
NAME:THANGANEDZO ESHIRLY MAPHUTHA
DATA ANALYST INTERN CAREER
Task 1: YouTube Streamer Analysis

In [71]:
```
#importing the necessery libabries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [ ]:
```
# Data Exploration:
```

In [72]:
```
 #let our data stored in data
data=pd.read_csv("desktop\DS intern\youtubers_df.csv")
data
```

Out[72]:

	Rank	Username	Categories	Suscribers	Country	Visits	Likes	Comme
0	1	tseries	Música y baile	249500000	India	86200.0	2700	
1	2	MrBeast	Videojuegos, Humor	183500000	Estados Unidos	117400000.0	5300000	18
2	3	CoComelon	Educación	165500000	Unknown	7000000.0	24700	
3	4	SETIndia	NaN	162600000	India	15600.0	166	
4	5	KidsDianaShow	Animación, Juguetes	113500000	Unknown	3900000.0	12400	
...	...	...	...	...	...	...	...	...
995	996	hamzymukbang	NaN	11700000	Estados Unidos	397400.0	14000	
996	997	Adaahqueen	NaN	11700000	India	1100000.0	92500	
997	998	LittleAngelIndonesia	Música y baile	11700000	Unknown	211400.0	745	
998	999	PenMultiplex	NaN	11700000	India	14000.0	81	
999	1000	OneindiaHindi	Noticias y Política	11700000	India	2200.0	31	

1000 rows × 9 columns

In [ ]:
```
# This data contains 1000 rows and 9 unique columns
```

In [73]:
```
data.shape
```

Out[73]: (1000, 9)

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   Suscribers   1000 non-null   int64  
 4   Country      1000 non-null   object 
 5   Visits       1000 non-null   float64
 6   Likes        1000 non-null   int64  
 7   Comments     1000 non-null   int64  
 8   Links        1000 non-null   object 
dtypes: float64(1), int64(4), object(4)
memory usage: 70.4+ KB
```

In [ ]:
```
# checking for the column contains null value
```

In [6]:
```
data.isnull().sum()
```

Out[6]:
```
Rank            0
Username        0
Categories    306
Suscribers      0
Country         0
Visits          0
Likes           0
Comments        0
Links           0
dtype: int64
```

In [ ]:
```
# replacing null value with unknown
```

In [38]:
```
data['Categories'].fillna('Unknown', inplace=True)
data.isnull().sum()
```

Out[38]:
```
Rank          0
Username      0
Categories    0
Suscribers    0
Country       0
Visits        0
Likes         0
Comments      0
Links         0
dtype: int64
```

In [ ]:
```
#Removing ''Like'' column which is not key variable
```

In [40]:
```
data.drop(columns=['Links'], inplace=True)
data
```

Out[40]:

	Rank	Username	Categories	Suscribers	Country	Visits	Likes	Comme
0	1	tseries	Música y baile	249500000	India	86200.0	2700	
1	2	MrBeast	Videojuegos, Humor	183500000	Estados Unidos	117400000.0	5300000	18
2	3	CoComelon	Educación	165500000	Unknown	7000000.0	24700	
3	4	SETIndia	Unknown	162600000	India	15600.0	166	
4	5	KidsDianaShow	Animación, Juguetes	113500000	Unknown	3900000.0	12400	
...	...	...	...	...	...	...	...	
995	996	hamzymukbang	Unknown	11700000	Estados Unidos	397400.0	14000	
996	997	Adaahqueen	Unknown	11700000	India	1100000.0	92500	
997	998	LittleAngelIndonesia	Música y baile	11700000	Unknown	211400.0	745	
998	999	PenMultiplex	Unknown	11700000	India	14000.0	81	
999	1000	OneindiaHindi	Noticias y Política	11700000	India	2200.0	31	

1000 rows × 8 columns

In [7]:
```
Maths_desc= data.describe()
Maths_desc
```

Out[7]:

	Rank	Suscribers	Visits	Likes	Comments
count	1000.000000	1.000000e+03	1.000000e+03	1.000000e+03	1000.000000
mean	500.500000	2.189440e+07	1.209446e+06	5.363259e+04	1288.768000
std	288.819436	1.682775e+07	5.229942e+06	2.580457e+05	6778.188308
min	1.000000	1.170000e+07	0.000000e+00	0.000000e+00	0.000000
25%	250.750000	1.380000e+07	3.197500e+04	4.717500e+02	2.000000
50%	500.500000	1.675000e+07	1.744500e+05	3.500000e+03	67.000000
75%	750.250000	2.370000e+07	8.654750e+05	2.865000e+04	472.000000
max	1000.000000	2.495000e+08	1.174000e+08	5.300000e+06	154000.000000

In [44]:
```
min_values=Maths_desc.loc['min']
min_values
```

Out[44]:
```
Rank                  1.0
Suscribers     11700000.0
Visits                0.0
Likes                 0.0
Comments              0.0
Name: min, dtype: float64
```

In [45]:
```
max_values=Maths_desc.loc['max']
max_values
```

Out[45]:
```
Rank                1000.0
Suscribers     249500000.0
Visits         117400000.0
Likes            5300000.0
Comments          154000.0
Name: max, dtype: float64
```

In [46]:
```
quartiles=Maths_desc.loc[['25%','50%','75%']]
quartiles
```

Out[46]:

	Rank	Suscribers	Visits	Likes	Comments
25%	250.75	13800000.0	31975.0	471.75	2.0
50%	500.50	16750000.0	174450.0	3500.00	67.0
75%	750.25	23700000.0	865475.0	28650.00	472.0

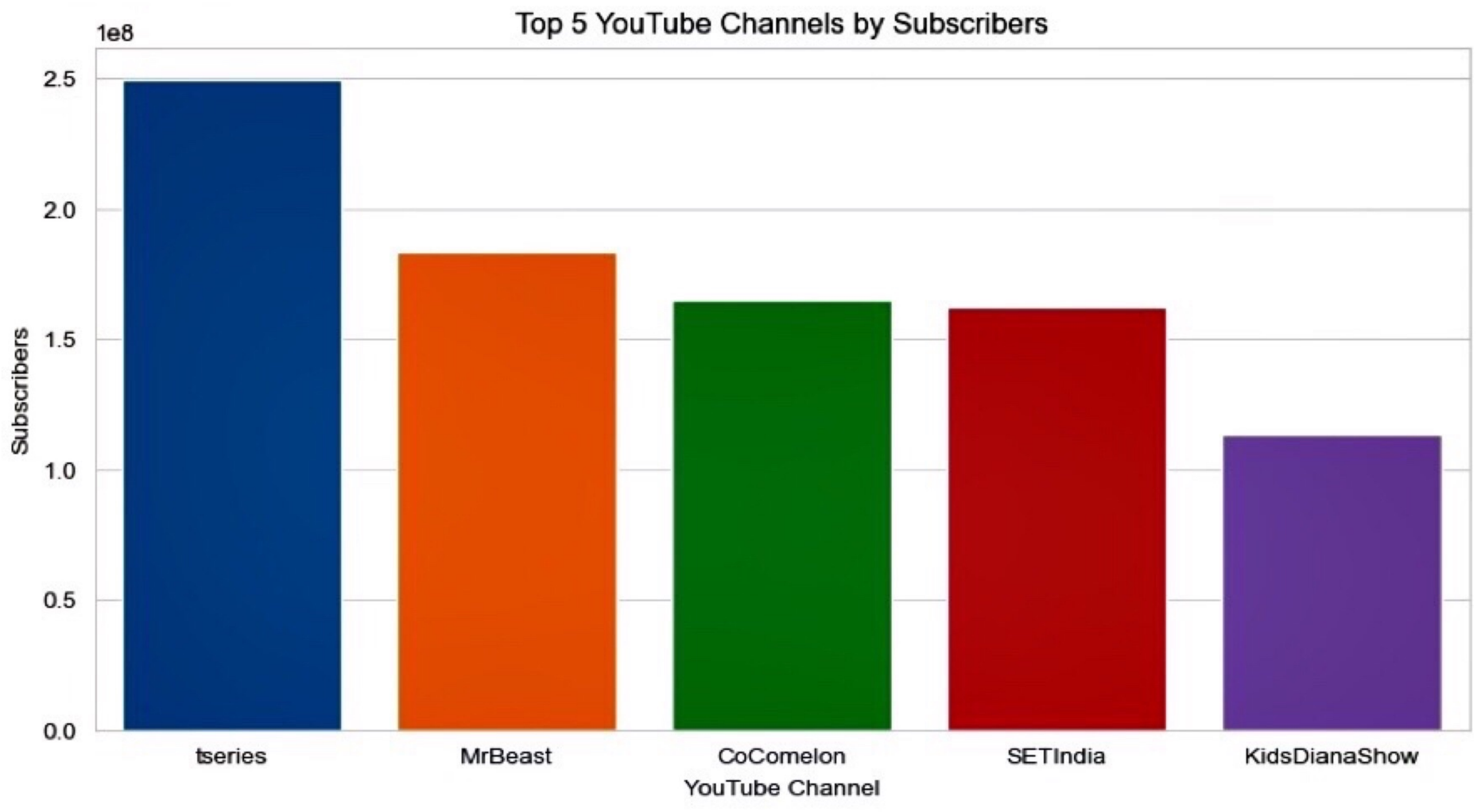
In [ ]:
```
#Trend Analysis:
```

In [11]:
```
data['Rank'] = pd.to_numeric(data['Rank'], errors='coerce')
```

In [12]:
```
sns.set_style("whitegrid")
```

In [13]:
```
#Identify top streamers interms of number of subscibers

top_five_subscribers = data.sort_values(by='Suscribers', ascending=False).head
plt.figure(figsize=(10, 5))
sns.barplot(x='Username', y='Suscribers', data=top_five_subscribers)
plt.title('Top 5 YouTube Channels by Subscribers')
plt.xlabel('YouTube Channel')
plt.ylabel('Subscribers')
plt.show()
```

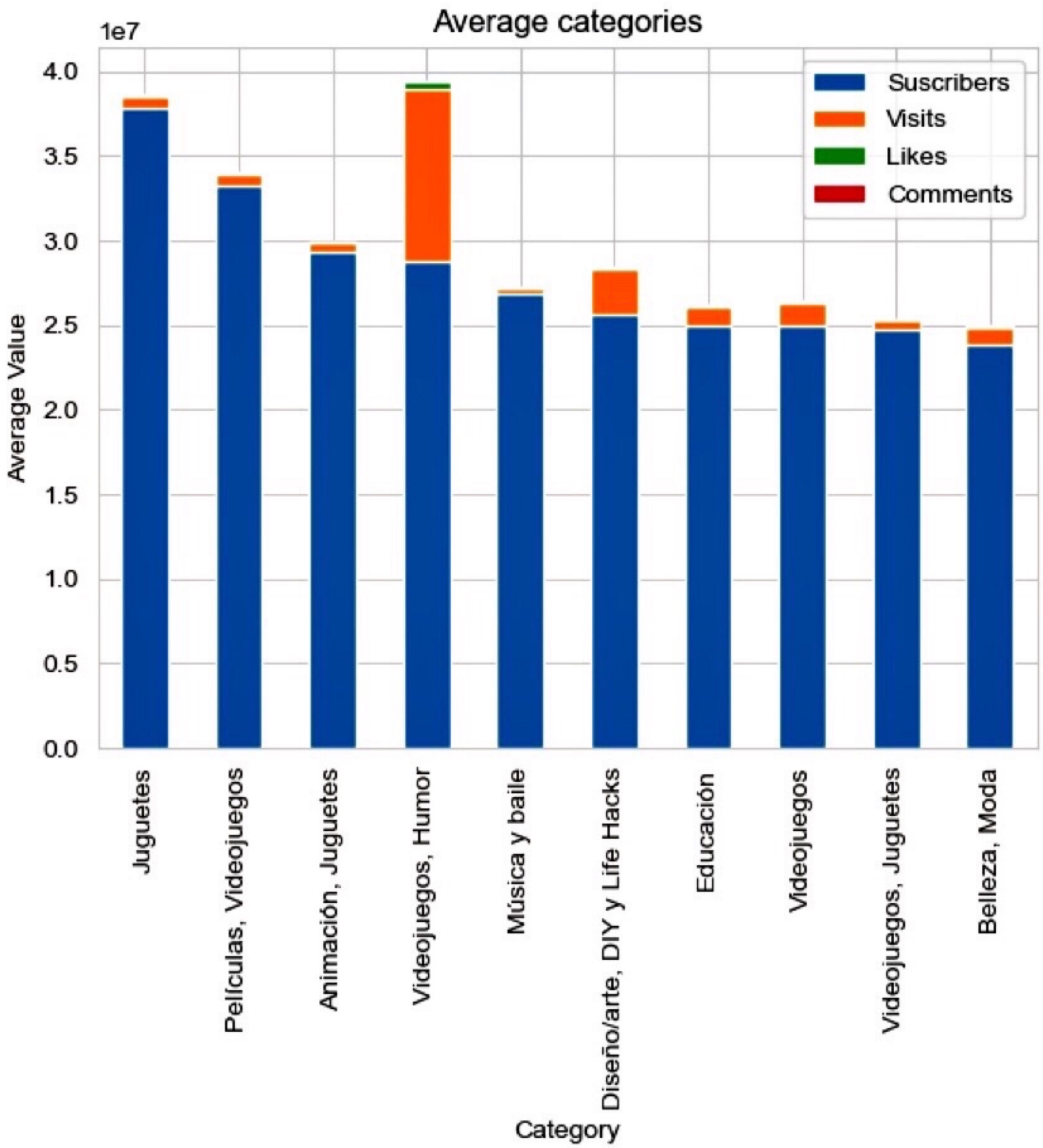


tseries is the top streamer interms of number of subscribers.

In [15]:
```
#Identify top streamers interms of categories
numeric_columns=['Suscribers','Visits','Likes','Comments']
```

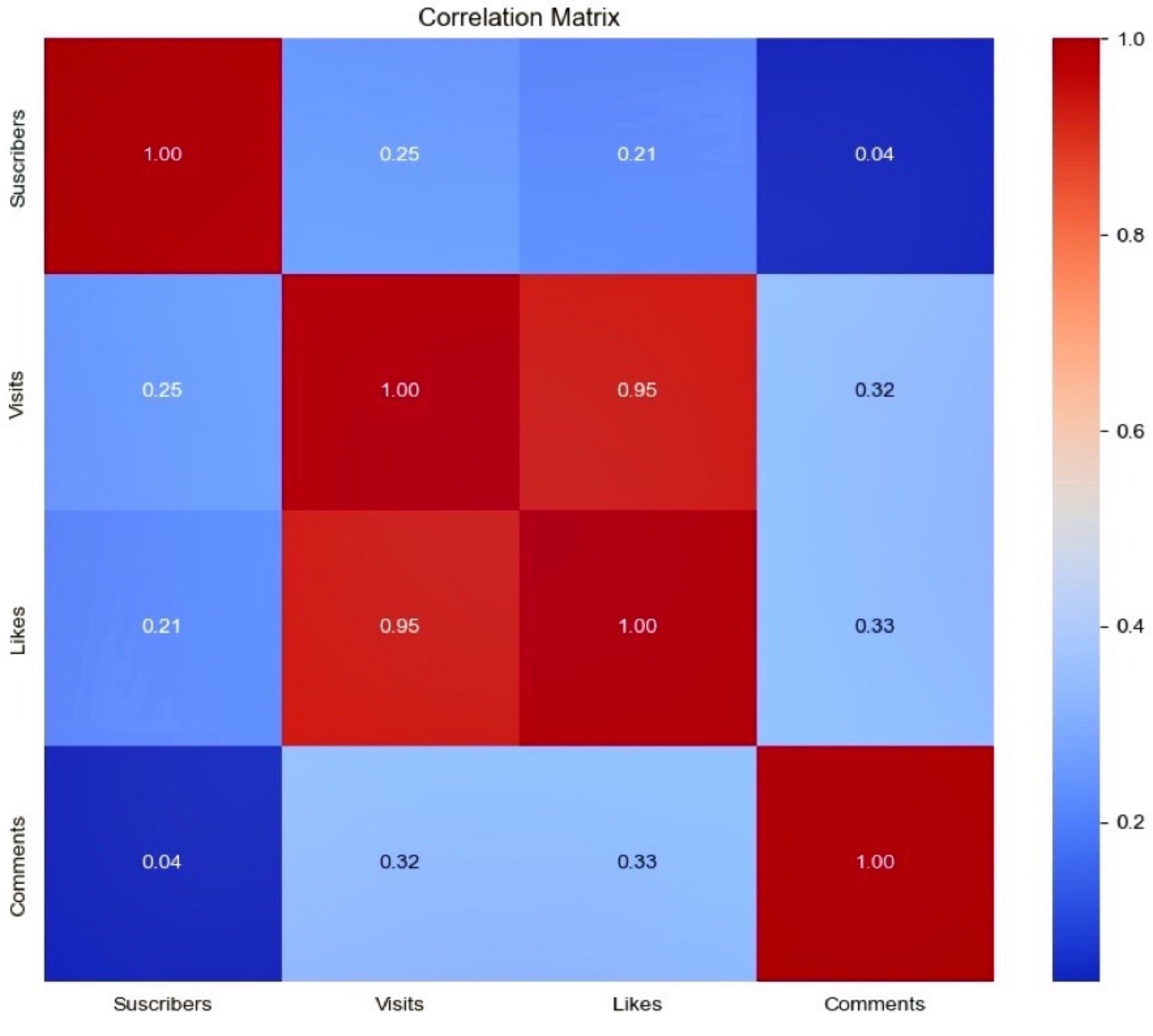
In [16]:
```
category_analysis=data.groupby('Categories')[numeric_columns].mean()
category_analysis=category_analysis.sort_values(by='Suscribers', ascending=Fal
plt.figure(figsize=(12, 8))
category_analysis.plot(kind='bar', stacked=True)
plt.ylabel('Average Value')
plt.xlabel('Category')
plt.title('Average categories')
plt.show()
```

```
<Figure size 1200x800 with 0 Axes>
```



In [18]:
```
# identify correlation interms of 'Suscribers','Visits','Likes', and 'Comments
numeric_columns=['Suscribers','Visits','Likes','Comments']
correlation_matrix = data[numeric_columns].corr()
```

In [19]:
```
#correlation matrix
plt.figure(figsize=(10,8))
sns.heatmap(correlation_matrix, annot=True, cmap="coolwarm",fmt=".2f")
plt.title("Correlation Matrix")
plt.show()
```

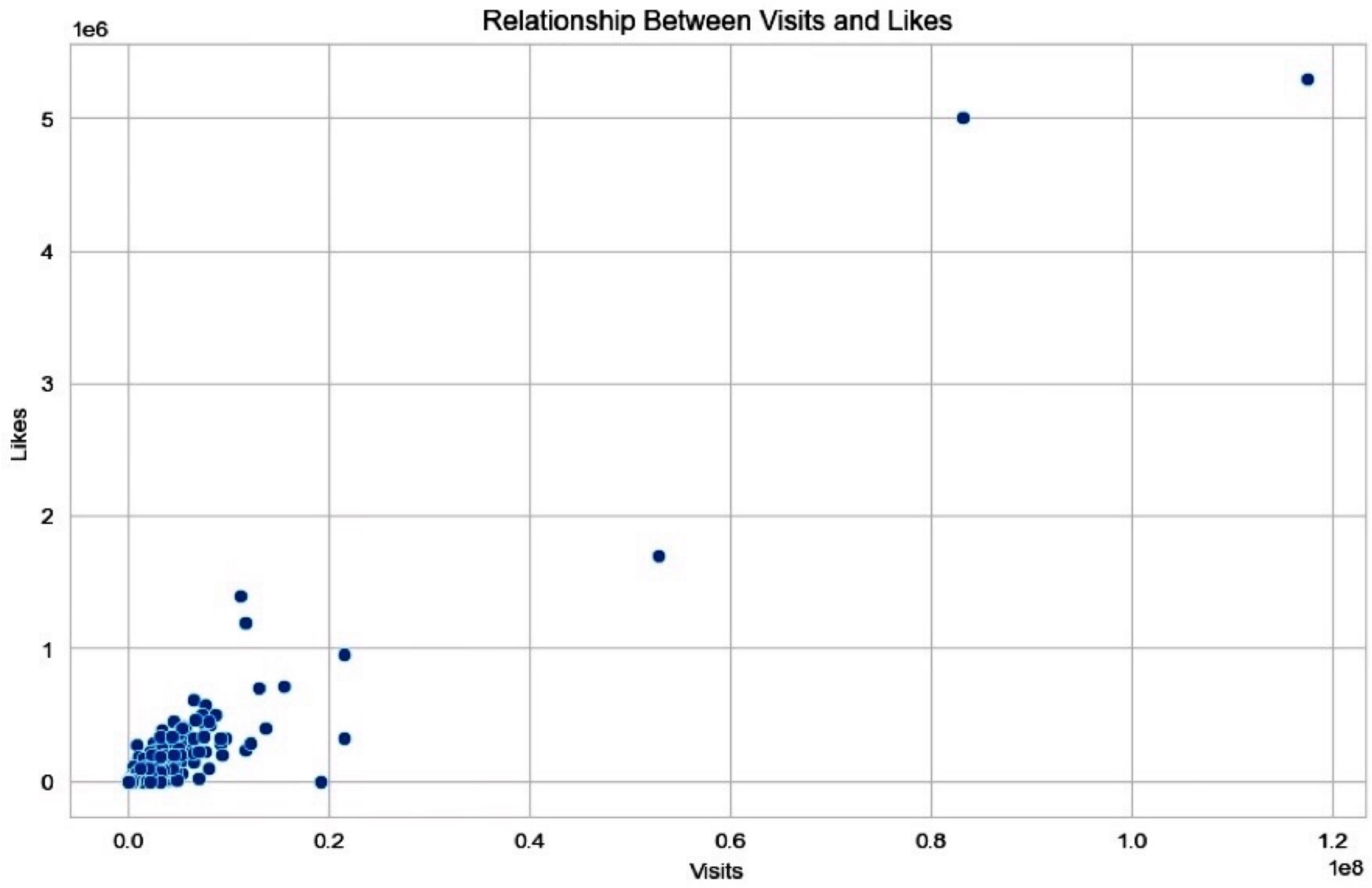


The above heatmap depicts that the streamer which have more visits contains highest likes.

In [20]:
```
total_categories = data["Categories"].value_counts()
total_categories.head(15)
```

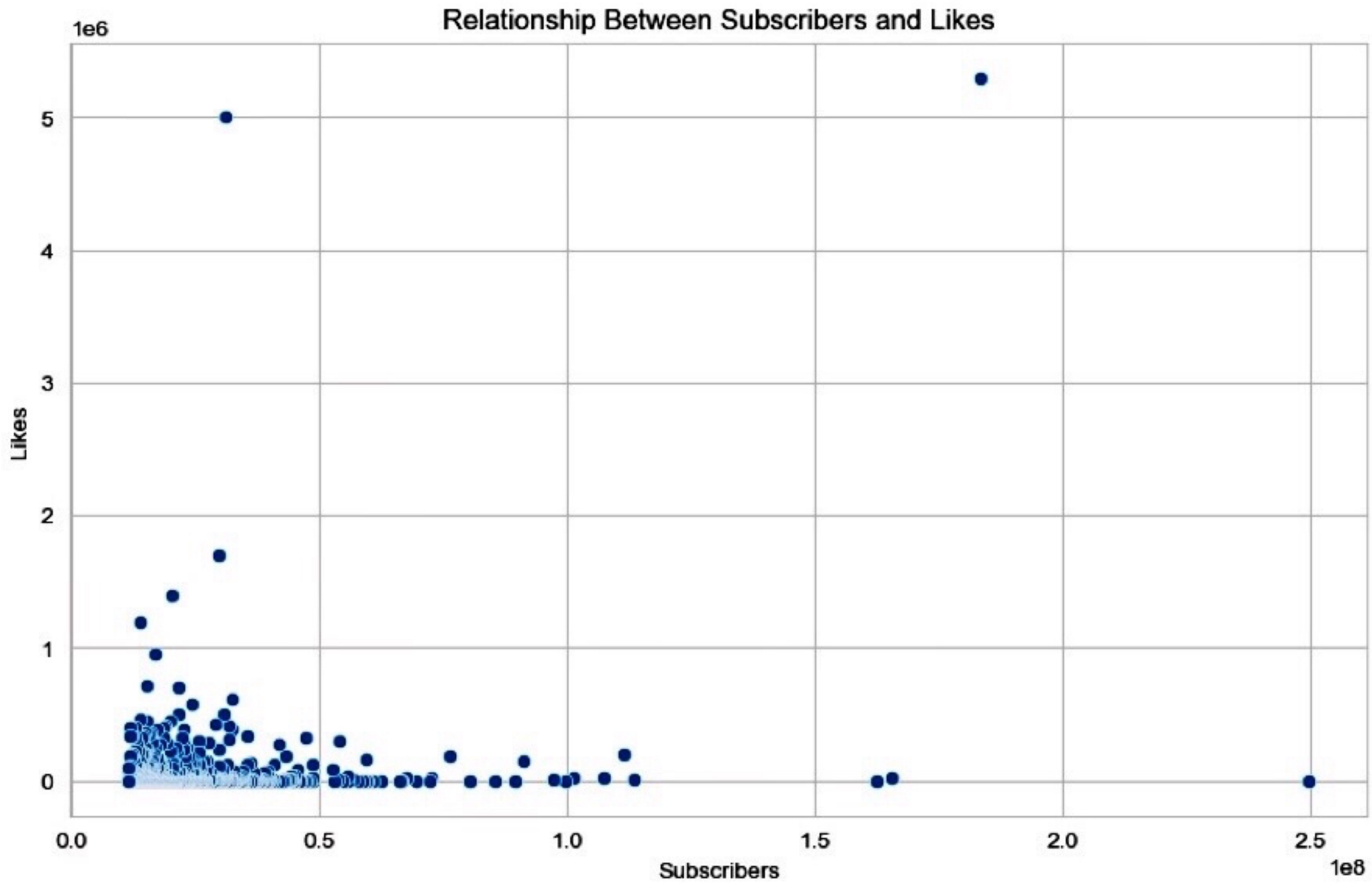
Out[20]:
```
Categories
Música y baile                160
Películas, Animación           61
Música y baile, Películas      41
Vlogs diarios                  37
Noticias y Política            36
Películas, Humor               34
Animación, Videojuegos         34
Animación, Juguetes            29
Animación, Humor               27
Películas                      24
Educación                      24
Animación                      22
Videojuegos                    19
Videojuegos, Humor             17
Música y baile, Animación      16
Name: count, dtype: int64
```

In [69]:
```
# correlation between subscriber & Likes
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Visits', y='Likes', data=data)
plt.title('Relationship Between Visits and Likes')
plt.xlabel('Visits')
plt.ylabel('Likes')
plt.show()
```

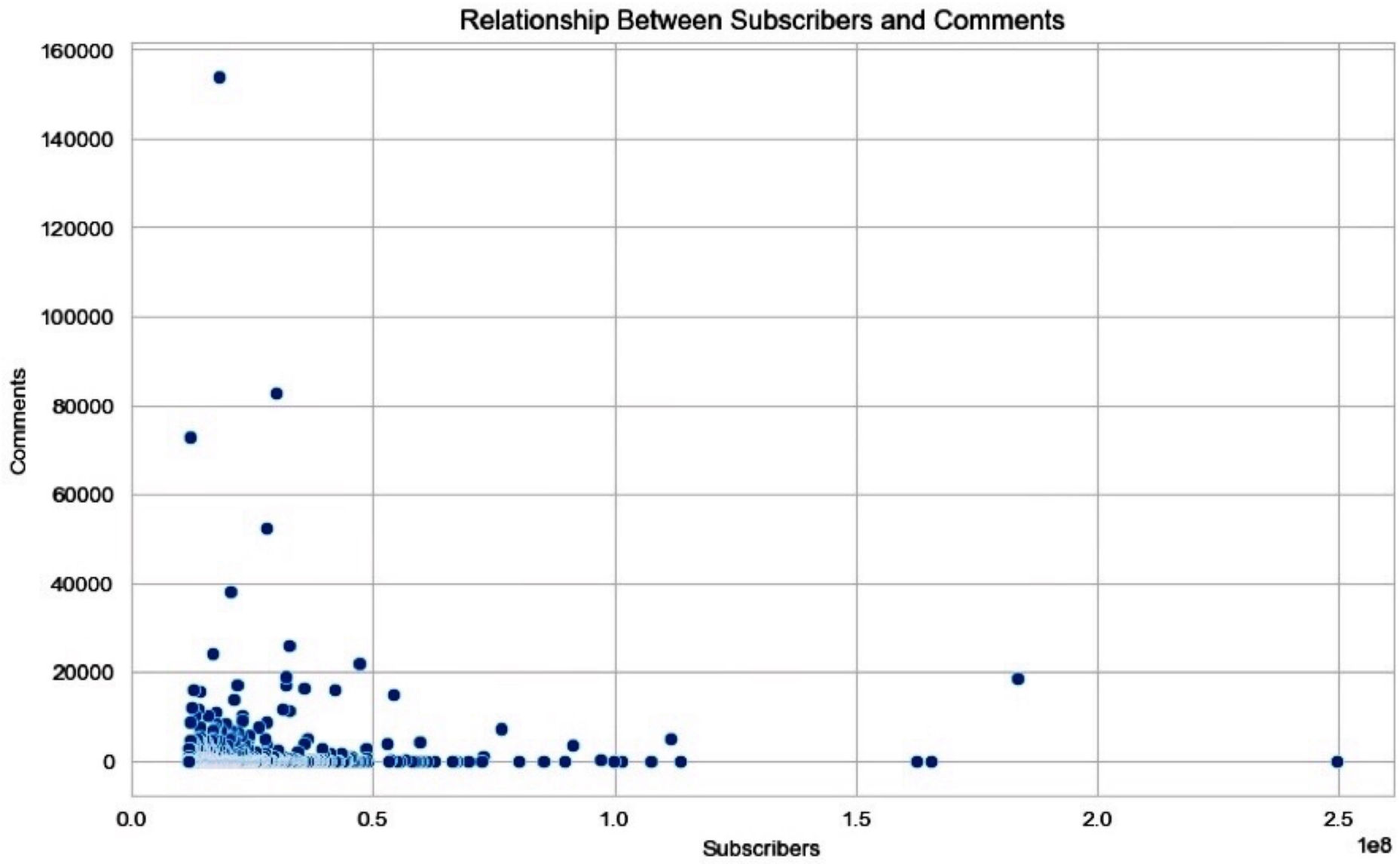


Good realtionship between visits and likes

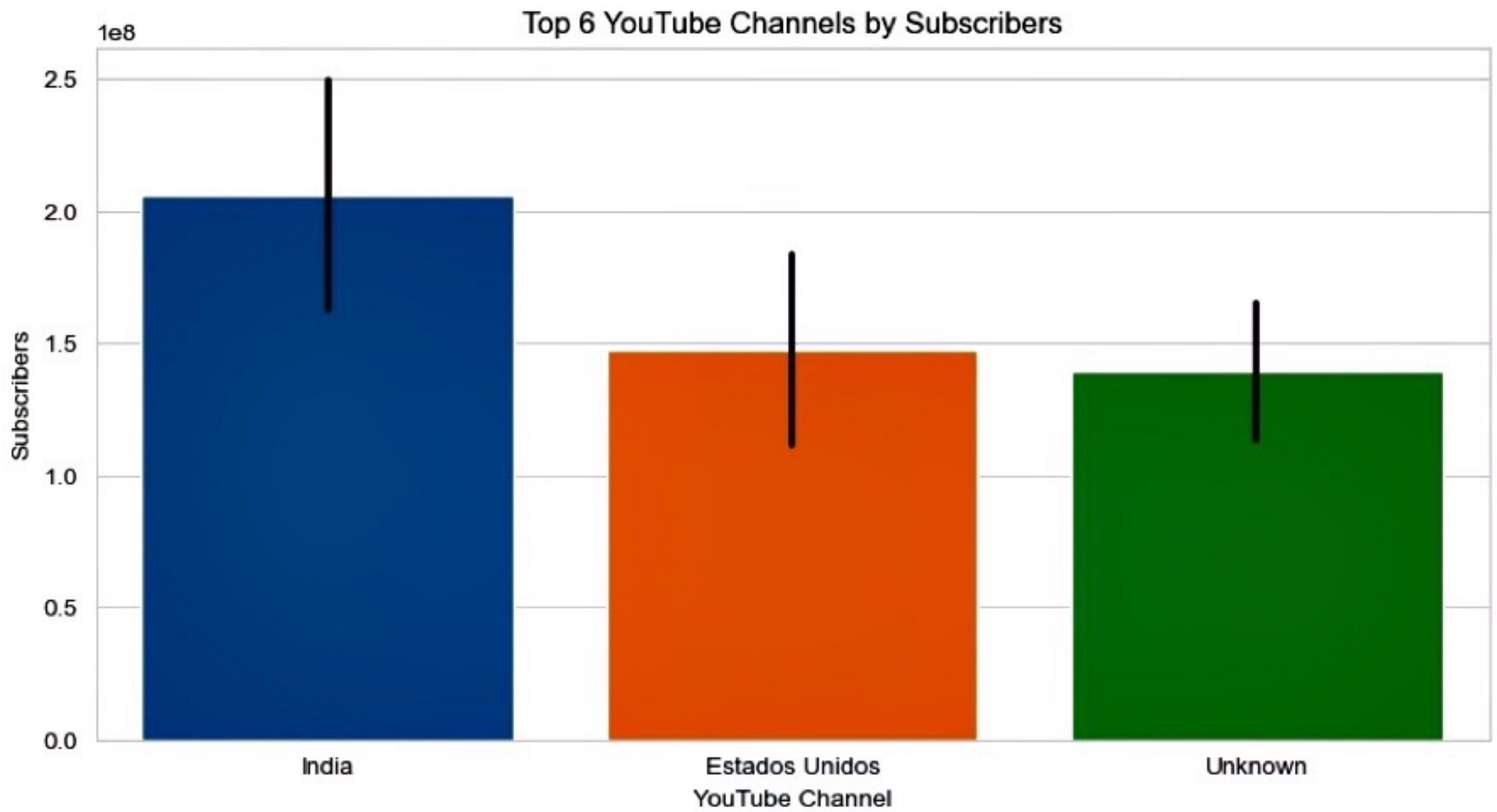
In [63]:
```
# correlation between subscriber & Likes
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Suscribers', y='Likes', data=data)
plt.title('Relationship Between Subscribers and Likes')
plt.xlabel('Subscribers')
plt.ylabel('Likes')
plt.show()
```



In [64]:
```
# correlation between subscriber & comments                    #  # week r
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Suscribers', y='Comments', data=data)
plt.title('Relationship Between Subscribers and Comments')
plt.xlabel('Subscribers')
plt.ylabel('Comments')
plt.show()
```

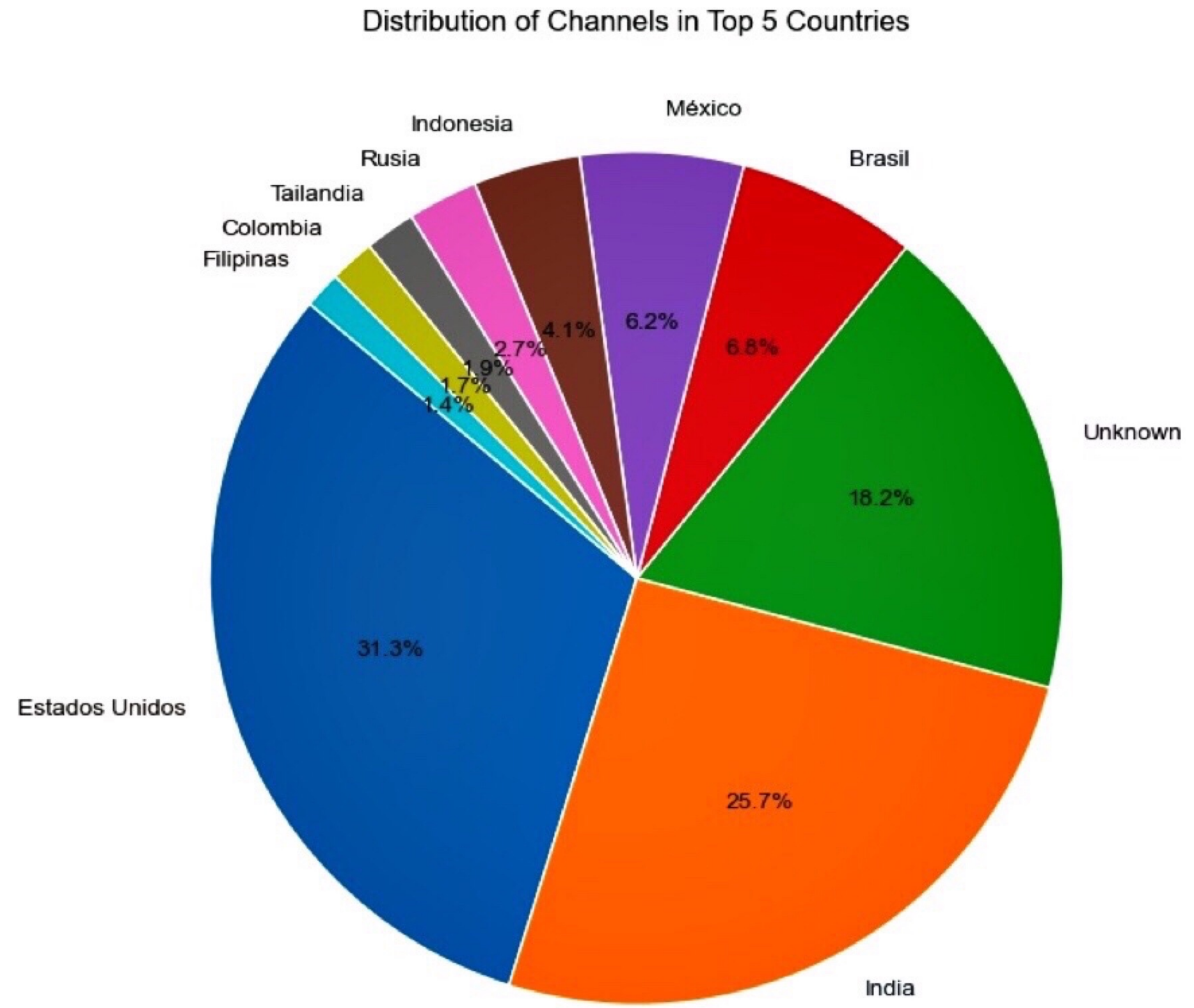


In [23]:
```
top_six_subscribers = data.sort_values(by='Suscribers', ascending=False).head(
plt.figure(figsize=(10, 5))
sns.barplot(x='Country', y='Suscribers', data=top_six_subscribers)
plt.title('Top 6 YouTube Channels by Subscribers')
plt.xlabel('YouTube Channel')
plt.ylabel('Subscribers')
plt.show()
```



In [ ]:
```
#Audience study
```

In [24]:
```
top_countries = data['Country'].value_counts().head(10)  # Top 10 countries
plt.figure(figsize=(10, 8))
plt.pie(top_countries, labels=top_countries.index, autopct='%1.1f%%', startang
plt.title('Distribution of Channels in Top 5 Countries')
plt.show()
```



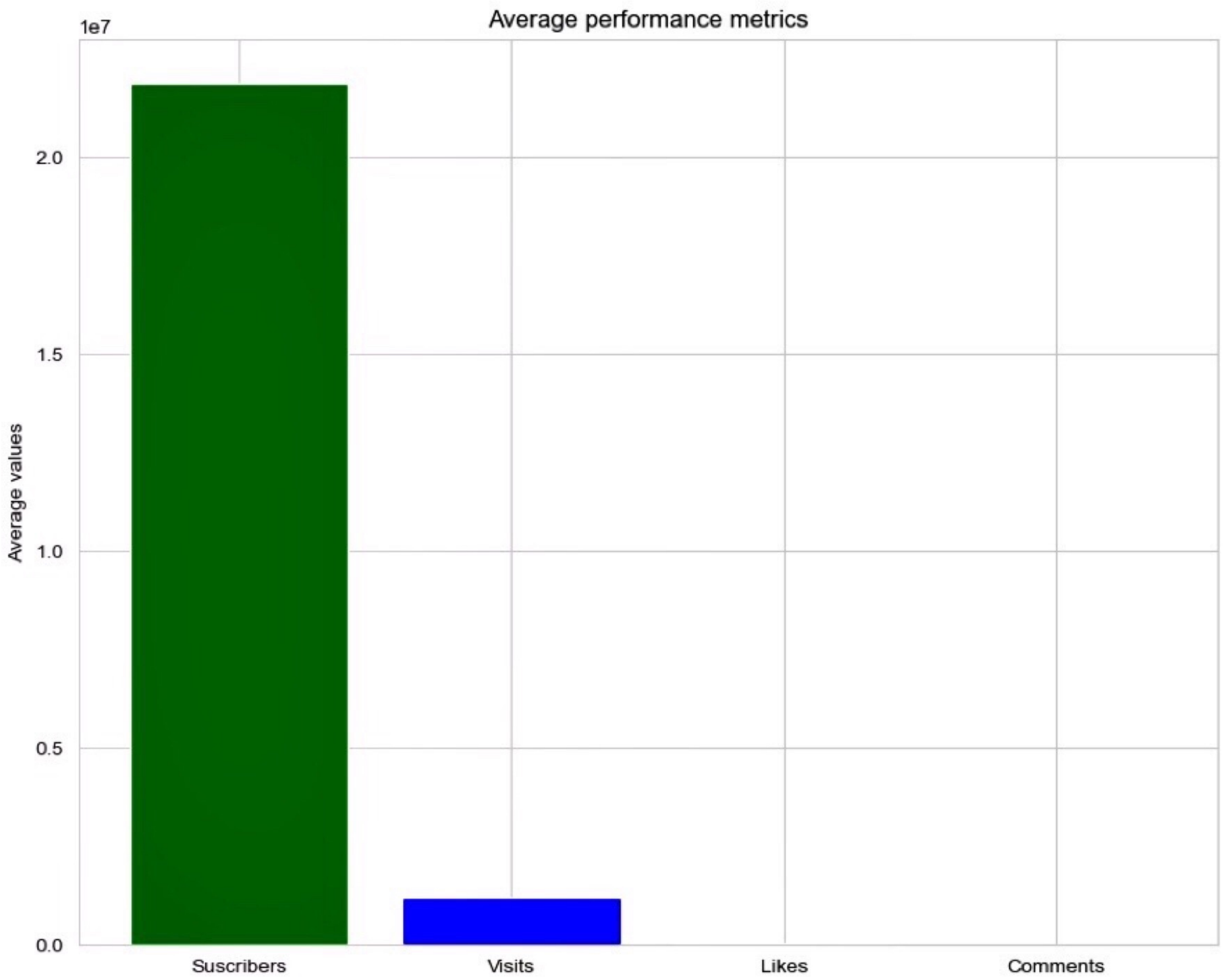
The above pie chart depicts that top distriburion of channels interms of country. So, Estados Unidos is the top country which have highest channels.

In [ ]:
```
#Anlyse regional preferences for specific content categories
```

In [26]:
```
plt.figure(figsize=(15, 8))
sns.countplot(data=data, x='Country', hue='Categories')
plt.ylabel('Username')
plt.xlabel('Country')
plt.title('Regional preferences for content categories')
plt.legend(title='Category',bbox_to_anchor=(1.50, 1), loc='upper left')
plt.show()
```

In [ ]:
```
#Calculate Average values
```

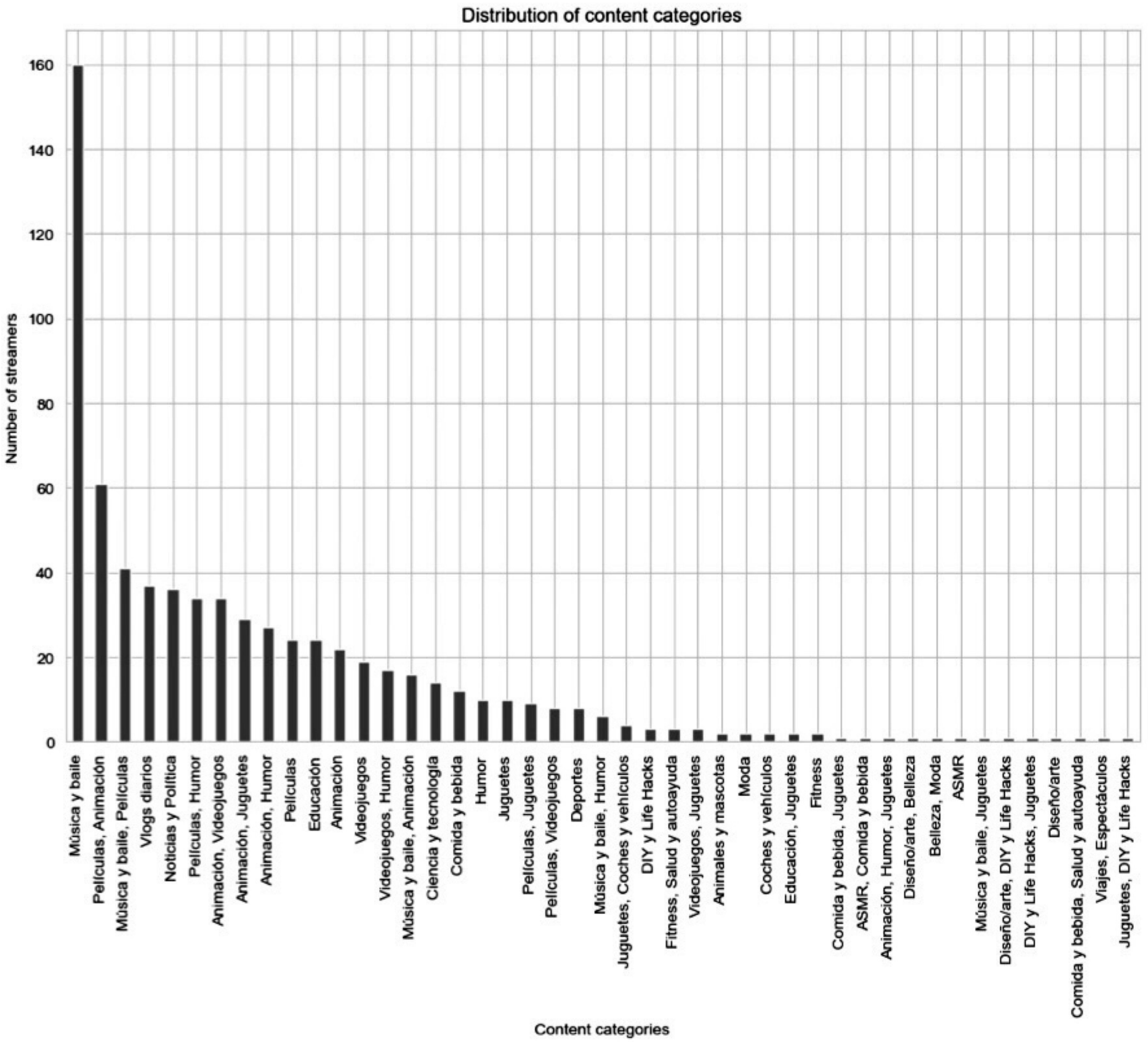
In [70]:
```
average_Suscribers = data['Suscribers'].mean()
average_Visits = data['Visits'].mean()
average_Likes= data['Likes'].mean()
average_Comments= data['Comments'].mean()
metrix=['Suscribers','Visits','Likes','Comments']
average_values=[average_Suscribers,average_Visits,average_Likes,average_Commen
plt.figure(figsize=(10,8),)
plt.bar(metrix,average_values, color=['green','blue','red','grey'])
plt.ylabel('Average values')
plt.title('Average performance metrics')
plt.show()
```



If a metric unusually high or low relative others, then it's anomalies

In [31]:
```
#Content Categories

Categories_distributions=data['Categories'].value_counts()
plt.figure(figsize=(12,8))
Categories_distributions.plot(kind='bar', color='grey')
plt.ylabel('Number of streamers')
plt.xlabel('Content categories')
plt.title('Distribution of content categories')
plt.show()
```

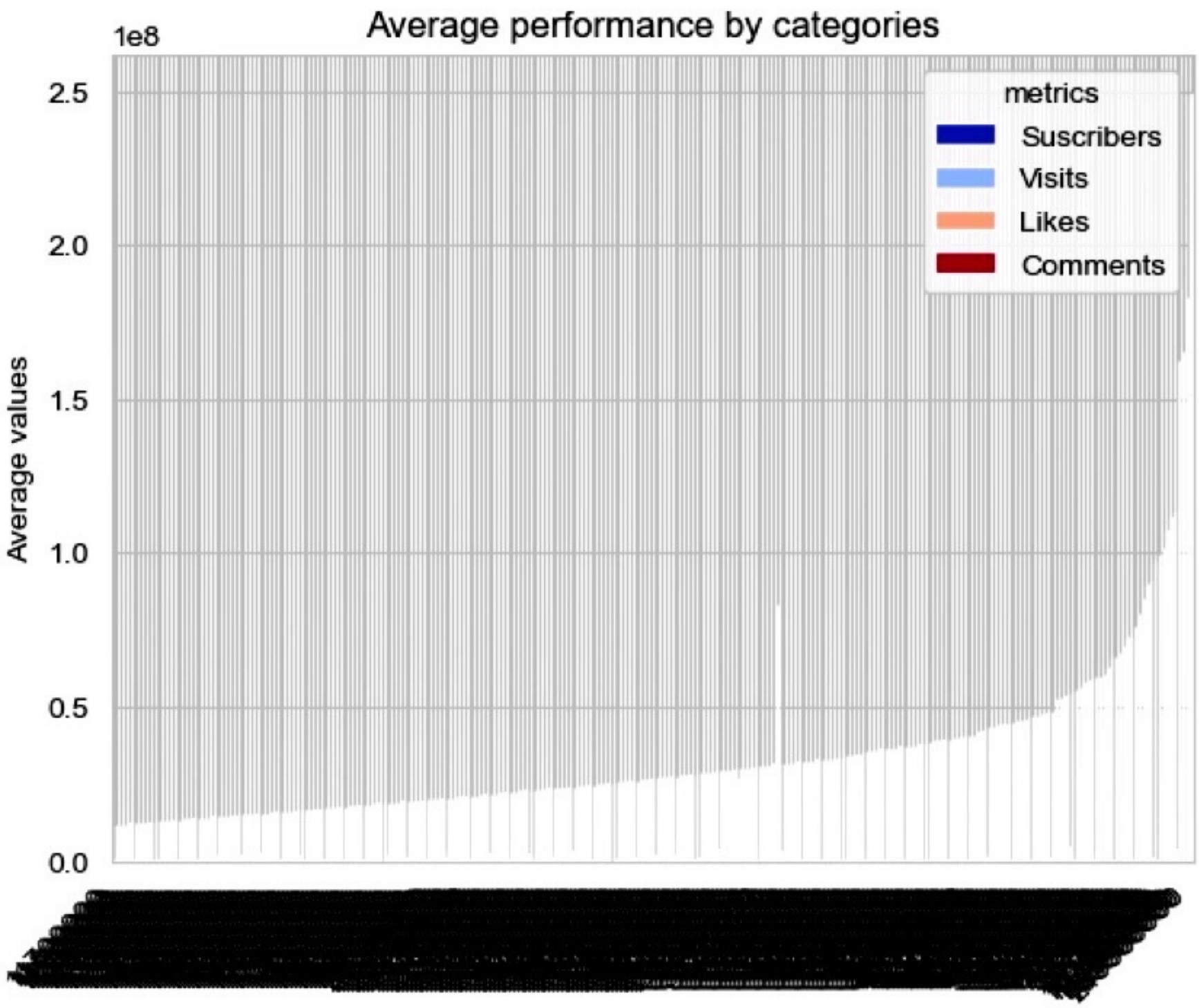


From the above categories distribution, Musica y baile has the highest streamers.

In [38]:
```
#Brands and Collaborations:

list_columns=['Suscribers', 'Visits', 'Likes', 'Comments']
Avregae_by_categories=data.groupby('Suscribers')[list_columns].mean()
plt.figure(figsize=(12,8))
Avregae_by_categories.plot(kind='bar', colormap='coolwarm')
plt.ylabel('Average values')
plt.xlabel('Content categories')
plt.title('Average performance by categories')
plt.xticks(rotation=45, ha='right')
plt.legend(title='metrics')
plt.show()
```

```
<Figure size 1200x800 with 0 Axes>
```



Benchmarking:

Identify streamers with above-average performance in terms of subscribers, visits, likes, and comments.
Who are the top-performing content creators?

In [59]:
```
lists_columns=['Suscribers', 'Visits', 'Likes', 'Comments']
Average_by_categories=data[lists_columns].mean()

Above_average_streamers=data[(data['Suscribers']>Average_by_categories['Suscri
                              &(data['Visits']>Average_by_categories['Visits'])
                              &(data['Likes']>Average_by_categories['Likes'])
                              &(data['Comments']>Average_by_categories['Comment

Top_performing_creaters=Above_average_streamers[['Username','Suscribers','Visi
Top_performing_creaters=Top_performing_creaters.sort_values(by='Suscribers', a

Top_performing_creaters
```

Out[59]:

	Username	Suscribers	Visits	Likes	Comments
1	MrBeast	183500000	117400000.0	5300000	18500
5	PewDiePie	111500000	2400000.0	197300	4900
26	dudeperfect	59700000	5300000.0	156500	4200
34	TaylorSwift	54100000	4300000.0	300400	15000
39	JuegaGerman	48600000	2000000.0	117100	3000
43	A4a4a4a4	47300000	9700000.0	330400	22000
58	Mikecrack	43400000	2200000.0	183400	1800
62	KimberlyLoaiza	42100000	5300000.0	271300	16000
64	luisitocomunica	41100000	2500000.0	128900	1800
70	JessNoLimit	39600000	1300000.0	73500	1600
96	TotalGaming093	36300000	1500000.0	129400	4900
98	TechnoGamerzOfficial	35600000	6200000.0	341800	16500
100	markiplier	35500000	2100000.0	126500	3800
122	AboFlah	32700000	3300000.0	382000	11400
123	MRINDIANHACKER	32600000	6500000.0	617400	26000
131	fedevigevani	32000000	7700000.0	412200	17000
132	dream	31900000	3300000.0	309200	19000
136	MrBeast2	31300000	83100000.0	5000000	11600
145	jacksepticeye	30400000	1600000.0	83400	2300
153	DaFuqBoom	29800000	52700000.0	1700000	82800
176	CrazyXYZ	27800000	4200000.0	284100	8600
177	DanTDM	27800000	3500000.0	285000	52500
179	brentrivera	27600000	6400000.0	154100	5000
180	NichLmao	27500000	1500000.0	85800	1600
195	nickiminaj	26100000	1600000.0	98300	7600
206	AlejoIgoa	25700000	5700000.0	208400	1700

	Username	Suscribers	Visits	Likes	Comments
207	ZHCYT	25700000	2600000.0	127300	2200
234	rug	24300000	3200000.0	85300	5100
238	alanbecker	24300000	7600000.0	582600	5900
241	juandediospantojaa	24000000	3000000.0	133200	3600
266	DrossRotzank	23100000	1700000.0	105900	3900
272	AmiRodrigueZZ	22900000	4300000.0	294400	1300
278	StokesTwins	22700000	11700000.0	235000	10000
281	SSundee	22700000	1700000.0	59800	1800
282	souravjoshivlogs7028	22700000	5600000.0	382300	8900
288	VillageCookingChannel	22500000	21500000.0	321500	5900
300	alfredolarin	21900000	12900000.0	707600	2100
302	royaltyfam	21900000	4700000.0	67000	6600