Global Youtube Statistics-2022

In [1]:
```
# importing the libraries;
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
D:\anaconda\lib\site-packages\scipy\__init__.py:138: UserWarning: A NumPy v
ersion >=1.16.5 and <1.23.0 is required for this version of SciPy (detected
version 1.24.4)
  warnings.warn(f"A NumPy version >={np_minversion} and <{np_maxversion} is
required for this version of "
```

Loading dataset;

In [2]:
```
# import dataset;
df = pd.read_csv(r'C:\Users\WINDOWS 10\Documents\Datasets\global\Global YouT
print(df)
```

```
     Rank                    Youtuber  Subscribers   Video Views  \
0       1                    T-Series    245000000  2.280000e+11
1       2              YouTube Movies    170000000  0.000000e+00
2       3                     MrBeast    166000000  2.836884e+10
3       4  Cocomelon - Nursery Rhymes    162000000  1.640000e+11
4       5                   SET India    159000000  1.480000e+11
..    ...                         ...          ...           ...
990   991                Natan por Aï¿     12300000  9.029610e+09
991   992    Free Fire India Official     12300000  1.674410e+09
992   993                       Panda     12300000  2.214684e+09
993   994                  RobTopGames     12300000  3.741235e+08
994   995                 Make Joke Of     12300000  2.129774e+09

             Category                        Title  Uploads         Countr
y  \
0               Music                      T-Series    20082           Indi
a
1    Film & Animation                 youtubemovies        1   United State
s
2       Entertainment                       MrBeast      741   United State
```

Understanding the dataset;

In [3]:
```
# print top 5 rows;
df.head()
```

Out[3]:

	Rank	Youtuber	Subscribers	Video Views	Category	Title	Uploads	Counti
0	1	T-Series	245000000	2.280000e+11	Music	T-Series	20082	Ind
1	2	YouTube Movies	170000000	0.000000e+00	Film & Animation	youtubemovies	1	Unite State
2	3	MrBeast	166000000	2.836884e+10	Entertainment	MrBeast	741	Unite State
3	4	Cocomelon - Nursery Rhymes	162000000	1.640000e+11	Education	Cocomelon - Nursery Rhymes	966	Unite State
4	5	SET India	159000000	1.480000e+11	Shows	SET India	116536	Ind

5 rows × 28 columns

In [4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 995 entries, 0 to 994
Data columns (total 28 columns):
 #   Column                                  Non-Null Count  Dtype  
---  ------                                  --------------  -----  
 0   Rank                                    995 non-null    int64  
 1   Youtuber                                995 non-null    object 
 2   Subscribers                             995 non-null    int64  
 3   Video Views                             995 non-null    float64
 4   Category                                949 non-null    object 
 5   Title                                   995 non-null    object 
 6   Uploads                                 995 non-null    int64  
 7   Country                                 873 non-null    object 
 8   Abbreviation                            873 non-null    object 
 9   Channel Type                            965 non-null    object 
 10  Video Views Rank                        994 non-null    float64
 11  Country Rank                            879 non-null    float64
 12  Channel Type Rank                       962 non-null    float64
 13  Video Views for the last 30 days        939 non-null    float64
 14  Lowest Monthly Earnings                 995 non-null    float64
 15  Highest Monthly Earnings                995 non-null    float64
 16  Lowest Yearly Earnings                  995 non-null    float64
 17  Highest Yearly Earnings                 995 non-null    float64
 18  Subscribers for last 30 days            658 non-null    float64
 19  Created Year                            990 non-null    float64
 20  Created Month                           990 non-null    object 
 21  Created Date                            990 non-null    float64
 22  Gross Tertiary Education Enrollment (%) 872 non-null    float64
 23  Population                              872 non-null    float64
 24  Unemployment Rate                       872 non-null    float64
 25  Urban Population                        872 non-null    float64
 26  Latitude                                872 non-null    float64
 27  Longitude                               872 non-null    float64
dtypes: float64(18), int64(3), object(7)
memory usage: 217.8+ KB
```

In [5]:
```
# last 5 rows of dataset;
df.tail()
```

Out[5]:

	Rank	Youtuber	Subscribers	Video Views	Category	Title	Uploads	Cou
990	991	Natan por Aï¿	12300000	9.029610e+09	Sports	Natan por Aï¿	1200	Bı
991	992	Free Fire India Official	12300000	1.674410e+09	People & Blogs	Free Fire India Official	1500	Iı
992	993	Panda	12300000	2.214684e+09	NaN	HybridPanda	2452	Un Kingc
993	994	RobTopGames	12300000	3.741235e+08	Gaming	RobTopGames	39	Swe
994	995	Make Joke Of	12300000	2.129774e+09	Comedy	Make Joke Of	62	Iı

5 rows × 28 columns

In [6]:
```
# converting suscribers , video views and population, urban population in mi
df['Subscribers'] = (df['Subscribers'] / 1000000).round(2)
df['Video Views'] = (df['Video Views'] / 1000000).round(2)
df["Population"] = (df['Population'] / 1000000).round(2)
df['Urban Population'] = (df['Urban Population']/1000000).round(2)
df.tail()
```

Out[6]:

	Rank	Youtuber	Subscribers	Video Views	Category	Title	Uploads	Country
990	991	Natan por Aï¿	12.3	9029.61	Sports	Natan por Aï¿	1200	Brazil
991	992	Free Fire India Official	12.3	1674.41	People & Blogs	Free Fire India Official	1500	India
992	993	Panda	12.3	2214.68	NaN	HybridPanda	2452	United Kingdom
993	994	RobTopGames	12.3	374.12	Gaming	RobTopGames	39	Sweden
994	995	Make Joke Of	12.3	2129.77	Comedy	Make Joke Of	62	India

5 rows × 28 columns

In [7]:
```
# checking of null values
df.isnull().sum()
```

Out[7]:
```
Rank                                    0
Youtuber                                0
Subscribers                             0
Video Views                             0
Category                               46
Title                                   0
Uploads                                 0
Country                               122
Abbreviation                          122
Channel Type                           30
Video Views Rank                        1
Country Rank                          116
Channel Type Rank                      33
Video Views for the last 30 days       56
Lowest Monthly Earnings                 0
Highest Monthly Earnings                0
Lowest Yearly Earnings                  0
Highest Yearly Earnings                 0
Subscribers for last 30 days          337
Created Year                            5
Created Month                           5
Created Date                            5
Gross Tertiary Education Enrollment (%) 123
Population                            123
Unemployment Rate                     123
Urban Population                      123
Latitude                              123
Longitude                             123
dtype: int64
```

In [8]:
```
df.shape
```

Out[8]:
```
(995, 28)
```

Data Cleaning;

In [9]:
```
# finding the duplicates values;
df.duplicated().sum()
```

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

In [10]:
```
# finding number of unique rows;
df[-df.duplicated()].shape[0]
```

Out[10]:
```
995
```

In [11]:
```
# handling missing values in dataset
NaNnumerical= df.select_dtypes(include=["float64","int64"] ).columns
NaNcategorical= df.select_dtypes(include=["object"]).columns
df[NaNnumerical]= df[NaNnumerical].fillna(0)
df[NaNcategorical]= df[NaNcategorical].fillna("Unknown")
```

In [12]:
```
df.isnull().sum()
```

Out[12]:
```
Rank                                  0
Youtuber                              0
Subscribers                           0
Video Views                           0
Category                              0
Title                                 0
Uploads                               0
Country                               0
Abbreviation                          0
Channel Type                          0
Video Views Rank                      0
Country Rank                          0
Channel Type Rank                     0
Video Views for the last 30 days      0
Lowest Monthly Earnings               0
Highest Monthly Earnings              0
Lowest Yearly Earnings                0
Highest Yearly Earnings               0
Subscribers for last 30 days          0
Created Year                          0
Created Month                         0
Created Date                          0
Gross Tertiary Education Enrollment (%)    0
Population                            0
Unemployment Rate                     0
Urban Population                      0
Latitude                              0
Longitude                             0
dtype: int64
```

In [13]:
```
# analysis of value in statistical format;
describe_df=df.describe()
print(describe_df)
```

```
            Rank  Subscribers    Video Views        Uploads  Video Views Ra
nk  \
count  995.00000   995.000000     995.000000     995.000000      9.950000e+
02
mean   498.00000    22.982412   11039.536724    9187.125628      5.536919e+
05
std    287.37606    17.526105   14110.844384   34151.352254      1.362210e+
06
min      1.00000    12.300000       0.000000       0.000000      0.000000e+
00
25%    249.50000    14.500000    4288.145000     194.500000      3.175000e+
02
50%    498.00000    17.700000    7760.820000     729.000000      9.130000e+
02
75%    746.50000    24.600000   13554.700000    2667.500000      3.579000e+
03
max    995.00000   245.000000  228000.000000  301308.000000      4.057944e+
06

       Country Rank  Channel Type Rank  Video Views for the last 30 days  \
count    995.000000         995.000000                      9.950000e+02
mean     341.046231         720.986935                      1.657267e+08
std     1164.728018        1916.500613                      4.065010e+08
min        0.000000           0.000000                      0.000000e+00
25%        5.000000          24.000000                      1.351700e+07
50%       34.000000          62.000000                      5.635800e+07
75%      114.000000         137.000000                      1.585655e+08
max     7741.000000        7741.000000                      6.589000e+09

       Lowest Monthly Earnings  Highest Monthly Earnings  ...  \
count               995.000000              9.950000e+02  ...
mean              36886.148281              5.898078e+05  ...
std               71858.724092              1.148622e+06  ...
min                   0.000000              0.000000e+00  ...
25%                2700.000000              4.350000e+04  ...
50%               13300.000000              2.127000e+05  ...
75%               37900.000000              6.068000e+05  ...
max              850900.000000              1.360000e+07  ...

       Highest Yearly Earnings  Subscribers for last 30 days  Created Year
\
count             9.950000e+02                  9.950000e+02    995.000000
mean              7.081814e+06                  2.308483e+05   2002.516583
std               1.379704e+07                  5.261092e+05    142.455353
min               0.000000e+00                  0.000000e+00      0.000000
25%               5.217500e+05                  0.000000e+00   2009.000000
50%               2.600000e+06                  1.000000e+05   2013.000000
75%               7.300000e+06                  2.000000e+05   2016.000000
max               1.634000e+08                  8.000000e+06   2022.000000

       Created Date  Gross Tertiary Education Enrollment (%)   Population
\
count    995.000000                                995.000000   995.000000
mean      15.667337                                 55.762211   377.183980
std        8.826000                                 32.191178   464.717802
min        0.000000                                  0.000000     0.000000
25%        8.000000                                 28.100000    50.340000
50%       16.000000                                 60.000000   270.200000
75%       23.000000                                 88.200000   328.240000
max       31.000000                                113.100000  1397.720000
```

```
       Unemployment Rate  Urban Population     Latitude    Longitude
count         995.000000        995.000000  995.000000  995.000000
mean            8.132191        196.496844   23.340489  -12.381652
std             5.502432        162.538385   21.150562   79.479723
min             0.000000          0.000000  -38.416097 -172.104629
25%             3.850000         40.830000    4.570868  -95.712891
50%             5.360000        183.240000   23.634501   -3.435973
75%            14.700000        270.660000   37.090240   78.962880
max            14.720000        842.930000   61.924110  138.252924

[8 rows x 21 columns]
```

Data manipulation;

In [14]:
```
# rename the column name ;
df.rename(columns={"Unemployment rate": "Unemployment rate(%)"}, inplace=Tru

# Display the modified DataFrame
df.head()
```

Out[14]:

	Rank	Youtuber	Subscribers	Video Views	Category	Title	Uploads	Country
0	1	T-Series	245.0	228000.00	Music	T-Series	20082	India
1	2	YouTube Movies	170.0	0.00	Film & Animation	youtubemovies	1	United States
2	3	MrBeast	166.0	28368.84	Entertainment	MrBeast	741	United States
3	4	Cocomelon - Nursery Rhymes	162.0	164000.00	Education	Cocomelon - Nursery Rhymes	966	United States
4	5	SET India	159.0	148000.00	Shows	SET India	116536	India

5 rows × 28 columns

In [15]:
```
# filter the rows where the videos views are not equal to 0;
df[df['Video Views']!=0]
```

Out[15]:

	Rank	Youtuber	Subscribers	Video Views	Category	Title	Uploads	Cou
0	1	T-Series	245.0	228000.00	Music	T-Series	20082	I
2	3	MrBeast	166.0	28368.84	Entertainment	MrBeast	741	Ur St
3	4	Cocomelon - Nursery Rhymes	162.0	164000.00	Education	Cocomelon - Nursery Rhymes	966	Ur St
4	5	SET India	159.0	148000.00	Shows	SET India	116536	I
6	7	ýýý Kids Diana Show	112.0	93247.04	People & Blogs	ýýý Kids Diana Show	1111	Ur St
...	...	...	...	...	...	...	...	
990	991	Natan por Aï¿	12.3	9029.61	Sports	Natan por Aï¿	1200	B
991	992	Free Fire India Official	12.3	1674.41	People & Blogs	Free Fire India Official	1500	I
992	993	Panda	12.3	2214.68	Unknown	HybridPanda	2452	Ur King
993	994	RobTopGames	12.3	374.12	Gaming	RobTopGames	39	Swe
994	995	Make Joke Of	12.3	2129.77	Comedy	Make Joke Of	62	I

986 rows × 28 columns

In [16]:
```
# creating new column in dataset named 'AvarageYearlyEarnings'
df['AverageYearlyEarnings'] = (df['Lowest Yearly Earnings'] + df['Highest Ye
df['AverageYearlyEarnings'] = (df['AverageYearlyEarnings'] / 1000000).round(
df.head()
```

Out[16]:

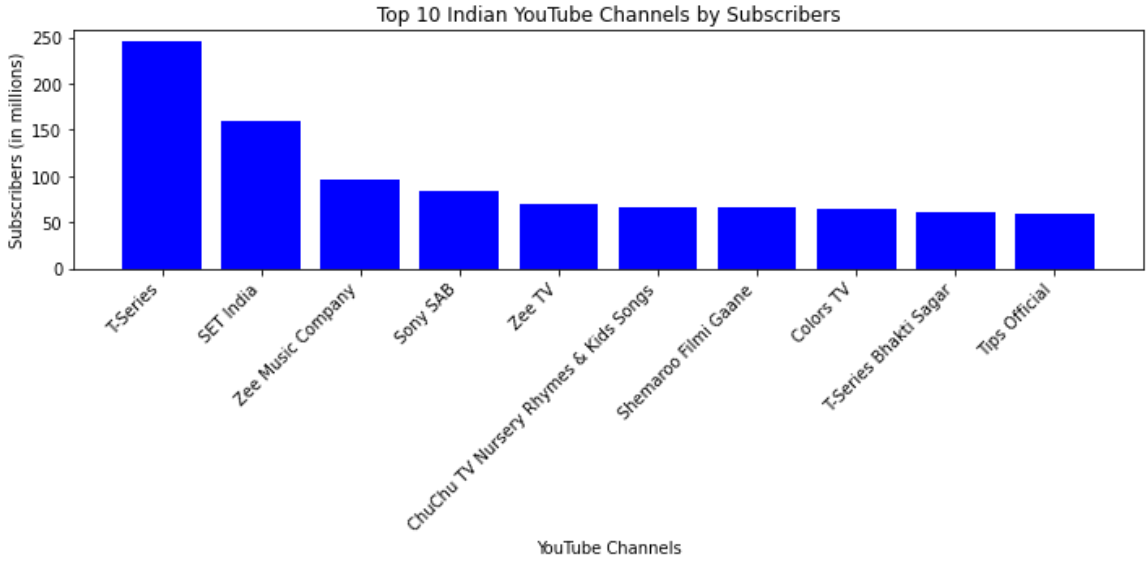
	Rank	Youtuber	Subscribers	Video Views	Category	Title	Uploads	Country
0	1	T-Series	245.0	228000.00	Music	T-Series	20082	India
1	2	YouTube Movies	170.0	0.00	Film & Animation	youtubemovies	1	United States
2	3	MrBeast	166.0	28368.84	Entertainment	MrBeast	741	United States
3	4	Cocomelon - Nursery Rhymes	162.0	164000.00	Education	Cocomelon - Nursery Rhymes	966	United States
4	5	SET India	159.0	148000.00	Shows	SET India	116536	India

5 rows × 29 columns

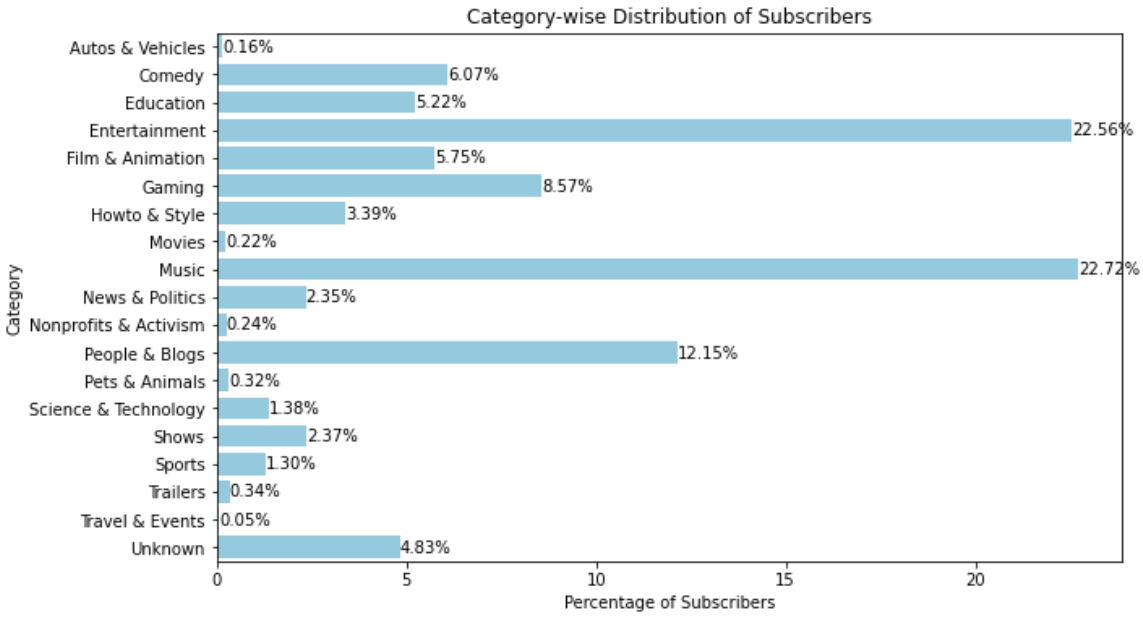
Data analysis and visualization;

In [17]:
```
#Insight 1: T-Series leads with the highest YouTube subscribers in India,
# followed by entertainment and music channels like SET India and Zee Music
#Root Cause:Dominance of music and entertainment content suggests a strong p
#these genres among Indian viewers.
#Provable Solution: Content creators targeting the Indian audience should co
# entertainment content to tap into the popularity of these genres and poten

# find the top 10 indian youtube subscribers;
indian_channels = df[df['Abbreviation'] == 'IN']
top_10_indian_channels = indian_channels.sort_values(by='Subscribers', ascen

print(top_10_indian_channels[['Youtuber', 'Subscribers']])

import matplotlib.pyplot as plt

plt.figure(figsize=(10, 5))
plt.bar(top_10_indian_channels['Youtuber'], top_10_indian_channels['Subscrib
plt.xlabel('YouTube Channels')
plt.ylabel('Subscribers (in millions)')
plt.title('Top 10 Indian YouTube Channels by Subscribers')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()

# Display the plot
plt.show()
```

	Youtuber	Subscribers
0	T-Series	245.0
4	SET India	159.0
10	Zee Music Company	96.7
15	Sony SAB	83.0
21	Zee TV	70.5
24	ChuChu TV Nursery Rhymes & Kids Songs	65.9
25	Shemaroo Filmi Gaane	65.6
26	Colors TV	64.6
27	T-Series Bhakti Sagar	61.0
30	Tips Official	59.3

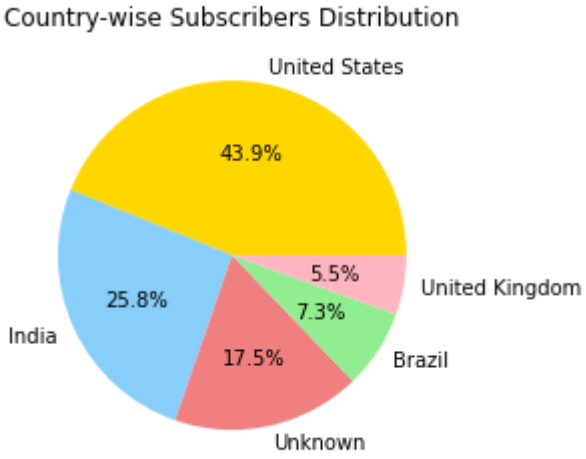


In [18]:
```
# category wise subscribers in percentage in horizontal bar chart;
#Insight 2:Music and Entertainment categories have the highest percentage of
#indicating a strong audience preference for these genres.
#Root Cause:The popularity of music and entertainment content may be due to
#ability to engage a wide audience.
#Provable Solution:Content creators should strategically incorporate music a
#aligning with the preferences of the majority of subscribers, to maximize e

import matplotlib.pyplot as plt
import seaborn as sns

category_subscribers = df.groupby('Category')['Subscribers'].sum()
total_subscribers = df['Subscribers'].sum()

percentage_subscribers = (category_subscribers / total_subscribers) * 100
percentage_subscribers = percentage_subscribers.reset_index()

plt.figure(figsize=(10, 6))
bar_plot = sns.barplot(x='Subscribers', y='Category', data=percentage_subscr

# Add data labels
for index, value in enumerate(percentage_subscribers['Subscribers']):
    bar_plot.text(value, index, f'{value:.2f}%', va='center')

plt.xlabel('Percentage of Subscribers')
plt.ylabel('Category')
plt.title('Category-wise Distribution of Subscribers')

# Horizontal bar chart with data labels
plt.show()
```

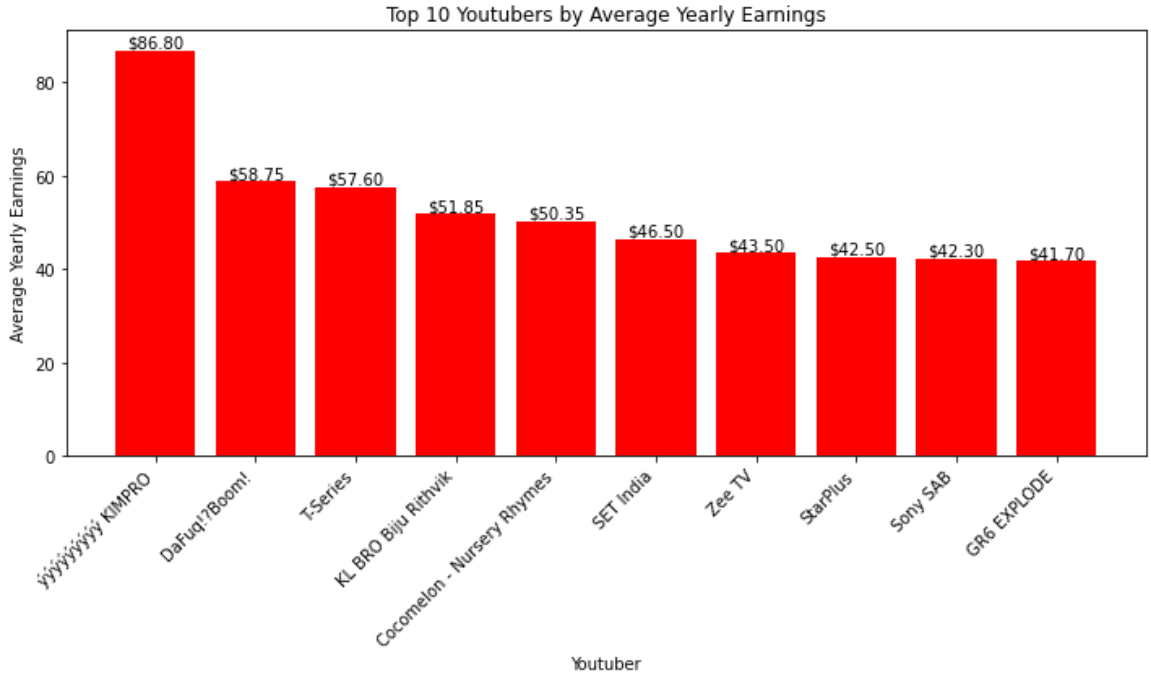


In [ ]:

In [19]:
```
# Insight 3:The distribution of country-wise subscribers reveals that the Un
# has the highest percentage of subscribers at 43.9%, followed by India at 2
# Brazil at 7.3%, and the United Kingdom at 5.5%.
# Root Cause: Variations in subscriber percentages can be attributed to fact
# internet penetration, cultural interests, and the popularity of YouTube co
# Provable Solution:Localized Content Develop Create content that aligns wit
# interests of the dominant market
# Pie chart of country-wise subscribers
country_subscribers = df.groupby('Country')['Subscribers'].sum().sort_values
plt.pie(country_subscribers, labels=country_subscribers.index, autopct='%1.1
        colors=['gold', 'lightskyblue', 'lightcoral', 'lightgreen', 'lightpi

plt.title('Country-wise Subscribers Distribution')
plt.show()
```

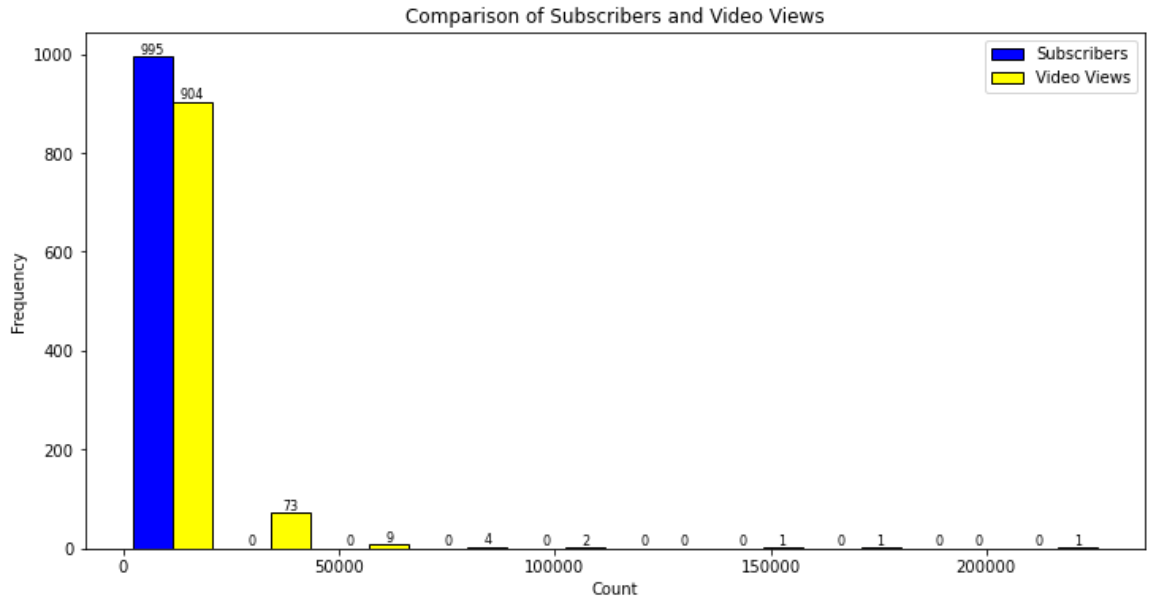


In [20]:
```
#Insight 4:KIMPRO tops the list with the highest average yearly earnings, fo
# indicating varied revenue streams and successful monetization strategies
#Root Cause:Diverse revenue sources, including sponsorships, merchandise, an
#contribute to the high average earnings.
#Provable Solution:Content creators should explore multiple revenue streams
#such as collaborations, merchandise sales, and affiliate partnerships, to o
# Top 10 youtubers by average yearly earnings:
import matplotlib.pyplot as plt


temp = df[['Youtuber', 'AverageYearlyEarnings']].sort_values(by='AverageYear

plt.figure(figsize=(10, 6))
plt.bar(temp['Youtuber'], temp['AverageYearlyEarnings'], color='red')
plt.xlabel('Youtuber')
plt.ylabel('Average Yearly Earnings')
plt.title('Top 10 Youtubers by Average Yearly Earnings')
plt.xticks(rotation=45, ha='right')

# Data labels
for index, value in enumerate(temp['AverageYearlyEarnings']):
    plt.text(index, value, f"${value:.2f}", ha='center', va='bottom')

plt.tight_layout()
plt.show()
```



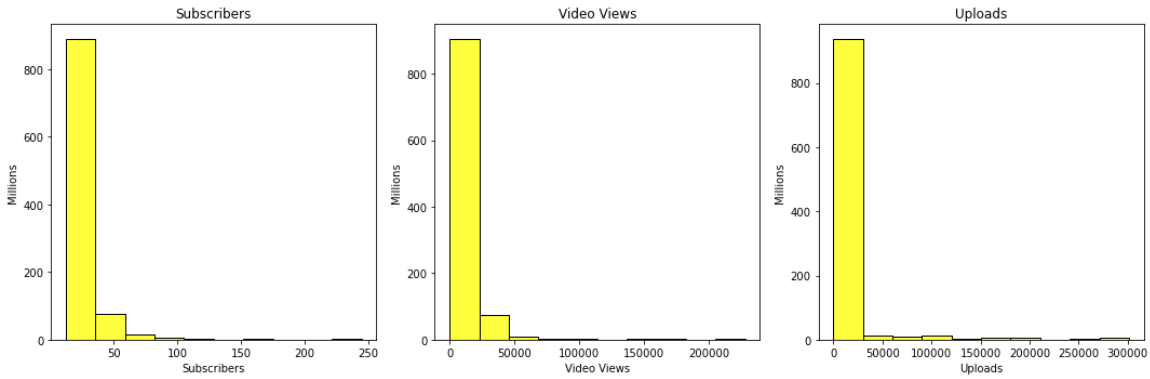
In [21]:

```
#Insight 5: There is a positive correlation between the number of subscriber
# suggesting that channels with more subscribers tend to attract higher view
#Root Cause:Engaged subscribers are more likely to watch videos regularly, c
#Provable Solution:Focus on building a loyal subscriber base through consist
# higher video views. Encourage user interactions, such as likes, comments,
# Comparison of Subscriber and Video Views:
import matplotlib.pyplot as plt
import pandas as pd

plt.figure(figsize=(12, 6))

_, _, patches = plt.hist([df['Subscribers'], df['Video Views']], bins=10, co

# Add data labels on the bar columns
for i in range(len(patches[0])):
    plt.text(patches[0][i].get_x() + patches[0][i].get_width() / 2, patches[
             str(int(patches[0][i].get_height())), ha='center', va='bottom',

for i in range(len(patches[1])):
    plt.text(patches[1][i].get_x() + patches[1][i].get_width() / 2, patches[
             str(int(patches[1][i].get_height())), ha='center', va='bottom',
# Histogram

plt.xlabel('Count')
plt.ylabel('Frequency')
plt.title('Comparison of Subscribers and Video Views')
plt.legend()

plt.show()
```



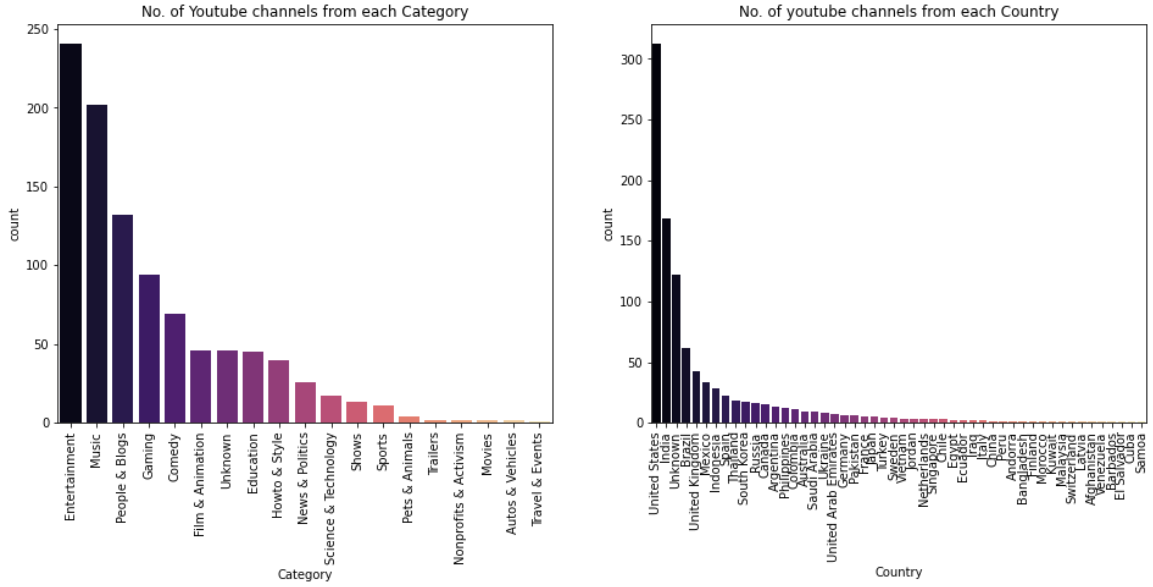
In [22]:

```
# Histograms for the 'Subscribers,' 'Video Views,' and 'Uploads' columns.
# Insight 6: The distribution of subscribers, Video views and uploads likely
# a significantly higher number of subscribers,suggesting that a few videos
# Root Cause: The concentration of subscribers may be influenced by the type
# Channels with a lower number of uploads may prioritize quality content, wh
# Provable Solution: Implement targeted marketing strategies to increase cha
# Focus on creating high-quality and engaging content that resonates with th
#Strike a balance between quality and quantity based on audience preferences
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd

columns_to_plot =['Subscribers', 'Video Views', 'Uploads']



plt.figure(figsize=(15, 5))
for i, column in enumerate(columns_to_plot, 1):
    plt.subplot(1, 3, i)
    sns.histplot(df[column], bins=10, kde=False, color='yellow', edgecolor='
    plt.title(f'{column}')
    plt.xlabel(column)
    plt.ylabel('Millions')

plt.tight_layout()
plt.show()
```

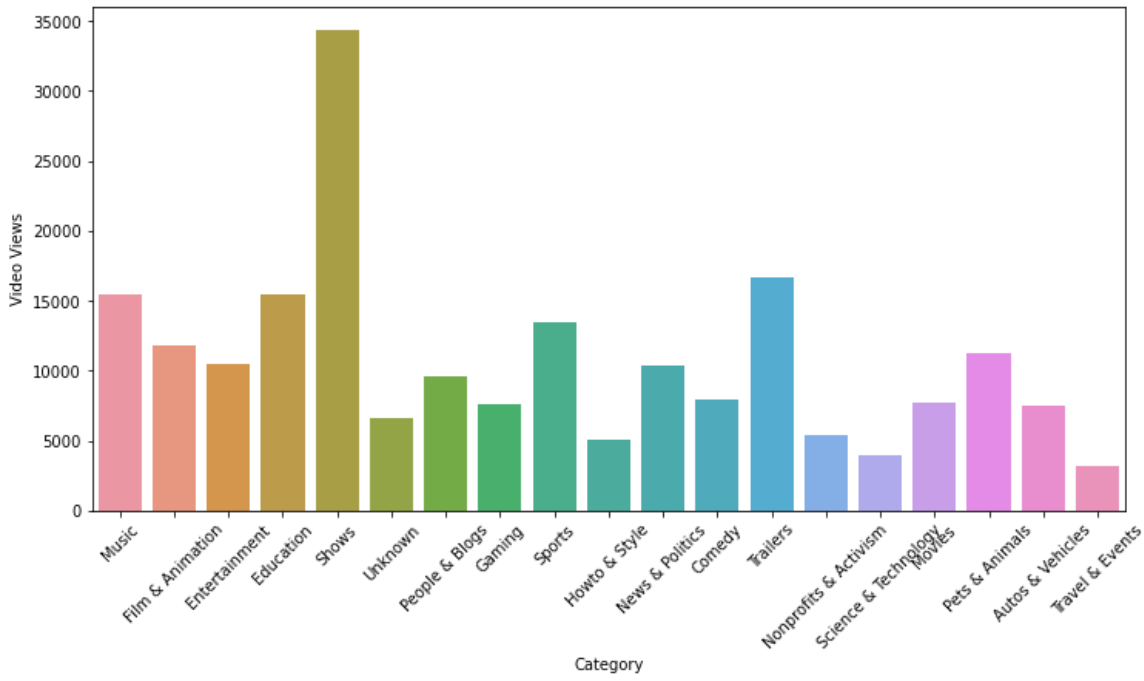


In [23]:
```
#Insight 7:The count plots reveal that certain categories and countries domi
#indicating popular content and geographical concentrations.
#Root Cause:Variances in the number of channels across categories and countr
# regional trends, and audience demographics.
#Provable Solution:Content creators should analyze popular categories and ta
#potentially expanding into underrepresented categories and countries to div
# No. of youtube channels from each category and country:
fig,ax=plt.subplots(nrows=1,ncols=2,figsize=(16,6))
sns.countplot(x='Category',data=df,order=df['Category'].value_counts(ascendi
sns.countplot(x='Country',data=df,order=df['Country'].value_counts(ascending
ax[0].set_xticklabels(list(df['Category'].value_counts(ascending=False).inde
ax[1].set_xticklabels(list(df['Country'].value_counts(ascending=False).index
ax[0].set_title('No. of Youtube channels from each Category')
ax[1].set_title('No. of youtube channels from each Country')
```

Out[23]:
```
Text(0.5, 1.0, 'No. of youtube channels from each Country')
```

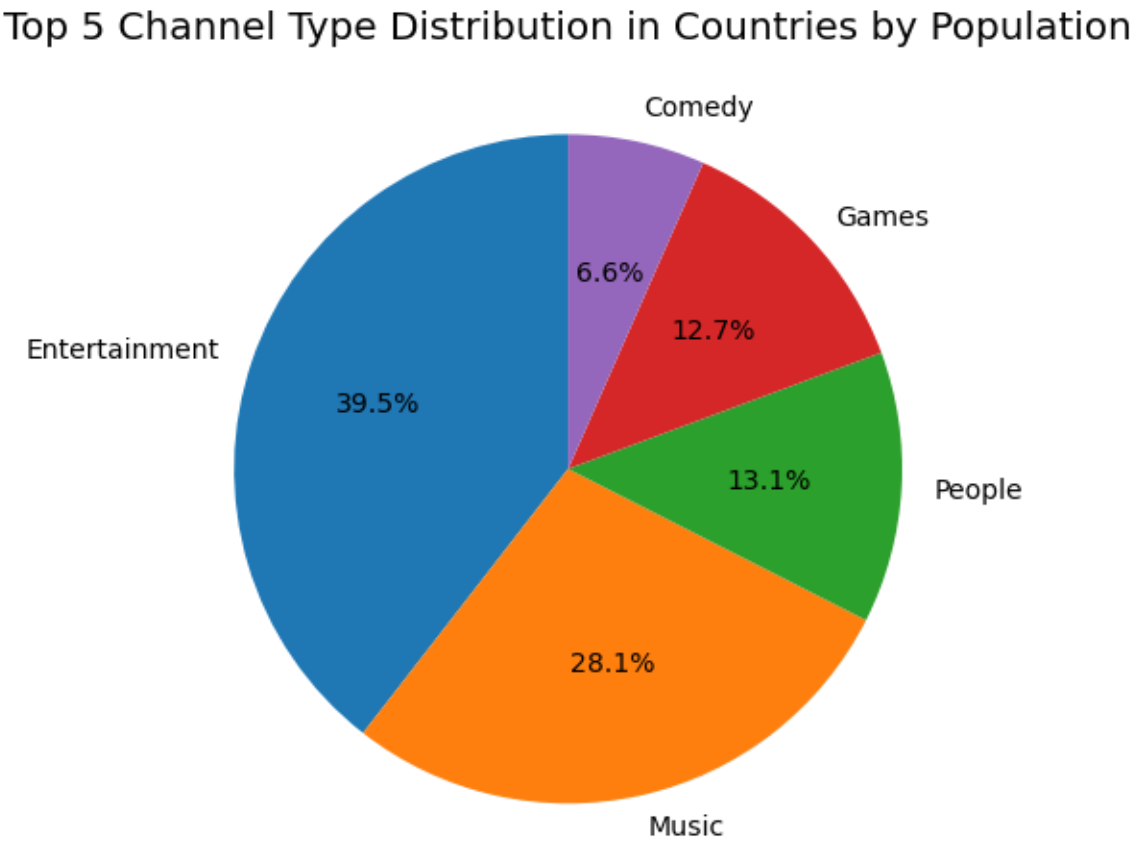


In [24]:
```
#Insight 8:The bar plot illustrates the distribution of video views across d
# highlighting variations in popularity.
#Root Cause:Differences in video views by category may stem from varying aud
#content quality, or trends in specific categories.
#Provable Solution:Content creators should analyze the performance of each c
#and strategically align content creation with high-performing categories to
#Consistent monitoring of category-wise performance can guide content strate
#-----Video Views Distribution by Category
plt.figure(figsize=(12, 6))
sns.barplot(x='Category', y='Video Views', data=df, ci=None)
plt.xticks(rotation=45)
plt.show()
```

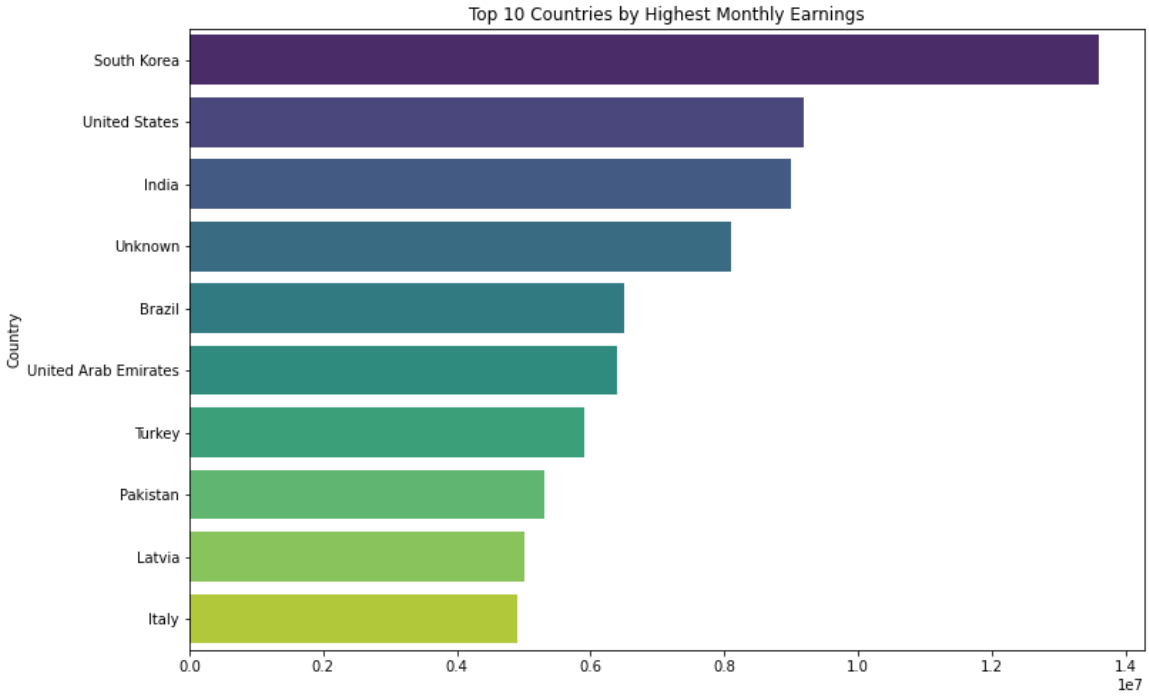


In [25]:
```
#Insight 9:The pie chart visually represents the distribution of Top 5 chann
# revealing the dominant channel types in these high-population regions.
#Root Cause:Variances in channel type distribution may be influenced by cult
#or content consumption habits in countries with higher populations.
#Provable Solution:Tailor content strategies based on the dominant channel t
# adapting to local preferences and trends to maximize audience engagement a
#Analyzing successful channels within each type can provide insights for con
# top 5 Channel type distribution in countries by population
import matplotlib.pyplot as plt
import seaborn as sns

top_countries = df.groupby('Country')['Population'].max().sort_values(ascend
print(f'Countries with Highest Population: {top_countries}')

plt.figure(figsize=(20, 8))  # Adjust figure size
channel_type_counts_top5 = df[df['Country'].isin(top_countries)]['Channel Ty

# Increase label size and display percentage values
plt.pie(channel_type_counts_top5, labels=channel_type_counts_top5.index, aut
plt.title('Top 5 Channel Type Distribution in Countries by Population', font
plt.show()
```

```
Countries with Highest Population: ['China', 'India', 'United States', 'Ind
onesia', 'Pakistan', 'Brazil', 'Bangladesh', 'Russia', 'Japan', 'Mexico',
'Philippines', 'Egypt', 'Vietnam', 'Turkey', 'Germany', 'Thailand', 'Franc
e', 'United Kingdom', 'Italy', 'South Korea', 'Colombia', 'Spain', 'Argenti
na', 'Ukraine', 'Iraq', 'Afghanistan', 'Canada', 'Morocco', 'Saudi Arabia',
'Peru', 'Malaysia', 'Venezuela', 'Australia', 'Chile', 'Ecuador', 'Netherla
nds', 'Cuba', 'Sweden', 'Jordan', 'United Arab Emirates', 'Switzerland', 'E
l Salvador', 'Singapore', 'Finland', 'Kuwait', 'Latvia', 'Barbados', 'Samo
a', 'Andorra', 'Unknown']
```



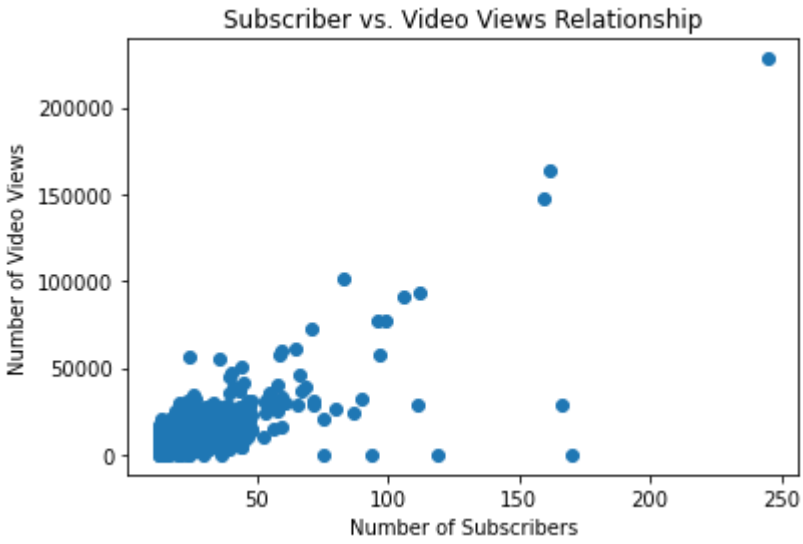
In [26]:
```
#Insight 10: The top earners are not concentrated in a specific region,
#indicating a global distribution of successful YouTube content creators.
#Root Cause: Variances in the availability of monetization opportunities, su
#Provable Solution: Encourage content creators to tailor their content to lo
#Provide resources and guidance to creators on understanding and adapting co

top_countries = df.groupby('Country')['Highest Monthly Earnings'].max().sort
plt.figure(figsize=(12, 8))
sns.barplot(x=top_countries.values, y=top_countries.index, palette='viridis'
plt.title('Top 10 Countries by Highest Monthly Earnings')
plt.show()
```



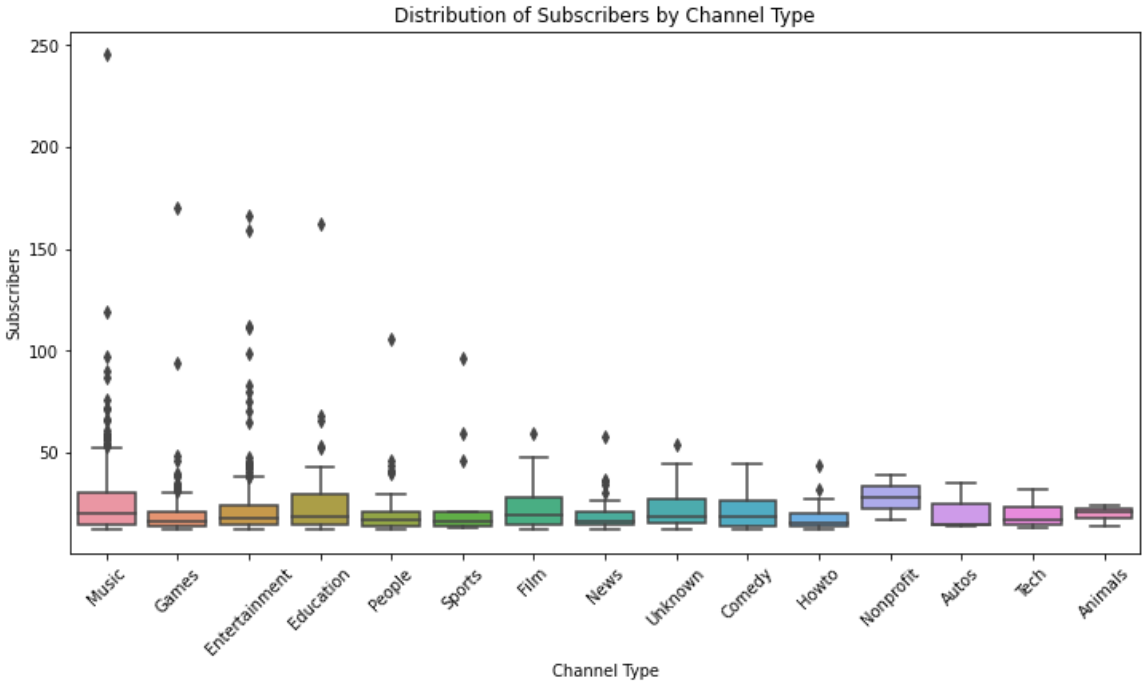
In [27]:
```
# In[11]:
# Filter rows where Country is India
india_data = df[df['Country'] == 'India']

# Find the category with the maximum occurrence
max_category = india_data['Category'].value_counts().idxmax()

print(f"The maximum category for India is: {max_category}")
```

```
The maximum category for India is: Entertainment
```

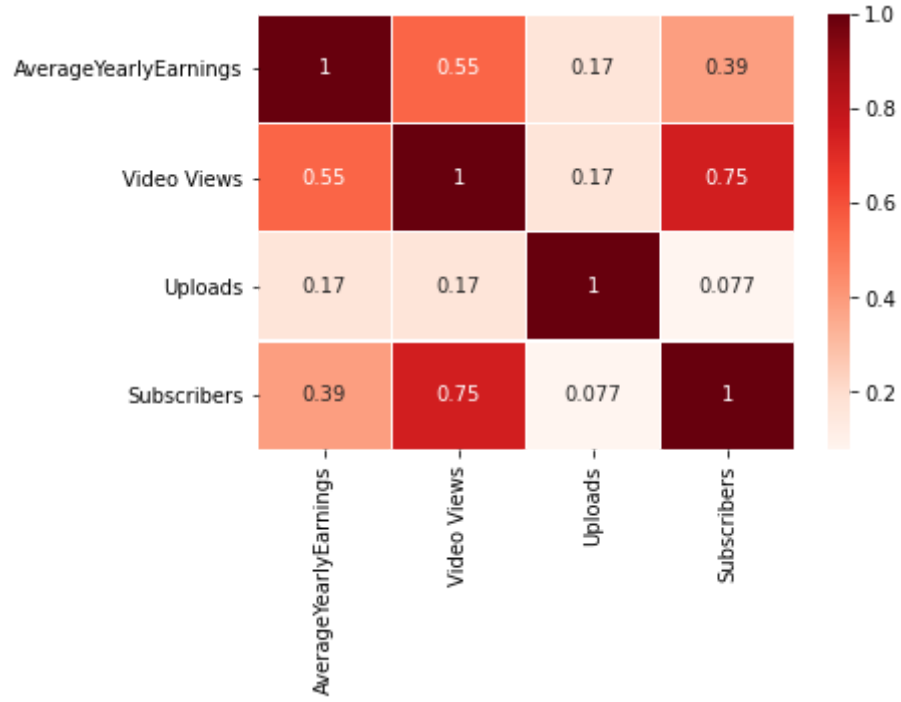
In [28]:
```
#Insights 12: As the number of subscribers increases, there is a correspondi
# Root cause: Channels with broad appeal content attract a diverse audience,
# Versatile content, trending topics, Quality content, community engagement.
# Provable solution: Active Community Interaction through Regular uploads, c
import matplotlib.pyplot as plt
import numpy as np
plt.scatter(df['Subscribers'], df['Video Views'], alpha=1)
plt.title('Subscriber vs. Video Views Relationship')
plt.xlabel('Number of Subscribers')
plt.ylabel('Number of Video Views')
plt.show()
```



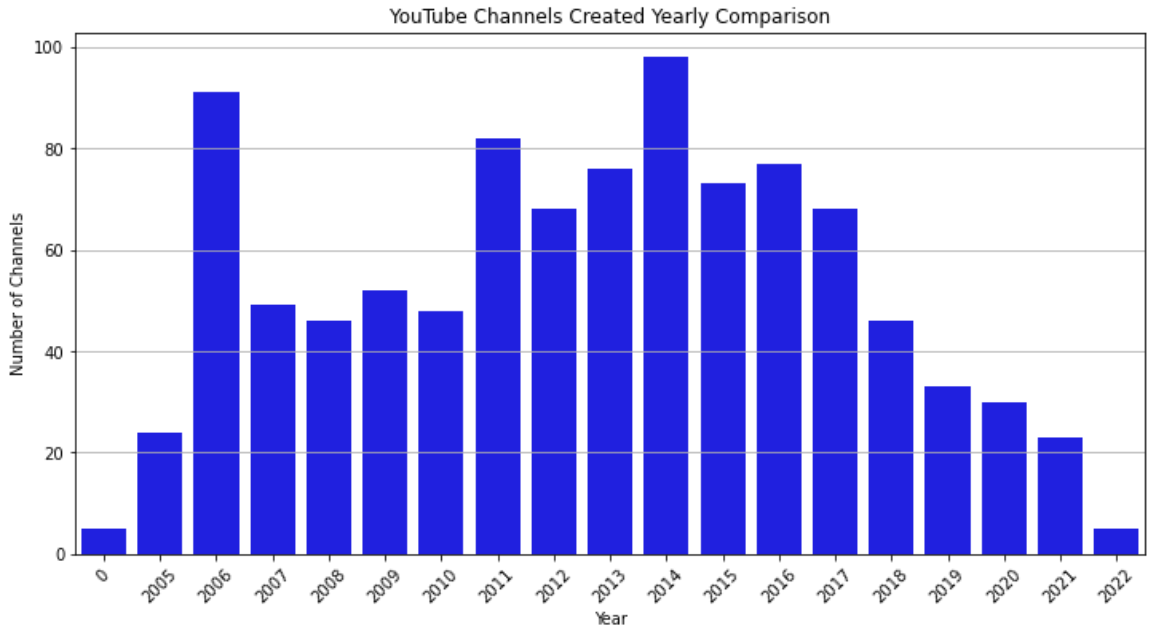
In [29]:
```
# Insights 13: Different channel types exhibit varied distributions of subsc
# root cause: The type of content offered by different channel types influen
# content format, and audience preferences.
# Provable solution: Analyze successful channels within each type, identify
# position content to maximize subscriber growth.
plt.figure(figsize=(12, 6))
sns.boxplot(x='Channel Type', y='Subscribers', data=df)
plt.title('Distribution of Subscribers by Channel Type')
plt.xticks(rotation=45)
plt.show()
```



In [30]:
```
# Insights 14: Video views are highly correlated with yearly earnings, and v
# root cause: Channels with a larger subscriber base often have a more engag
#             Channels with a higher frequency of uploads may attract viewers
# Provable solution: Optimize monetization strategies based on viewership pa
# Action: Explore revenue streams beyond ads, such as sponsorships, affiliat
# Align monetization efforts with the factors contributing to higher video v

df_corr = df[['AverageYearlyEarnings', 'Video Views', 'Uploads', 'Subscriber
corr = df_corr.corr()
sns.heatmap(corr, annot=True, linewidth=0.5, cmap='Reds')
plt.show()
```

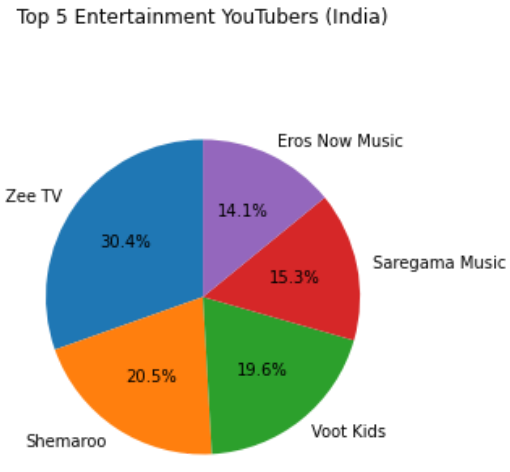


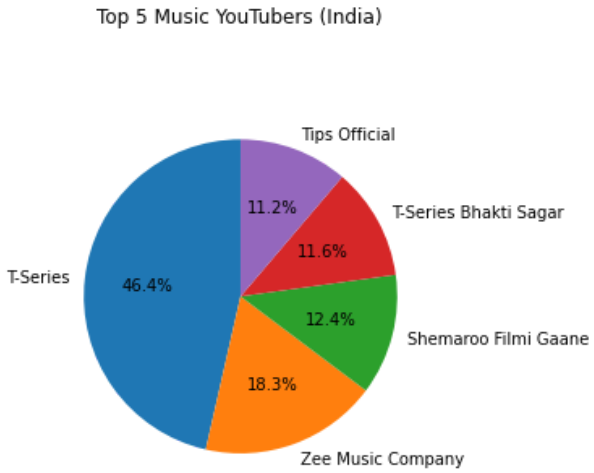
In [31]:
```
# Insights 15:Understanding trends in channel creation over the years is cru
# growth and content creator engagement.Peaks may indicate periods of increa
# Root cause: Platform updates, algorithm changes, and shifts in user behavi
# of new creators during specific periods.
# Provable solution: Regularly update content strategies based on platform f
# and emerging trends to maximize channel growth.
# Provide continuous educational resources, community support, and opportuni
# to maintain a healthy and stable creator ecosystem.
# How many channels are created in particular years:
df['Created Year'] = df['Created Year'].astype(int)
df_filtered = df[df['Created Year'] != 1970]

year_counts = df_filtered['Created Year'].value_counts().sort_index()

plt.figure(figsize=(12, 6))
sns.barplot(x=year_counts.index, y=year_counts.values, color='blue')

plt.title('YouTube Channels Created Yearly Comparison')
plt.xlabel('Year')
plt.ylabel('Number of Channels')
plt.xticks(rotation=45)
plt.grid(axis='y')
plt.show()
```

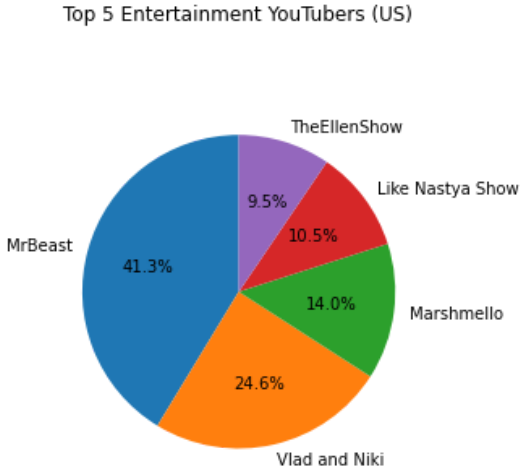


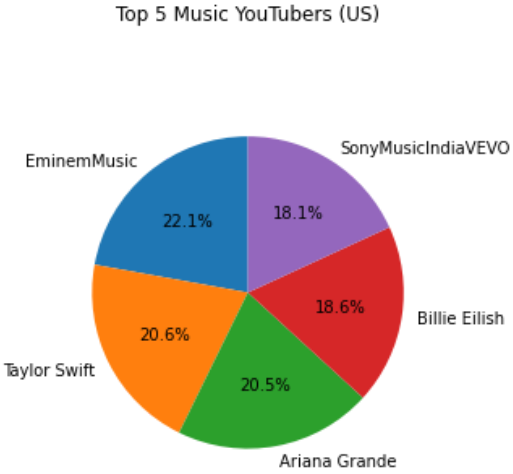
In [43]:
```
#Insights 16: Comparison between India and US with Entertainment and Music c

# Dominance of Music in India: The top 5 YouTubers in the Music category hav
# Music content resonates strongly with the Indian audience.

# Root Cause:Cultural Preference for Music India has a rich cultural heritag
# Factors: Cultural influence, diverse music genres, and widespread music ap

# Provable Solution: Strategy: Content creators targeting the Indian audienc
# Action: Explore collaborations with musicians, cover popular songs, and pr


# Both Entertainment and Music categories have diverse genres represented in
# US audiences have varied interests across different content genres.

# Root cause: Hollywood influence, celebrity culture, and the prominence of

# Provable Solution: Collaborate with influencers, leverage popular trends,
# maintain and grow the subscriber base.Content creators targeting the US au
# content.



# Filter rows for Entertainment and Music categories in India and US
filtered_df = df[df['Category'].isin(['Entertainment', 'Music']) & df['Abbre

# Separate the data for each country and category
india_entertainment = filtered_df[(filtered_df['Abbreviation'] == 'IN') & (f
india_music = filtered_df[(filtered_df['Abbreviation'] == 'IN') & (filtered_
us_entertainment = filtered_df[(filtered_df['Abbreviation'] == 'US') & (filt
us_music = filtered_df[(filtered_df['Abbreviation'] == 'US') & (filtered_df[

# Define a function to get the top 5 YouTubers
def get_top_5(dataframe):
    return dataframe.nlargest(5, 'Subscribers')[['Youtuber', 'Subscribers',

# Get the top 5 YouTubers for each category and country
top_5_india_entertainment = get_top_5(india_entertainment)
top_5_india_music = get_top_5(india_music)
top_5_us_entertainment = get_top_5(us_entertainment)
top_5_us_music = get_top_5(us_music)

# create figure:
fig, axes = plt.subplots(nrows=2, ncols=2, figsize=(10, 10))

# pie charts each category and country
def plot_pie(ax, labels, sizes, title):
    ax.pie(sizes, labels=labels, autopct='%1.1f%%', startangle=90)
    ax.axis('equal')
    ax.set_title(title)

plot_pie(axes[0, 0], top_5_india_entertainment['Youtuber'], top_5_india_ente
plot_pie(axes[0, 1], top_5_india_music['Youtuber'], top_5_india_music['Subsc
plot_pie(axes[1, 0], top_5_us_entertainment['Youtuber'], top_5_us_entertainm
plot_pie(axes[1, 1], top_5_us_music['Youtuber'], top_5_us_music['Subscribers


plt.tight_layout()
```

```
plt.show()
```


Top 5 Entertainment YouTubers (India)
Eros Now Music
Zee TV
14.1%
30.4%
Saregama Music
15.3%
20.5%
19.6%
Shemaroo
Voot Kids


Top 5 Music YouTubers (India)
Tips Official
11.2%
T-Series Bhakti Sagar
11.6%
T-Series
46.4%
12.4%
Shemaroo Filmi Gaane
18.3%
Zee Music Company


Top 5 Entertainment YouTubers (US)
TheEllenShow
9.5%
Like Nastya Show
10.5%
MrBeast
41.3%
14.0%
Marshmello
24.6%
Vlad and Niki


Top 5 Music YouTubers (US)
EminemMusic
22.1%
SonyMusicIndiaVEVO
18.1%
18.6%
Billie Eilish
20.6%
Taylor Swift
20.5%
Ariana Grande

In [ ]: