In [23]:

```python
# 1

import pandas as pd

df = pd.read_excel('NETFLIX.xlsx')
df
# Get basic information about the dataset
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9425 entries, 0 to 9424
Data columns (total 29 columns):
 #   Column                 Non-Null Count  Dtype         
---  ------                 --------------  -----         
 0   Title                  9425 non-null   object        
 1   Genre                  9400 non-null   object        
 2   Tags                   9389 non-null   object        
 3   Languages              9255 non-null   object        
 4   Series or Movie        9425 non-null   object        
 5   Hidden Gem Score       9415 non-null   float64       
 6   Country Availability   9414 non-null   object        
 7   Runtime                9424 non-null   object        
 8   Director               7120 non-null   object        
 9   Writer                 7615 non-null   object        
 10  Actors                 9314 non-null   object        
 11  View Rating            6827 non-null   object        
 12  IMDb Score             9417 non-null   float64       
 13  Rotten Tomatoes Score  5445 non-null   float64       
 14  Metacritic Score       4082 non-null   float64       
 15  Awards Received        5226 non-null   float64       
 16  Awards Nominated For   6376 non-null   float64       
 17  Boxoffice              3754 non-null   float64       
 18  Release Date           9217 non-null   datetime64[ns]
 19  Netflix Release Date   9425 non-null   datetime64[ns]
 20  Production House       4393 non-null   object        
 21  Netflix Link           9425 non-null   object        
 22  IMDb Link              9101 non-null   object        
 23  Summary                9420 non-null   object        
 24  IMDb Votes             9415 non-null   float64       
 25  Image                  9425 non-null   object        
 26  Poster                 8487 non-null   object        
 27  TMDb Trailer           9425 non-null   object        
 28  Trailer Site           9424 non-null   object        
dtypes: datetime64[ns](2), float64(8), object(19)
memory usage: 2.1+ MB
```

In [24]:

```
# 2

# Identify missing values
missing_values = df.isnull()

# Count the number of missing values for each column
missing_values_count = missing_values.sum()

# Display the count of missing values for each column
print(missing_values_count)
```

```
Title                           0
Genre                          25
Tags                           36
Languages                     170
Series or Movie                 0
Hidden Gem Score               10
Country Availability           11
Runtime                         1
Director                     2305
Writer                       1810
Actors                        111
View Rating                  2598
IMDb Score                      8
Rotten Tomatoes Score        3980
Metacritic Score             5343
Awards Received              4199
Awards Nominated For         3049
Boxoffice                    5671
Release Date                  208
Netflix Release Date            0
Production House             5032
Netflix Link                    0
IMDb Link                     324
Summary                         5
IMDb Votes                     10
Image                           0
Poster                        938
TMDb Trailer                    0
Trailer Site                    1
dtype: int64
```

In [26]:

```
df=df.dropna()
df
```

Out[26]:

	Title	Genre	Tags	Languages	Series or Movie	Hidden Gem Score	
0	Lets Fight Ghost	Crime, Drama, Fantasy, Horror, Romance	Comedy Programmes,Romantic TV Comedies,Horror ...	Swedish, Spanish	Series	4.3	
9	Joker	Crime, Drama, Thriller	Dark Comedies,Crime Comedies,Dramas,Comedies,C...	English	Movie	3.5	Lithuania
10	I	Action, Adventure, Fantasy, Sci-Fi	Dramas,Swedish Movies	English, Sanskrit	Movie	2.8	Lithuania,
11	Harrys Daughters	Adventure, Drama, Fantasy, Mystery	Dramas,Swedish Movies	English	Movie	4.4	Lithuania,
17	The Closet	Comedy	Korean Movies,Horror Movies,Mysteries	French	Movie	3.8	
...	...	...	...	...	...	...	
9411	50 First Dates	Comedy, Drama, Romance	Romantic Favourites,Romantic Comedies,Comedies...	English, Hawaiian, Mandarin	Movie	2.7	
9412	21	Crime, Drama, History, Thriller	Dramas,Dramas based on a book,Police Dramas,Po...	English	Movie	2.5	
9414	One Chance	Biography, Comedy, Drama, Music	Dramas,Biographical Dramas,Dramas based on rea...	English, Italian	Movie	3.0	
9415	The Twilight Saga: Breaking Dawn: Part 1	Adventure, Drama, Fantasy, Romance, Thriller	Dramas,Romantic Dramas,Dramas based on a book,...	English, Portuguese	Movie	2.0	Car
9416	One for the Money	Action, Comedy, Crime, Thriller	Romantic Comedies,Action Comedies,Comedies,Pol...	English	Movie	1.3	

2155 rows × 29 columns

In [27]:

```
df.isnull().sum()
```

Out[27]:

```
Title                    0
Genre                    0
Tags                     0
Languages                0
Series or Movie          0
Hidden Gem Score         0
Country Availability     0
Runtime                  0
Director                 0
Writer                   0
Actors                   0
View Rating              0
IMDb Score               0
Rotten Tomatoes Score    0
Metacritic Score         0
Awards Received          0
Awards Nominated For     0
Boxoffice                0
Release Date             0
Netflix Release Date     0
Production House         0
Netflix Link             0
IMDb Link                0
Summary                  0
IMDb Votes               0
Image                    0
Poster                   0
TMDb Trailer             0
Trailer Site             0
dtype: int64
```

In [28]:
```
# 3

import pandas as pd

# Generate summary statistics for numerical columns
summary_stats = df.describe()

print(summary_stats)
```

```
       Hidden Gem Score   IMDb Score  Rotten Tomatoes Score  Metacritic Score  
\
count       2155.000000  2155.000000            2155.000000       2155.000000
mean           3.396659     6.788538              65.759165         60.961021
min            0.600000     2.200000               0.000000          6.000000
25%            2.700000     6.300000              49.000000         50.000000
50%            3.500000     6.900000              72.000000         62.000000
75%            4.000000     7.400000              86.000000         73.000000
max            8.700000     9.300000             100.000000        100.000000
std            1.090777     0.908366              25.199188         16.927377

       Awards Received  Awards Nominated For     Boxoffice  \
count      2155.000000           2155.000000  2.155000e+03
mean         13.547564             27.378654  6.950284e+07
min           1.000000              1.000000  5.090000e+02
25%           2.000000              5.000000  8.551992e+06
50%           4.000000             12.000000  4.321839e+07
75%          13.000000             29.000000  1.002433e+08
max         300.000000            355.000000  6.523856e+08
std          25.693355             41.910874  8.403720e+07

                        Release Date           Netflix Release Date  \
count                           2155                           2155
mean   2007-09-02 10:54:10.858468608  2016-09-21 21:10:16.426914048
min              1936-02-25 00:00:00            2015-04-14 00:00:00
25%              2002-07-29 12:00:00            2015-04-14 00:00:00
50%              2010-08-20 00:00:00            2015-07-02 00:00:00
75%              2015-09-07 12:00:00            2017-12-29 00:00:00
max              2020-06-19 00:00:00            2021-03-04 00:00:00
std                              NaN                            NaN

         IMDb Votes
count  2.155000e+03
mean   1.917973e+05
min    5.560000e+02
25%    4.267700e+04
50%    1.102020e+05
75%    2.406775e+05
max    2.354197e+06
std    2.433967e+05
```

In [29]:
```python
# 4

# Identify categorical columns using select_dtypes()
categorical_columns = df.select_dtypes(include='object').columns

# Iterate through the categorical columns and print unique values
for column in categorical_columns:
    print(f"Unique values in column '{column}':")
    print(df[column].unique())
    print("\n")
```

```
Unique values in column 'Title':
['Lets Fight Ghost' 'Joker' 'I' ... 'One Chance'
 'The Twilight Saga: Breaking Dawn: Part 1' 'One for the Money']


Unique values in column 'Genre':
['Crime, Drama, Fantasy, Horror, Romance' 'Crime, Drama, Thriller'
 'Action, Adventure, Fantasy, Sci-Fi' 'Adventure, Drama, Fantasy, Mystery'
 'Comedy' 'Comedy, Romance' 'Drama' 'Adventure, Drama'
 'Adventure, Drama, Mystery' 'Crime, Drama, Mystery, Thriller'
 'Action, Comedy, Crime, Thriller'
 'Action, Comedy, Crime, Sci-Fi, Thriller'
 'Crime, Drama, Horror, Thriller' 'Action, Adventure, Comedy, Sci-Fi'
 'Drama, Mystery, Thriller' 'Drama, Mystery' 'Drama, Thriller, Western'
 'Biography, Crime, Drama' 'Documentary, Action, Crime, Drama'
 'Comedy, Drama, Family, Romance' 'Biography, Drama'
 'Adventure, Drama, Horror, Mystery, Sci-Fi, Thriller'
 'Biography, Drama, Romance' 'Comedy, Drama'
 'Comedy, Drama, Music, Romance' 'Documentary, Biography, Comedy'
 'Comedy, Family' 'Biography, Drama, Family, Musical' 'Adventure, Comedy'
```

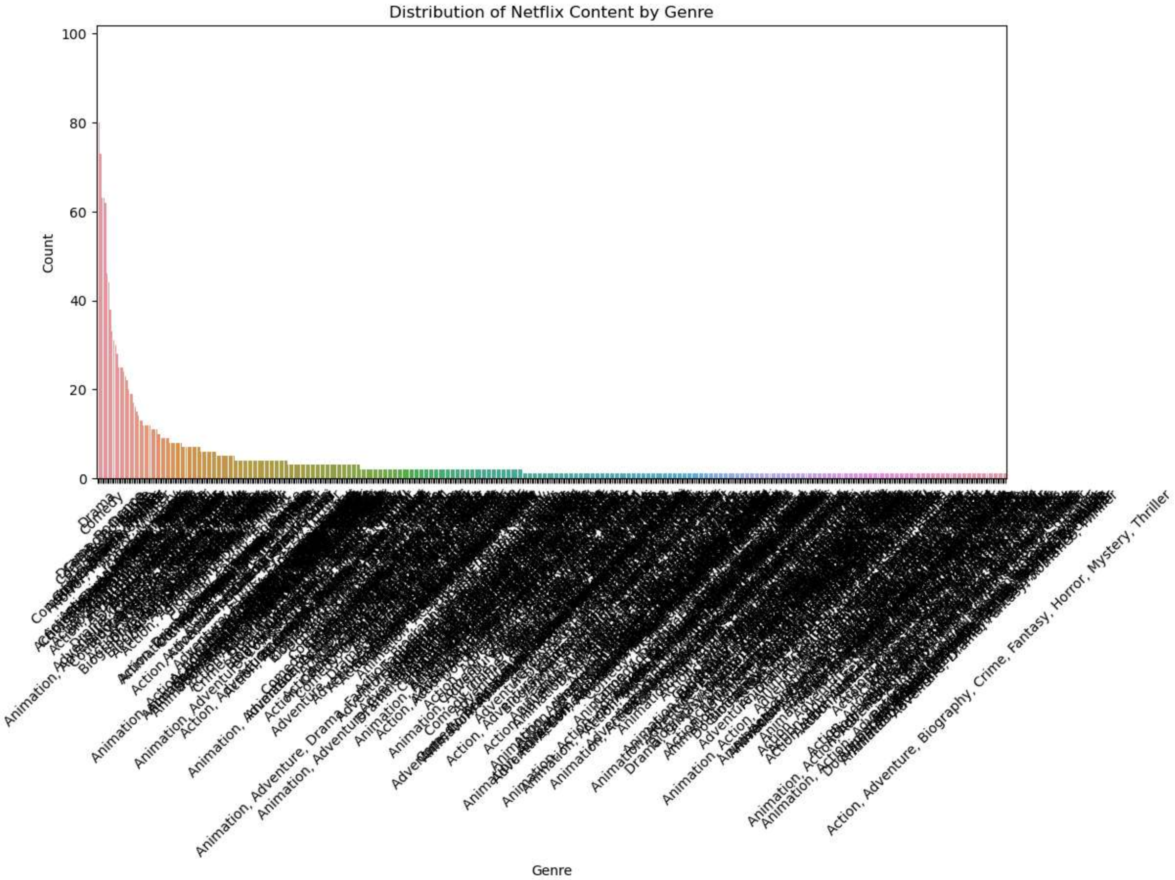
In [16]:

```
df.isnull().sum()
```

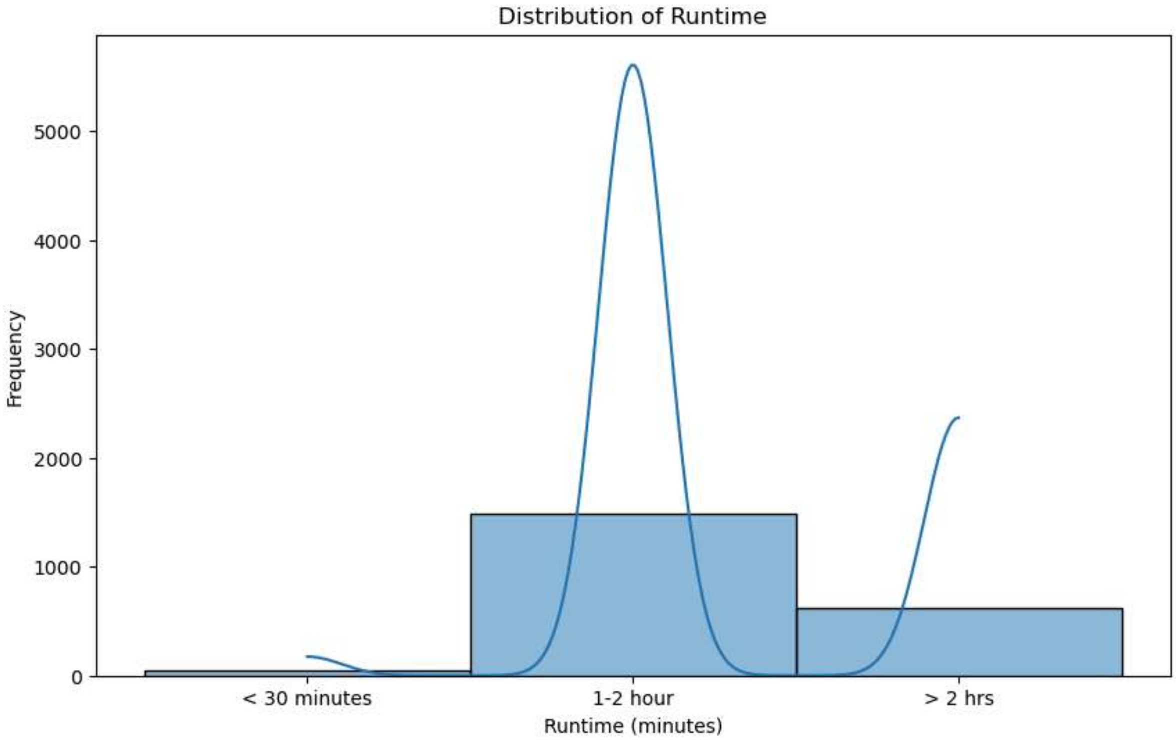
Out[16]:

```
Title                    0
Genre                    0
Tags                     0
Languages                0
Series or Movie          0
Hidden Gem Score         0
Country Availability     0
Runtime                  0
Director                 0
Writer                   0
Actors                   0
View Rating              0
IMDb Score               0
Rotten Tomatoes Score    0
Metacritic Score         0
Awards Received          0
Awards Nominated For     0
Boxoffice                0
Release Date             0
Netflix Release Date     0
Production House         0
Netflix Link             0
IMDb Link                0
Summary                  0
IMDb Votes               0
Image                    0
Poster                   0
TMDb Trailer             0
Trailer Site             0
dtype: int64
```

In [17]:
```
# 5

import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Assuming the target variable is 'genre'
# Create a count plot to visualize the distribution of the 'genre' column
plt.figure(figsize=(12, 6))
sns.countplot(data=df, x='Genre', order=df['Genre'].value_counts().index)
plt.title('Distribution of Netflix Content by Genre')
plt.xlabel('Genre')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.show()
```

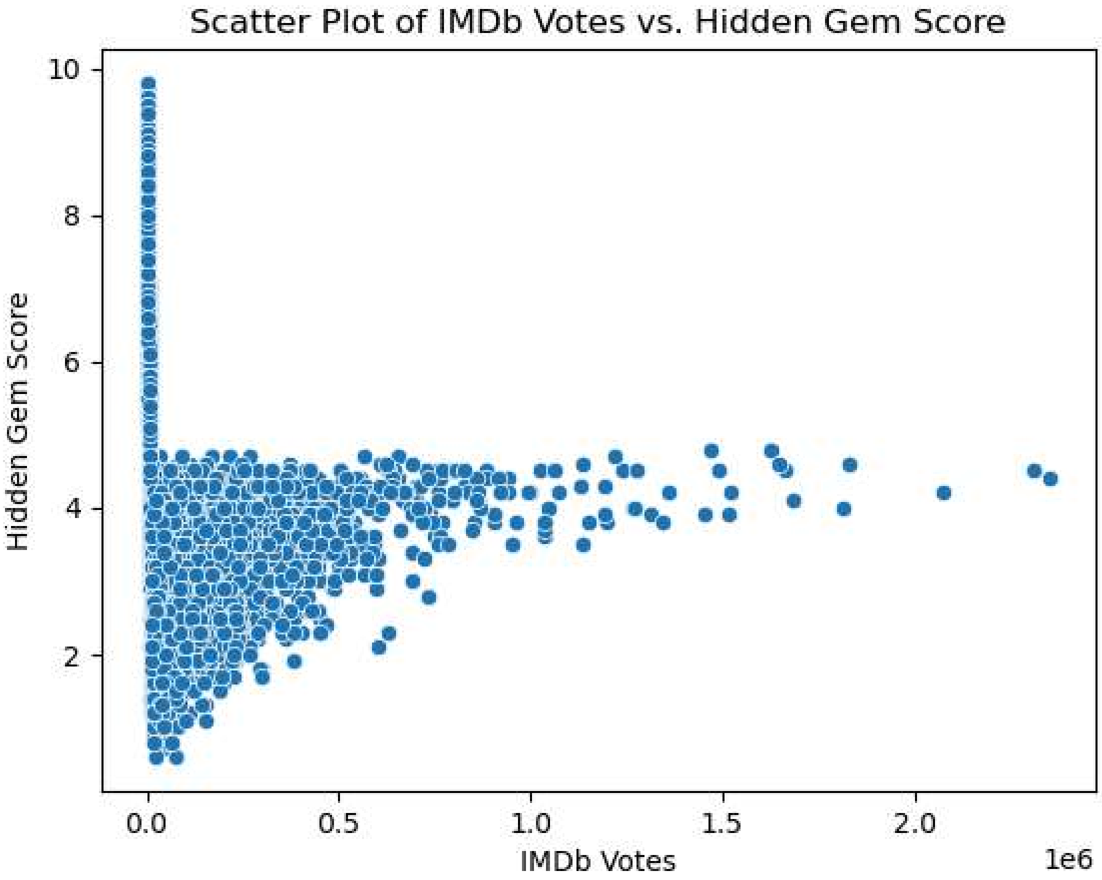


In [30]:
```
# 6

import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')


# Visualize the distribution of the 'Runtime' column
plt.figure(figsize=(10, 6))
sns.histplot(df['Runtime'], bins=20, kde=True)
plt.title('Distribution of Runtime')
plt.xlabel('Runtime (minutes)')
plt.ylabel('Frequency')
plt.show()
```



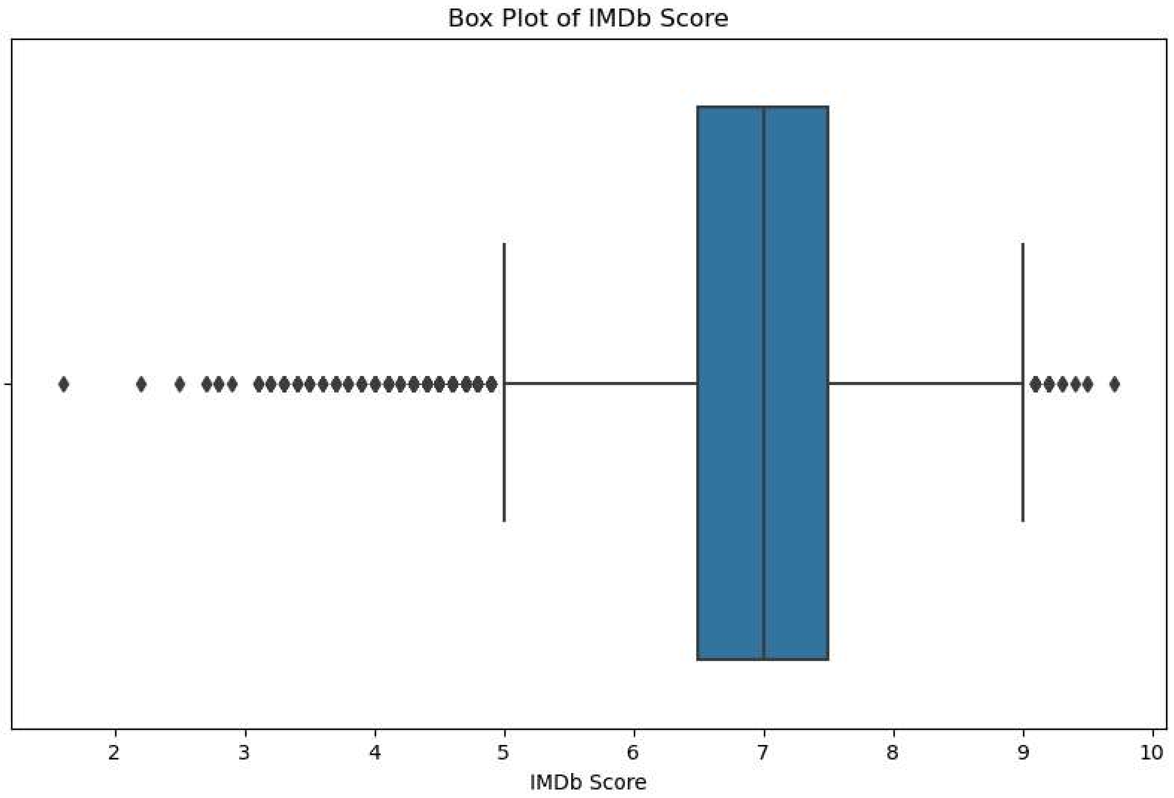
In [31]:

```
# 7

import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd

df = pd.read_excel('netflix.xlsx')
df

# Create scatter plot
# plt.figure(figsize=(10, 6))
sns.scatterplot(x='IMDb Votes', y='Hidden Gem Score', data=df)
plt.title('Scatter Plot of IMDb Votes vs. Hidden Gem Score')
plt.xlabel('IMDb Votes')
plt.ylabel('Hidden Gem Score')
plt.show()
```



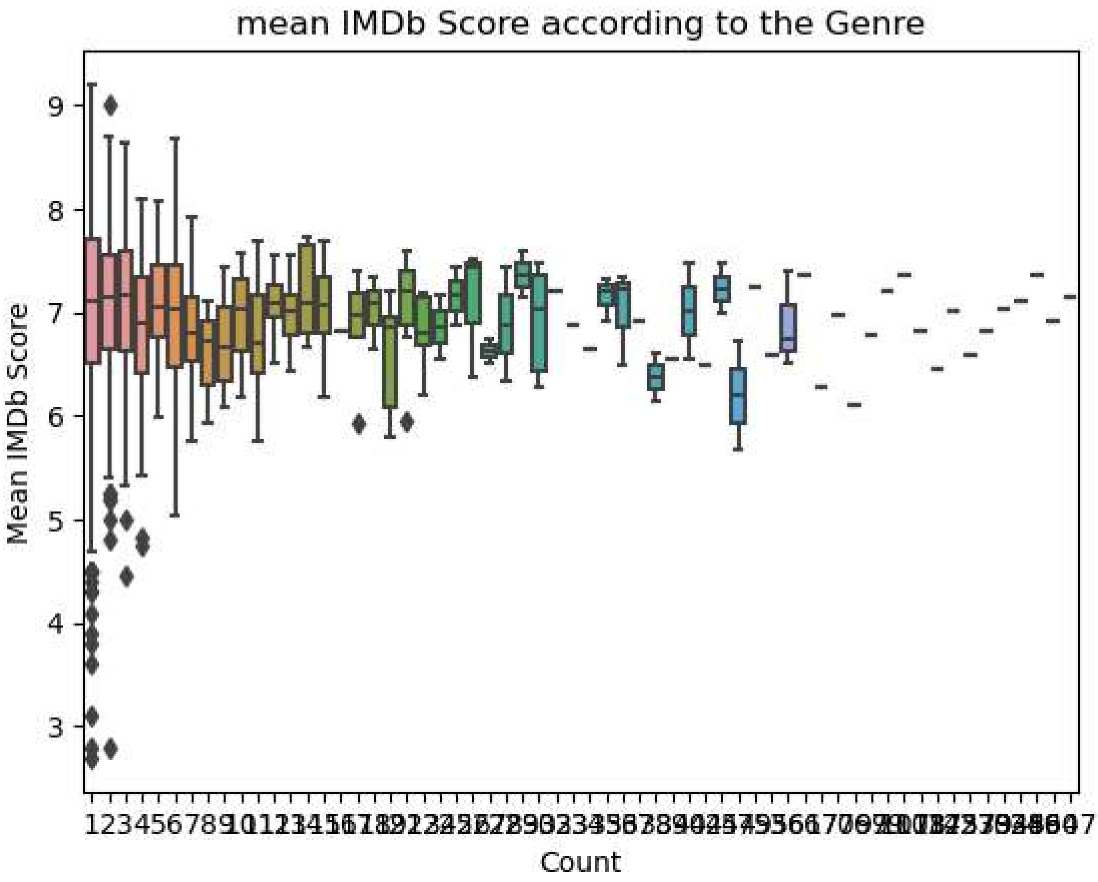
In [32]:

```
# 8

import seaborn as sns
import matplotlib.pyplot as plt

df = pd.read_excel('netflix.xlsx')
df

# Assuming 'df' is a DataFrame containing your data and 'column' is the column
plt.figure(figsize=(10, 6))
sns.boxplot(data=df, x='IMDb Score')
plt.title('Box Plot of IMDb Score')
plt.xlabel('IMDb Score')
plt.show()
```



In [33]:
```
# 9

df1=df.groupby('Genre')['IMDb Score'].agg(['mean','count']).reset_index()
df1.columns=['Genre','mean IMDb Score','count']
print(df1.head(1))
sns.boxplot(x='count',y='mean IMDb Score',data=df1)
plt.title('mean IMDb Score according to the Genre')
plt.xlabel('Count')
plt.ylabel('Mean IMDb Score')
plt.show()
```

```
    Genre  mean IMDb Score  count
0  Action              6.8     23
```



In [ ]:
```
#10 conclusion:

The count plot of cuisine types reveals the popularity of various cuisines.
Identifying the most and least popular cuisines can help Zomato focus on custo
Restaurant Ratings Distribution:

The histogram of restaurant ratings shows how ratings are spread across the da
If the ratings distribution is skewed, it might indicate a general trend towar
Relationship Between Average Cost for Two and Ratings:

The scatter plot reveals any potential correlation between the cost of dining 
Identifying such correlations can help in understanding whether more expensive
Average Ratings by City:

The bar plot showing average ratings by city provides insights into how differ
This can help Zomato tailor their marketing and restaurant acquisition strateg
Missing Values:

Analysis of missing values helps to understand data quality.
Columns with significant missing values might need imputation or exclusion fro
```