In [2]:
```
import pandas as pd
data=pd.read_csv("youtubers_df.csv")
data.head()
```

Out[2]:

	Rank	Username	Categories	Suscribers	Country	Visits	Likes	Comments
0	1	tseries	Música y baile	249500000.0	India	86200.0	2700.0	78.0
1	2	MrBeast	Videojuegos, Humor	183500000.0	Estados Unidos	117400000.0	5300000.0	18500.0
2	3	CoComelon	Educación	165500000.0	Unknown	7000000.0	24700.0	0.0
3	4	SETIndia	NaN	162600000.0	India	15600.0	166.0	9.0
4	5	KidsDianaShow	Animación, Juguetes	113500000.0	Unknown	3900000.0	12400.0	0.0

In [3]:
```
data.rename(columns={'Suscribers': 'Subscribers'}, inplace=True)
```

In [3]:
```
data.head()
```

Out[3]:

	Rank	Username	Categories	Subscribers	Country	Visits	Likes	Comments
0	1	tseries	Música y baile	249500000.0	India	86200.0	2700.0	78.0
1	2	MrBeast	Videojuegos, Humor	183500000.0	Estados Unidos	117400000.0	5300000.0	18500.0
2	3	CoComelon	Educación	165500000.0	Unknown	7000000.0	24700.0	0.0
3	4	SETIndia	NaN	162600000.0	India	15600.0	166.0	9.0
4	5	KidsDianaShow	Animación, Juguetes	113500000.0	Unknown	3900000.0	12400.0	0.0

In [5]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 9 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   Rank         1000 non-null   int64  
 1   Username     1000 non-null   object 
 2   Categories   694 non-null    object 
 3   Subscribers  1000 non-null   float64
 4   Country      1000 non-null   object 
 5   Visits       1000 non-null   float64
 6   Likes        1000 non-null   float64
 7   Comments     1000 non-null   float64
 8   Links        1000 non-null   object 
dtypes: float64(4), int64(1), object(4)
memory usage: 70.4+ KB
```

In [7]:

```
data['Subscribers'] = data['Subscribers'].astype(int)
data['Visits'] = data['Visits'].astype(int)
data['Likes'] = data['Likes'].astype(int)
data['Comments'] = data['Comments'].astype(int)
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 9 columns):
 #   Column       Non-Null Count  Dtype 
---  ------       --------------  ----- 
 0   Rank         1000 non-null   int64 
 1   Username     1000 non-null   object
 2   Categories   694 non-null    object
 3   Subscribers  1000 non-null   int32 
 4   Country      1000 non-null   object
 5   Visits       1000 non-null   int32 
 6   Likes        1000 non-null   int32 
 7   Comments     1000 non-null   int32 
 8   Links        1000 non-null   object
dtypes: int32(4), int64(1), object(4)
memory usage: 54.8+ KB
```

In [15]:

```
data=data.dropna(axis=0)
data.shape
```

Out[15]:

```
(694, 9)
```

In [16]:

```
data.describe()
```

Out[16]:

	Rank	Subscribers	Visits	Likes	Comments
count	694.000000	6.940000e+02	6.940000e+02	6.940000e+02	694.000000
mean	495.298271	2.241556e+07	1.210730e+06	5.347360e+04	1558.793948
std	289.222212	1.824123e+07	6.038274e+06	2.979711e+05	7967.470234
min	1.000000	1.170000e+07	0.000000e+00	0.000000e+00	0.000000
25%	244.250000	1.380000e+07	3.692500e+04	5.685000e+02	2.000000
50%	492.500000	1.680000e+07	1.587000e+05	3.550000e+03	78.000000
75%	746.750000	2.390000e+07	8.339000e+05	2.377500e+04	499.750000
max	1000.000000	2.495000e+08	1.174000e+08	5.300000e+06	154000.000000

In [17]:

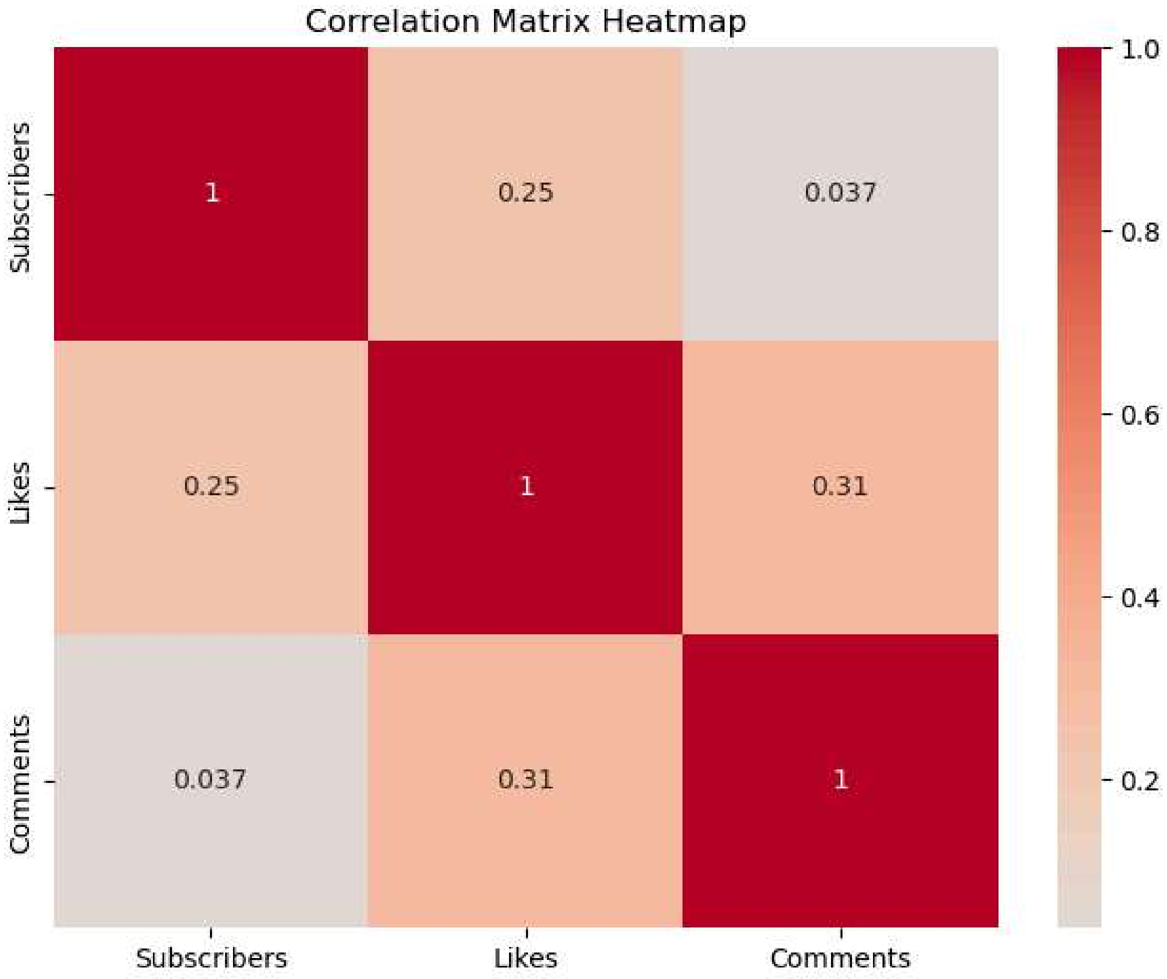
```
correlation_matrix = data[['Subscribers', 'Likes', 'Comments']].corr()
correlation_matrix
```

Out[17]:

	Subscribers	Likes	Comments
Subscribers	1.000000	0.248389	0.037293
Likes	0.248389	1.000000	0.311424
Comments	0.037293	0.311424	1.000000

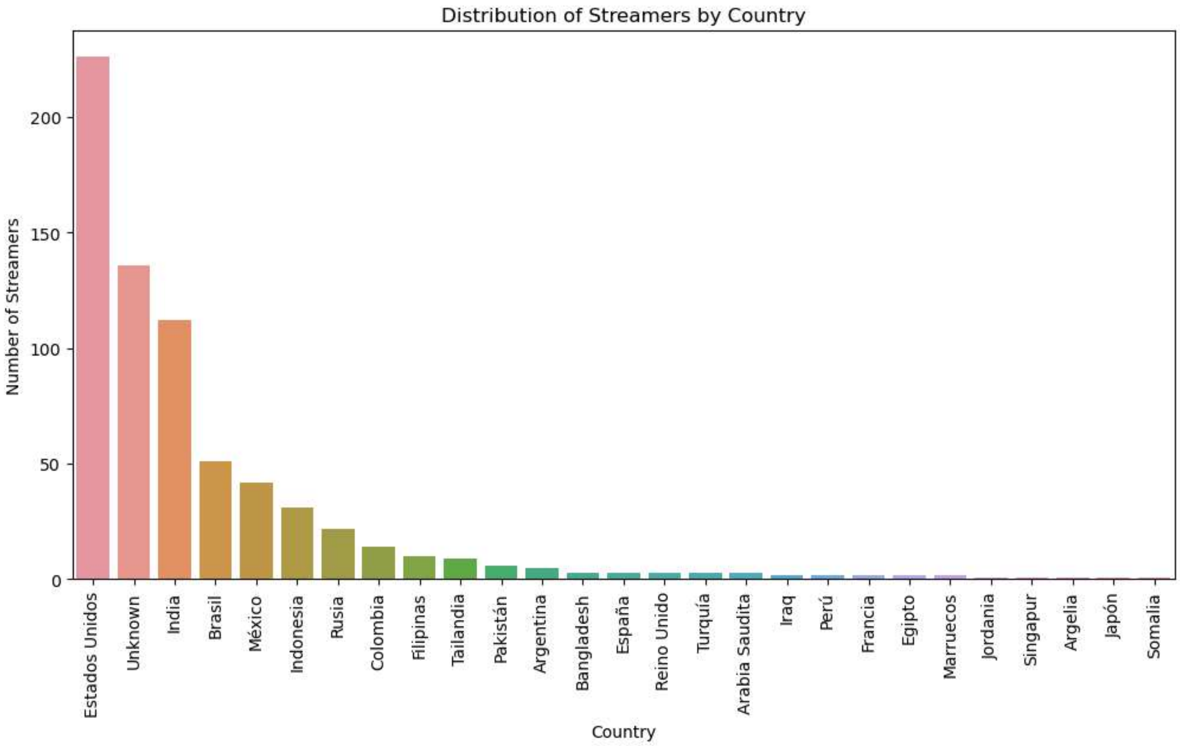
The value is 0.248389. This indicates a positive but weak correlation. This means that as the number of subscribers increases, the number of likes tends to increase, but the relationship is not very strong. The value is 0.037293. This indicates a very weak positive correlation. This suggests that there is almost no relationship between the number of subscribers and the number of comments.

In [18]:
```
import seaborn as sns
import matplotlib.pyplot as plt
plt.figure(figsize=(8, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', center=0)
plt.title('Correlation Matrix Heatmap')
plt.show()
```



In [26]:

```
country_counts = data['Country'].value_counts()
plt.figure(figsize=(12, 6))
sns.barplot(x=country_counts.index, y=country_counts.values)
plt.title('Distribution of Streamers by Country')
plt.xlabel('Country')
plt.ylabel('Number of Streamers')
plt.xticks(rotation=90)
plt.show()
```



In [37]:

```
average_metrics = data[['Subscribers', 'Visits', 'Likes', 'Comments']].mean()
average_metrics
```

Out[37]:

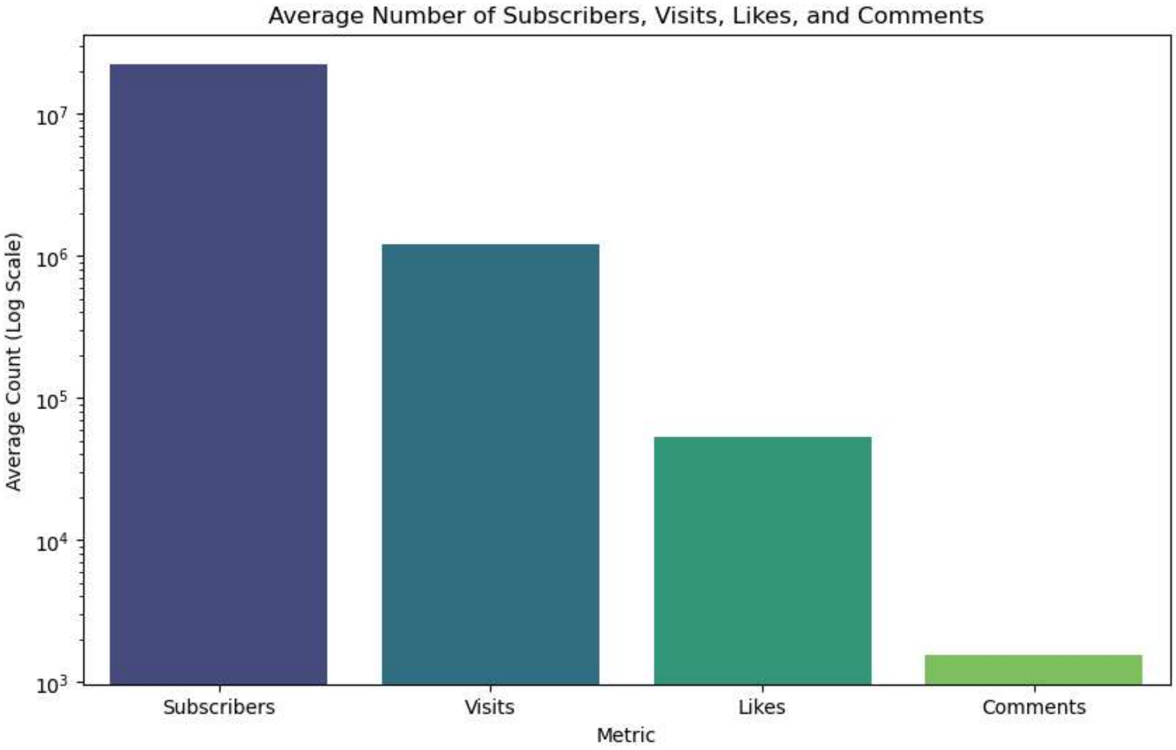
```
Subscribers    2.241556e+07
Visits         1.210730e+06
Likes          5.347360e+04
Comments       1.558794e+03
dtype: float64
```

The average number of subscribers is extremely high (22 million). This suggests that the dataset consists of very popular YouTube streamers The average number of visits (1.2 million) is significantly lower than the number of subscribers.This could imply that while these streamers have a large subscriber base, only a portion of the subscribers regularly visit the streamer's content. The average number of likes (53,473) is much lower than the number of visits.This pattern is expected since not all viewers who visit the content will like it. The average number of comments (1,559) is the lowest among the metrics. This suggests that a very small percentage of viewers engage with the content by commenting. This is typical as commenting requires more effort compared to liking a video.

In [39]:

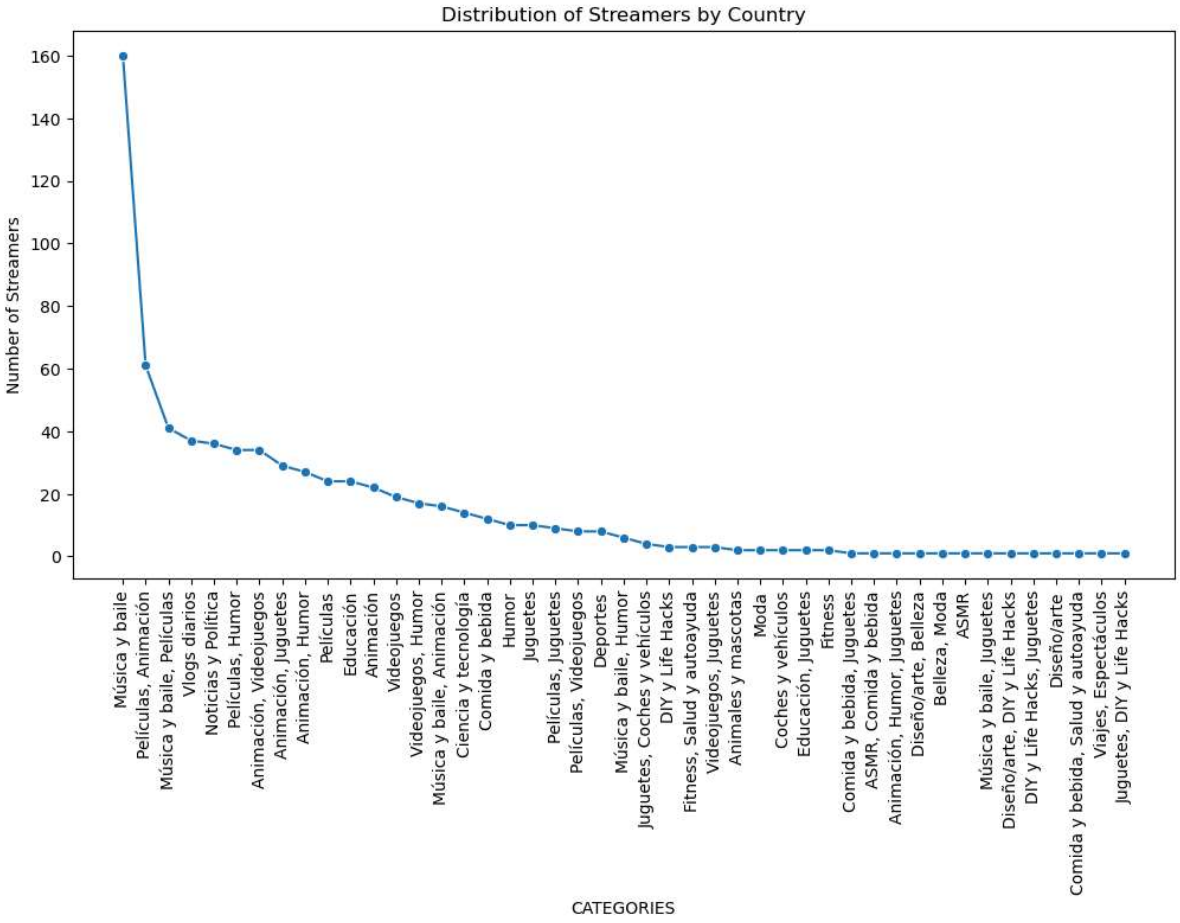
```
# Define the average metrics
average_metrics = pd.Series({
    'Subscribers': 2.241556e+07,
    'Visits': 1.210730e+06,
    'Likes': 5.347360e+04,
    'Comments': 1.558794e+03
})

plt.figure(figsize=(10, 6))
sns.barplot(x=average_metrics.index, y=average_metrics.values, palette='viridi
plt.yscale('log')  # Using log scale for better visualization of large differe
plt.title('Average Number of Subscribers, Visits, Likes, and Comments')
plt.xlabel('Metric')
plt.ylabel('Average Count (Log Scale)')
plt.show()
```



There is a clear pattern of decreasing engagement from subscribers to visits to likes to comments. This is expected as the level of engagement effort increases.

In [52]:
```
counts = data['Categories'].value_counts()
plt.figure(figsize=(12, 6))
sns.lineplot(x=counts.index, y=counts.values, marker='o')
plt.title('Distribution of Streamers by Country')
plt.xlabel('CATEGORIES')
plt.ylabel('Number of Streamers')
plt.xticks(rotation=90)
plt.show()
```



In [54]:

```
above_average_streamers = data[
    (data['Subscribers'] > average_metrics['Subscribers']) &
    (data['Visits'] > average_metrics['Visits']) &
    (data['Likes'] > average_metrics['Likes']) &
    (data['Comments'] > average_metrics['Comments'])
]
top_performers = above_average_streamers.sort_values(by='Subscribers', ascendi
top_performers
```

Out[54]:

	Rank	Username	Categories	Subscribers	Country	Visits	Likes	Co
1	2	MrBeast	Videojuegos, Humor	183500000	Estados Unidos	117400000	5300000	
5	6	PewDiePie	Películas, Videojuegos	111500000	Estados Unidos	2400000	197300	
26	27	dudeperfect	Videojuegos	59700000	Estados Unidos	5300000	156500	
34	35	TaylorSwift	Música y baile	54100000	Estados Unidos	4300000	300400	
39	40	JuegaGerman	Películas, Animación	48600000	México	2000000	117100	
43	44	A4a4a4a4	Animación, Humor	47300000	Rusia	9700000	330400	
58	59	Mikecrack	Películas, Animación	43400000	México	2200000	183400	

MrBeast is the top on the basis of Subscribers as compare to other content creaters

In [ ]: