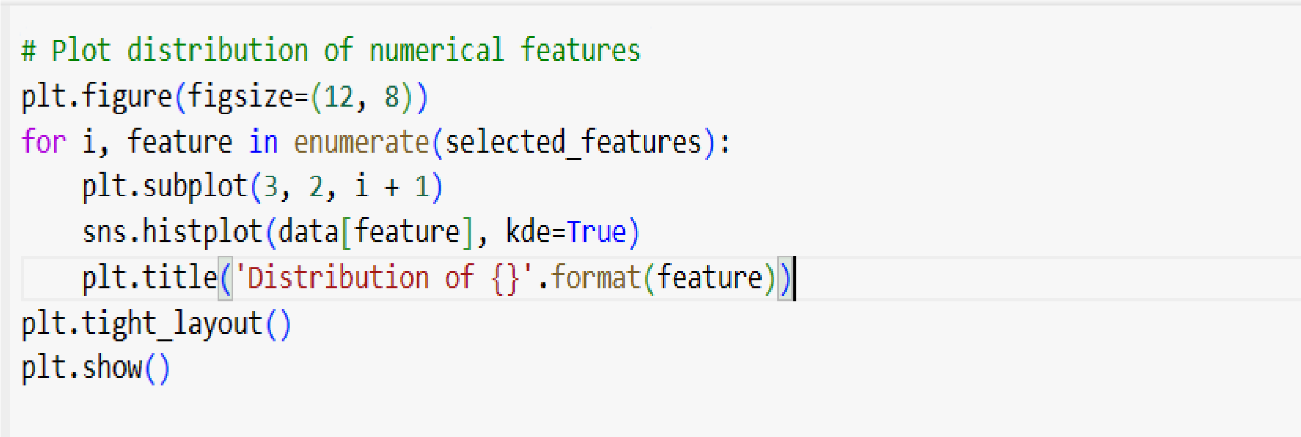
- Comparison of Feature Distributions: By creating separate boxplots for each numerical feature grouped by the categorical feature ('color'), we can compare the distributions of these features across different categories of 'color'. This comparison helps in understanding if there are any notable differences or patterns in the distribution of each feature based on the color category.
- Identification of Outliers: Boxplots allow us to identify outliers within each category of the categorical feature. Outliers are data points that fall significantly below or above the whiskers of the boxplot. Detecting outliers can be important for understanding the presence of unusual data points within specific categories.
- Understanding Central Tendency and Spread: The central line within each box represents the median (Q2) of the data distribution for each feature within a category of 'color'. The length of the box (interquartile range) provides insights into the spread of the data within each category. Comparing the lengths of the boxes across different categories helps in understanding variations in data spread.
- Observing Skewness and Symmetry: Boxplots also provide visual cues about the skewness and symmetry of the data distribution within each category. For instance, if one box appears longer on one side compared to the other, it indicates skewness in the data distribution.
- Identifying Potential Relationships: By analyzing the boxplots collectively, we can identify potential relationships or differences between the numerical features and the categorical feature ('color'). These insights can guide further analysis or decision-making processes, such as feature selection or model building.

DISTRIBUTION PLOT:

Distribution plots, also known as histograms, provide visualizations of the distribution of data within a dataset. Here's what they offer:

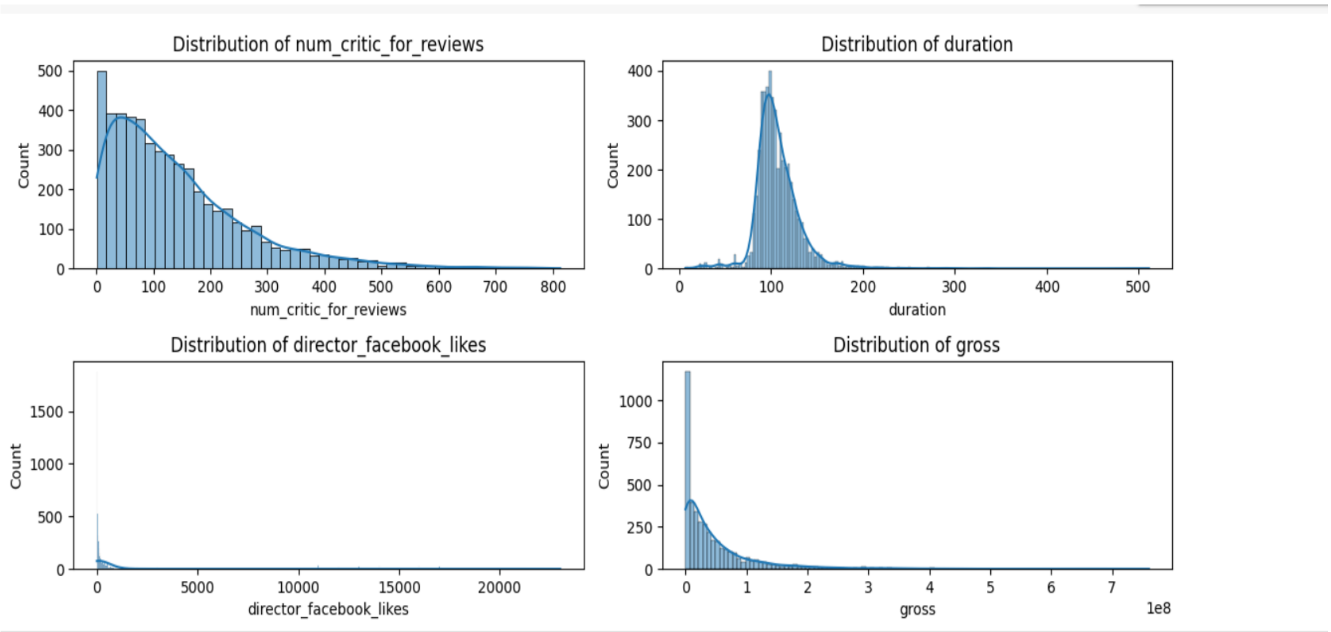
- Data Distribution: Distribution plots show how data points are distributed across different values or intervals of a numerical variable.
- Frequency of Values: They illustrate the frequency or count of data points falling within each bin or interval.
- Central Tendency: They give insights into the central tendency of the data, such as the mean, median, and mode, based on the shape of the distribution.
- Spread and Variation: They indicate the spread and variation of the data, including measures like range, variance, and standard deviation.
- Skewness and Kurtosis: Distribution plots can reveal the skewness (asymmetry) and kurtosis (peakedness) of the data distribution.
- Outlier Detection: They help identify potential outliers or anomalies in the dataset.

CODE:



```
# Plot distribution of numerical features
plt.figure(figsize=(12, 8))
for i, feature in enumerate(selected_features):
    plt.subplot(3, 2, i + 1)
    sns.histplot(data[feature], kde=True)
    plt.title('Distribution of {}'.format(feature))
plt.tight_layout()
plt.show()
```

RESULT:



INSIGHTS:

In the first graph, we observe that the distribution is not normally distributed and exhibits left skewness. This skewness indicates that the data is concentrated towards the higher end of the scale, with a tail extending towards the lower values. To address this skewness, we can apply transformations such as the inverse logarithm to normalize the data distribution. Therefore, distribution plots play a crucial role in identifying and addressing skewness in the data, facilitating normalization and enhancing the reliability of statistical analyses.

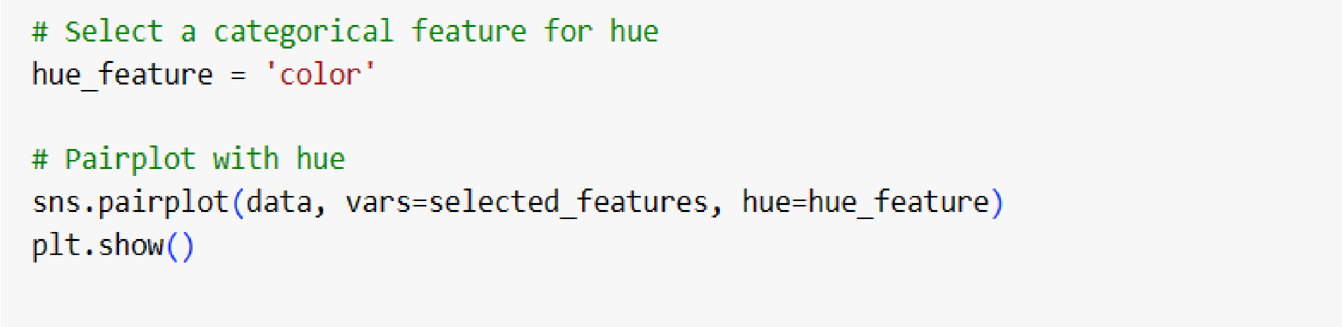
PAIRPLOT FOR CATEGORICAL FEATURES AS HUE:

When using a categorical feature as the hue in a pairplot, each unique category of the categorical feature is assigned a different color, allowing for visual differentiation between data points belonging to different categories. This enables the exploration of relationships between numerical features while considering the categorical variable as a factor.

Here's what the pairplot with a categorical feature as the hue provides:

- Visualizing relationships: The pairplot creates scatterplots for each pair of numerical features. However, with the addition of hue, data points are colored differently based on the categories of the chosen categorical feature.
- Understanding interactions: By incorporating hue, the pairplot facilitates the identification of how the relationships between numerical features vary across different categories of the categorical variable. This helps in understanding if there are any distinct patterns or interactions based on the categorical feature.
- Comparing distributions: It also allows for the comparison of the distributions of numerical features across different categories of the categorical variable. This can reveal insights into how the numerical variables behave within each category of the categorical feature.

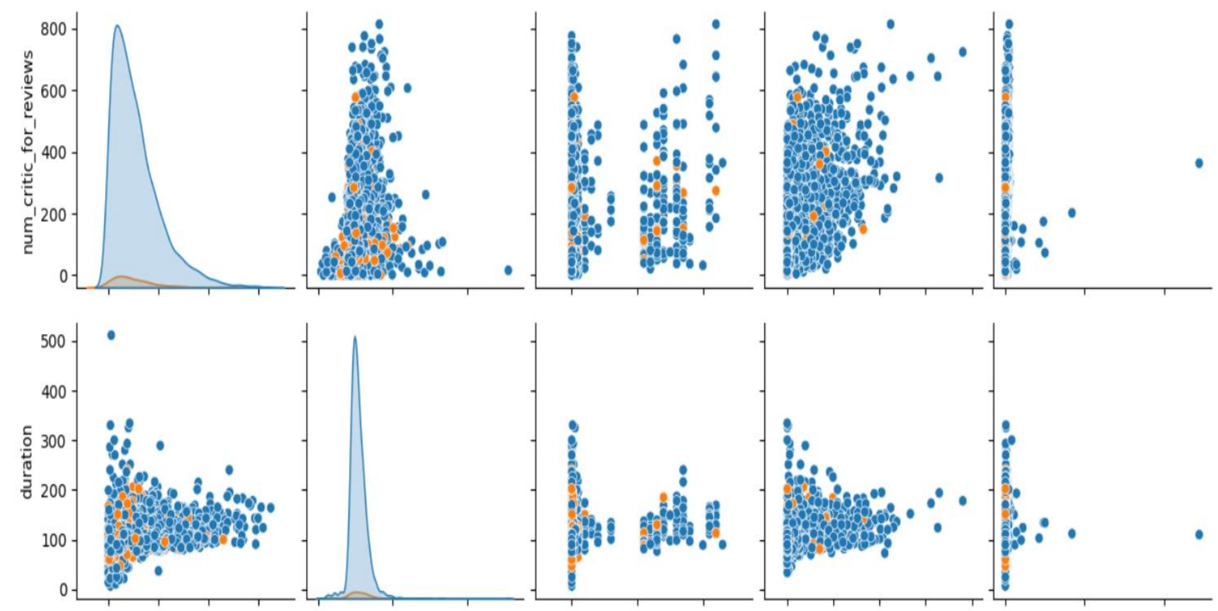
CODE:



```
# Select a categorical feature for hue
hue_feature = 'color'

# Pairplot with hue
sns.pairplot(data, vars=selected_features, hue=hue_feature)
plt.show()
```

RESULT:



INSIGHTS:

- Distribution Variation: We can observe how the distribution of numerical features varies across different categories of the categorical feature ('color'). For example, we can compare the distributions of features like 'num_critic_for_reviews', 'duration', 'director_facebook_likes', etc., across different movie colors.
- Pattern Differences: We can identify any patterns or trends in the relationships between numerical features based on the categorical feature ('color'). This could help in understanding how certain movie colors might influence various aspects of the movie metadata.
- Outlier Detection: It can also aid in detecting any outliers or anomalies in the numerical features within each category of the categorical feature ('color'). This could provide insights into potential data quality issues or interesting outliers specific to certain movie colors.

CONCLUSION:

In this visualization analysis report, we explored various visualizations and insights derived from a dataset containing information about movies. Through visualizations such as histograms, boxplots, pairplots, and correlation matrices, we gained valuable insights into the relationships between different features and their distributions.

Here are some key conclusions drawn from the visualization analysis:

- Distribution of Numerical Features:
  We observed that some numerical features, such as budget and gross earnings, are right-skewed, indicating that a few movies have significantly higher values for these attributes.
  Other features, such as duration and IMDb scores, exhibit more normal distributions, with data clustered around the mean.
- Correlation Analysis:
  By visualizing the correlation matrix heatmap, we identified strong positive correlations between certain pairs of numerical features, such as budget and gross earnings.
  Features like actor Facebook likes and movie Facebook likes showed relatively weak correlations with other numerical attributes.
- Relationship between Numerical and Categorical Features:
  Using boxplots grouped by categorical features like colour, we explored the distribution of numerical features across different categories.
  For example, we observed variations in gross earnings across different movie colours, with some colours having higher median earnings compared to others.
- Pairplot with Categorical Feature as Hue:
  By incorporating a categorical feature as the hue in pairplots, we visualized how the distribution of numerical features varies based on different categories.
  This allowed us to identify patterns and relationships between numerical attributes while considering the categorical aspect.
- Insights from Pairplot with Hue:
  The pairplot with color as the hue provided insights into how numerical features vary across different movie colours.
  For instance, we observed differences in IMDb scores and budget distributions for movies of different colours.

Overall, the visualization analysis provided valuable insights into the dataset, allowing us to understand the distributions, relationships, and trends present in movie attributes. These insights can be leveraged for further analysis and decision-making in the movie industry, such as understanding audience preferences, optimizing budgets, and predicting movie success.