

ADITYA MENON

MEDIA AND TECHNOLOGY PROJECT: YOUTUBE STATISITCS

Media and technology

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
df = pd.read_csv('Global YouTube Statistics.csv')

df.columns

Index(['rank', 'Youtuber', 'subscribers', 'video views', 'category',
      'Title',
      'uploads', 'Country of origin', 'Country', 'Abbreviation',
      'channel_type', 'video_views_rank', 'country_rank',
      'channel_type_rank',
      'video_views_for_the_last_30_days', 'lowest_monthly_earnings',
      'highest_monthly_earnings', 'lowest_yearly_earnings',
      'highest_yearly_earnings', 'subscribers_for_last_30_days',
      'created_year', 'created_month', 'created_date',
      'Gross tertiary education enrollment (%)', 'Population',
      'Unemployment rate', 'Urban_population', 'Latitude',
      'Longitude'],
      dtype='object')
```

1. Top 10 YouTube channels based on the number of subscribers

```
top_10_subscribers = df.nlargest(10, 'subscribers')[['Youtuber',  
'subscribers']]  
print("The top 10 YouTube channels based on subscribers are:")  
print(top_10_subscribers)
```

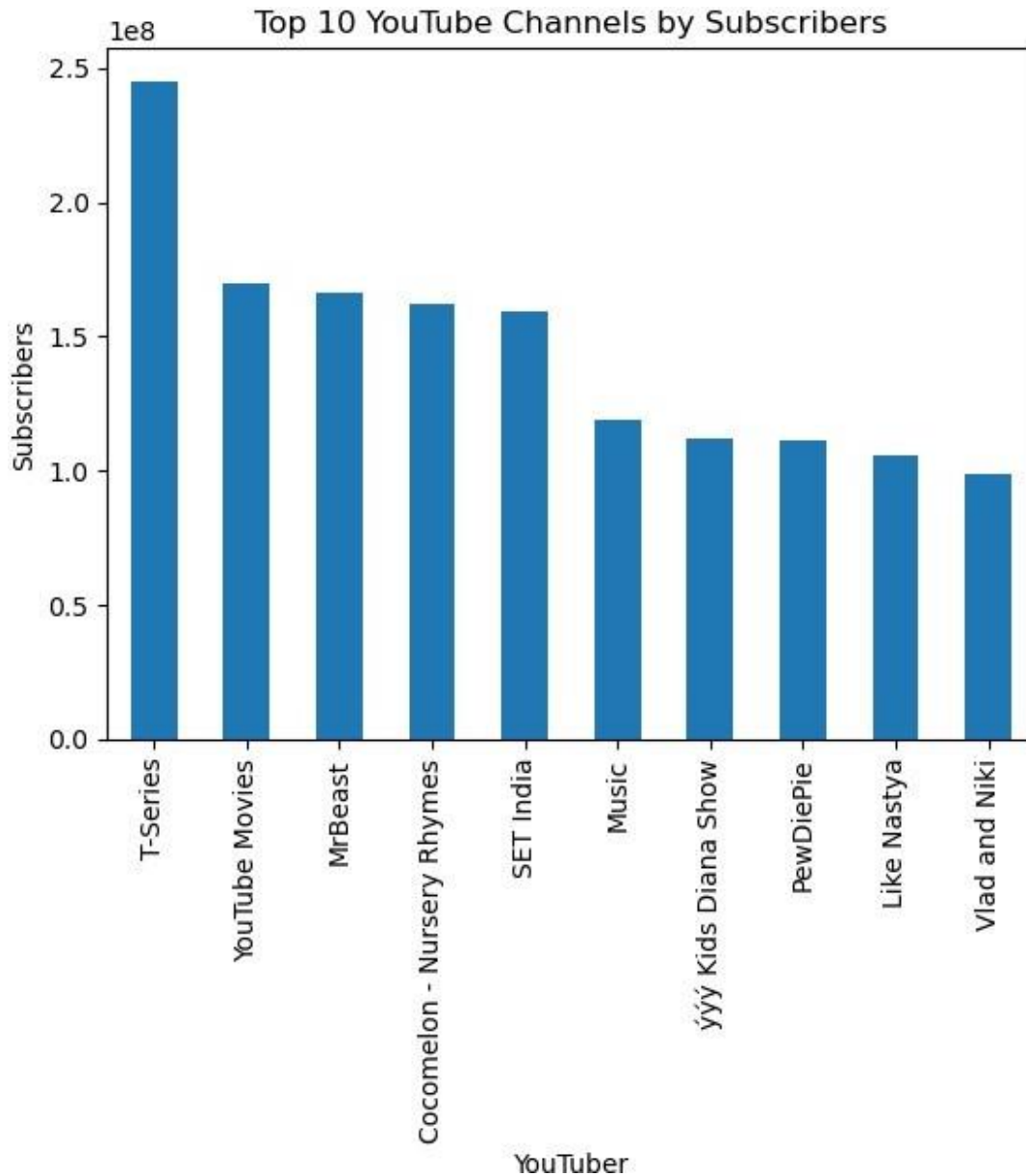
The top 10 YouTube channels based on subscribers are:

	Youtuber	subscribers
0	T-Series	245000000.0
1	YouTube Movies	170000000.0
2	MrBeast	166000000.0
3	Cocomelon - Nursery Rhymes	162000000.0
4	SET India	159000000.0
5	Music	119000000.0

```
6      ЫЫЫ Kids Diana Show  112000000.0
7              PewDiePie    111000000.0
8              Like Nastya  106000000.0
9              Vlad and Niki  98900000.0
```

```
plt.figure(figsize=(10, 6))
top_10_subscribers.plot(kind='bar', x='Youtuber', y='subscribers',
legend=False)
plt.title('Top 10 YouTube Channels by Subscribers')
plt.xlabel('YouTuber')
plt.ylabel('Subscribers')
plt.show()
```

```
<Figure size 1000x600 with 0 Axes>
```



2. Category with the highest average number of subscribers

```
avg_subscribers_per_category = df.groupby('category')  
['subscribers'].mean().idxmax()  
print("\nCategory with highest average number of subscribers:",  
avg_subscribers_per_category)
```

Category with highest average number of subscribers: Shows

3. Average number of videos uploaded by YouTube channels in each category

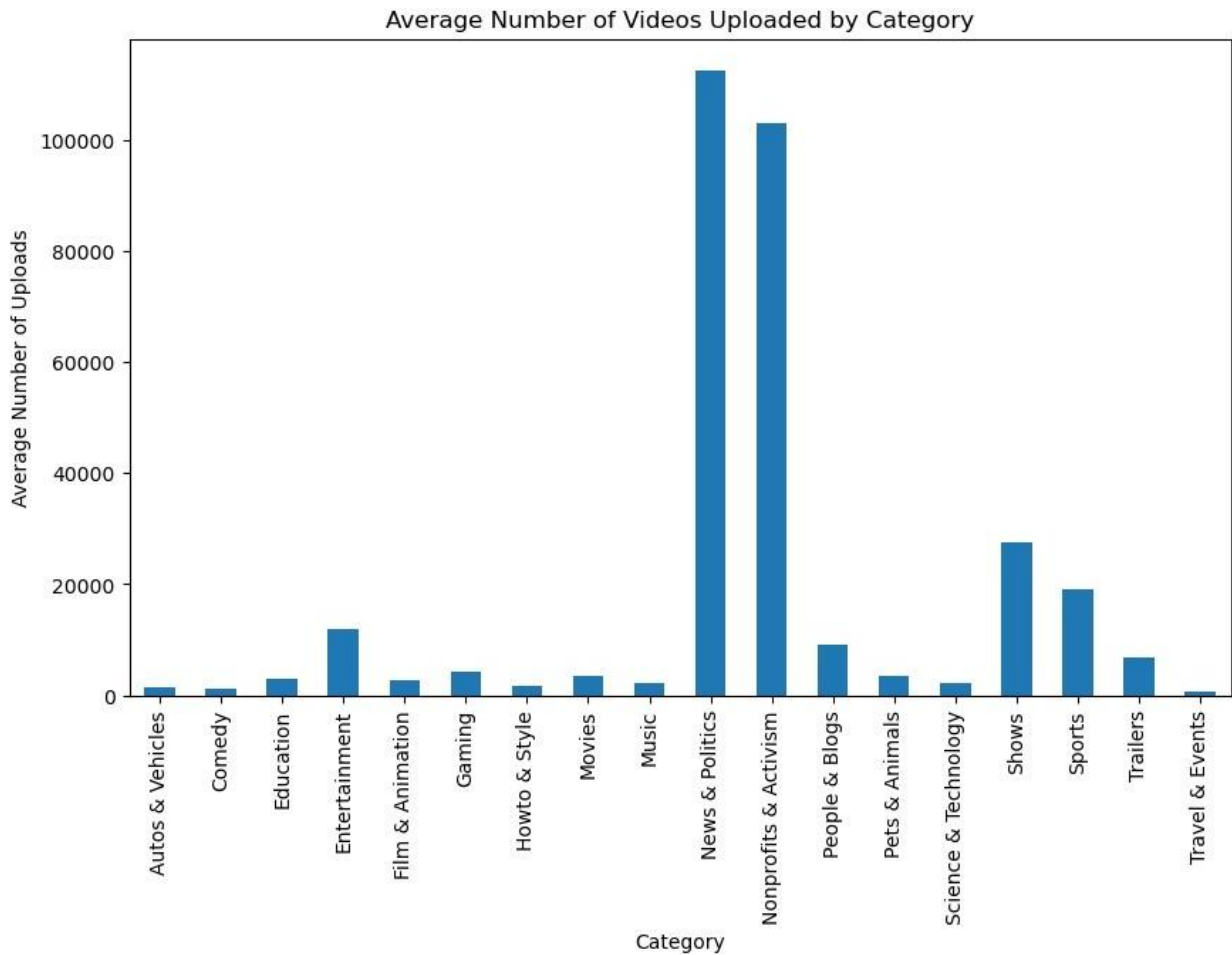
```
avg_videos_per_category = df.groupby('category')['uploads'].mean()  
print("\nAverage videos uploaded per category:")  
print(avg_videos_per_category)
```

Average videos uploaded per category:

category	
Autos & Vehicles	1550.666667
Comedy	1202.557143
Education	3087.086957
Entertainment	12052.445378
Film & Animation	2861.844444
Gaming	4285.273684
Howto & Style	1695.500000
Movies	3553.000000
Music	2325.945813
News & Politics	112484.384615
Nonprofits & Activism	102912.000000
People & Blogs	9256.793893
Pets & Animals	3562.800000
Science & Technology	2114.058824
Shows	27443.692308
Sports	19129.833333
Trailers	6839.000000
Travel & Events	766.000000

Name: uploads, dtype: float64

```
plt.figure(figsize=(10, 6))  
avg_videos_per_category.plot(kind='bar')  
plt.title('Average Number of Videos Uploaded by Category')  
plt.xlabel('Category')  
plt.ylabel('Average Number of Uploads')  
plt.show()
```



4. Top 5 countries with the highest number of YouTube channels

```
top_5_countries = df['Country'].value_counts().nlargest(5)
print("\nTop 5 countries with highest number of YouTube channels:")
print(top_5_countries)
```

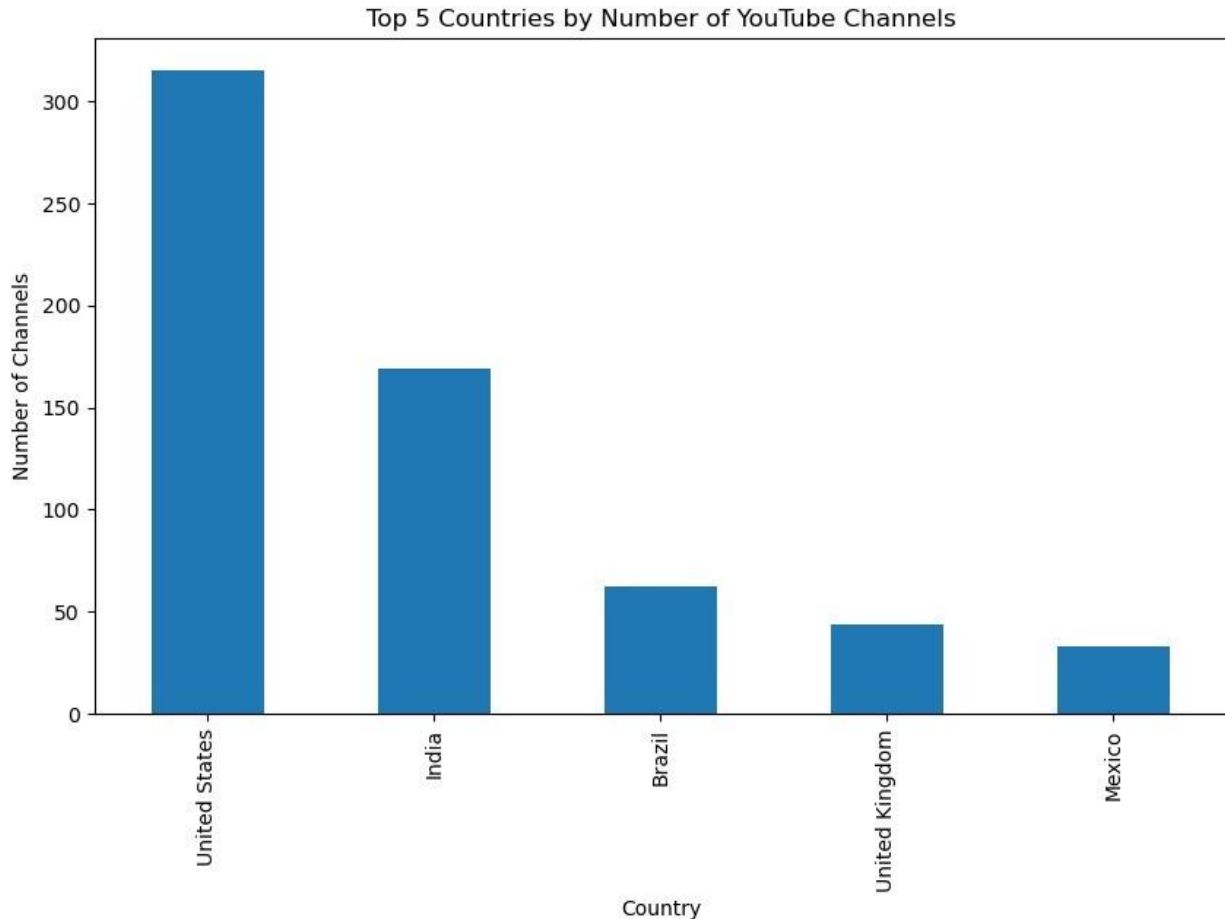
Top 5 countries with highest number of YouTube channels:

United States	315
India	169
Brazil	62
United Kingdom	44
Mexico	33

Name: Country, dtype: int64

```
plt.figure(figsize=(10, 6))
top_5_countries.plot(kind='bar')
```

```
plt.title('Top 5 Countries by Number of YouTube Channels')
plt.xlabel('Country')
plt.ylabel('Number of Channels')
plt.show()
```



5. Distribution of channel types across different categories

```
channel_type_distribution = df.groupby(['category',
'channel_type']).size().unstack(fill_value=0)
print("\nDistribution of channel types across categories:")
print(channel_type_distribution)
```

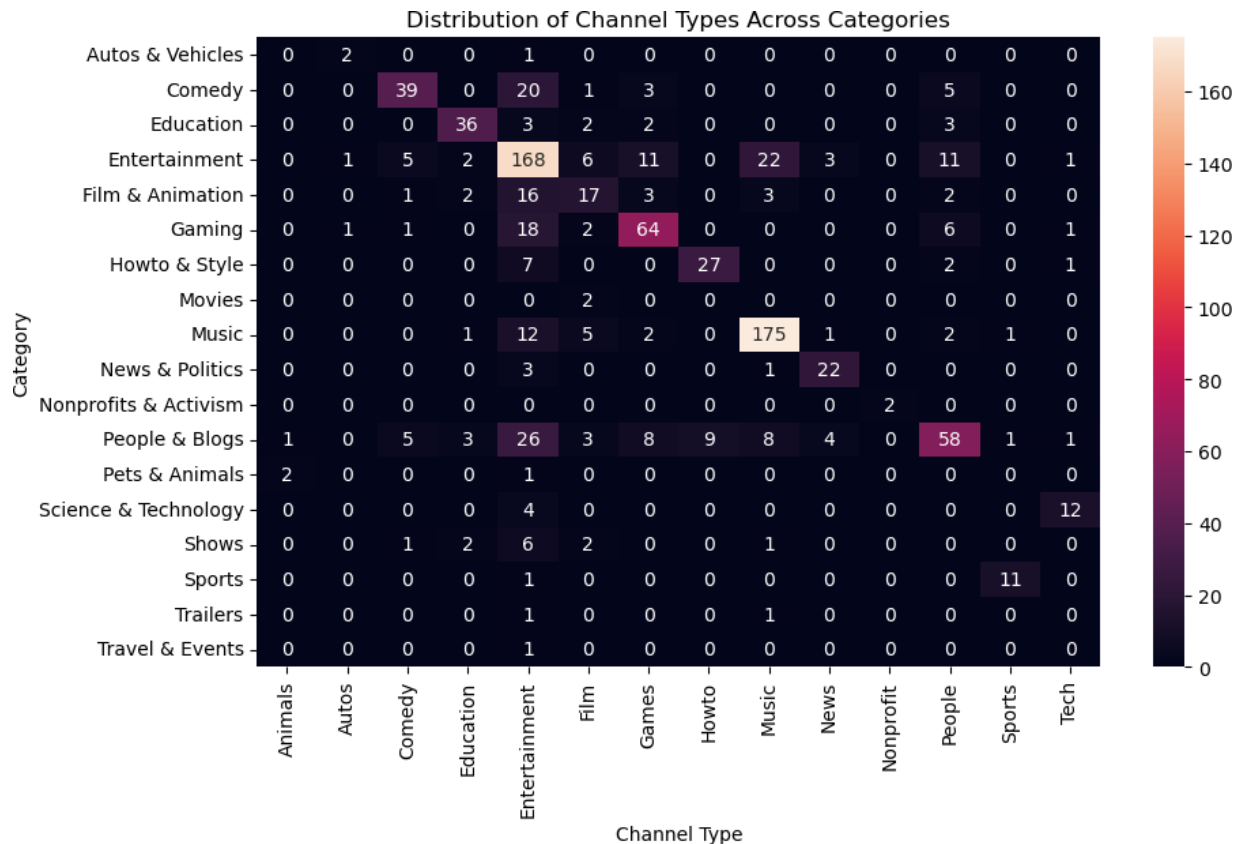
```
Distribution of channel types across categories:
channel_type      Animals Autos Comedy Education
Entertainment Film \
category
```

Autos & Vehicles	0	2	0	0
1 0				
Comedy	0	0	39	0
20 1				
Education	0	0	0	36
3 2				
Entertainment	0	1	5	2
168 6				
Film & Animation	0	0	1	2
16 17				
Gaming	0	1	1	0
18 2				
Howto & Style	0	0	0	0
7 0				
Movies	0	0	0	0
0 2				
Music	0	0	0	1
12 5				
News & Politics	0	0	0	0
3 0				
Nonprofits & Activism	0	0	0	0
0 0				
People & Blogs	1	0	5	3
26 3				
Pets & Animals	2	0	0	0
1 0				
Science & Technology	0	0	0	0
4 0				
Shows	0	0	1	2
6 2				
Sports	0	0	0	0
1 0				
Trailers	0	0	0	0
1 0				
Travel & Events	0	0	0	0
1 0				

channel type	Games	Howto	Music	News	Nonprofit	People
Sports \ category						
Autos & Vehicles	0	0	0	0	0	0
0						
Comedy	3	0	0	0	0	5
0						
Education	2	0	0	0	0	3
0						
Entertainment	11	0	22	3	0	11
0						

Film & Animation	3	0	3	0	0	2
0						
Gaming	64	0	0	0	0	6
0						
Howto & Style	0	27	0	0	0	2
0						
Movies	0	0	0	0	0	0
0						
Music	2	0	175	1	0	2
1						
News & Politics	0	0	1	22	0	0
0						
Nonprofits & Activism	0	0	0	0	2	0
0						
People & Blogs	8	9	8	4	0	58
1						
Pets & Animals	0	0	0	0	0	0
0						
Science & Technology	0	0	0	0	0	0
0						
Shows	0	0	1	0	0	0
0						
Sports	0	0	0	0	0	0
11						
Trailers	0	0	1	0	0	0
0						
Travel & Events	0	0	0	0	0	0
0						
channel_type						
Tech						
category						
Autos & Vehicles	0					
Comedy	0					
Education	0					
Entertainment	1					
Film & Animation	0					
Gaming	1					
Howto & Style	1					
Movies	0					
Music	0					
News & Politics	0					
Nonprofits & Activism	0					
People & Blogs	1					
Pets & Animals	0					
Science & Technology	12					
Shows	0					
Sports	0					
Trailers	0					
Travel & Events	0					

```
plt.figure(figsize=(10, 6))
sns.heatmap(channel_type_distribution, annot=True, fmt='d')
plt.title('Distribution of Channel Types Across Categories')
plt.xlabel('Channel Type')
plt.ylabel('Category')
plt.show()
```



6. Correlation between subscribers and total video views

```
correlation_subscribers_views = df['subscribers'].corr(df['video
views'])
print("\nCorrelation between subscribers and total video views:",
correlation_subscribers_views)
```

```
Correlation between subscribers and total video views:
0.7481786016237688
```

7. Monthly earnings variation throughout different categories

```
monthly_earnings_variation = df.groupby('category')  
[['lowest_monthly_earnings', 'highest_monthly_earnings']].mean()  
print("\nMonthly earnings variation throughout categories:")  
print(monthly_earnings_variation)
```

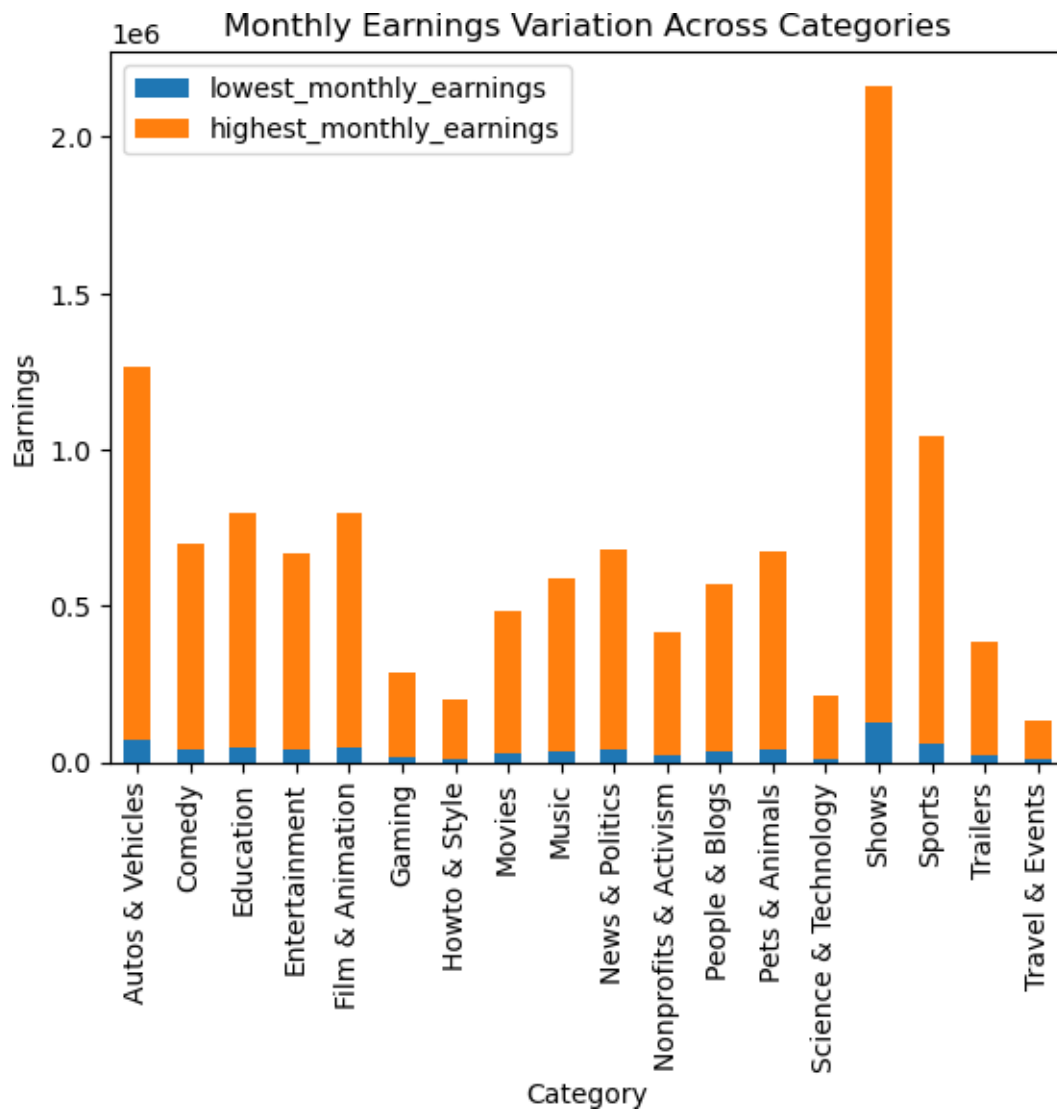
Monthly earnings variation throughout categories:

	lowest_monthly_earnings
highest_monthly_earnings	
category	
Autos & Vehicles	74966.666667
1.190900e+06	
Comedy	41332.285714
6.597635e+05	
Education	46863.239348
7.518043e+05	
Entertainment	39393.724370
6.293549e+05	
Film & Animation	46802.533556
7.489841e+05	
Gaming	16965.790316
2.720201e+05	
Howto & Style	12022.350500
1.920441e+05	
Movies	28400.000000
4.547000e+05	
Music	34671.696798
5.535735e+05	
News & Politics	40192.625000
6.426320e+05	
Nonprofits & Activism	24400.000000
3.904000e+05	
People & Blogs	33485.993969
5.357493e+05	
Pets & Animals	39980.800000
6.355516e+05	
Science & Technology	12635.411765
2.020432e+05	
Shows	126961.538462
2.037662e+06	
Sports	60783.333333
9.813583e+05	
Trailers	22600.000000
3.619000e+05	

```
Travel & Events                7800.000000
1.240000e+05
```

```
plt.figure(figsize=(10, 6))
monthly_earnings_variation.plot(kind='bar', stacked=True)
plt.title('Monthly Earnings Variation Across Categories')
plt.xlabel('Category')
plt.ylabel('Earnings')
plt.show()
```

<Figure size 1000x600 with 0 Axes>



8. Relationship between gross tertiary education enrollment and the number of YouTube channels in a country

```
channels_per_country = df['country'].value_counts()
edu_enrollment_vs_channels = df.groupby('country')
['gross_tertiary_education_enrollment_(%)'].mean().to_frame()
edu_enrollment_vs_channels['channels'] = channels_per_country
edu_enrollment_vs_channels.dropna(inplace=True)
edu_enrollment_vs_channels_corr =
```

```
edu_enrollment_vs_channels.corr().iloc[0, 1]
print("edu_enrollment_vs_channels_corr:", edu_enrollment_vs_channels_co
rr)
```

```
edu_enrollment_vs_channels_corr: 0.11114034153558282
```

9. Unemployment rate variation among the top 10 countries with the highest number of YouTube channels

```
top_10_countries = channels_per_country.nlargest(10).index
unemployment_top_10 =
df[df['country'].isin(top_10_countries)].groupby('country')
['unemployment_rate'].mean()
print("unemployment_top_10: ",unemployment_top_10)
```

```
unemployment_top_10: country
Brazil                12.08
India                 5.36
Indonesia             4.69
Mexico                3.42
Russia                4.59
South Korea           4.15
Spain                13.96
Thailand              0.75
United Kingdom        3.85
United States        14.70
Name: unemployment_rate, dtype: float64
```

10. Average urban population percentage in countries with YouTube channels

```
avg_urban_population_percentage = df.groupby('country')
['urban_population'].mean()
print("avg_urban_population_percentage:",avg_urban_population_percenta
ge)
```

```
avg_urban_population_percentage: country
Afghanistan           9797273.0
Andorra               NaN
Argentina             41339571.0
Australia             21844756.0
Bangladesh            60987417.0
Barbados              89431.0
Brazil               183241641.0
Canada               30628482.0
Chile                 16610135.0
```

China	842933962.0
Colombia	40827302.0
Cuba	8739135.0
Ecuador	11116711.0
Egypt	42895824.0
El Salvador	4694702.0
Finland	4716888.0
France	54123364.0
Germany	64324835.0
India	471031528.0
Indonesia	151509724.0
Iraq	27783368.0
Italy	42651966.0
Japan	115782416.0
Jordan	9213048.0
Kuwait	4207083.0
Latvia	1304943.0
Malaysia	24475766.0
Mexico	102626859.0
Morocco	22975026.0
Netherlands	15924729.0
Pakistan	79927762.0
Peru	25390339.0
Philippines	50975903.0
Russia	107683889.0
Samoa	35588.0
Saudi Arabia	28807838.0
Singapore	5703569.0
South Korea	42106719.0
Spain	37927409.0
Sweden	9021165.0
Switzerland	6332428.0
Thailand	35294600.0
Turkey	63097818.0
Ukraine	30835699.0
United Arab Emirates	8479744.0
United Kingdom	55908316.0
United States	270663028.0
Venezuela	25162368.0
Vietnam	35332140.0
india	471031528.0

Name: urban_population, dtype: float64

11. Correlation between the number of subscribers and the population of a country

```
subscribers_vs_population = df.groupby('country').agg({'subscribers':  
'sum', 'population': 'mean'}).dropna()  
subscribers_vs_population_corr =  
subscribers_vs_population.corr().iloc[0, 1]  
print("subscribers_vs_population_corr:", subscribers_vs_population_corr  
)  
  
subscribers_vs_population_corr: 0.3412198958571251
```


12. Comparison of top 10 countries with the highest number of YouTube channels in terms of their total population

```
top_10_countries_population =  
df[df['country'].isin(top_10_countries)].groupby('country')  
['population'].mean()  
print("top_10_countries_population:",top_10_countries_population)
```

```
top 10 countries population: country  
Brazil          2.125594e+08  
India            1.366418e+09  
Indonesia        2.702039e+08  
Mexico           1.260140e+08  
Russia           1.443735e+08  
South Korea      5.170910e+07  
Spain            4.707678e+07  
Thailand          6.962558e+07  
United Kingdom   6.683440e+07  
United States     3.282395e+08  
Name: population, dtype: float64
```

13. Correlation between the number of subscribers gained in the last 30 days and the unemployment rate in a country

```
subscribers_30days_vs_unemployment =  
df.groupby('country').agg({'subscribers_for_last_30_days': 'sum',  
                           'unemployment_rate': 'mean'}).dropna()  
subscribers_30days_vs_unemployment_corr =  
subscribers_30days_vs_unemployment.corr().iloc[0, 1]  
print("subscribers_30days_vs_unemployment_corr:",subscribers_30days_vs_  
_unemployment_corr)  
  
subscribers_30days_vs_unemployment_corr: 0.21049865229143547
```

14. Distribution of video views for the last 30 days across different channel types

```
video_views_30days_channel_type = df.groupby('channel_type')  
['video_views_for_the_last_30_days'].sum()
```

```
print("video_views_30days_channel_type:",video_views_30days_channel_type)
```

```
video_views_30days_channel_type: channel_type
```

Animals	2.122043e+09
Autos	7.065205e+08
Comedy	9.618215e+09
Education	1.003686e+10
Entertainment	6.386184e+10
Film	5.763538e+09
Games	8.668806e+09
Howto	2.170354e+09
Music	3.773526e+10
News	5.250126e+09
Nonprofit	1.951810e+08
People	1.720830e+10
Sports	2.843109e+09
Tech	9.365210e+08

```
Name: video_views_for_the_last_30_days, dtype: float64
```

