Movie Data Analysis

For this project, you will use exploratory data analysis to generate insights for a business stakeholder.

```
#Begin by importing all the essential libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

#  loading the data sets into dataframes

imdb_title_basics = pd.read_csv(r"C:\Users\Mau\Documents\Flatiron\dsc-data-science-env-config\Movie Review\imdb.title.basics.csv")
imdb_title_ratings = pd.read_csv(r"C:\Users\Mau\Documents\Flatiron\dsc-data-science-env-config\Movie Review\imdb.title.ratings.csv")
bom_movie_gross = pd.read_csv(r"C:\Users\Mau\Documents\Flatiron\dsc-data-science-env-config\Movie Review\bom.movie_gross.csv")


# Lets now merge our dataframes based on their unique identifiers

# Merge imdb.title.basics and imdb.title.ratings based on tconst
imdb_merged = pd.merge(imdb_title_basics, imdb_title_ratings,
on='tconst', how='inner')

# Merge the datasets
merged_data = pd.merge(imdb_merged, bom_movie_gross,
left_on='original_title', right_on='title', how='inner')

# Display the merged dataset
df = merged_data
df
```

```
         tconst                     primary_title  \
0     tt0315642                             Wazir
1     tt0337692                       On the Road
2     tt4339118                       On the Road
3     tt5647250                       On the Road
4     tt0359950  The Secret Life of Walter Mitty
...         ...                               ...
2443  tt8097306                     Nobody's Fool
2444  tt8108198                         Andhadhun
2445  tt8427036                    Helicopter Eela
2446  tt8549902  Oolong Courtyard: KungFu School
2447  tt9151704        Burn the Stage: The Movie
```

```
                        original_title  start_year  runtime_minutes  \
0                                Wazir        2016            103.0
1                          On the Road        2012            124.0
2                          On the Road        2014             89.0
3                          On the Road        2016            121.0
4     The Secret Life of Walter Mitty        2013            114.0
...                                ...         ...              ...
2443                     Nobody's Fool        2018            110.0
2444                         Andhadhun        2018            139.0
2445                   Helicopter Eela        2018            135.0
2446                  Oolong Courtyard        2018            103.0
2447        Burn the Stage: The Movie        2018             84.0

                        genres  averagerating  numvotes  \
0          Action,Crime,Drama            7.1     15378
1     Adventure,Drama,Romance            6.1     37886
2                       Drama            6.0         6
3                       Drama            5.7       127
4      Adventure,Comedy,Drama            7.3    275300
...                       ...            ...       ...
2443     Comedy,Drama,Romance            4.6      3618
2444           Crime,Thriller            8.5     43409
2445                    Drama            5.4       673
2446                   Comedy            4.6        61
2447        Documentary,Music            8.8      2067

                                title     studio  domestic_gross  \
0                                Wazir    Relbig.       1100000.0
1                          On the Road        IFC        744000.0
2                          On the Road        IFC        744000.0
3                          On the Road        IFC        744000.0
4     The Secret Life of Walter Mitty        Fox      58200000.0
...                                ...        ...             ...
2443                     Nobody's Fool       Par.      31700000.0
2444                         Andhadhun       Eros       1200000.0
2445                   Helicopter Eela       Eros         72000.0
2446                  Oolong Courtyard         CL         37700.0
2447        Burn the Stage: The Movie  Trafalgar       4200000.0

     foreign_gross  year
0              NaN  2016
1          8000000  2012
2          8000000  2012
3          8000000  2012
4        129900000  2013
...            ...   ...
2443       1800000  2018
2444           NaN  2018
2445           NaN  2018
2446           NaN  2018
```

```
2447      16100000  2018

[2448 rows x 13 columns]
```

Data Inspection

```
#Display the first 5 and last five rows of the merged data frame
df.head()
```

```
      tconst                    primary_title  \
0  tt0315642                            Wazir
1  tt0337692                      On the Road
2  tt4339118                      On the Road
3  tt5647250                      On the Road
4  tt0359950  The Secret Life of Walter Mitty

                    original_title  start_year  runtime_minutes  \
0                            Wazir        2016            103.0
1                      On the Road        2012            124.0
2                      On the Road        2014             89.0
3                      On the Road        2016            121.0
4  The Secret Life of Walter Mitty        2013            114.0

                    genres  averagerating  numvotes  \
0       Action,Crime,Drama            7.1     15378
1  Adventure,Drama,Romance            6.1     37886
2                    Drama            6.0         6
3                    Drama            5.7       127
4   Adventure,Comedy,Drama            7.3    275300

                             title   studio  domestic_gross
foreign_gross  \
0                            Wazir  Relbig.       1100000.0
NaN
1                      On the Road      IFC        744000.0
8000000
2                      On the Road      IFC        744000.0
8000000
3                      On the Road      IFC        744000.0
8000000
4  The Secret Life of Walter Mitty      Fox      58200000.0
129900000

   year
0  2016
1  2012
2  2012
```

```
3  2012
4  2013
```

```
df.tail()
```

```
         tconst                    primary_title
original_title  \
2443  tt8097306                    Nobody's Fool              Nobody's
Fool
2444  tt8108198                        Andhadhun
Andhadhun
2445  tt8427036                  Helicopter Eela            Helicopter
Eela
2446  tt8549902  Oolong Courtyard: KungFu School           Oolong
Courtyard
2447  tt9151704        Burn the Stage: The Movie  Burn the Stage: The
Movie

      start_year  runtime_minutes                genres  averagerating
\
2443        2018            110.0  Comedy,Drama,Romance            4.6

2444        2018            139.0        Crime,Thriller            8.5

2445        2018            135.0                 Drama            5.4

2446        2018            103.0                Comedy            4.6

2447        2018             84.0     Documentary,Music            8.8


      numvotes                      title     studio
domestic_gross  \
2443      3618              Nobody's Fool       Par.      31700000.0

2444     43409                  Andhadhun       Eros       1200000.0

2445       673            Helicopter Eela       Eros         72000.0

2446        61           Oolong Courtyard         CL         37700.0

2447      2067  Burn the Stage: The Movie  Trafalgar       4200000.0


     foreign_gross  year
2443       1800000  2018
2444           NaN  2018
2445           NaN  2018
2446           NaN  2018
2447      16100000  2018
```

```python
#Lets derive some basic information regarding our data frame
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2448 entries, 0 to 2447
Data columns (total 13 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   tconst           2448 non-null   object 
 1   primary_title    2448 non-null   object 
 2   original_title   2448 non-null   object 
 3   start_year       2448 non-null   int64  
 4   runtime_minutes  2403 non-null   float64
 5   genres           2444 non-null   object 
 6   averagerating    2448 non-null   float64
 7   numvotes         2448 non-null   int64  
 8   title            2448 non-null   object 
 9   studio           2445 non-null   object 
 10  domestic_gross   2430 non-null   float64
 11  foreign_gross    1574 non-null   object 
 12  year             2448 non-null   int64  
dtypes: float64(3), int64(3), object(7)
memory usage: 248.8+ KB
```

The DataFrame has a total of 2448 entries and 13 columns. It includes various attributes related to movies, such as titles, release years, genres, ratings, box office earnings, and other relevant details. We can see that some data is missing. The columns have ranging data types from float, integers, and object

```python
# Check for the missing values
missing_values_sum = df.isnull().sum()
missing_values_sum
```

```
tconst               0
primary_title        0
original_title       0
start_year           0
runtime_minutes     45
genres               4
averagerating        0
numvotes             0
title                0
studio               3
domestic_gross      18
foreign_gross      874
year                 0
dtype: int64
```

There are a number of missing values such as; runtime_minutes = 45 genres = 4 studio = 3 domestic_gross = 18 foreign_gross = 874 .I am considering dropping some of these column that have no direct impact on our variable

```
# Replace the missing values in runtime_minutes with the median value
median_runtime_value = df['runtime_minutes'].median()

# Replacing
df['runtime_minutes'].fillna(median_runtime_value, inplace = True)


#Replace the genre and studio missing values with a placeholder
df['genres'].fillna('Unknown', inplace = True)
df['studio'].fillna('Unknown', inplace = True)

# Lets check if the missing values have been filled
df.isnull().sum()
```

```
tconst               0
primary_title        0
original_title       0
start_year           0
runtime_minutes      0
genres               0
averagerating        0
numvotes             0
title                0
studio               0
domestic_gross      18
foreign_gross      874
year                 0
dtype: int64
```

```
# Fill the null values in domestic gross column with median

median_domestic_gross_value =
df['domestic_gross'].astype(float).median()


df['domestic_gross'].fillna(median_domestic_gross_value, inplace=True)

# Drop 'foreign_gross' column

df.drop('foreign_gross', axis=1, inplace=True)

# Check the changes
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2448 entries, 0 to 2447
Data columns (total 12 columns):
```

```
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   tconst           2448 non-null   object 
 1   primary_title    2448 non-null   object 
 2   original_title   2448 non-null   object 
 3   start_year       2448 non-null   int64  
 4   runtime_minutes  2403 non-null   float64
 5   genres           2444 non-null   object 
 6   averagerating    2448 non-null   float64
 7   numvotes         2448 non-null   int64  
 8   title            2448 non-null   object 
 9   studio           2445 non-null   object 
 10  domestic_gross   2448 non-null   float64
 11  year             2448 non-null   int64  
dtypes: float64(3), int64(3), object(6)
memory usage: 229.6+ KB
```

```
#Cleaned dataframe
df
```

```
         tconst                    primary_title  \
0     tt0315642                            Wazir   
1     tt0337692                      On the Road   
2     tt4339118                      On the Road   
3     tt5647250                      On the Road   
4     tt0359950  The Secret Life of Walter Mitty   
...         ...                              ...   
2443  tt8097306                    Nobody's Fool   
2444  tt8108198                        Andhadhun   
2445  tt8427036                  Helicopter Eela   
2446  tt8549902  Oolong Courtyard: KungFu School   
2447  tt9151704        Burn the Stage: The Movie   

                       original_title  start_year  runtime_minutes  \
0                               Wazir        2016            103.0   
1                         On the Road        2012            124.0   
2                         On the Road        2014             89.0   
3                         On the Road        2016            121.0   
4     The Secret Life of Walter Mitty        2013            114.0   
...                               ...         ...              ...   
2443                    Nobody's Fool        2018            110.0   
2444                        Andhadhun        2018            139.0   
2445                  Helicopter Eela        2018            135.0   
2446                 Oolong Courtyard        2018            103.0   
2447        Burn the Stage: The Movie        2018             84.0   

                       genres  averagerating  numvotes  \
0          Action,Crime,Drama            7.1     15378   
1     Adventure,Drama,Romance            6.1     37886   
2                       Drama            6.0         6   
```

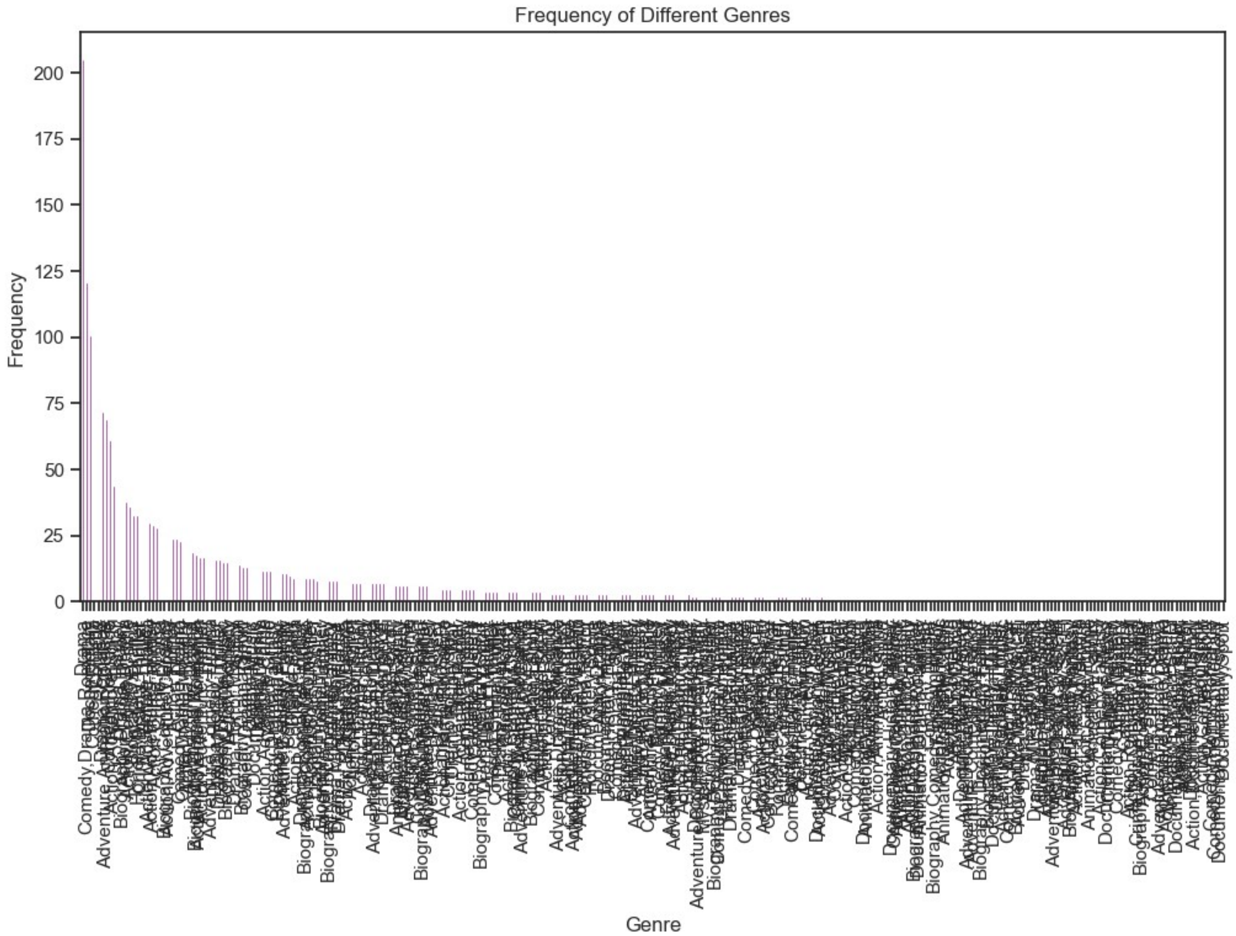
```
3                          Drama            5.7      127
4      Adventure,Comedy,Drama            7.3   275300
...                          ...            ...      ...
2443     Comedy,Drama,Romance            4.6     3618
2444           Crime,Thriller            8.5    43409
2445                    Drama            5.4      673
2446                   Comedy            4.6       61
2447        Documentary,Music            8.8     2067

                                   title     studio  domestic_gross  year

0                                  Wazir    Relbig.       1100000.0  2016

1                            On the Road        IFC        744000.0  2012

2                            On the Road        IFC        744000.0  2012

3                            On the Road        IFC        744000.0  2012

4     The Secret Life of Walter Mitty        Fox      58200000.0  2013

...                                  ...        ...             ...   ...

2443                       Nobody's Fool       Par.      31700000.0  2018

2444                           Andhadhun       Eros       1200000.0  2018

2445                     Helicopter Eela       Eros         72000.0  2018

2446                    Oolong Courtyard         CL         37700.0  2018

2447        Burn the Stage: The Movie  Trafalgar       4200000.0  2018


[2448 rows x 12 columns]
```

Exploratory Data Analysis

```
# Dataframe summary
df.describe()

        start_year  runtime_minutes  averagerating      numvotes  \
count  2448.000000      2403.000000    2448.000000  2.448000e+03
mean   2013.773284       106.799834       6.406454  7.270063e+04
std       2.496518        20.063935       1.044846  1.345679e+05
min    2010.000000         3.000000       1.600000  5.000000e+00
25%    2012.000000        94.000000       5.800000  3.772000e+03
50%    2014.000000       104.000000       6.500000  2.071850e+04
75%    2016.000000       118.000000       7.100000  8.058950e+04
```

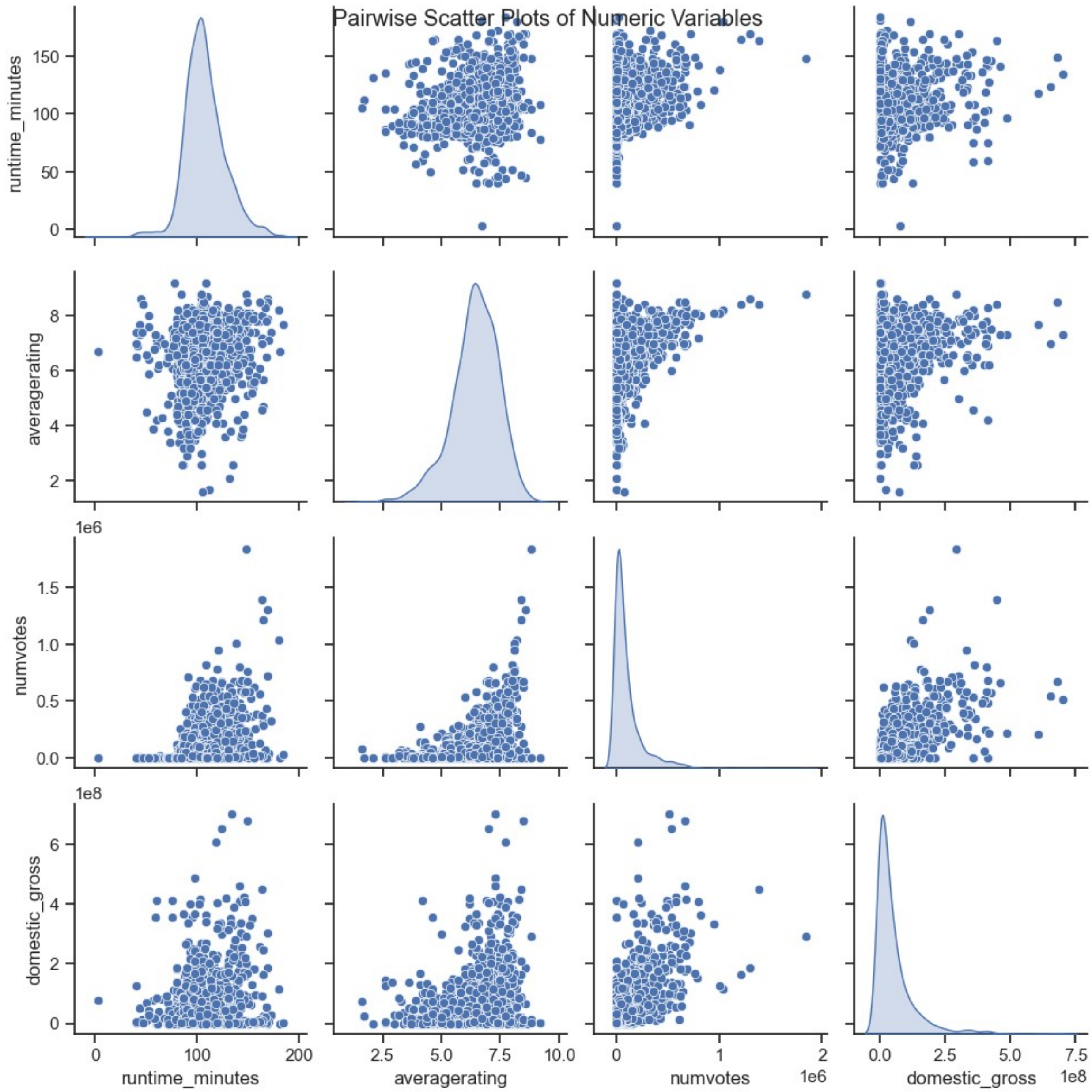
```
max    2019.000000      186.000000      9.200000  1.841066e+06

       domestic_gross         year
count    2.448000e+03  2448.000000
mean     3.588117e+07  2014.000408
std      6.930797e+07     2.465040
min      1.000000e+02  2010.000000
25%      3.040000e+05  2012.000000
50%      5.050000e+06  2014.000000
75%      4.262500e+07  2016.000000
max      7.001000e+08  2018.000000
```

```python
 #Count the frequency of different genres
genre_counts = df['genres'].value_counts()

# Create bar plot for genre frequency
plt.figure(figsize=(12, 6))
genre_counts.plot(kind='bar', color='purple')
plt.title('Frequency of Different Genres')
plt.xlabel('Genre')
plt.ylabel('Frequency')

plt.show()
```



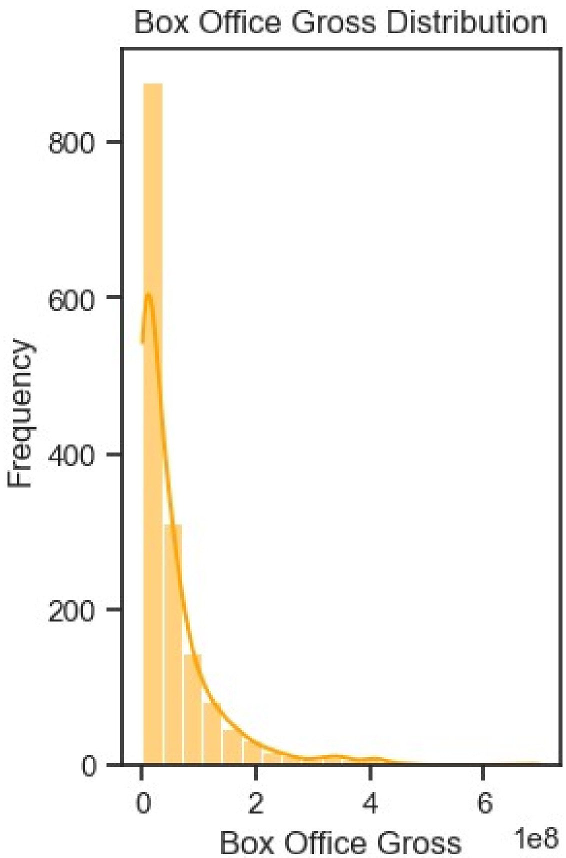

```
# Overview of the numeric variables

numeric_variables = ['runtime_minutes', 'averagerating', 'numvotes', 
'domestic_gross']

# Creating scatter plots for pairs of numeric variables
sns.set(style="ticks")
sns.pairplot(df[numeric_variables].dropna(), kind='scatter', 
diag_kind='kde')
plt.suptitle('Pairwise Scatter Plots of Numeric Variables')
plt.show()
```

```
C:\Users\Mau\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: 
UserWarning: The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```



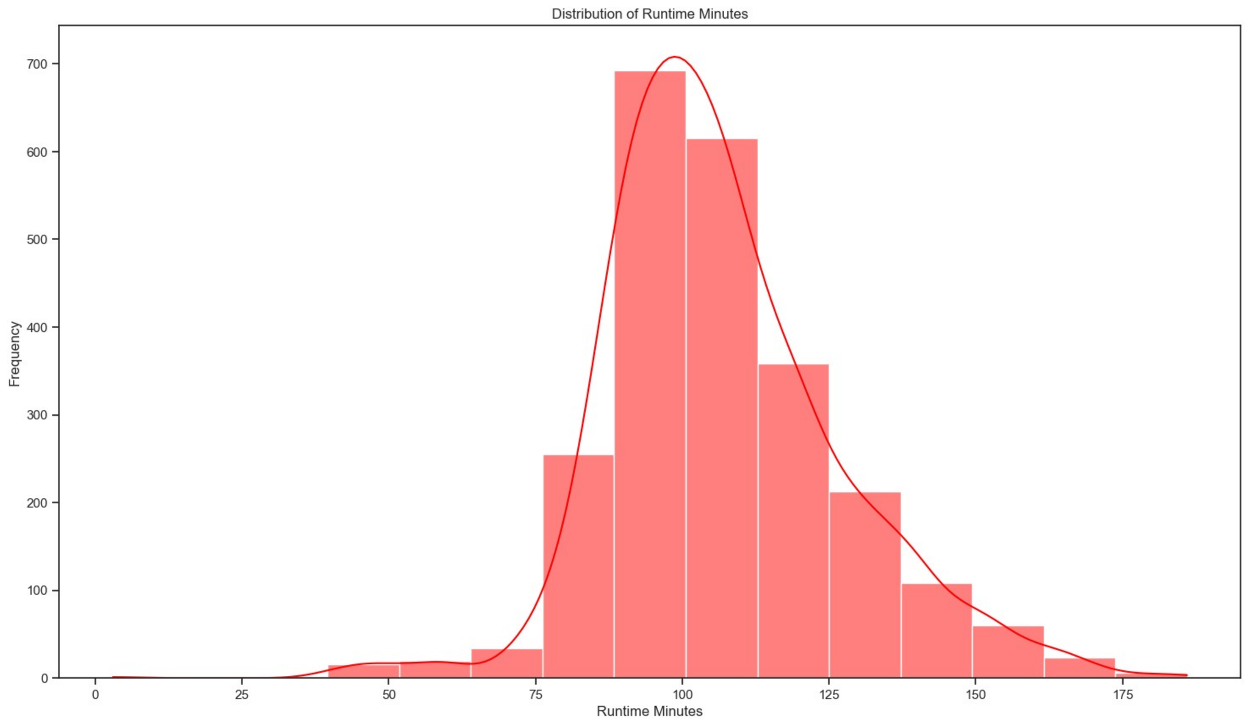

```
# Box Office Gross Distribution
plt.subplot(1, 2, 1)
sns.histplot(df['domestic_gross'], bins=20, kde=True, color='orange')
plt.title('Box Office Gross Distribution')
plt.xlabel('Box Office Gross')
plt.ylabel('Frequency')
```

```
Text(0, 0.5, 'Frequency')
```



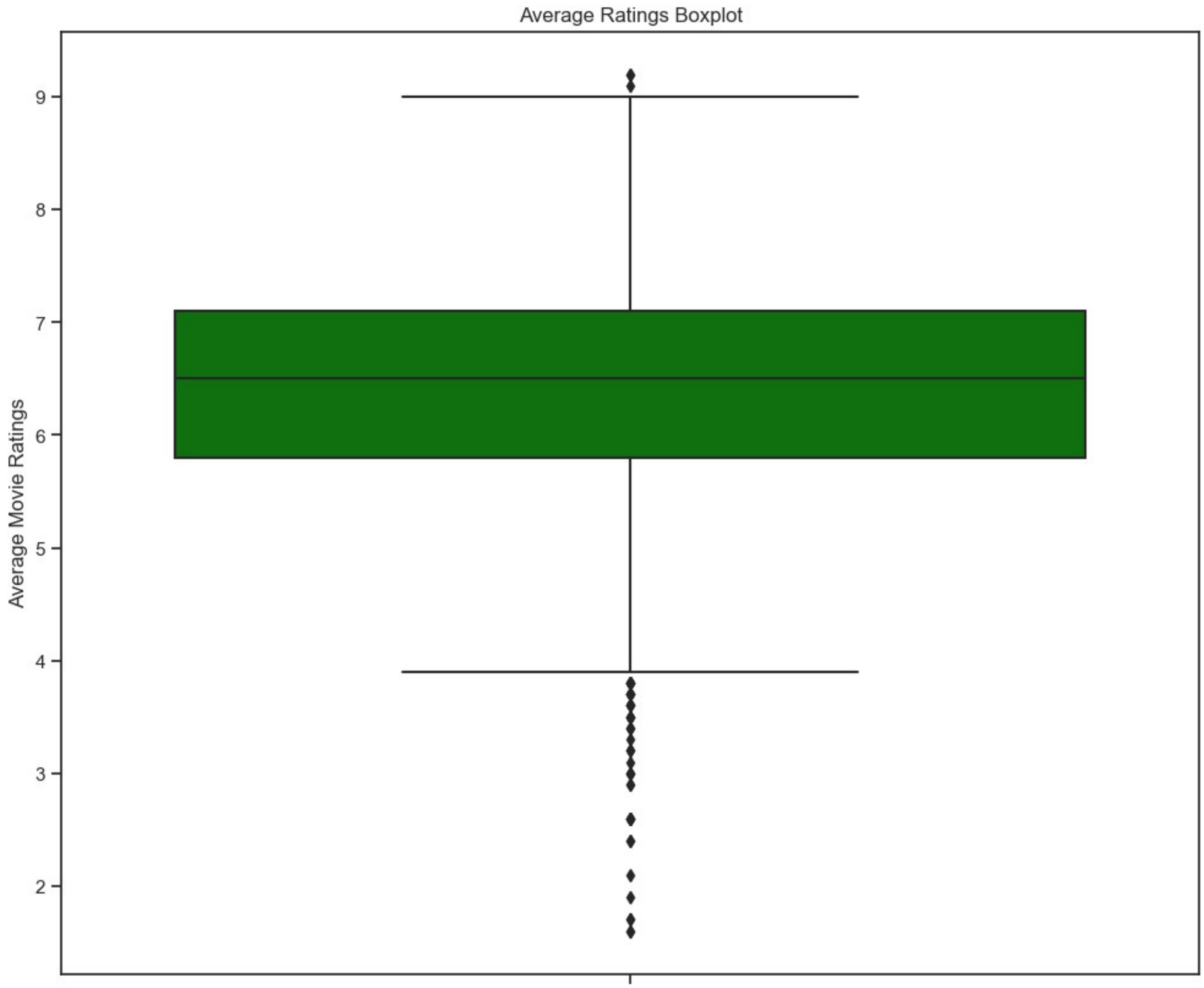
The distribution of this plot is positively skewed, implying that there are more movies with lower box office gross earnings than with higher earnings.

```
# Visualize the runtime minutes in a histogram

plt.figure(figsize=(18, 10))
sns.histplot(df['runtime_minutes'], bins=15, kde=True, color='red')
plt.title('Distribution of Runtime Minutes')
plt.xlabel('Runtime Minutes')
plt.ylabel('Frequency')
plt.show()
```



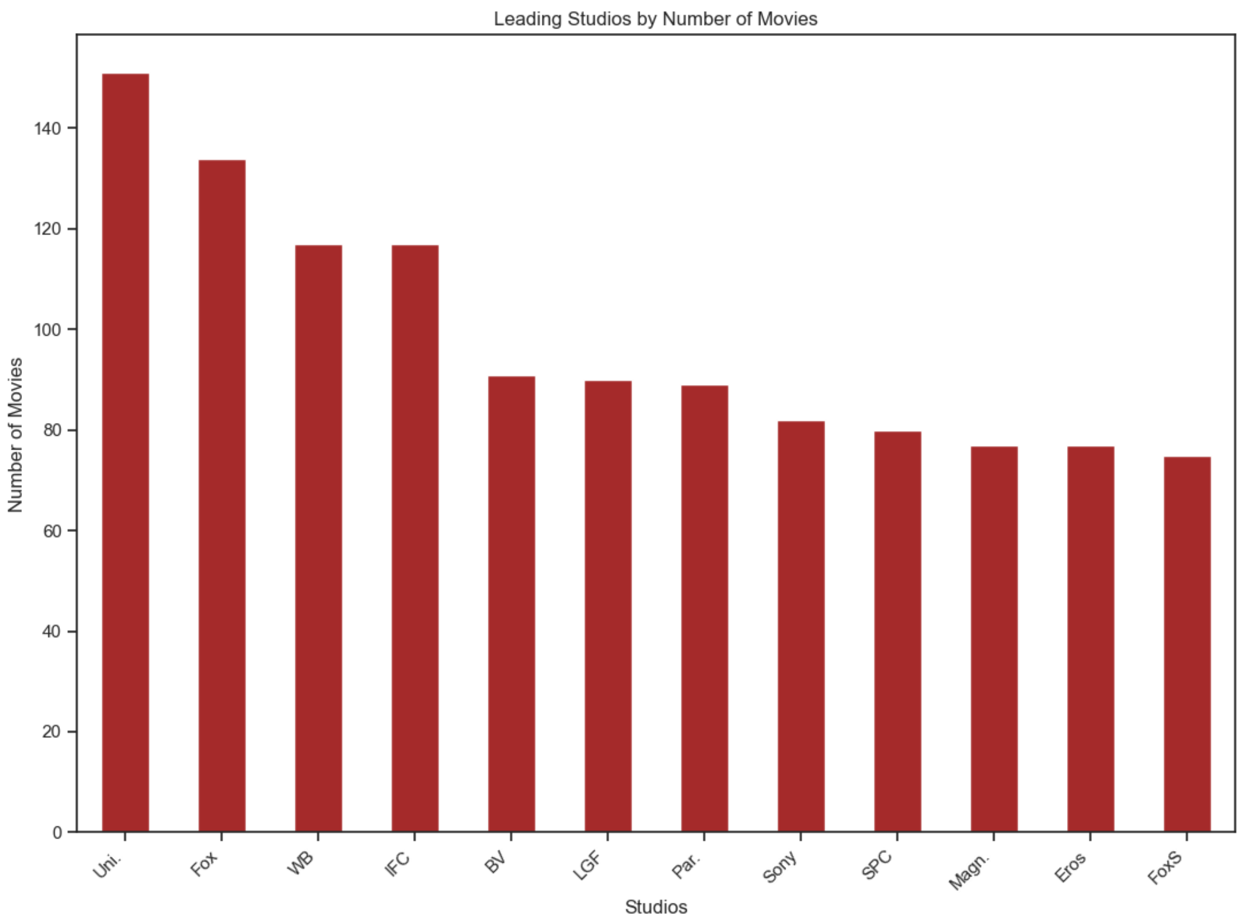
Many movies have a runtime ranging between 90-120 minutes. This is a common characteristic of films. This shows a normal distribution. The distribution of the runtime shows that it is positively skewed, meaning that more movies have a shorter runtime, than the ones with a longer runtime.

```
# Box plot for average ratings
plt.figure(figsize=(12, 10))
sns.boxplot(y='averagerating', data=df, color = 'green')
plt.title('Average Ratings Boxplot')
plt.ylabel('Average Movie Ratings')
plt.show()
```



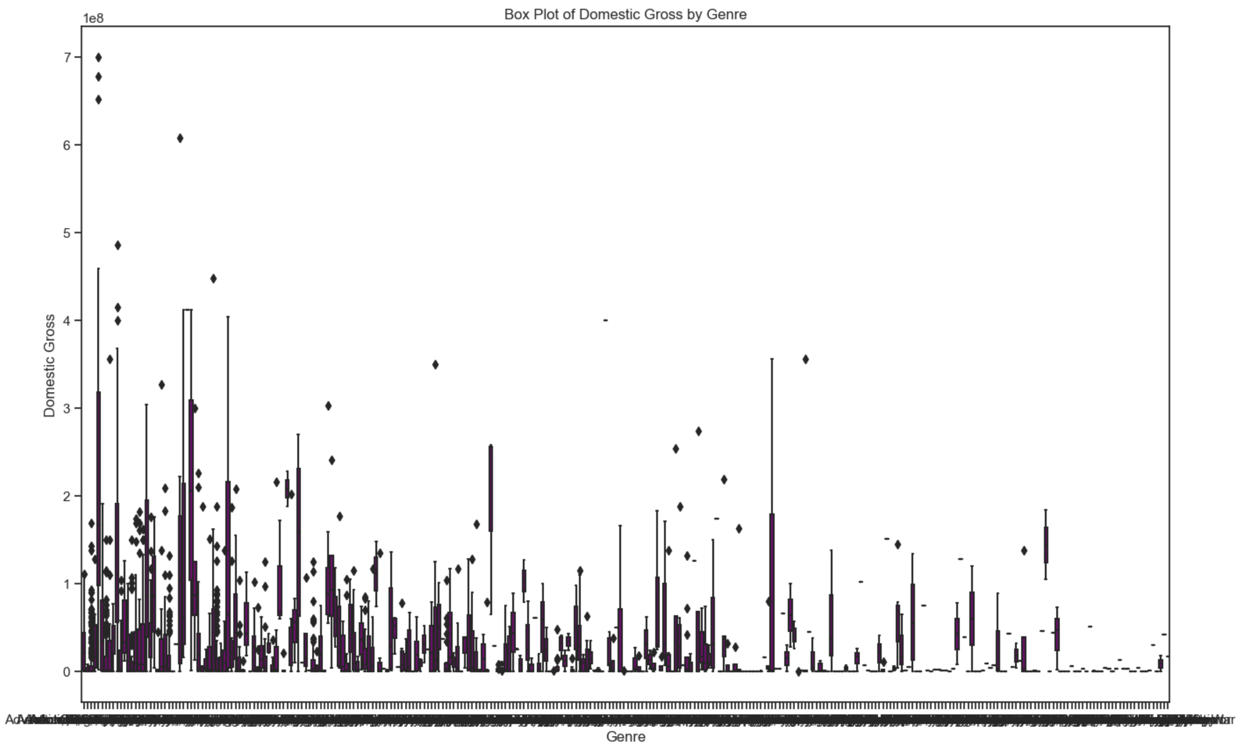
There are individual data points outside the whiskers, which denotes the presence of outliers in the data. Movie ratings are evenly distributed across the median.

```
#lets check the leading studios by the number of movies
leading_studios = df['studio'].value_counts().head(12)

plt.figure(figsize=(13, 9))
leading_studios.plot(kind='bar', color='brown')
plt.title('Leading Studios by Number of Movies')
plt.xlabel('Studios')
plt.ylabel('Number of Movies')
plt.xticks(rotation=45, ha='right')
plt.show()
```



From the above bar chart, it is clear that Uni., Fox, IFC and WB are some of the studios with the highest number of movies produced. Factors such as the size of the studio, the marketing strategy, budgets used during production, and resources allocated for production could be the reason behind the high number of movies produced. strategies. Some studios rank small in the studio market, which can actually be a contributing factor to lower procuction in movies.

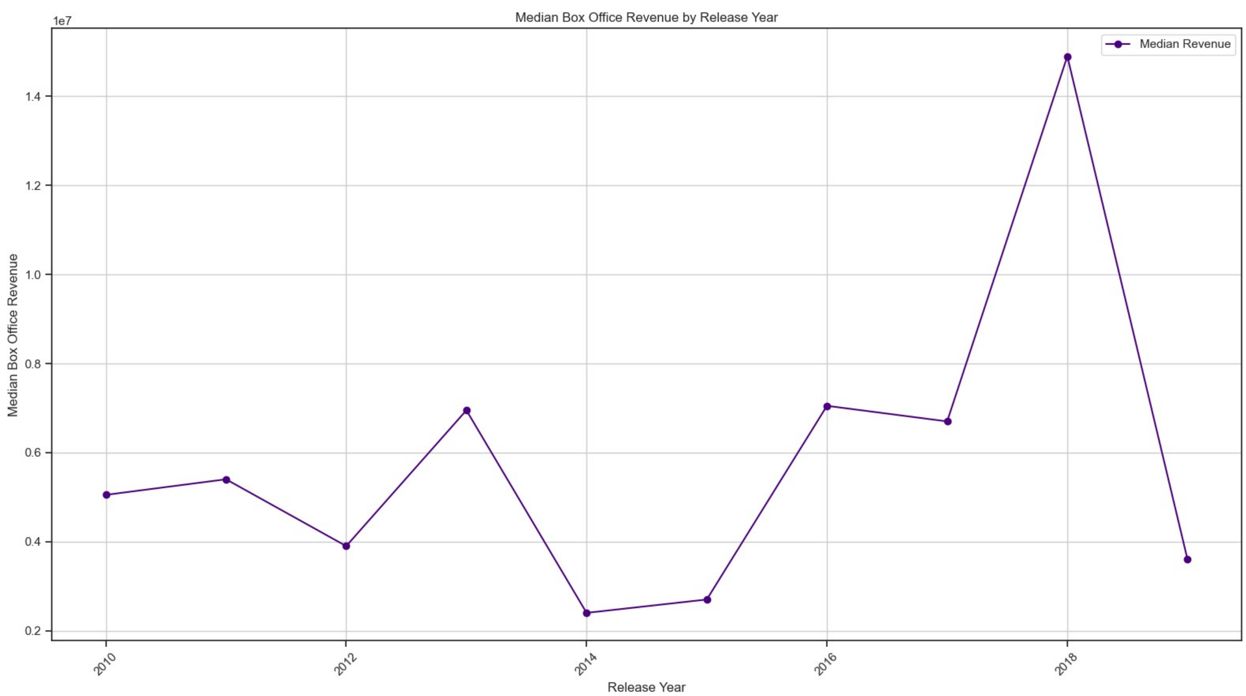
```
# Lets compare, the domestic_gross and genre using a box plot

import seaborn as sns
import matplotlib.pyplot as plt


plt.figure(figsize=(16, 10))
sns.boxplot(x='genres', y='domestic_gross', color = 'purple', data=df)

plt.title('Box Plot of Domestic Gross by Genre')
plt.xlabel('Genre')
plt.ylabel('Domestic Gross')

plt.show()
```



Some genres are exhibiting a wider range of domestic gross earnings, while others have a more concentrated distribution. Genres with higher median domestic gross indicate that, movies within these genres tend to perform better at the box office compared to ones with a lower median domestic gross.

```
df['domestic_gross'] = df['domestic_gross'].astype(str)


df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2448 entries, 0 to 2447
Data columns (total 12 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   tconst           2448 non-null   object 
 1   primary_title    2448 non-null   object 
 2   original_title   2448 non-null   object 
 3   start_year       2448 non-null   int64  
 4   runtime_minutes  2403 non-null   float64
 5   genres           2444 non-null   object 
 6   averagerating    2448 non-null   float64
 7   numvotes         2448 non-null   int64  
 8   title            2448 non-null   object 
 9   studio           2445 non-null   object 
 10  domestic_gross   2448 non-null   float64
 11  year             2448 non-null   int64  
```

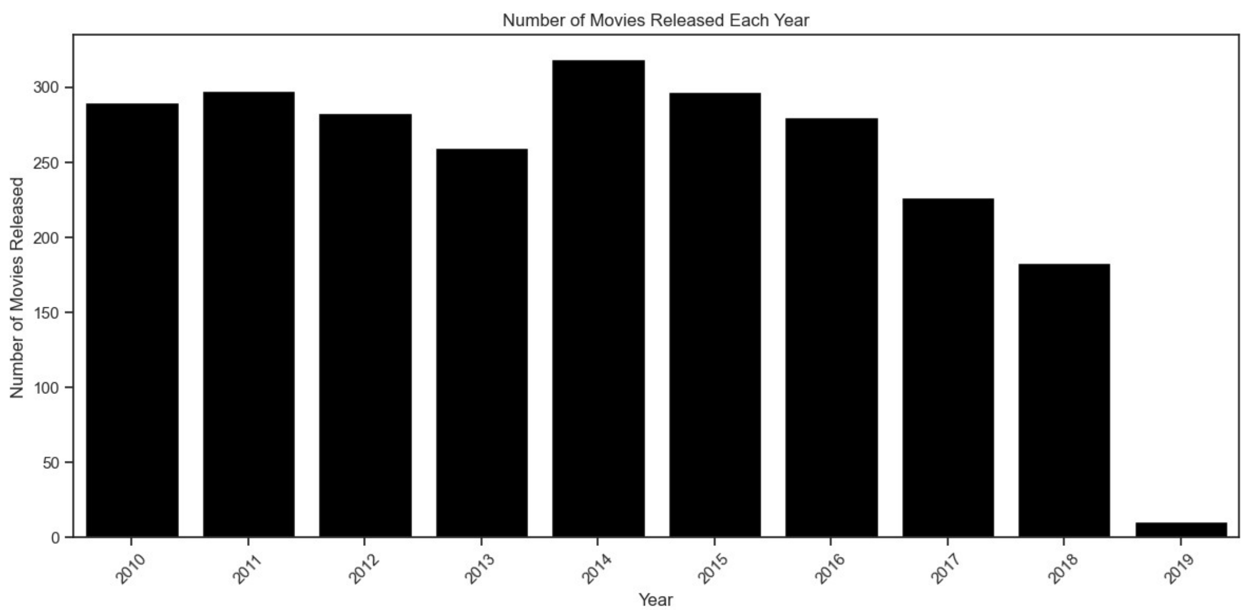
```
dtypes: float64(3), int64(3), object(6)
memory usage: 229.6+ KB

median_revenue_per_year = df.groupby('start_year')
['domestic_gross'].median()

# line plot for median box office revenue per year
plt.figure(figsize=(16, 9))
plt.plot(median_revenue_per_year.index,
median_revenue_per_year.values, marker='o', color='indigo',
label='Median Revenue')
plt.title('Median Box Office Revenue by Release Year')
plt.xlabel('Release Year')
plt.ylabel('Median Box Office Revenue')
plt.xticks(rotation=45)
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```

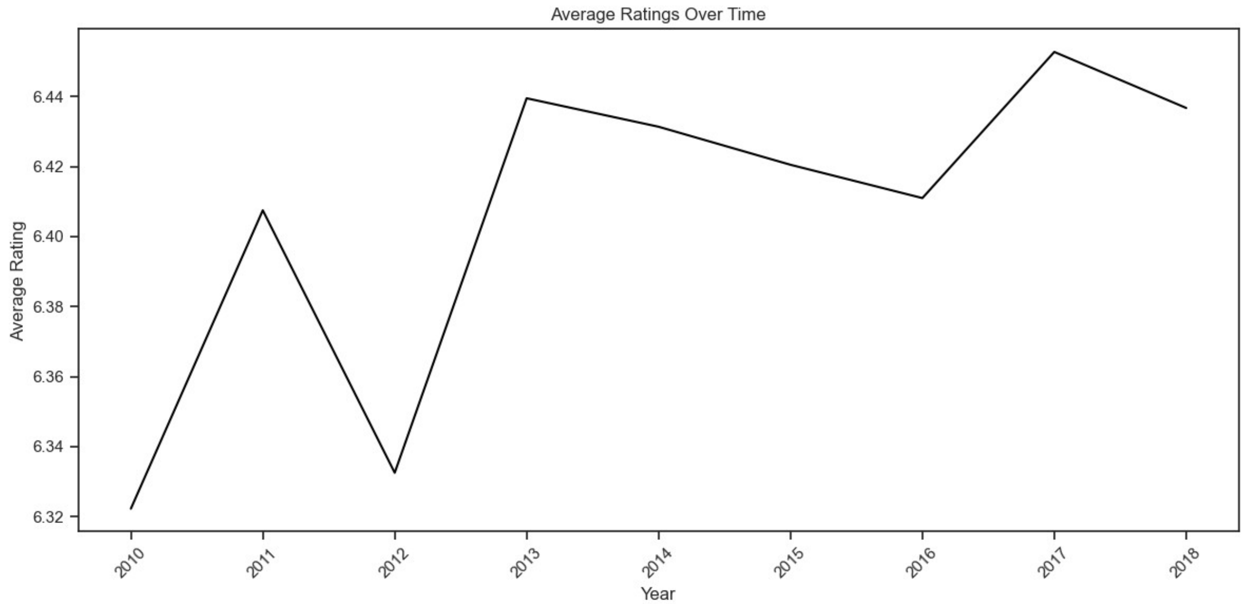


The above line plot shows how the median box office revenue for movies has changed over different release years, indicating variability in box office performance over time. The presence of outliers could influence the median and affect the overall trend.

```
movies_per_year = df['start_year'].value_counts().sort_index()

plt.figure(figsize=(12, 6))
sns.barplot(x=movies_per_year.index, y=movies_per_year.values,
```

```
color='black')
plt.title('Number of Movies Released Each Year')
plt.xlabel('Year')
plt.ylabel('Number of Movies Released')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()

# Average ratings over time
avg_ratings_per_year = df.groupby('year')['averagerating'].mean()

plt.figure(figsize=(12, 6))
sns.lineplot(x=avg_ratings_per_year.index,
y=avg_ratings_per_year.values, color='black')
plt.title('Average Ratings Over Time')
plt.xlabel('Year')
plt.ylabel('Average Rating')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```





The number of movies released each year seems to change. There is an increase in the number of movies released in recent years compared to earlier years. This suggests a potential growth in the film industry and changes in production trends.

Average ratings over time are relatively stable across different years, with some changes in average ratings from year to year. The stability in average ratings shows a consistent quality level of movies being produced over the years.

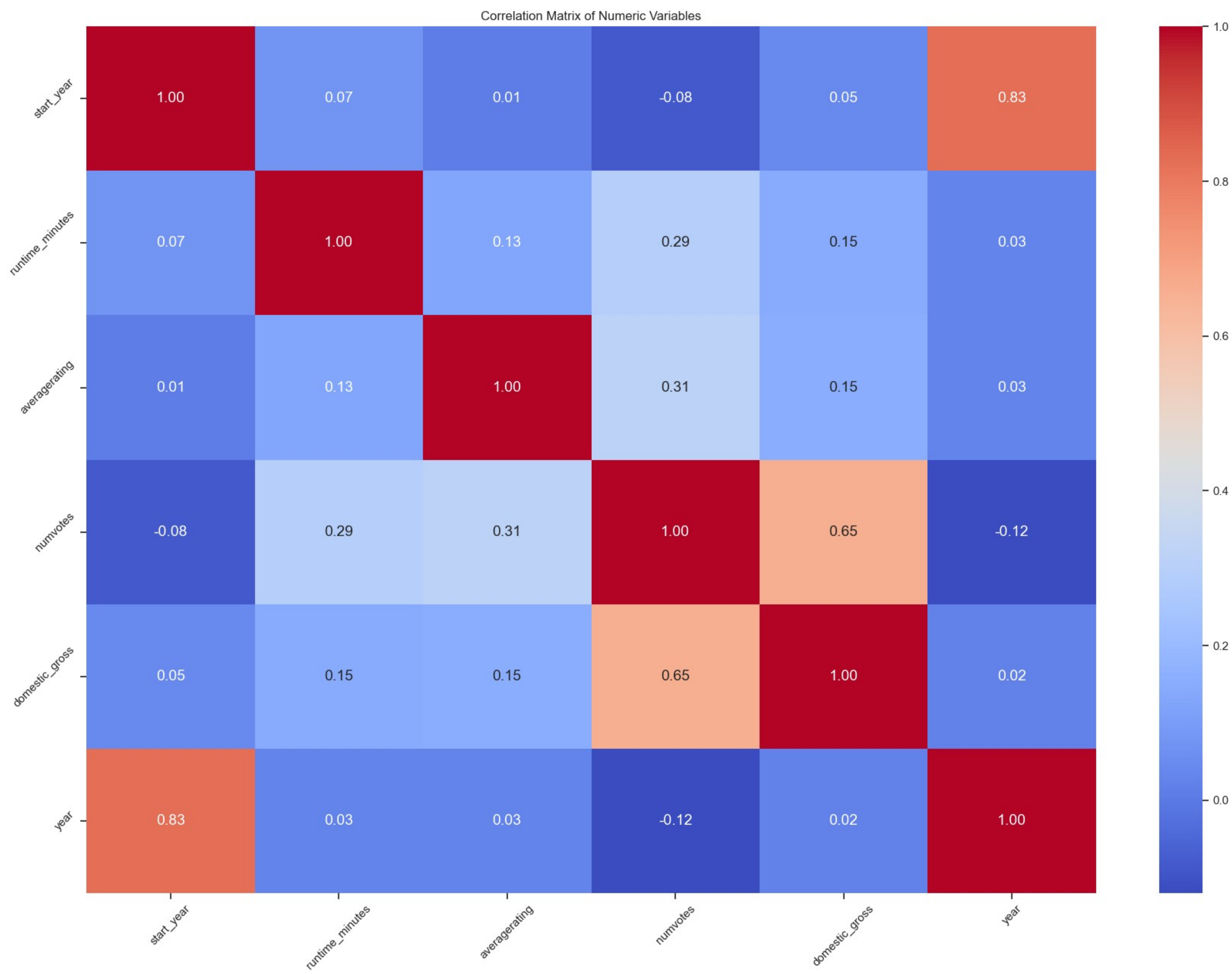
Correlation Analysis

```
print(df.dtypes)
```

```
tconst                object
primary_title         object
original_title        object
start_year             int64
runtime_minutes      float64
genres                object
averagerating        float64
numvotes               int64
title                 object
studio                object
domestic_gross       float64
year                   int64
dtype: object
```

```
# Numeric columns for correlation analysis
numeric_columns = ['start_year', 'runtime_minutes', 'averagerating', 
'numvotes', 'domestic_gross', 'year']
```

```python
# Computing the correlation matrix
correlation_matrix = df[numeric_columns].corr()

# Create a heatmap
plt.figure(figsize=(18, 13))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm',
fmt=".2f", annot_kws={"size": 14})
plt.title('Correlation Matrix of Numeric Variables')
plt.xticks(rotation=45)
plt.yticks(rotation=45)
plt.tight_layout()
plt.show()
```



There is a strong positive correlation observed between num_votes and domestic votes, hinting that movies with higher numbers of votes tend to have higher domestic gross earnings, meaning the more the attention a movie gets the more that its likelyto perfrom better at box office. Average rating, the year of release, the runtime, and the start year of the movie may not have a significant impact on its box office performance a they have no strong correlation with domestic gross.

Conclusion

With regards to the analysis above:

Insights into the types of films currently performing well at the box office were derived from correlation analysis, genre distribution, and box office gross trends. Observations indicate that certain genres, such as action, adventure, fantasy, and science fiction, tend to perform better at the box office. Understanding audience preferences and industry dynamics is crucial for establishing a successful movie studio.

Recommendations

Recommendations When making the decision to open a new movie studio,

Consider focusing on genres that are currently popular and have a track record of success at the box office, success in terms of audience interraction. Consider partnerships with established studios, filmmakers, and distribution networks to leverage expertise and resources in the industry. Have some effective marketing and promotional strategies, such as social media, digital platforms to build anticipation and generate word to many when there is a movie releases.