IMPORTING LIBRARIES

In [1]:
```
#Importing Libraries:
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
# Loading the csv file into a pandas dataframe:
df = pd.read_csv("youtubers_df.csv")
```

DATA EXPLORATION:

First five rows of the dataset:

In [3]:
```
# First five rows of the dataset
df.head()
```

Out[3]:

	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

Last five rows of the dataset:

In [4]:
```
# Last five rows of the dataset
df.tail()
```

Out[4]:

	Rank	Username	Categories	Suscribers	Country	Visits	Likes	Comments
995	996	hamzymukbang	NaN	11700000.0	Estados Unidos	397400.0	14000.0	124.0
996	997	Adaahqueen	NaN	11700000.0	India	1100000.0	92500.0	164.0
997	998	LittleAngelIndonesia	Música y baile	11700000.0	Unknown	211400.0	745.0	0.0
998	999	PenMultiplex	NaN	11700000.0	India	14000.0	81.0	1.0
999	1000	OneindiaHindi	Noticias y Política	11700000.0	India	2200.0	31.0	1.0

Overview of the dataset:

In [5]:
```
# Overview of the dataset
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
```

Shape of the dataset:

In [6]:
```
# Shape of the dataset
df.shape
```

Out[6]: (1000, 9)

Column Names

In [7]:
```
# Available column names:
df.columns
```

Out[7]:
```
Index(['Rank', 'Username', 'Categories', 'Suscribers', 'Country', 'Visits',
       'Likes', 'Comments', 'Links'],
      dtype='object')
```

Data Types in Each Column

In [8]:
```
# The data type of the available column
df.dtypes
```

Out[8]:
```
Rank            int64
Username       object
Categories     object
Suscribers    float64
Country        object
Visits        float64
Likes         float64
Comments      float64
Links          object
dtype: object
```

Checking for Duplicate data:

In [9]:
```
# Duplicate data

df.duplicated().sum()
```

Out[9]:
```
0
```

```
There is no Duplicate values present in the dataset
```

DEALING WITH MISSING VALUES

In [10]:
```
# Missing value
df.isnull().sum()
```

Out[10]:
```
Rank            0
Username        0
Categories    306
Suscribers      0
Country         0
Visits          0
Likes           0
Comments        0
Links           0
dtype: int64
```

In [11]:
```
df[df['Categories'].isnull()==True]
```

Out[11]:

	Rank	Username	Categories	Suscribers	Country	Visits	Likes	Comm
3	4	SETIndia	NaN	162600000.0	India	15600.0	166.0	
11	12	GoldminesTelefilms	NaN	89700000.0	India	34600.0	421.0	
16	17	zeetv	NaN	72500000.0	India	7300.0	144.0	
20	21	colorstv	NaN	66600000.0	India	14700.0	413.0	
38	39	shemaroo	NaN	48800000.0	India	22600.0	163.0	
...	...	...	...	...	...	...	...	
993	994	RedChilliesEntertainment	NaN	11700000.0	India	848300.0	66700.0	8
994	995	VYRLOriginals	NaN	11700000.0	India	48800.0	949.0	1
995	996	hamzymukbang	NaN	11700000.0	Estados Unidos	397400.0	14000.0	1
996	997	Adaahqueen	NaN	11700000.0	India	1100000.0	92500.0	1
998	999	PenMultiplex	NaN	11700000.0	India	14000.0	81.0	

306 rows × 9 columns

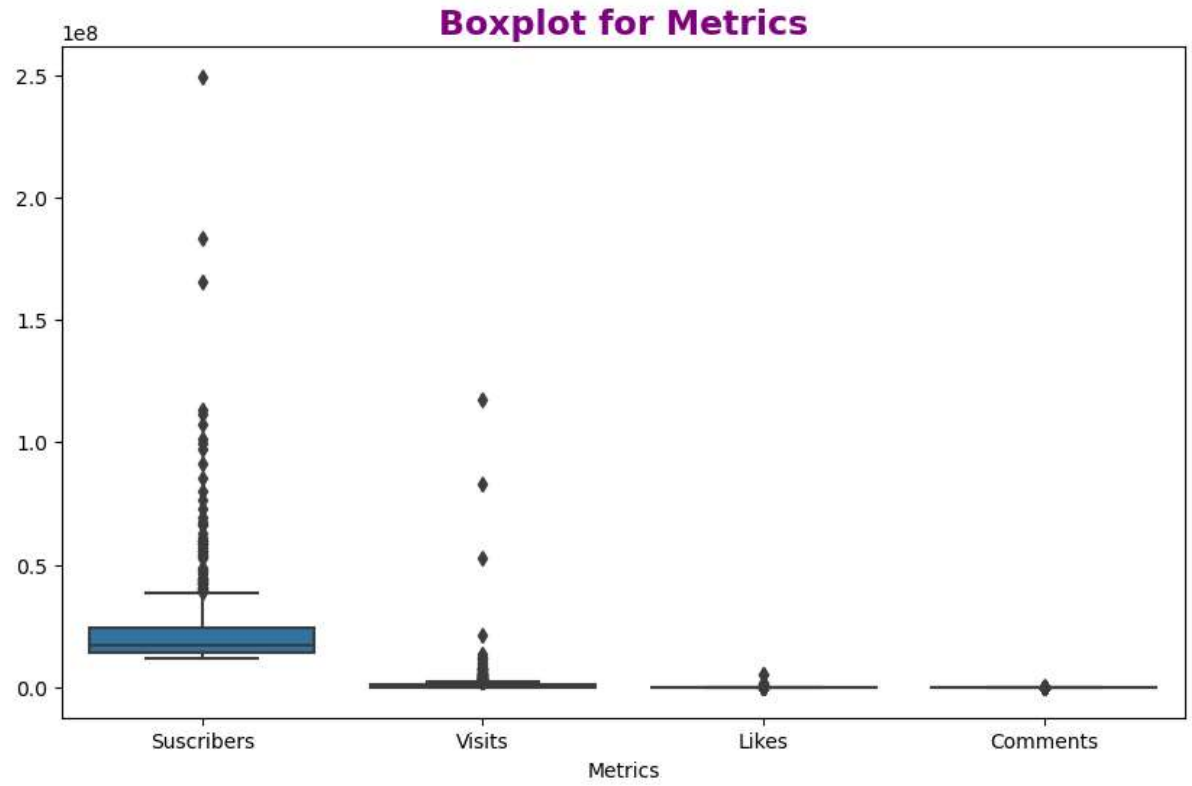
In [12]:
```
# Dealing with missing values
df = df.dropna()
```

In [13]:
```
df.shape
```

Out[13]:
```
(694, 9)
```

FINDING OUTLIERS:

In [14]:
```
# Finding outlier
plt.figure(figsize=(10,6))
sns.boxplot(data=df[['Suscribers', 'Visits','Likes','Comments']])
plt.title('Boxplot for Metrics', color = 'purple', fontweight='bold', fontsize=
plt.xlabel('Metrics')
plt.show()
```



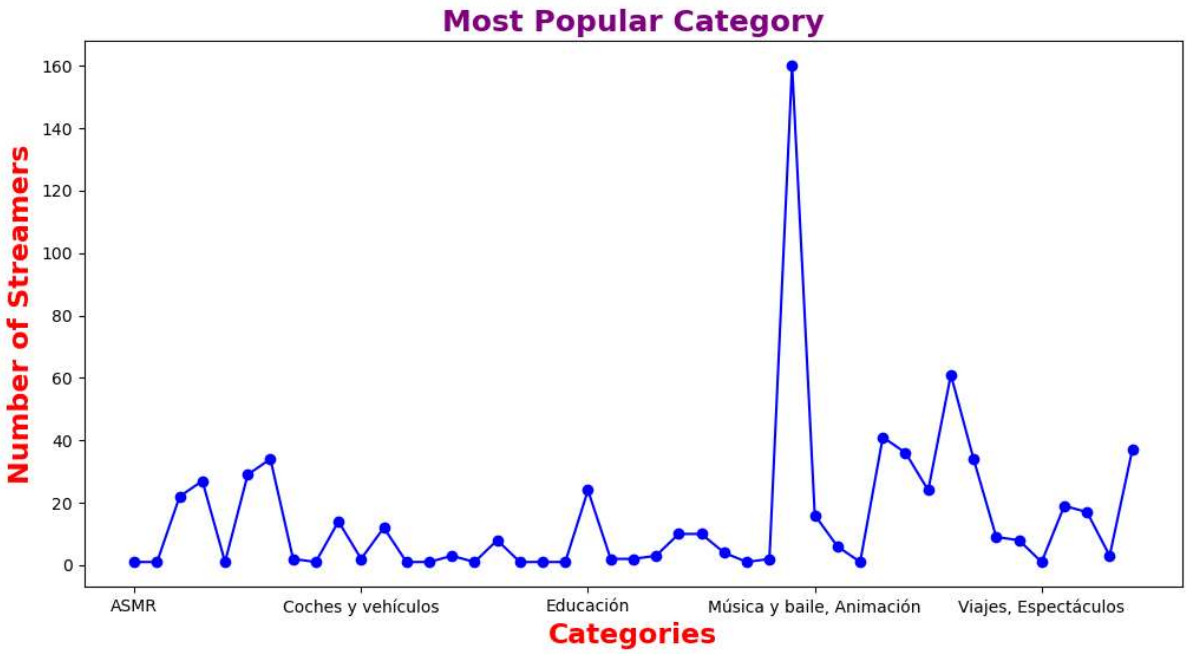
TREND ANALYSIS:

In [15]:
```
#Categories count
a = df.groupby('Categories')['Categories'].count()
a
```

Out[15]:
```
Categories
ASMR                                    1
ASMR, Comida y bebida                   1
Animación                              22
Animación, Humor                       27
Animación, Humor, Juguetes              1
Animación, Juguetes                    29
Animación, Videojuegos                 34
Animales y mascotas                     2
Belleza, Moda                           1
Ciencia y tecnología                   14
Coches y vehículos                      2
Comida y bebida                        12
Comida y bebida, Juguetes               1
Comida y bebida, Salud y autoayuda      1
DIY y Life Hacks                        3
DIY y Life Hacks, Juguetes              1
Deportes                                8
Diseño/arte                             1
Diseño/arte, Belleza                    1
Diseño/arte, DIY y Life Hacks           1
Educación                              24
Educación, Juguetes                     2
Fitness                                 2
Fitness, Salud y autoayuda              3
Humor                                  10
Juguetes                               10
Juguetes, Coches y vehículos            4
Juguetes, DIY y Life Hacks              1
Moda                                    2
Música y baile                        160
Música y baile, Animación              16
Música y baile, Humor                   6
Música y baile, Juguetes                1
Música y baile, Películas              41
Noticias y Política                    36
Películas                              24
Películas, Animación                   61
Películas, Humor                       34
Películas, Juguetes                     9
Películas, Videojuegos                  8
Viajes, Espectáculos                    1
Videojuegos                            19
Videojuegos, Humor                     17
Videojuegos, Juguetes                   3
Vlogs diarios                          37
Name: Categories, dtype: int64
```

Most Popular Category

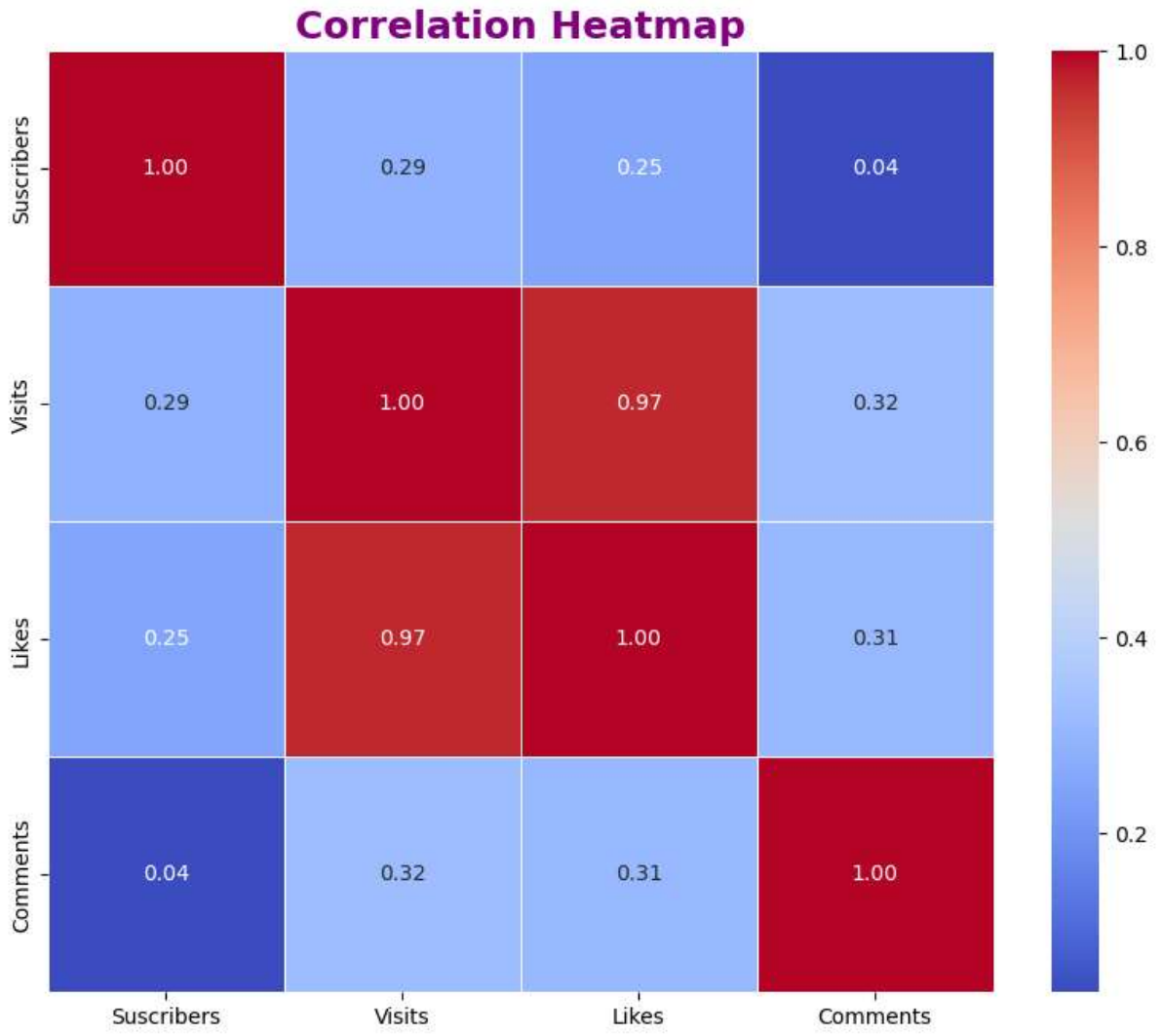
In [16]:
```
plt.figure(figsize=(12,6))
a.plot(kind='line',marker='o',linestyle='-',color = 'blue')

plt.title('Most Popular Category', color = 'purple', fontweight='bold', fontsi
plt.xlabel('Categories', color = 'red', fontweight='bold', fontsize=17)
plt.ylabel('Number of Streamers', color = 'red', fontweight='bold', fontsize=1

plt.show()
```



Correlation between Subscribers, Visits, LIkes, and Comments:

In [17]:
```
correlation_matrix = df[['Suscribers','Visits','Likes','Comments']].corr()

plt.figure(figsize=(10,8))
sns.heatmap(correlation_matrix, annot = True, cmap='coolwarm',fmt='.2f', linew
plt.title('Correlation Heatmap', color = 'purple',fontweight='bold',fontsize=1

plt.show
```

Out[17]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



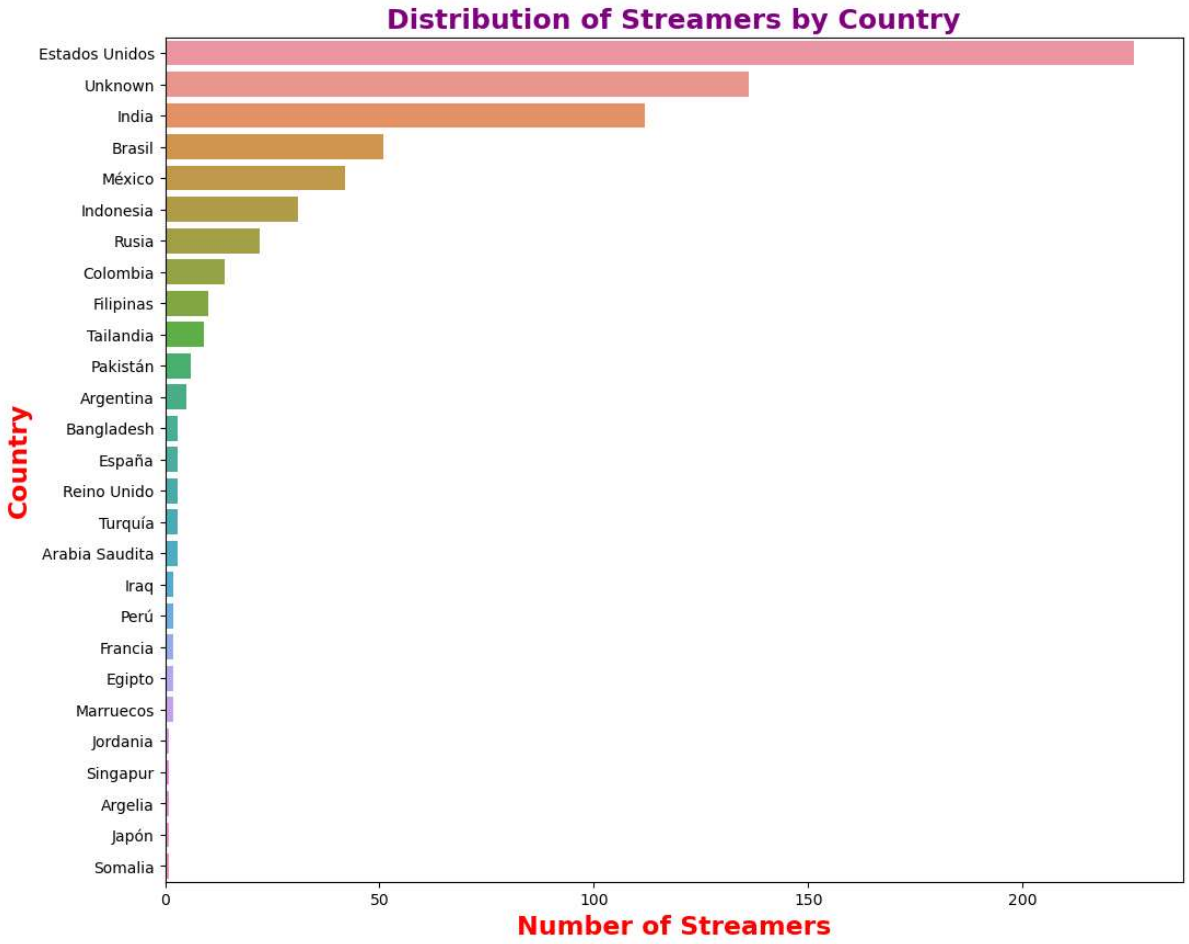
Distribution of streamers audience by country

In [18]:
```
country_category_distribution=df.groupby('Country')['Categories'].count().sort_
country_category_distribution
```

Out[18]:

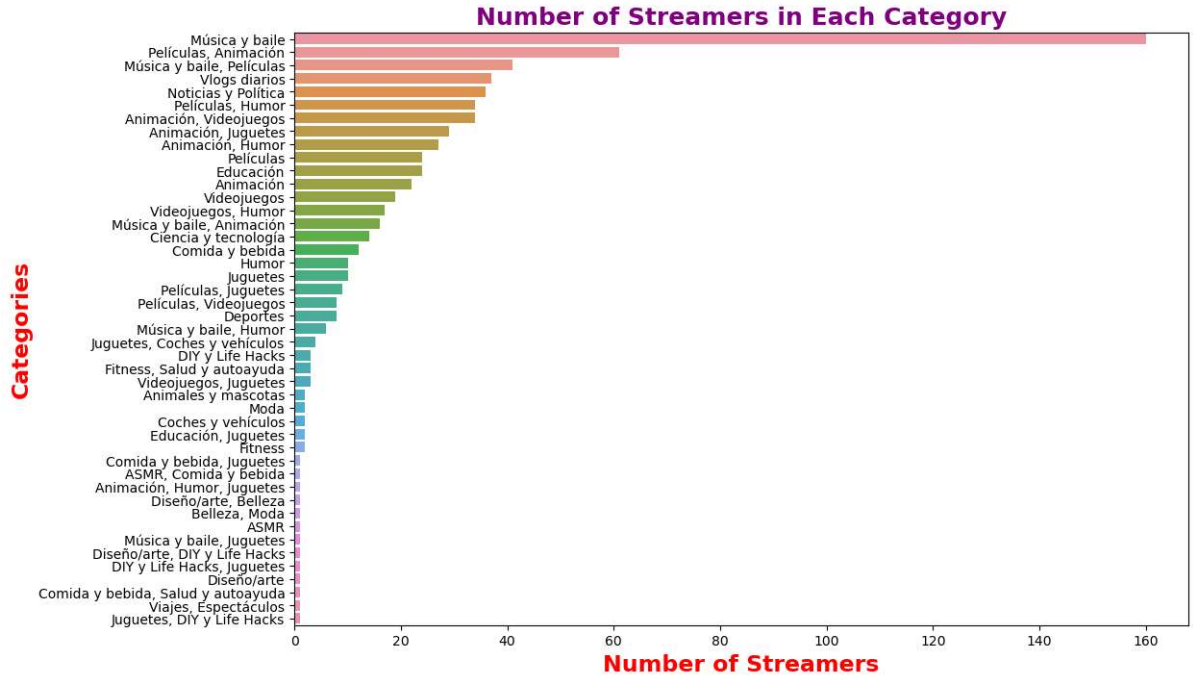
	Country	Categories
0	Estados Unidos	226
1	Unknown	136
2	India	112
3	Brasil	51
4	México	42
5	Indonesia	31
6	Rusia	22
7	Colombia	14
8	Filipinas	10
9	Tailandia	9
10	Pakistán	6
11	Argentina	5
12	Turquía	3
13	Reino Unido	3
14	Arabia Saudita	3
15	España	3
16	Bangladesh	3
17	Marruecos	2
18	Perú	2
19	Francia	2
20	Egipto	2
21	Iraq	2
22	Jordania	1
23	Japón	1
24	Argelia	1
25	Singapur	1
26	Somalia	1

In [19]:
```
plt.figure(figsize=(12, 10))
sns.countplot(y='Country', data=df, order=df['Country'].value_counts().index)

plt.title('Distribution of Streamers by Country', color = 'purple',fontweight=
plt.xlabel('Number of Streamers', color = 'red',fontweight='bold',fontsize=17)
plt.ylabel('Country', color = 'red',fontweight='bold',fontsize=17)

plt.show()
```



Number of Streamers in Each Category

In [20]:
```
plt.figure(figsize=(12, 8))
sns.countplot(y='Categories', data=df, order=df['Categories'].value_counts().ir
plt.title('Number of Streamers in Each Category',color = 'purple',fontweight='l
plt.xlabel('Number of Streamers', color = 'red',fontweight='bold',fontsize=17)
plt.ylabel('Categories', color = 'red',fontweight='bold',fontsize=17)
plt.show()
```



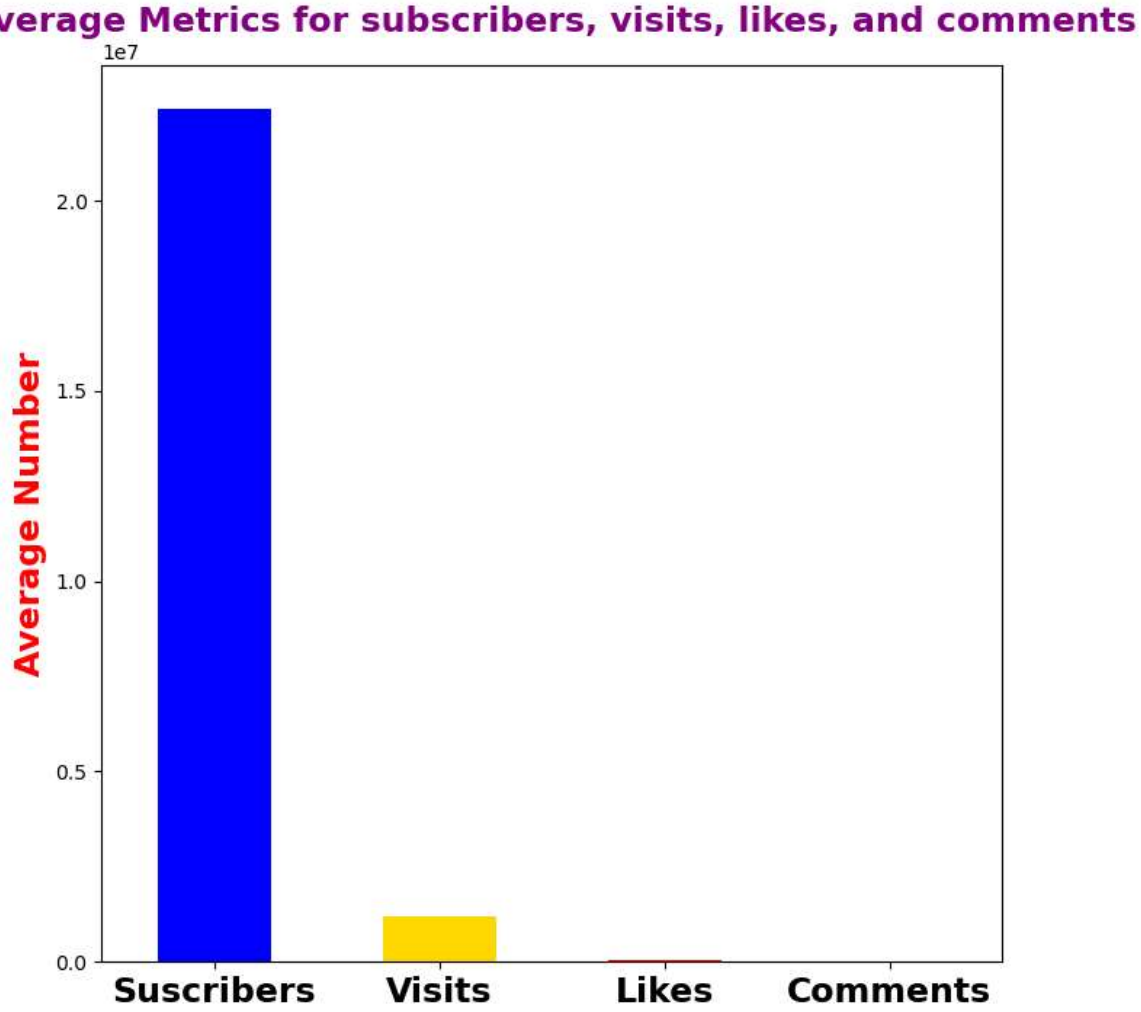
PERFORMANCE METRICS:

Average Metrics For Subscribers, Visits, Likes, And Comments:

In [21]:
```
#Average number of suscribers, visits, likes and comment
average_metrics = df[['Suscribers','Visits','Likes','Comments']].mean()
average_metrics
```

Out[21]:
```
Suscribers    2.241556e+07
Visits        1.210730e+06
Likes         5.347360e+04
Comments      1.558794e+03
dtype: float64
```

In [22]:
```
plt.figure(figsize=(8, 8))
average_metrics.plot(kind='bar',color=['blue','gold','red','yellow'])

plt.title('Average Metrics for subscribers, visits, likes, and comments', colo
plt.ylabel('Average Number',color = 'red',fontweight='bold',fontsize=17)
plt.xticks(rotation=0,fontweight='bold',fontsize=17)

plt.show()
```



CONTENT CATEGORIES:

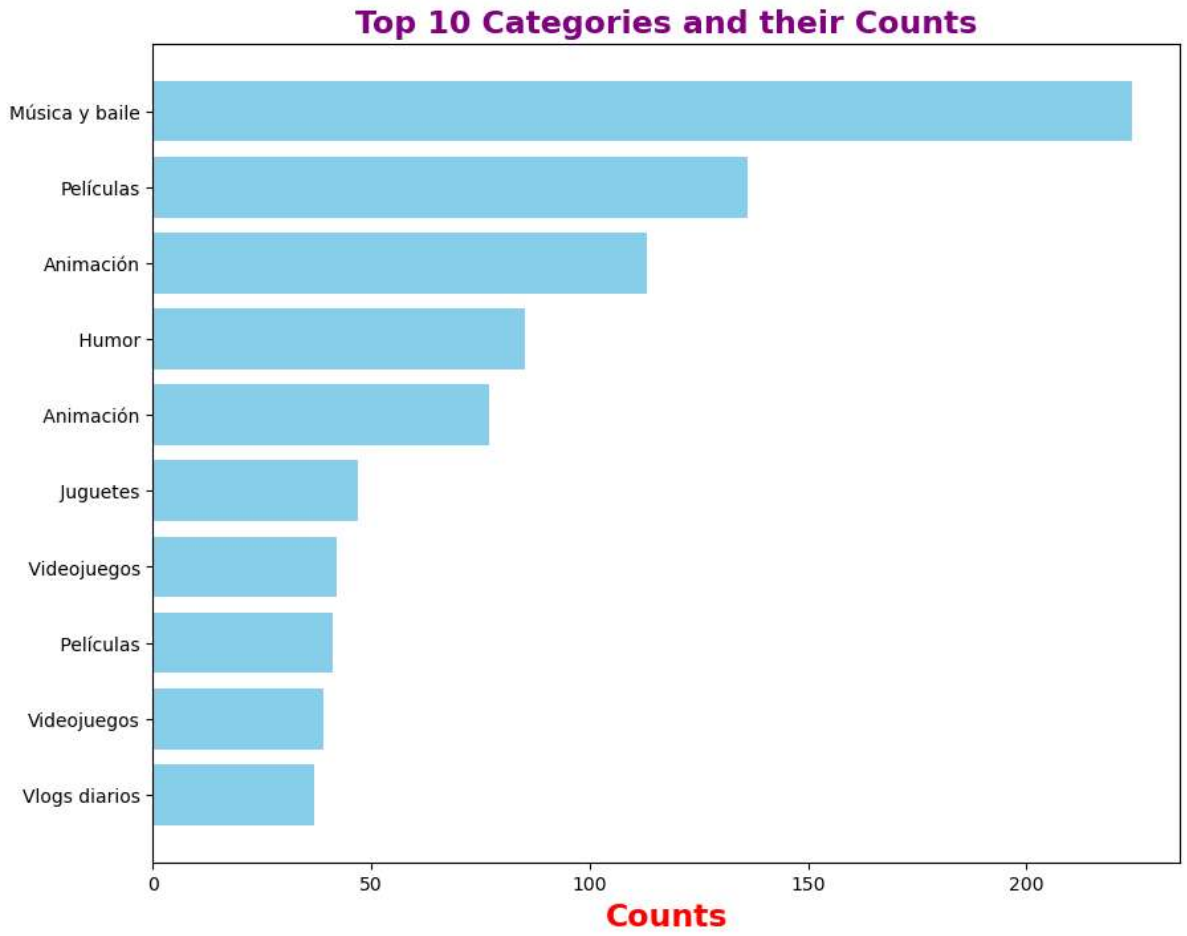
Distribution of Content Categories:

In [23]:
```
a = df['Categories'].str.split(',')

categories = [category for categories in a for category in categories]

category_counts = pd.Series(categories).value_counts()

category_counts_df = pd.DataFrame({'categories': category_counts.index, 'counts

print(category_counts_df)
```

```
              categories  counts
0         Música y baile     224
1              Películas     136
2              Animación     113
3                  Humor      85
4              Animación      77
5               Juguetes      47
6            Videojuegos      42
7              Películas      41
8            Videojuegos      39
9          Vlogs diarios      37
10   Noticias y Política      36
11             Educación      26
12              Juguetes      15
13      Comida y bebida      14
14  Ciencia y tecnología      14
15                 Humor      10
16              Deportes       8
17               Fitness       5
18    Salud y autoayuda       4
19     DIY y Life Hacks       4
20   Coches y vehículos       4
21          Diseño/arte       3
22     DIY y Life Hacks       2
23   Animales y mascotas       2
24                  ASMR       2
25                 Moda       2
26   Coches y vehículos       2
27              Belleza       1
28                 Moda       1
29               Viajes       1
30         Espectáculos       1
31              Belleza       1
32      Comida y bebida       1
```

Top 10 Categories and their Counts:

In [24]:
```
# Plotting the top categories and their counts
plt.figure(figsize=(10, 8))
category_counts_df = category_counts_df.sort_values(by='counts',ascending=Fals
plt.barh(category_counts_df['categories'][:10], category_counts_df['counts'][:

plt.xlabel('Counts',color = 'red',fontweight='bold',fontsize=17)
plt.title('Top 10 Categories and their Counts',color = 'purple',fontweight='bo
plt.gca().invert_yaxis()
# Invert y-axis to display the most frequent category on top
plt.show()
```



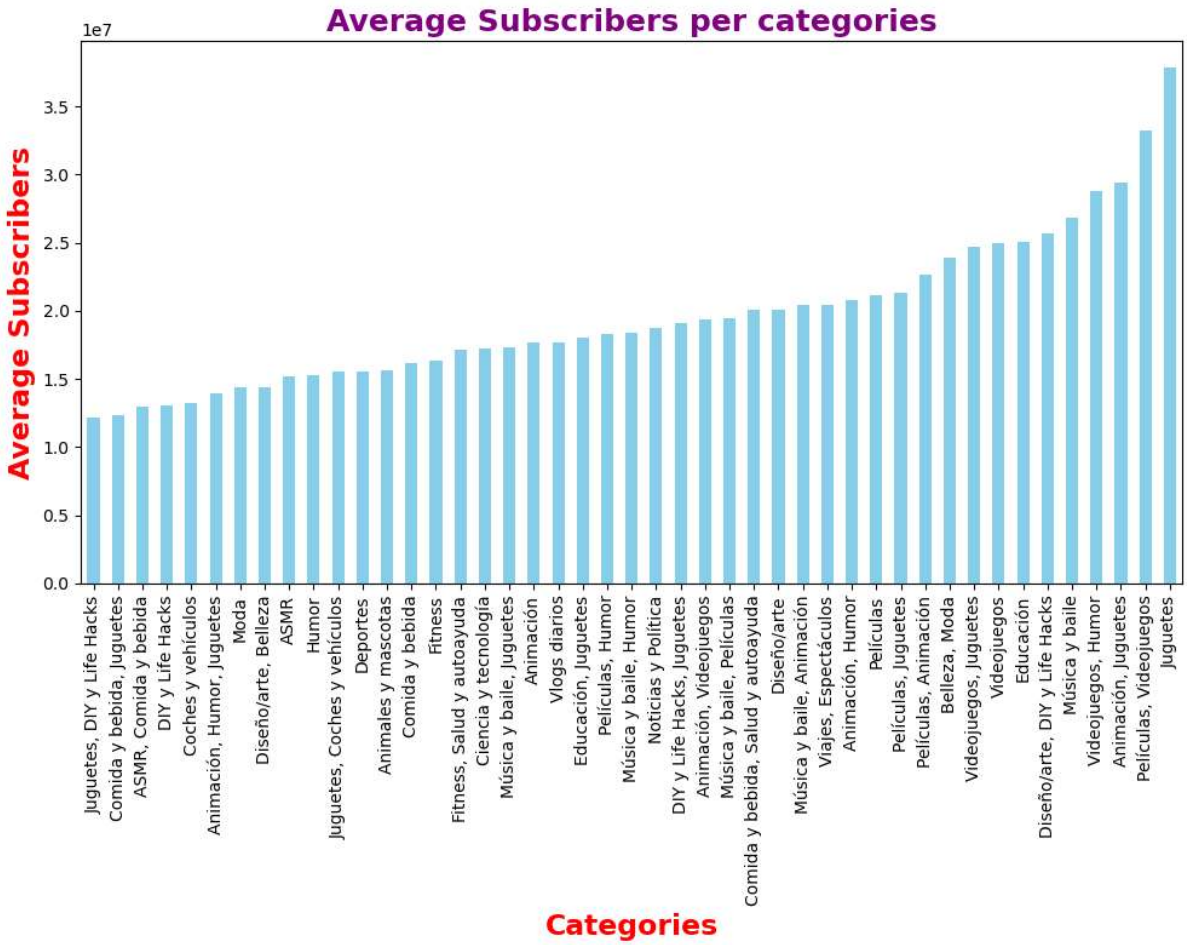
Categories with exceptional performance metrics:

In [25]:
```
#Calculate average subscribers per category
average_subscribers_per_category =df.groupby('Categories')['Suscribers'].mean()

#Sort the average subcribers for each category in ascending order
average_subscribers_per_category_sorted = average_subscribers_per_category.sort

# Display the average subscribers for each category in ascening order
print(average_subscribers_per_category_sorted)
```

```
Categories
Juguetes, DIY y Life Hacks            1.220000e+07
Comida y bebida, Juguetes             1.230000e+07
ASMR, Comida y bebida                 1.300000e+07
DIY y Life Hacks                      1.306667e+07
Coches y vehículos                    1.320000e+07
Animación, Humor, Juguetes            1.390000e+07
Moda                                  1.440000e+07
Diseño/arte, Belleza                  1.440000e+07
ASMR                                  1.520000e+07
Humor                                 1.525000e+07
Juguetes, Coches y vehículos          1.550000e+07
Deportes                              1.552500e+07
Animales y mascotas                   1.560000e+07
Comida y bebida                       1.612500e+07
Fitness                               1.635000e+07
Fitness, Salud y autoayuda            1.710000e+07
Ciencia y tecnología                  1.726429e+07
Música y baile, Juguetes              1.730000e+07
Animación                             1.764091e+07
Vlogs diarios                         1.770000e+07
Educación, Juguetes                   1.805000e+07
Películas, Humor                      1.829706e+07
Música y baile, Humor                 1.838333e+07
Noticias y Política                   1.878056e+07
DIY y Life Hacks, Juguetes            1.910000e+07
Animación, Videojuegos                1.939412e+07
Música y baile, Películas             1.947561e+07
Comida y bebida, Salud y autoayuda    2.010000e+07
Diseño/arte                           2.010000e+07
Música y baile, Animación             2.040000e+07
Viajes, Espectáculos                  2.040000e+07
Animación, Humor                      2.078519e+07
Películas                             2.114167e+07
Películas, Juguetes                   2.130000e+07
Películas, Animación                  2.269344e+07
Belleza, Moda                         2.390000e+07
Videojuegos, Juguetes                 2.473333e+07
Videojuegos                           2.498421e+07
Educación                             2.501250e+07
Diseño/arte, DIY y Life Hacks         2.570000e+07
Música y baile                        2.683688e+07
Videojuegos, Humor                    2.876471e+07
Animación, Juguetes                   2.937586e+07
Películas, Videojuegos                3.325000e+07
Juguetes                              3.788000e+07
Name: Suscribers, dtype: float64
```

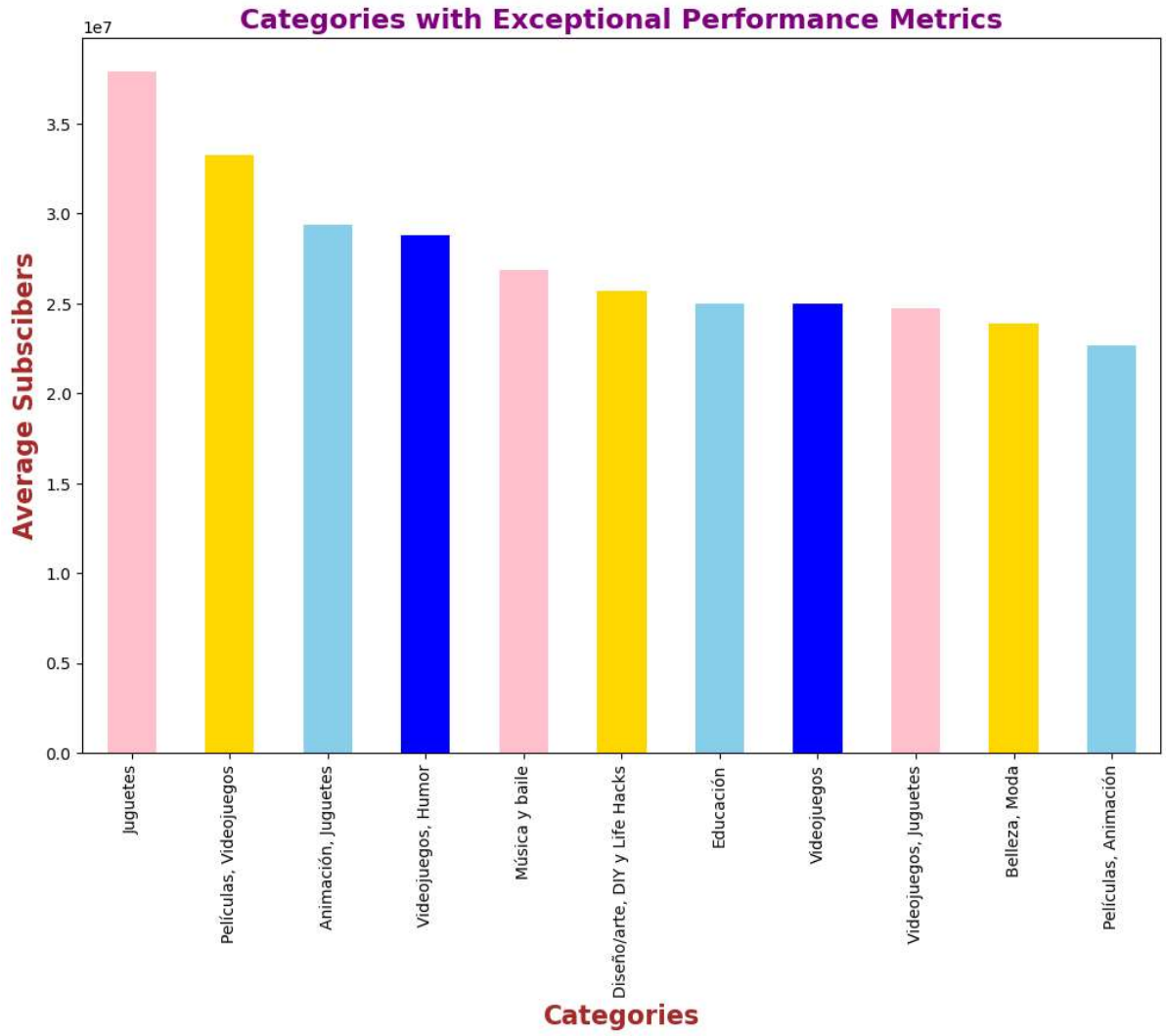
Average Subscribers per Category:

In [26]:
```
#Create a bar plot for average subscribers per category in ascending order
plt.figure(figsize=(10, 8))

average_subscribers_per_category_sorted.plot(kind='bar', color='skyblue')
plt.title('Average Subscribers per categories', color = 'purple',fontweight='b
plt.xlabel('Categories', color = 'red',fontweight='bold',fontsize=17)
plt.ylabel('Average Subscribers', color = 'red',fontweight='bold',fontsize=17)

plt.tight_layout()

plt.show()
```



Categories with Exceptional Performance Metrics:

In [27]:
```
plt.figure(figsize = (12,8))

metric = df['Suscribers'].mean()
b = df.groupby('Categories')['Suscribers'].mean().sort_values(ascending = Fals

colors = ['pink','gold','skyblue','blue']
b = b[b > metric].plot(kind = 'bar', color = colors)

b.set_ylabel('Average Subscibers', color='brown',fontweight='bold', fontsize=1
b.set_xlabel('Categories',color='brown',fontweight='bold', fontsize=16)
b.set_title('Categories with Exceptional Performance Metrics', color='purple',
plt.show()
```



BRANDS AND COLLABORATIONS:

Average Subscribers By Categories:

In [28]:
```
average_metrics = df[['Suscribers','Visits', 'Likes','Comments']].mean()
average_metrics
```

Out[28]:
```
Suscribers    2.241556e+07
Visits        1.210730e+06
Likes         5.347360e+04
Comments      1.558794e+03
dtype: float64
```

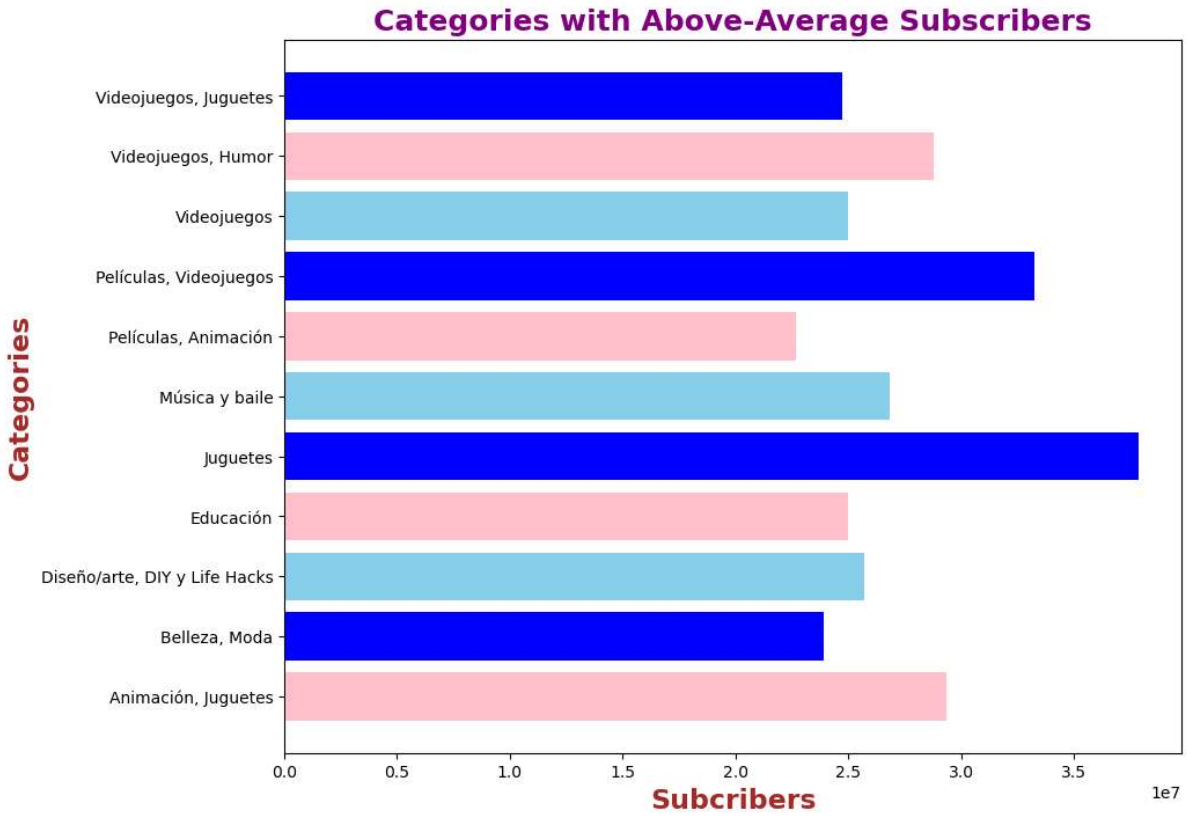
Above Average Subcribers by Categories:

In [29]:
```
a = df[['Categories', 'Suscribers']]
b = a.groupby('Categories')['Suscribers'].mean().reset_index()
above_avg_subscribers = b[b['Suscribers']>average_metrics['Suscribers']][['Cate
above_avg_subscribers
```

Out[29]:

	Categories	Suscribers
5	Animación, Juguetes	2.937586e+07
8	Belleza, Moda	2.390000e+07
19	Diseño/arte, DIY y Life Hacks	2.570000e+07
20	Educación	2.501250e+07
25	Juguetes	3.788000e+07
29	Música y baile	2.683688e+07
36	Películas, Animación	2.269344e+07
39	Películas, Videojuegos	3.325000e+07
41	Videojuegos	2.498421e+07
42	Videojuegos, Humor	2.876471e+07
43	Videojuegos, Juguetes	2.473333e+07

In [30]:
```
plt.figure(figsize=(10, 8))
plt.barh(above_avg_subscribers['Categories'],above_avg_subscribers['Suscribers

plt.title('Categories with Above-Average Subscribers', color='purple',fontweigh
plt.xlabel('Subcribers', color='brown',fontweight='bold', fontsize=17)
plt.ylabel('Categories', color='brown',fontweight='bold', fontsize=17)

plt.show()
```



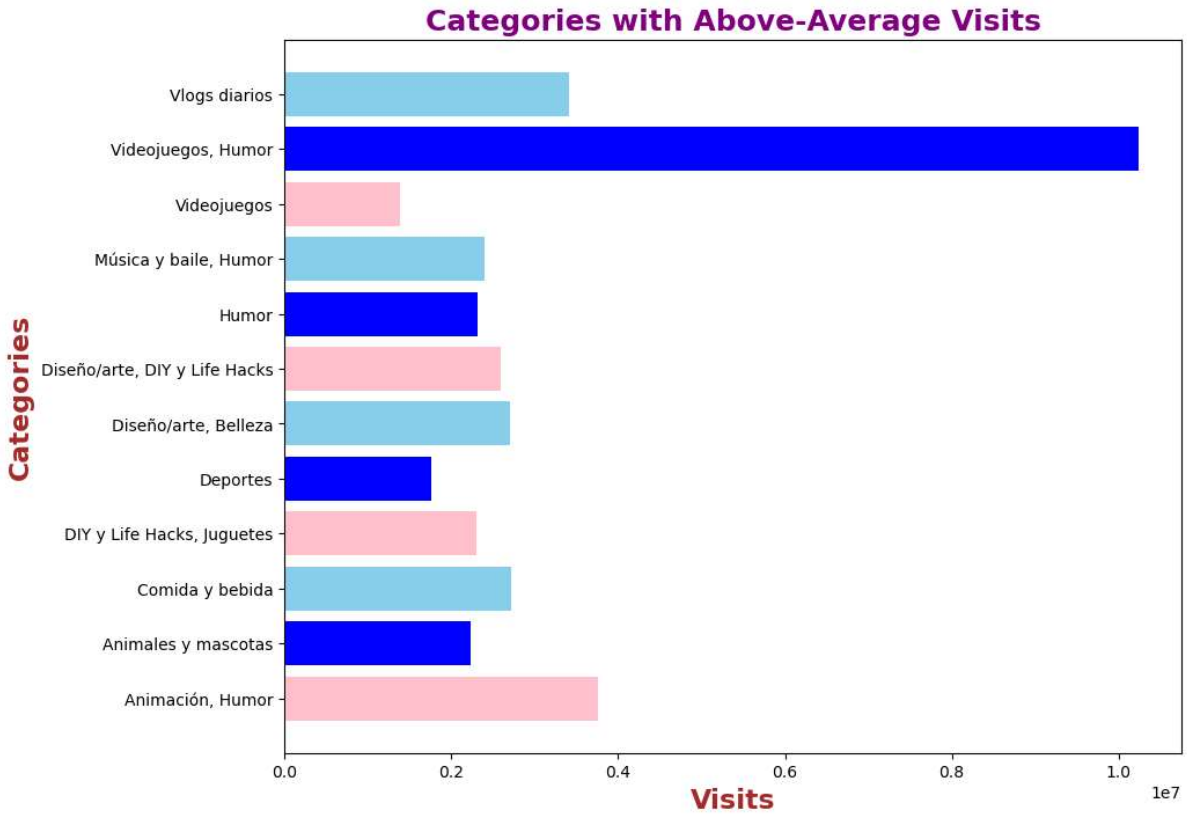
Above Average Visits by Categories:

In [31]:
```
a = df[['Categories', 'Visits']]
b = a.groupby('Categories')['Visits'].mean().reset_index()
above_avg_Visit = b[b['Visits']>average_metrics['Visits']][['Categories','Visi
above_avg_Visit
```

Out[31]:

	Categories	Visits
3	Animación, Humor	3.760126e+06
7	Animales y mascotas	2.231450e+06
11	Comida y bebida	2.722450e+06
15	DIY y Life Hacks, Juguetes	2.300000e+06
16	Deportes	1.759525e+06
18	Diseño/arte, Belleza	2.700000e+06
19	Diseño/arte, DIY y Life Hacks	2.600000e+06
24	Humor	2.310400e+06
31	Música y baile, Humor	2.402933e+06
41	Videojuegos	1.387137e+06
42	Videojuegos, Humor	1.023968e+07
44	Vlogs diarios	3.414338e+06

In [32]:
```
plt.figure(figsize=(10, 8))
plt.barh(above_avg_Visit['Categories'],above_avg_Visit['Visits'],color=['pink'
plt.title('Categories with Above-Average Visits', color='purple',fontweight='bo
plt.xlabel('Visits', color='brown',fontweight='bold', fontsize=17)
plt.ylabel('Categories', color='brown',fontweight='bold', fontsize=17)

plt.show()
```



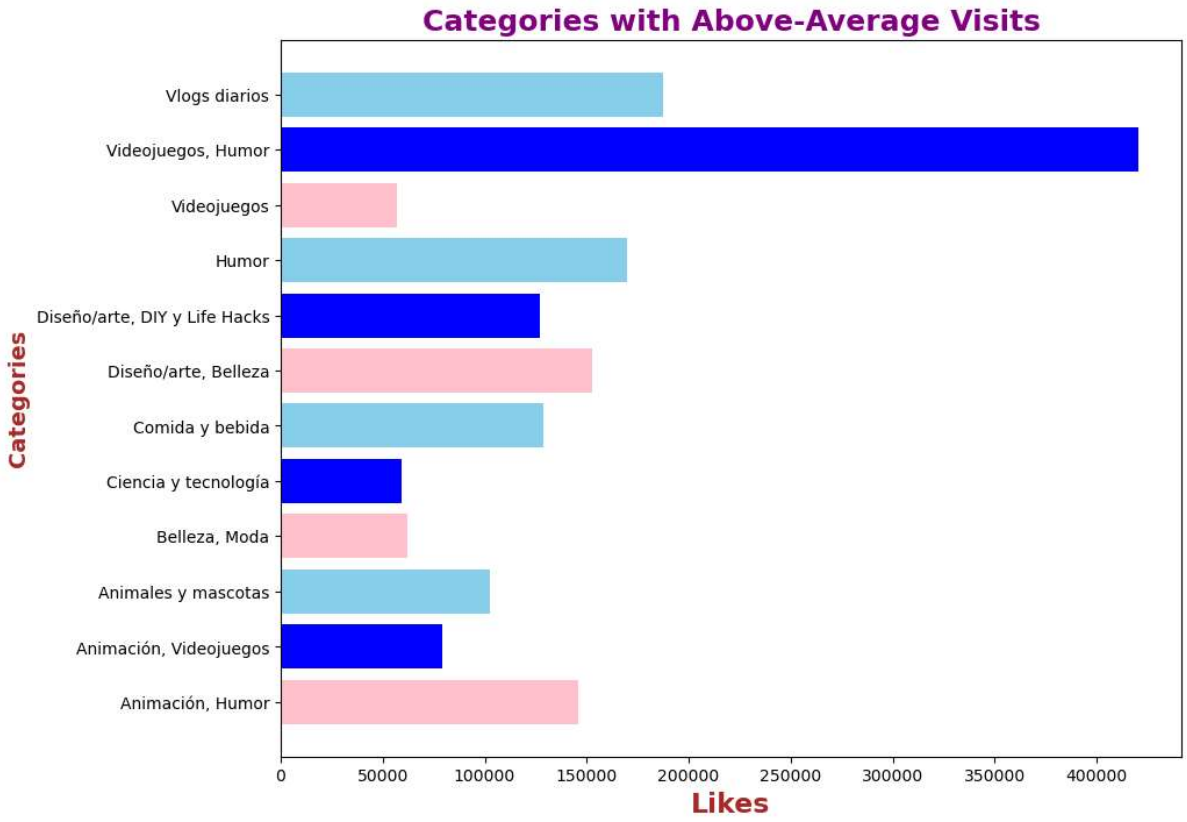
Above Average Subcribers by Likes:

In [33]:
```
a = df[['Categories', 'Likes']]
b = a.groupby('Categories')['Likes'].mean().reset_index()
above_avg_Likes = b[b['Likes']>average_metrics['Likes']][['Categories','Likes']
above_avg_Likes
```

Out[33]:

	Categories	Likes
3	Animación, Humor	145768.333333
6	Animación, Videojuegos	79294.029412
7	Animales y mascotas	102750.000000
8	Belleza, Moda	62300.000000
9	Ciencia y tecnología	59283.142857
11	Comida y bebida	128664.750000
18	Diseño/arte, Belleza	152400.000000
19	Diseño/arte, DIY y Life Hacks	127300.000000
24	Humor	169990.000000
41	Videojuegos	57121.052632
42	Videojuegos, Humor	420511.764706
44	Vlogs diarios	187244.945946

In [34]:
```
plt.figure(figsize=(10, 8))
plt.barh(above_avg_Likes['Categories'],above_avg_Likes['Likes'],color=['pink',

plt.title('Categories with Above-Average Visits', color='purple',fontweight='bo
plt.xlabel('Likes', color='brown',fontweight='bold', fontsize=17)
plt.ylabel('Categories', color='brown',fontweight='bold', fontsize=14)

plt.show()
```



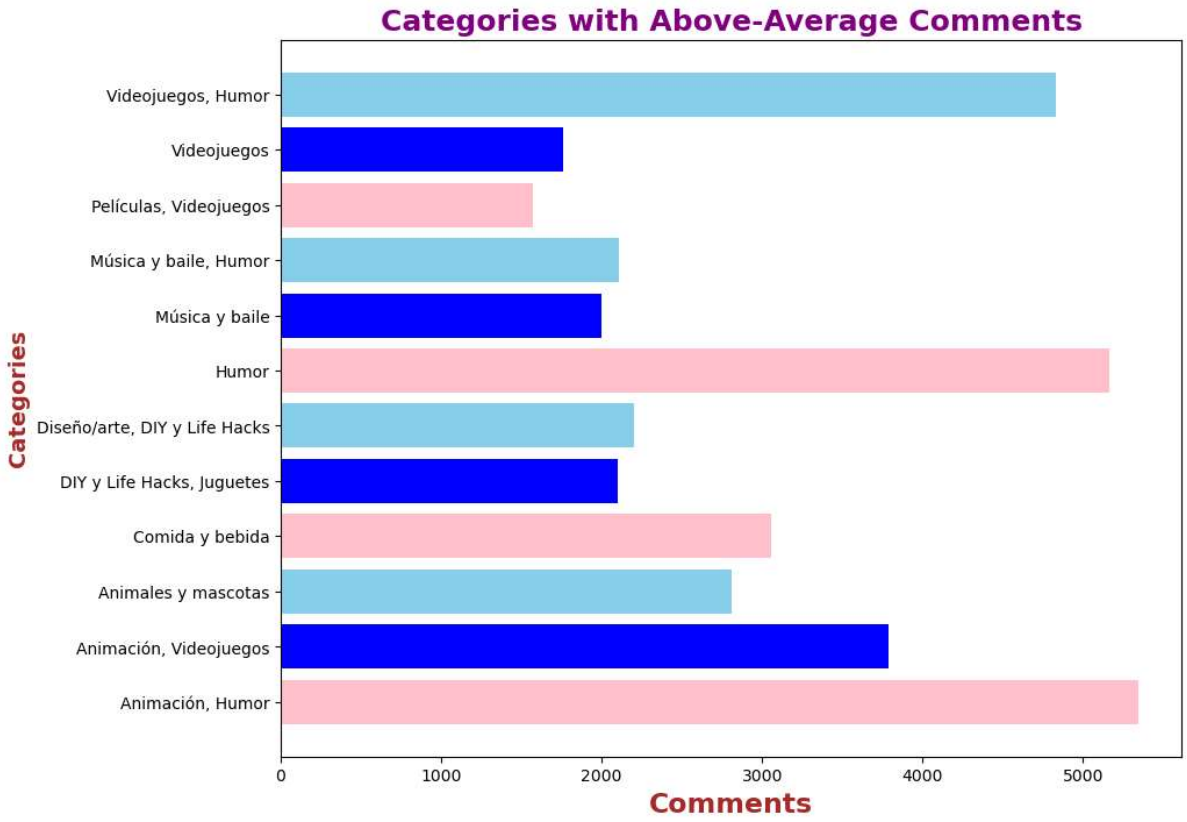
Above Average Subcribers by Comments:

In [35]:
```
a = df[['Categories', 'Comments']]
b = a.groupby('Categories')['Comments'].mean().reset_index()
above_avg_Comments = b[b['Comments']>average_metrics['Comments']][['Categories
above_avg_Comments
```

Out[35]:

	Categories	Comments
3	Animación, Humor	5344.962963
6	Animación, Videojuegos	3786.617647
7	Animales y mascotas	2806.000000
11	Comida y bebida	3053.416667
15	DIY y Life Hacks, Juguetes	2100.000000
19	Diseño/arte, DIY y Life Hacks	2200.000000
24	Humor	5159.800000
29	Música y baile	1998.931250
31	Música y baile, Humor	2110.500000
39	Películas, Videojuegos	1569.500000
41	Videojuegos	1760.157895
42	Videojuegos, Humor	4827.058824

In [36]:
```
plt.figure(figsize=(10, 8))
plt.barh(above_avg_Comments['Categories'],above_avg_Comments['Comments'],color=

plt.title('Categories with Above-Average Comments', color='purple',fontweight=
plt.xlabel('Comments', color='brown',fontweight='bold', fontsize=17)
plt.ylabel('Categories', color='brown',fontweight='bold', fontsize=14)

plt.show()
```



Top performing content creators by Suscribers:

In [37]:
```
top_content_creators = df.groupby('Username')['Suscribers'].mean().sort_values(
top_content_creators
```

Out[37]:

	Username	Suscribers
0	tseries	249500000.0
1	MrBeast	183500000.0
2	CoComelon	165500000.0
3	KidsDianaShow	113500000.0
4	PewDiePie	111500000.0
...	...	...
684	OneindiaHindi	11700000.0
685	LittleAngelIndonesia	11700000.0
686	JoeHattab	11700000.0
687	cut	11700000.0
688	BeAmazed	11700000.0

689 rows × 2 columns

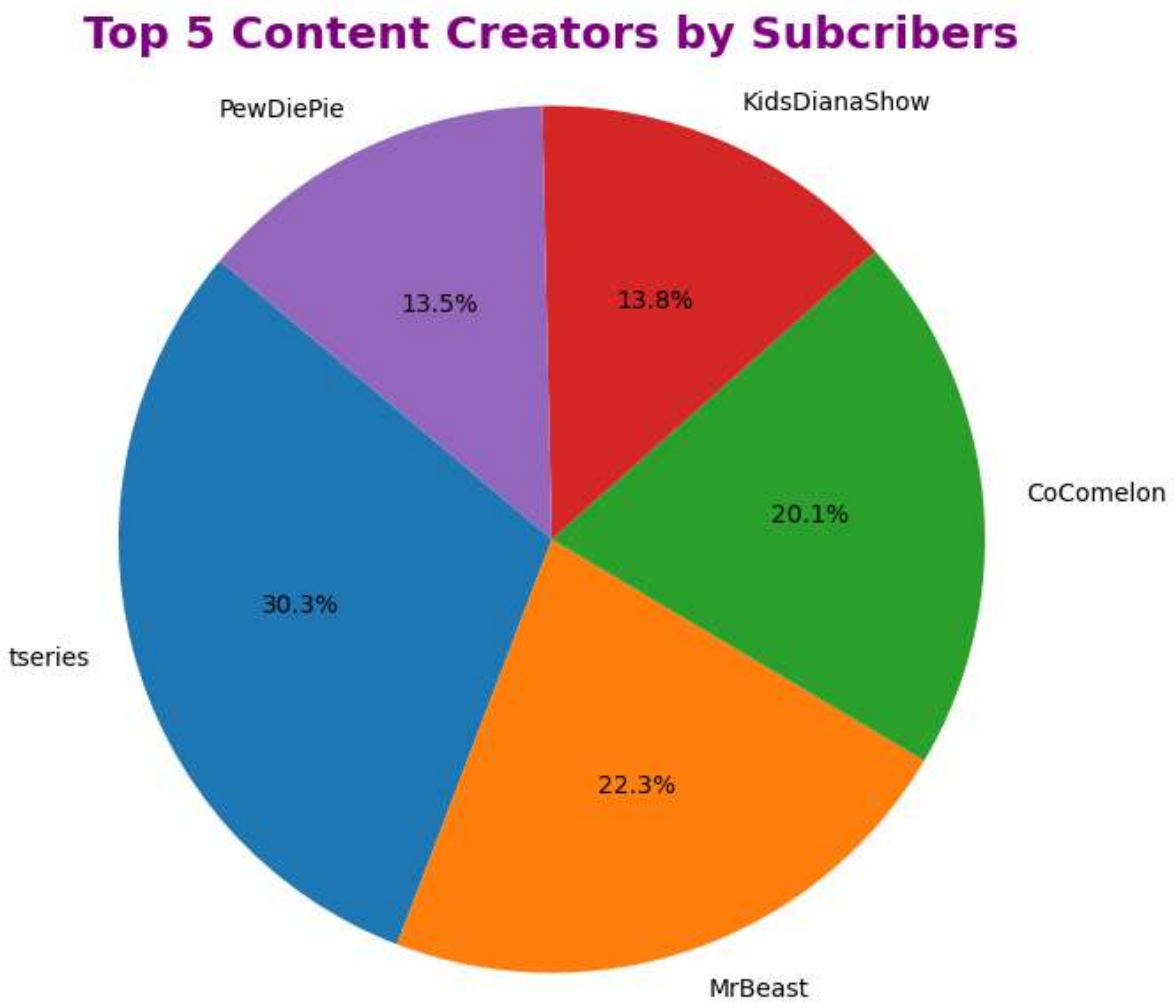
Top 5 Content Creators by Subcribers

In [38]:
```
top_content_creators = df.groupby('Username')['Suscribers'].mean().sort_values(
top_content_creators
```

Out[38]:

	Username	Suscribers
0	tseries	249500000.0
1	MrBeast	183500000.0
2	CoComelon	165500000.0
3	KidsDianaShow	113500000.0
4	PewDiePie	111500000.0

In [39]:
```
plt.figure(figsize=(8, 7))
#Create a pie chart to visualize the top 5 content creators by subscribers
plt.pie(top_content_creators['Suscribers'], labels= top_content_creators['User
plt.axis('equal')
plt.title('Top 5 Content Creators by Subcribers', color='purple',fontweight='b

plt.show()
```



CONTENT RECOMMENDATIONS:

Content recommendations based on streamers categories and performance metrics:

In [40]:
```
recommendations = df.groupby('Categories')['Categories'].count().sort_values(as
recommendations = pd.DataFrame({'Category_Count': recommendations})
recommendations.index.name = 'Categories'
content_recommendations = recommendations.reset_index()
content_recommendations
```

Out[40]:

	Categories	Category_Count
0	Música y baile	160
1	Películas, Animación	61
2	Música y baile, Películas	41
3	Vlogs diarios	37
4	Noticias y Política	36
5	Animación, Videojuegos	34
6	Películas, Humor	34
7	Animación, Juguetes	29
8	Animación, Humor	27
9	Educación	24

In [41]:

```
plt.figure(figsize=(10, 8))
plt.barh(content_recommendations['Categories'],content_recommendations['Categor

plt.title('Content Reecommendations',color = 'purple',fontweight='bold',fontsi
plt.xlabel('Category_Count', color = 'red',fontweight='bold',fontsize=17)
plt.ylabel('Categories', color = 'red',fontweight='bold',fontsize=17)
plt.show()
```