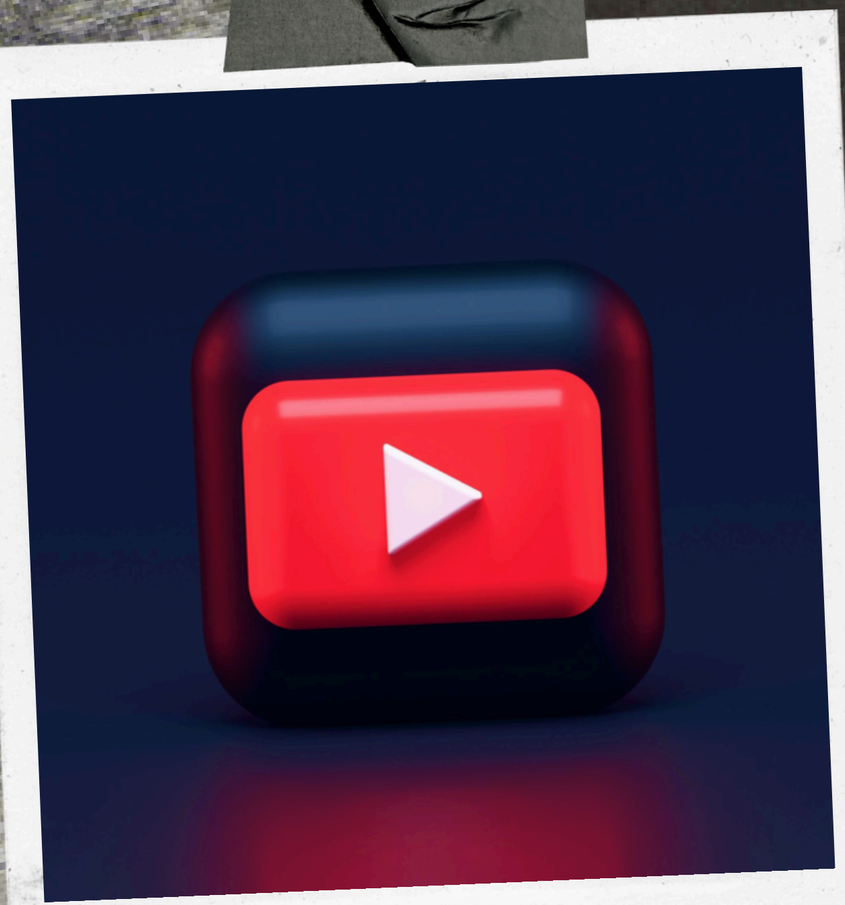
"WE SPEAK DATA"

Streamer Analysis

ASMAA BAZIGHE



Dataset : Top 1000 Youtubers statistics

Description: This dataset contains valuable information about the top YouTube streamers, including their ranking, categories, subscribers, country, visits, likes, comments, and more. Your task is to perform a comprehensive analysis of the dataset to extract insights about the top YouTube content creators.

1

Data Exploration:

- Start by exploring the dataset to understand its structure and identify key variables.
- Check for missing data and outliers.

```
# Load the dataset
df = pd.read_csv('Task 1 YouTube Streamer Analysis/youtubers_df.csv')
df
```

	Rank	Username	Categories	Suscribers	Country	Visits	Likes	Comments	Links
0	1	tseries	Música y baile	249500000.0	India	86200.0	2700.0	78.0	http://youtube.com/channel/UCq-Fj5jknLsUf-MWSy...
1	2	MrBeast	Videojuegos, Humor	183500000.0	Estados Unidos	117400000.0	5300000.0	18500.0	http://youtube.com/channel/UCX6OQ3DkcsbYNE6H8u...
2	3	CoComelon	Educación	165500000.0	Unknown	7000000.0	24700.0	0.0	http://youtube.com/channel/UCbCmjCuTUZos6Inko4...
3	4	SETIndia	NaN	162600000.0	India	15600.0	166.0	9.0	http://youtube.com/channel/UCpEhnqL0y41EpW2TvW...
4	5	KidsDianaShow	Animación, Juguetes	113500000.0	Unknown	3900000.0	12400.0	0.0	http://youtube.com/channel/UCk8GzjMOrta8yxDcKf...
...	...	...	...	...	...	...	...	...	...
995	996	hamzymukbang	NaN	11700000.0	Estados Unidos	397400.0	14000.0	124.0	http://youtube.com/channel/UCPKNKldggioffXPkSm...
996	997	Adaahqueen	NaN	11700000.0	India	1100000.0	92500.0	164.0	http://youtube.com/channel/UCk3fFpqI5kDMf__mUP...
997	998	LittleAngelIndonesia	Música y baile	11700000.0	Unknown	211400.0	745.0	0.0	http://youtube.com/channel/UCdrHrQf0o0TO8YDntX...
998	999	PenMultiplex	NaN	11700000.0	India	14000.0	81.0	1.0	http://youtube.com/channel/UCObyBrdrtQ20BU9PxH...
999	1000	OneindiaHindi	Noticias y Política	11700000.0	India	2200.0	31.0	1.0	http://youtube.com/channel/UCOjgc1p2hJ4GZi6pQQ...

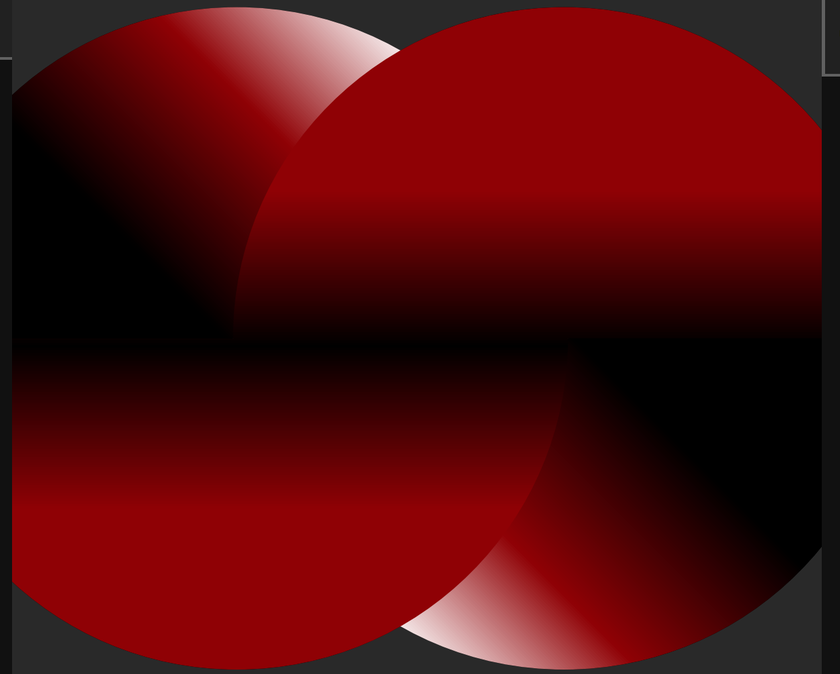
1000 rows × 9 columns

The dataset comprises information regarding the top YouTube streamers, encompassing 1000 entries across 9 columns. These columns encapsulate essential variables including 'Rank', 'Username', 'Categories', 'Subscribers', 'Country', 'Visits', 'Likes', 'Comments', and 'Links'. Understanding these variables lays the foundation for comprehensive analysis, as they serve as key indicators of streamers' popularity and engagement.

Upon inspecting the dataset, it is evident that the 'Categories' variable contains 306 missing values out of the total 1000 entries. This suggests that a significant portion of streamers have not specified their content categories, potentially hindering precise categorization and analysis. Conversely, other variables exhibit no missing values.

```
df.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   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
```



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

```
Rank          0
Username      0
Categories    306
Suscribers    0
Country       0
Visits        0
Likes         0
Comments      0
Links         0
dtype: int64
```

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

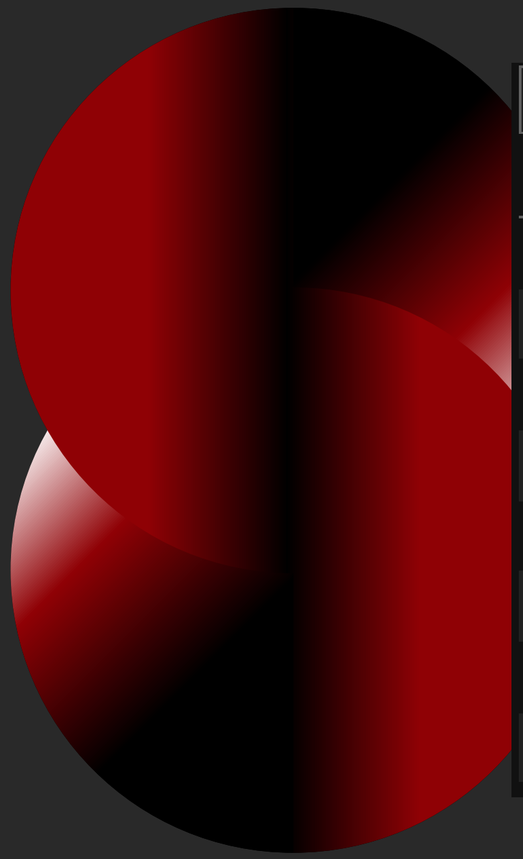
	Rank	Username	Categories	Suscribers	Country	Visits	Likes	Comments	Links
0	1	tseries	Música y baile	249500000.0	India	86200.0	2700.0	78.0	http://youtube.com/channel/UCq-Fj5jknLsUf-MWSy...
1	2	MrBeast	Videojuegos, Humor	183500000.0	Estados Unidos	117400000.0	5300000.0	18500.0	http://youtube.com/channel/UCX6OQ3DkcsbYNE6H8u...
2	3	CoComelon	Educación	165500000.0	Unknown	7000000.0	24700.0	0.0	http://youtube.com/channel/UCbCmjCuTUZos6Inko4...
4	5	KidsDianaShow	Animación, Juguetes	113500000.0	Unknown	3900000.0	12400.0	0.0	http://youtube.com/channel/UCk8GzjMOrta8yxDcKf...
5	6	PewDiePie	Películas, Videojuegos	111500000.0	Estados Unidos	2400000.0	197300.0	4900.0	http://youtube.com/channel/UC-IHJZR3Gqxm24_Vd_...
...	...	...	...	...	...	...	...	...	...
989	990	cut	Humor	11700000.0	Estados Unidos	359000.0	8800.0	342.0	http://youtube.com/channel/UCbaGn5VkOVlcRgJWAH...
990	991	JoeHattab	Películas	11700000.0	Somalia	1900000.0	98500.0	2900.0	http://youtube.com/channel/UCe6eisvsctSPvBhmin...
991	992	BeAmazed	Educación	11700000.0	Estados Unidos	477800.0	9900.0	556.0	http://youtube.com/channel/UCkQO3QsgTpNTsOw6uj...
997	998	LittleAngelIndonesia	Música y baile	11700000.0	Unknown	211400.0	745.0	0.0	http://youtube.com/channel/UCdrHrQf0o0TO8YDntX...
999	1000	OneindiaHindi	Noticias y Política	11700000.0	India	2200.0	31.0	1.0	http://youtube.com/channel/UCOjgc1p2hJ4GZi6pQQ...

694 rows × 9 columns

Note:

- By removing the rows with missing values, the shape of our data is now 694 rows × 9 columns.

However, summary statistics unveil a broad range of values for numeric variables such as 'Subscribers', 'Visits', 'Likes', and 'Comments', indicating potential outliers, particularly at the upper end of the distribution. The presence of outliers warrants thorough examination to ascertain their validity and potential impact on subsequent analyses.



```
df.describe()
```

	Rank	Suscribers	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

```python
# Identify unique values in 'Categories' variable
df['Categories'].unique()
```

```
array(['Música y baile', 'Videojuegos, Humor', 'Educación', nan,
       'Animación, Juguetes', 'Películas, Videojuegos', 'Juguetes',
       'Videojuegos', 'Películas, Animación', 'Películas',
       'Noticias y Política', 'Animación, Humor',
       'Música y baile, Animación', 'Música y baile, Películas',
       'Películas, Juguetes', 'Películas, Humor', 'Vlogs diarios',
       'Videojuegos, Juguetes', 'Animación, Videojuegos', 'Animación',
       'Música y baile, Humor', 'Diseño/arte, DIY y Life Hacks',
       'Ciencia y tecnología', 'Fitness, Salud y autoayuda',
       'Belleza, Moda', 'Humor', 'Comida y bebida', 'Deportes', 'Fitness',
       'Viajes, Espectáculos', 'Comida y bebida, Salud y autoayuda',
       'Diseño/arte', 'DIY y Life Hacks, Juguetes', 'Educación, Juguetes',
       'Juguetes, Coches y vehículos', 'Música y baile, Juguetes',
       'Animales y mascotas', 'ASMR', 'Moda', 'DIY y Life Hacks',
       'Diseño/arte, Belleza', 'Coches y vehículos',
       'Animación, Humor, Juguetes', 'ASMR, Comida y bebida',
       'Comida y bebida, Juguetes', 'Juguetes, DIY y Life Hacks'],
      dtype=object)
```

the analysis of unique values in the 'Categories' and 'Country' variables underscores the varied nature of content creation on YouTube and the platform's global appeal.

this understanding can facilitate targeted content strategies and audience engagement initiatives, thereby fostering growth and success for YouTube creators and the platform as a whole.

```python
# Identify unique values in 'Country' variable
df['Country'].unique()
```

```
array(['India', 'Estados Unidos', 'Unknown', 'Brasil', 'México', 'Rusia',
       'Pakistán', 'Filipinas', 'Indonesia', 'Tailandia', 'Francia',
       'Colombia', 'Iraq', 'Japón', 'Ecuador', 'Argentina', 'Turquía',
       'Arabia Saudita', 'El Salvador', 'Bangladesh', 'Reino Unido',
       'Argelia', 'España', 'Perú', 'Egipto', 'Jordania', 'Marruecos',
       'Singapur', 'Somalia'], dtype=object)
```

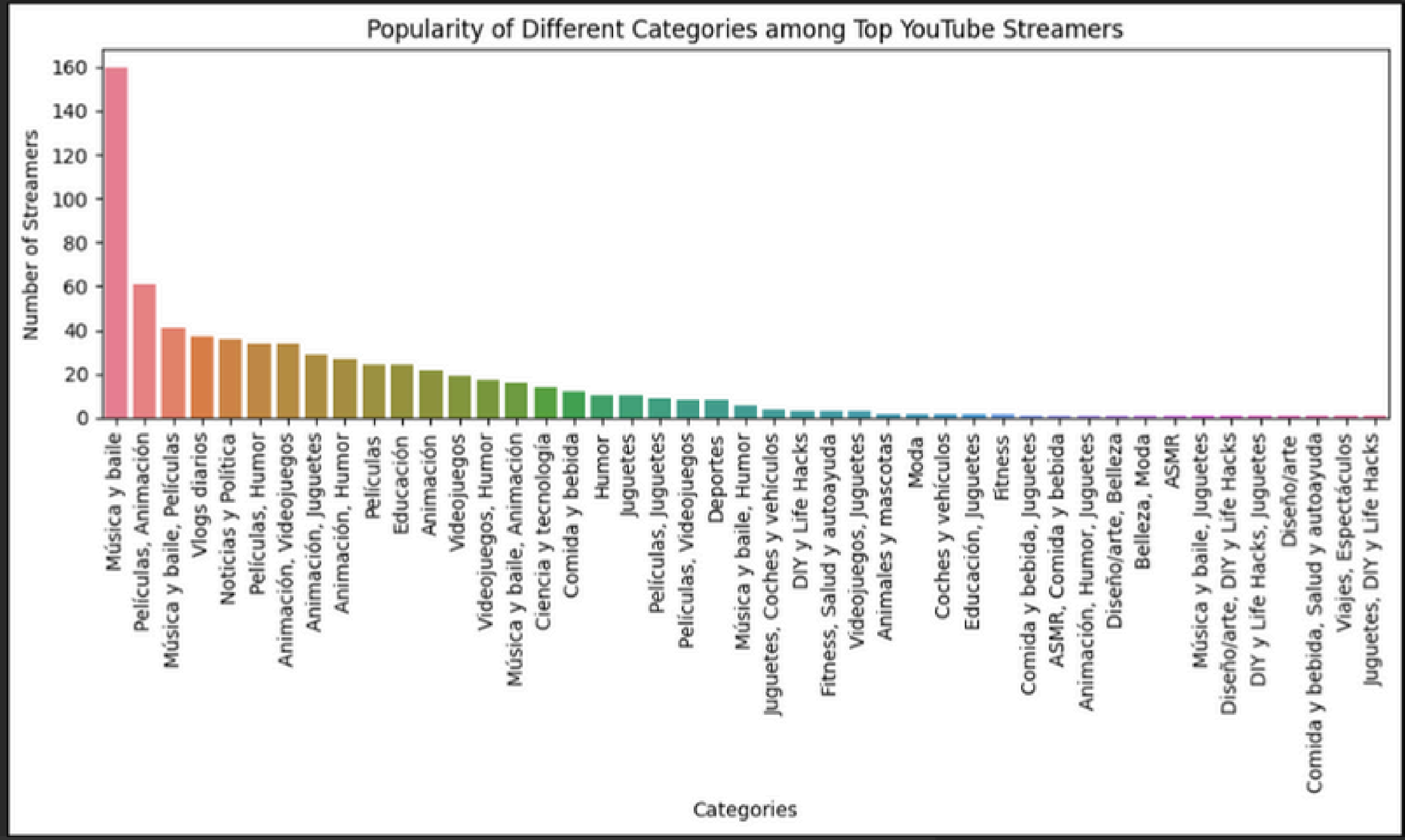
'WE SPEAK DATA'

02

Trend Analysis:

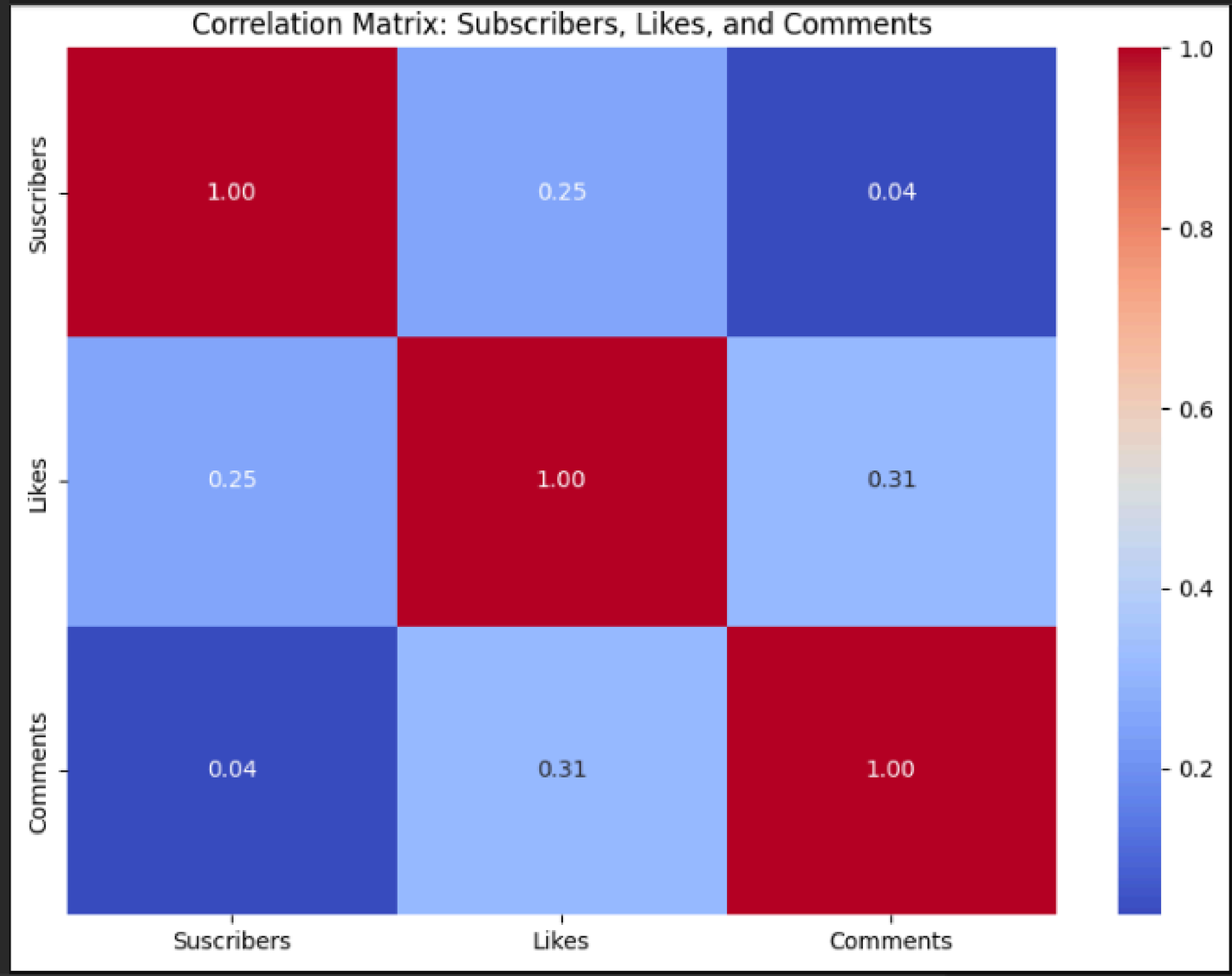
- Identify trends among the top YouTube streamers. Which categories are the most popular?
- Is there a correlation between the number of subscribers and the number of likes or comments?

02



- The category with the most popular is 'Mùsica y baile'

'WE SPEAK DATA'



- ✕- correlation between the number of subscribers and the number of likes: 0.25
- ✕- correlation between the number of subscribers and the number of comments: 0.04
- ✕- correlation between the number of comments and the number of like: 0.31

Audience Study:

Analyze the distribution of streamers' audiences by country. Are there regional preferences for specific content categories?

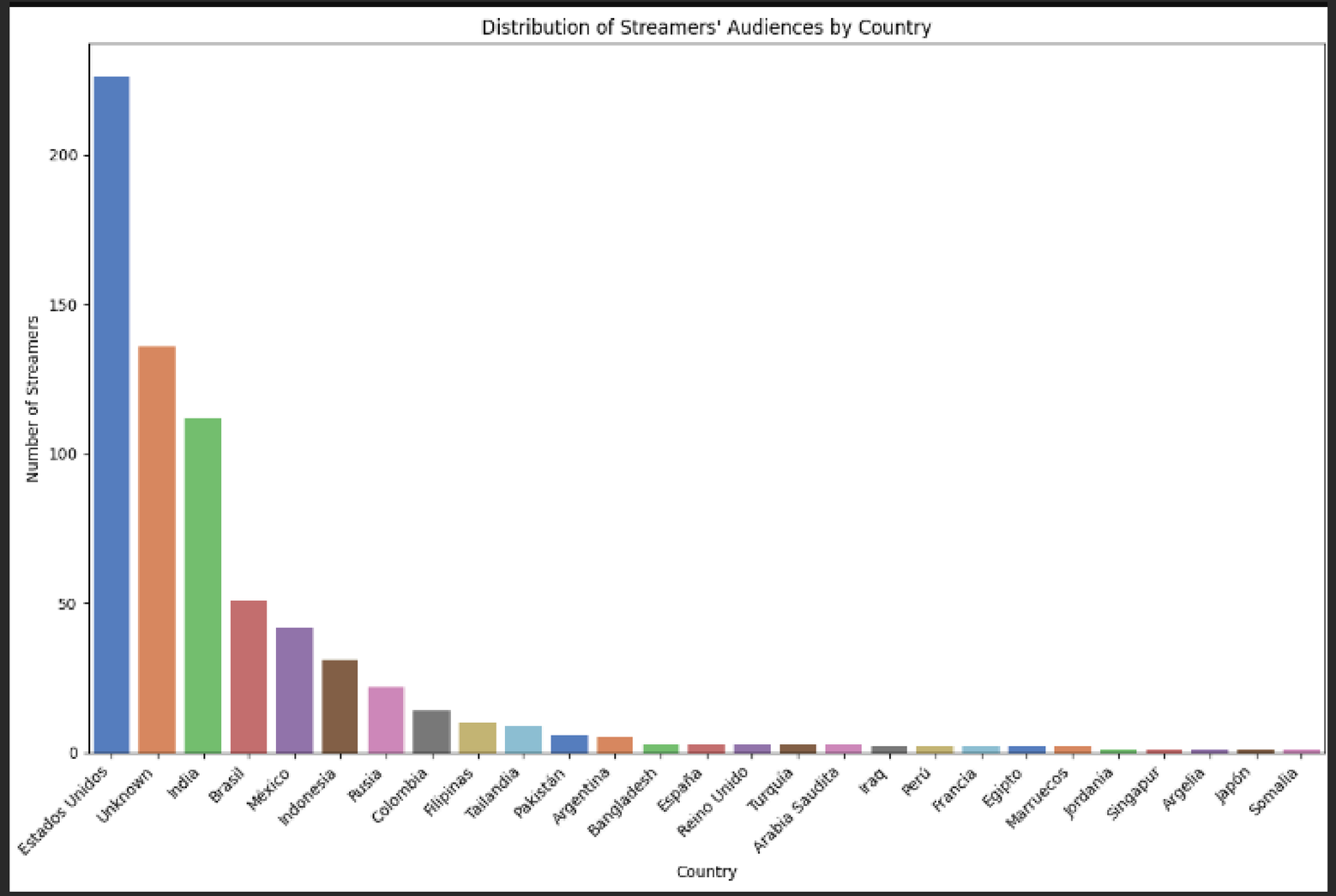
03

Here's an analysis of the distribution of streamers' audiences by country:

This analysis provides insights into the geographical distribution of streamers' audiences, highlighting the countries with the highest and lowest audience counts.

```
# Analyze the distribution of streamers' audiences by country
country_counts = df['Country'].value_counts()
country_counts
```

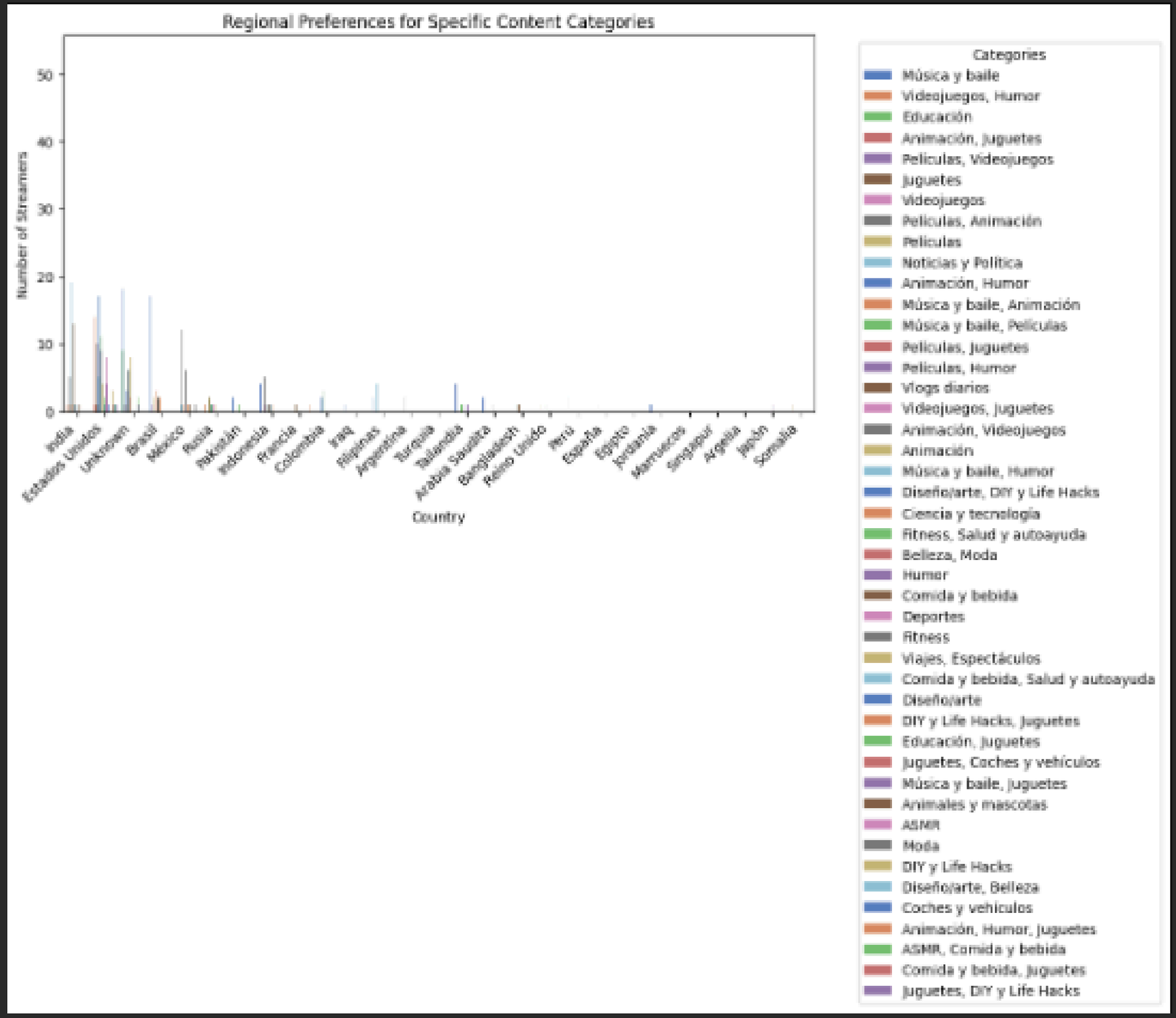
```
Country
Estados Unidos    226
Unknown           136
India             112
Brasil             51
México             42
Indonesia          31
Rusia              22
Colombia           14
Filipinas          10
Tailandia           9
Pakistán            6
Argentina           5
Bangladesh          3
España              3
Reino Unido         3
Turquía             3
Arabia Saudita      3
Iraq                2
Perú                2
Francia             2
Egipto              2
Marruecos           2
Jordania            1
Singapur            1
Argelia             1
Japón               1
Somalia             1
Name: count, dtype: int64
```



Bar plot that visualizes the distribution of streamers' audiences by country, making it easier to understand the geographical distribution of the audience across different countries.

'WE SPEAK DATA'

Plot that visualizes the regional preferences for specific content categories among streamers. The plot helps identify which categories are more popular in different countries, providing insights into regional content preferences.



	Country	Preferred_Category	Percentage
0	Arabia Saudita	Música y baile	66.666667
1	Argelia	Educación	100.000000
2	Argentina	Películas, Animación	40.000000
3	Bangladesh	Vlogs diarios	33.333333
4	Brasil	Música y baile	33.333333
5	Colombia	Música y baile	42.857143
6	Egipto	Películas	50.000000
7	España	Películas, Videojuegos	33.333333
8	Estados Unidos	Música y baile	23.451327
9	Filipinas	Noticias y Política	40.000000
10	Francia	Vlogs diarios	50.000000
11	India	Música y baile	37.500000
12	Indonesia	Música y baile, Películas	38.709677
13	Iraq	Animación, Videojuegos	50.000000
14	Japón	Humor	100.000000
15	Jordania	Música y baile	100.000000
16	Marruecos	Noticias y Política	100.000000
17	México	Películas, Animación	28.571429
18	Pakistán	Música y baile	33.333333
19	Perú	Música y baile	100.000000
20	Reino Unido	Música y baile, Películas	33.333333
21	Rusia	Videojuegos	36.363636
22	Singapur	Animación	100.000000
23	Somalia	Películas	100.000000
24	Tailandia	Música y baile	44.444444
25	Turquía	Vlogs diarios	66.666667
26	Unknown	Animación, Juguetes	20.588235

These insights provide a clear snapshot of the dominant content preferences in various countries. For instance, in Arabia Saudita, Música y baile reigns supreme, constituting approximately 66.67% of the preferred content. Similarly, in Argelia, the focus leans heavily towards content related to Education, encompassing 100% of the audience's preference.

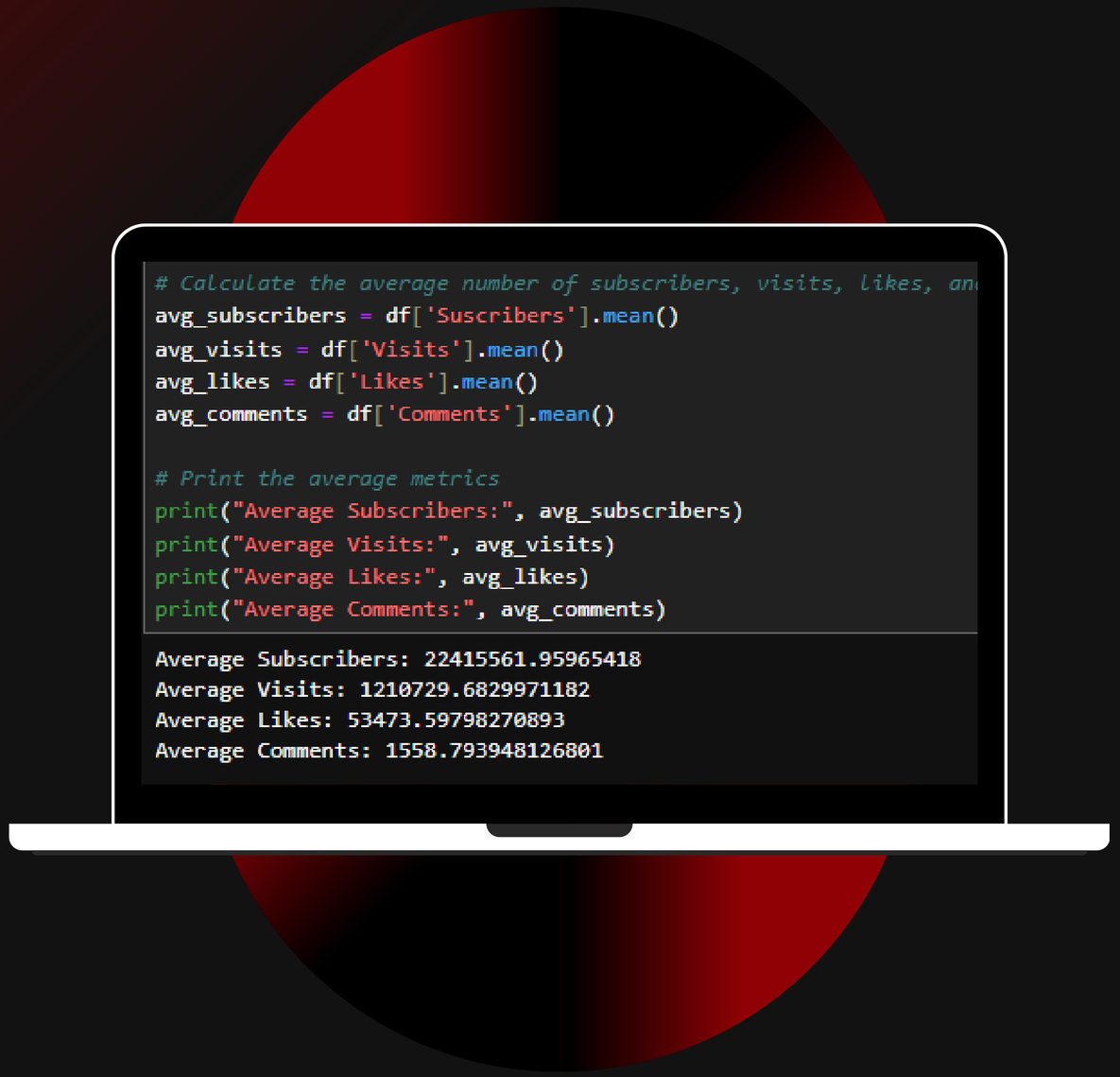
This tabular representation offers a concise and straightforward overview of the preferred content categories across different regions, facilitating easy interpretation and analysis.

Performance Metrics:

04

- Calculate and visualize the average number of subscribers, visits, likes, and comments.
- Are there patterns or anomalies in these metrics?

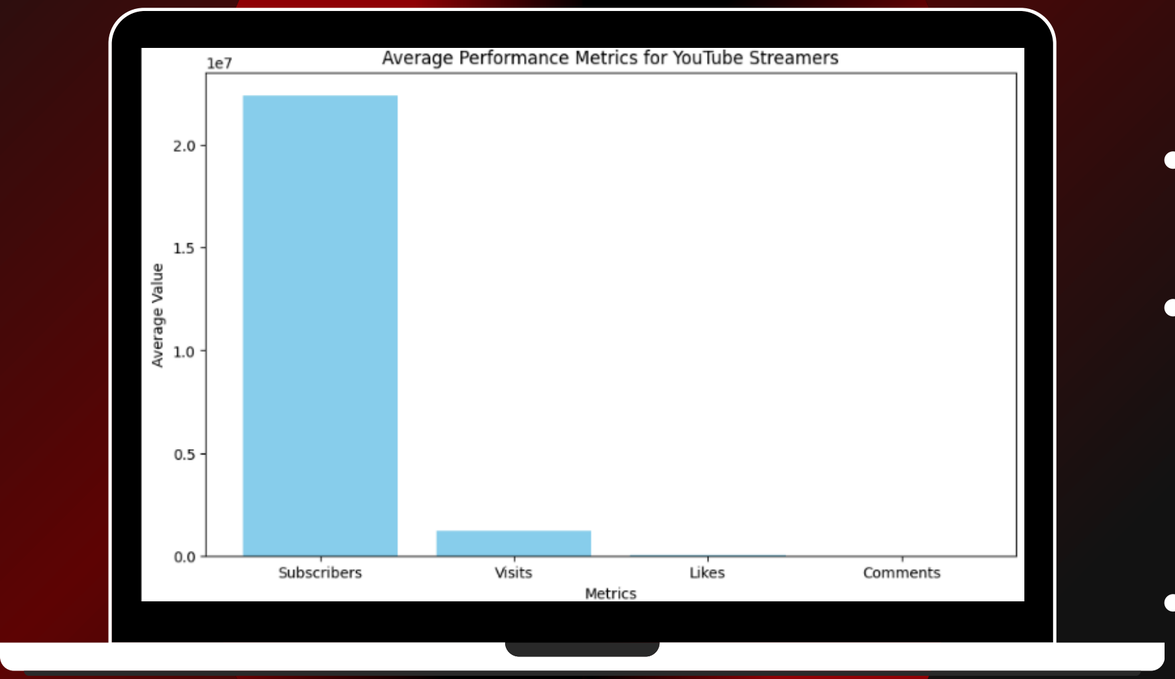
- **Average Subscribers**: Around 22,415,561.96 subscribers per streamer, indicating overall popularity.
- **Average Visits**: Approximately 1,210,729.68 visits per streamer, reflecting average traffic or viewership.
- **Average Likes**: About 53,473.60 likes per streamer, showing audience engagement and satisfaction.
- **Average Comments:** Roughly 1,558.79 comments per streamer, representing audience interaction and feedback.



```
# Calculate the average number of subscribers, visits, likes, an
avg_subscribers = df['Suscribers'].mean()
avg_visits = df['Visits'].mean()
avg_likes = df['Likes'].mean()
avg_comments = df['Comments'].mean()

# Print the average metrics
print("Average Subscribers:", avg_subscribers)
print("Average Visits:", avg_visits)
print("Average Likes:", avg_likes)
print("Average Comments:", avg_comments)
```

```
Average Subscribers: 22415561.95965418
Average Visits: 1210729.6829971182
Average Likes: 53473.59798270893
Average Comments: 1558.793948126801
```



- **Subscribers vs. Visits:** High subscribers with low visits may indicate content not effectively engaging the audience.
- **Likes vs. Comments:** More likes than comments suggest a preference for passive engagement.
- **Subscribers vs. Likes vs. Comments:** Differences between subscribers and engagement metrics may reveal issues with content quality or audience targeting.
- **Visits vs. Likes vs. Comments:** Low engagement metrics despite high visits may signify challenges in audience engagement.

05

Content Categories:

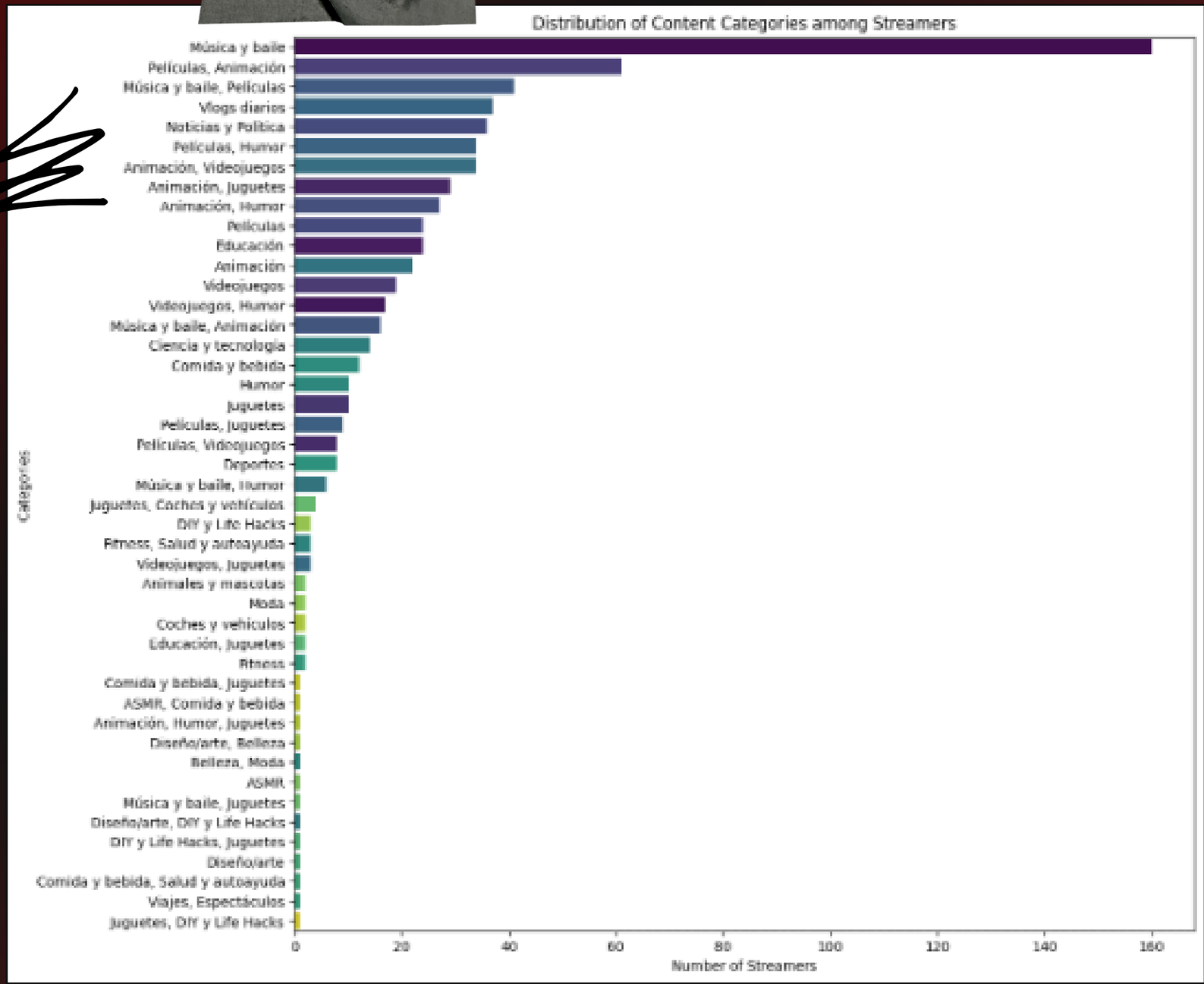
- Explore the distribution of content categories. Which categories have the highest number of streamers?
- Are there specific categories with exceptional performance metrics?

"WE SPEAK DATA"

Top Categories with the Highest Number of Streamers:

Categories	
Música y baile	160
Películas, Animación	61
Música y baile, Películas	41
Vlogs diarios	37
Noticias y Política	36

Name: count, dtype: int64



Performance Metrics for Each Content Category:

Categories	Suscribers	Visits	Likes	Comments
ASMR	1.520000e+07	3.685000e+05	4100.000000	148.000000
ASMR, Comida y bebida	1.300000e+07	5.575000e+05	8600.000000	349.000000
Animación	1.764091e+07	6.367182e+05	21413.454545	396.636364
Animación, Humor	2.078519e+07	3.760126e+06	145768.333333	5344.962963
Animación, Humor, Juguetes	1.390000e+07	8.000000e+03	37.000000	0.000000
Animación, Juguetes	2.937586e+07	5.254483e+05	2653.068966	0.517241
Animación, Videojuegos	1.939412e+07	1.200059e+06	79294.029412	3786.617647
Animales y mascotas	1.560000e+07	2.231450e+06	102750.000000	2806.000000
Belleza, Moda	2.390000e+07	9.645000e+05	62300.000000	1100.000000
Ciencia y tecnología	1.726429e+07	8.871286e+05	59283.142857	1363.571429
Coches y vehículos	1.320000e+07	2.664000e+05	18150.000000	439.500000
Comida y bebida	1.612500e+07	2.722450e+06	128664.750000	3053.416667
Comida y bebida, Juguetes	1.230000e+07	4.690000e+04	176.000000	0.000000
Comida y bebida, Salud y autoayuda	2.010000e+07	1.149000e+05	2800.000000	117.000000
DIY y Life Hacks	1.306667e+07	7.146667e+04	1616.333333	47.666667
DIY y Life Hacks, Juguetes	1.910000e+07	2.300000e+06	33200.000000	2100.000000
Deportes	1.552500e+07	1.759525e+06	44949.000000	136.500000
Diseño/arte	2.010000e+07	1.785000e+05	7300.000000	140.000000
Diseño/arte, Belleza	1.440000e+07	2.700000e+06	152400.000000	1100.000000
Diseño/arte, DIY y Life Hacks	2.570000e+07	2.600000e+06	127300.000000	2200.000000
Educación	2.501250e+07	1.106042e+06	45060.750000	1537.250000
Educación, Juguetes	1.805000e+07	4.697500e+05	2185.000000	0.000000
Fitness	1.635000e+07	8.620000e+04	3750.000000	63.500000
Fitness, Salud y autoayuda	1.710000e+07	1.946667e+05	7600.000000	532.000000
Humor	1.525000e+07	2.310400e+06	169990.000000	5159.800000
Juguetes	3.788000e+07	7.005100e+05	5290.200000	2.800000
Juguetes, Coches y vehículos	1.550000e+07	8.242500e+04	961.000000	0.000000
Juguetes, DIY y Life Hacks	1.220000e+07	6.260000e+04	256.000000	0.000000
Moda	1.440000e+07	1.726000e+05	8050.000000	218.500000
Música y baile	2.683688e+07	3.743881e+05	17405.681250	1998.931250
Música y baile, Animación	2.040000e+07	6.957188e+05	17155.000000	589.437500
Música y baile, Humor	1.838333e+07	2.402933e+06	45783.333333	2110.500000
Música y baile, Juguetes	1.730000e+07	5.250000e+04	129.000000	0.000000
Música y baile, Películas	1.947561e+07	4.405902e+05	17966.414634	457.195122
Noticias y Política	1.878056e+07	2.187333e+05	10353.222222	358.916667
Películas	2.114167e+07	7.758458e+05	28829.208333	1081.041667
Películas, Animación	2.269344e+07	5.513295e+05	25671.016393	645.655738
Películas, Humor	1.829706e+07	9.387235e+05	40684.617647	1008.794118
Películas, Juguetes	2.130000e+07	6.264667e+05	1332.888889	0.000000
Películas, Videojuegos	3.325000e+07	6.940375e+05	48083.375000	1569.500000
Viajes, Espectáculos	2.040000e+07	8.950000e+04	782.000000	49.000000
Videojuegos	2.498421e+07	1.387137e+06	57121.052632	1760.157895
Videojuegos, Humor	2.876471e+07	1.023968e+07	420511.764706	4827.058824
Videojuegos, Juguetes	2.473333e+07	5.741667e+05	6400.000000	337.000000
Vlogs diarios	1.770000e+07	3.414338e+06	187244.945946	980.405405

The performance metrics for each content category indicate variations in average subscribers, visits, likes, and comments across different categories. Some categories have notably higher or lower metrics compared to others. For example:

- ASMR category has relatively high average subscribers and visits but low average likes and comments, indicating potential room for improvement in audience engagement.
- Animación, Humor category stands out with significantly high average likes and comments, suggesting strong audience engagement and interaction.
- Juguetes category has exceptionally high average subscribers but low average likes and comments, indicating a large audience but relatively lower engagement levels.

These observations highlight the diversity in audience engagement and performance metrics across different content categories, suggesting that certain categories may resonate better with audiences and attract higher levels of engagement.

'WE SPEAK DATA'

06

Benchmarking:

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

06

```
# Filter streamers with above-average performance
top_streamers = df[
    (df['Suscribers'] > avg_subscribers) &
    (df['Visits'] > avg_visits) &
    (df['Likes'] > avg_likes) &
    (df['Comments'] > avg_comments)
]
top_streamers
```

	Rank	Username	Categories	Suscribers	Country	Visits	Likes	Comments	Links
1	2	MrBeast	Videojuegos, Humor	183500000.0	Estados Unidos	117400000.0	5300000.0	18500.0	http://youtube.com/channel/UCX6OQ3DkcsbYNE6H8u...
5	6	PewDiePie	Películas, Videojuegos	111500000.0	Estados Unidos	2400000.0	197300.0	4900.0	http://youtube.com/channel/UC-lHJZR3Gqxm24_Vd_...
26	27	dudeperfect	Videojuegos	59700000.0	Estados Unidos	5300000.0	156500.0	4200.0	http://youtube.com/channel/UCRijo3ddMTht_IHyNS...
34	35	TaylorSwift	Música y baile	54100000.0	Estados Unidos	4300000.0	300400.0	15000.0	http://youtube.com/channel/UCqECaJ8Gagnn7YCbPE...
39	40	JuegaGerman	Películas, Animación	48600000.0	México	2000000.0	117100.0	3000.0	http://youtube.com/channel/UCYiGq8XF7YQD00x7wA...
43	44	A4a4a4a4	Animación, Humor	47300000.0	Rusia	9700000.0	330400.0	22000.0	http://youtube.com/channel/UC2tsySbe9TNrI-xh2I...
58	59	Mikecrack	Películas, Animación	43400000.0	México	2200000.0	183400.0	1800.0	http://youtube.com/channel/UCqJ5zFEED1hWs0KNQC...
62	63	KimberlyLoaiza	Música y baile	42100000.0	México	5300000.0	271300.0	16000.0	http://youtube.com/channel/UCQZfFRohQ7UX-0CdXI...

As we can see, the top-performing content creators are:

- MrBeast
- PewDiePie
- Dude Perfect
- Taylor Swift
- JuegaGerman

07

Content Recommendations:

- Propose a system for enhancing content recommendations to YouTube users based on streamers categories and performance metrics.

'WE SPEAK DATA'

I start by selecting only the necessary columns from the original dataset (df). These columns include the streamer's username, their categories, and various performance metrics such as subscribers, visits, likes, and comments.

Next, we encode the categorical variable 'Categories' into numerical IDs. This is done to facilitate further analysis and modeling, as many machine learning algorithms require numerical inputs:

- We use the pd.merge function to merge the original data with a DataFrame called categories_df, which contains unique category names along with their corresponding IDs.
- The merge is performed based on the 'Categories' column in both DataFrames.
- After merging, we drop the original 'Categories' column and keep only the encoded numerical IDs, which are now named as 'category_ID'.

```
#use only necessary columns
data = df[['Username', 'Categories', 'Suscribers', 'Visits', 'Likes', 'Comments']]
# Encode categorical variables
data = pd.merge(data, categories_df, left_on='Categories', right_on='Categories', how='left')
data.drop(['Categories', 'Categories'], axis=1, inplace=True)
data
```

	Username	Suscribers	Visits	Likes	Comments	category_ID
0	tseries	249500000.0	86200.0	2700.0	78.0	1
1	MrBeast	183500000.0	117400000.0	5300000.0	18500.0	2
2	CoComelon	165500000.0	7000000.0	24700.0	0.0	3
3	KidsDianaShow	113500000.0	3900000.0	12400.0	0.0	4
4	PewDiePie	111500000.0	2400000.0	197300.0	4900.0	5
...	...	...	...	...	...	...
689	cut	11700000.0	359000.0	8800.0	342.0	25
690	JoeHattab	11700000.0	1900000.0	98500.0	2900.0	9
691	BeAmazed	11700000.0	477800.0	9900.0	556.0	3
692	LittleAngelIndonesia	11700000.0	211400.0	745.0	0.0	1
693	OneindiaHindi	11700000.0	2200.0	31.0	1.0	10

694 rows × 6 columns

'WE SPEAK DATA'

```
# Calculate engagement rate
data['EngagementRate'] = (data['Likes'] + data['Comments']) / data['Visits']
data
```

	Username	Suscribers	Visits	Likes	Comments	category_ID	EngagementRate
0	tseries	249500000.0	86200.0	2700.0	78.0	1	0.032227
1	MrBeast	183500000.0	117400000.0	5300000.0	18500.0	2	0.045302
2	CoComelon	165500000.0	7000000.0	24700.0	0.0	3	0.003529
3	KidsDianaShow	113500000.0	3900000.0	12400.0	0.0	4	0.003179
4	PewDiePie	111500000.0	2400000.0	197300.0	4900.0	5	0.084250
...	...	...	...	...	...	...	...
689	cut	11700000.0	359000.0	8800.0	342.0	25	0.025465
690	JoeHattab	11700000.0	1900000.0	98500.0	2900.0	9	0.053368
691	BeAmazed	11700000.0	477800.0	9900.0	556.0	3	0.021884
692	LittleAngelIndonesia	11700000.0	211400.0	745.0	0.0	1	0.003524
693	OneindiaHindi	11700000.0	2200.0	31.0	1.0	10	0.014545

694 rows × 7 columns

In this section of the code, I computed the engagement rate for each streamer in the dataset. The engagement rate is a metric that measures the level of interaction or engagement that viewers have with a streamer's content.

- I calculate the engagement rate by summing the number of likes and comments for each streamer and then dividing this sum by the total number of visits.
- This formula gives us a normalized measure of how engaging a streamer's content is, taking into account both the number of interactions (likes and comments) and the overall viewership (visits).
- The resulting 'EngagementRate' column in the DataFrame represents the computed engagement rate values for each streamer.

'WE SPEAK DATA'

- I used the **dropna()** function to remove any rows with missing values from the DataFrame **data**.
- By setting **inplace=True**, we perform the operation directly on the DataFrame, modifying it in place rather than creating a new DataFrame.
- After dropping the missing values, I then check for any remaining missing values using the **isnull().sum()** function.

This process ensures that the data is complete and ready for further analysis without the potential bias or errors introduced by missing values.

```
data.isnull().sum()
```

```
Username            0
Suscribers          0
Visits              0
Likes               0
Comments            0
category_ID         0
EngagementRate     17
dtype: int64
```

```
# Handle missing values
data.dropna(inplace=True)
data.isnull().sum()
```

```
Username            0
Suscribers          0
Visits              0
Likes               0
Comments            0
category_ID         0
EngagementRate      0
dtype: int64
```

'WE SPEAK DATA'

Implementing k-Nearest Neighbors (k-NN) Recommendation System

```python
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import NearestNeighbors

# Standardize numerical features
scaler = StandardScaler()
numerical_cols = ['Suscribers', 'Visits', 'Likes', 'Comments', 'category_ID', 'EngagementRate']
data[numerical_cols] = scaler.fit_transform(data[numerical_cols])

# Fit k-NN model
knn_model = NearestNeighbors(n_neighbors=4, metric='euclidean')
knn_model.fit(data[numerical_cols])

def recommend_similar_streamers(target_streamer_name):
    # Find index of target streamer
    if target_streamer_name in data['Username'].values:
        target_streamer_index = data[data['Username'] == target_streamer_name].index[0]

        # Query k-NN model to find similar streamers
        target_data = data.loc[[target_streamer_index], numerical_cols]  # Create DataFrame with single row
        distances, indices = knn_model.kneighbors(target_data)

        # Get names of similar streamers
        similar_streamers = data.loc[indices[0], 'Username'].tolist()

        # Remove target streamer from the list (if present)
        if target_streamer_name in similar_streamers:
            similar_streamers.remove(target_streamer_name)

        # Return recommended streamers
        return similar_streamers[:3]  # Return top 3 similar streamers
    else:
        print("We are sorry, but we don't have data about this streamer, so we can't presently provide you with recommendations.")
```

I utilize the k-Nearest Neighbors algorithm to recommend similar streamers to a given target streamer based on their numerical features.

- I import necessary modules from sklearn.preprocessing and sklearn.neighbors, including StandardScaler for standardizing numerical features and NearestNeighbors for implementing the k-NN algorithm.
- The numerical features in the DataFrame data are standardized using StandardScaler to ensure that all features have the same scale.
- I then initialize a k-NN model with NearestNeighbors and specify the number of neighbors (k=4) and the distance metric (euclidean).
- The recommend_similar_streamers function takes a target streamer's name as input and returns the names of similar streamers based on the k-NN algorithm.
- If the target streamer exists in the dataset, we find its index and query the k-NN model to obtain the indices of the most similar streamers.
- I retrieve the names of similar streamers, remove the target streamer from the list, and return the top 3 similar streamers.
- If the target streamer is not found in the dataset, we notify the user that recommendations cannot be provided.

This implementation enables us to recommend similar streamers facilitating personalized content suggestions for users.

Let's now test our model. First, I tried with MrBeast, who is present in my dataset, and I received the following recommendation:
"Recommended streamers for MrBeast: ['dream', 'TalkingTom', 'Willyrex']"

```
# Example usage:
target_streamer_name = input("Enter the name of your the target streamer: ")
recommended_streamers = recommend_similar_streamers(target_streamer_name)
print("Recommended streamers for", target_streamer_name, ":", recommended_streamers)
```

```
Enter the name of your the target streamer:  MrBeast
Recommended streamers for MrBeast : ['dream', 'TalkingTom', 'Willyrex']
```

```
# Example usage:
target_streamer_name = input("Enter the name of your the target streamer: ")
recommended_streamers = recommend_similar_streamers(target_streamer_name)
print("Recommended streamers for", target_streamer_name, ":", recommended_streamers)
```

```
Enter the name of your the target streamer:  bimo
We are sorry, but we don't have data about this streamer, so we can't presently provide you with recommendations.
Recommended streamers for bimo : None
```

Then, I tried with Bimo, who isn't present in my dataset, and I received the following message:
"We are sorry, but we don't have data about this streamer, so we can't presently provide you with recommendations.
Recommended streamers for Bimo: None"

"WE SPEAK DATA"

CONCLUSION:

Throughout this project, I analyzed a dataset of top YouTube streamers, delving into trends, audience preferences, performance metrics, and content categories. By exploring the dataset's structure, identifying popular content categories, and uncovering regional preferences, I gained valuable insights into the dynamics of YouTube content creation. The project also involved benchmarking top-performing creators and proposing a content recommendation system. This experience not only honed my analytical skills but also deepened my understanding of the digital content landscape, providing invaluable insights for future endeavors in the field.

CONTACT

Asmaa bazighe

https://github.com/AsmaaBazighe

asmaabazighe@gmail.com