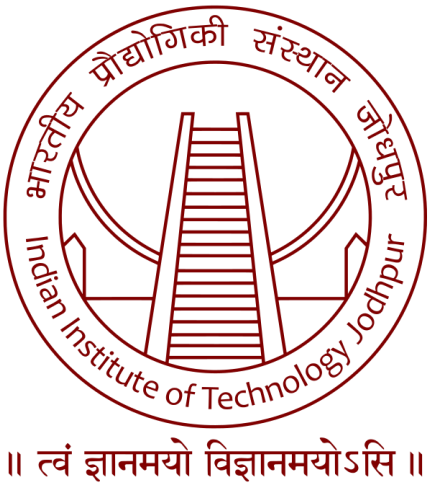
Indian Institute of Technology Jodhpur

Robotics and Mobility System

Finlatics Data Science



Project Report:

Media & Technology (Data Science)

By: Chaitanya Patil

Code Transcript and Solution

The programming for this project was conducted within Google Colab, a cloud-based platform that eliminates the need for software installation on local machines. This facilitated collaboration and streamlined the development process. To enable data analysis, essential libraries were incorporated, including Pandas, Matplotlib, NumPy, and Seaborn. These libraries function as pre-written code modules, offering efficient tools for data manipulation and visualization.

The csv file of "Global Youtube Statistics.csv" was added with encoding 'unicode_escape', the argument specifies the character encoding used to read the data from the csv file. It insurance proper interpretation of any special characters or symbol that may be present within the data.

```
# Data Science (Finlatics)(Youtube & Media)

# Libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np


# Adding the csv to the dataset
data = pd.read_csv('Global YouTube Statistics.csv' ,encoding='unicode_escape')
```

Data Preprocessing: In data preprocessing the , firstly the duplicate data was removed ,followed by verification of column headings and identification of missing value in the columns. After that, total number of missing value was also determined. The missing values were changed to '0'.

```
# Preprocessing

# To remove duplicate rows

data = data.drop_duplicates()



# To check and add '0' to all the missing places

print("Data Headers and missing values ")

print(data.isnull().sum() ,'\n')



print("Total number of missing values in the data")

print(data.isnull().sum().sum() ,'\n')
```

```
data= data.fillna(0)


print("Total number of missing values in the data ,After fillin the '0'")
print(data.isnull().sum().sum() ,'\n')
```

Output:

```
Data Headers and missing values
rank                                    0
Youtuber                                0
subscribers                             3
video views                             0
category                               53
Title                                   0
uploads                                 0
Country of origin                     122
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                          12
created_date                            5
Gross tertiary education enrollment (%)  123
Population                            123
Unemployment rate                     123
Urban_population                      123
Latitude                              123
Longitude                             123
dtype: int64

Total number of missing values in the data
1755

Total number of missing values in the data ,After fillin the '0'
0
```

Analysis and Questions Answering:

1. What are the top 10 YouTube channels based on the number of subscribers?

Output:

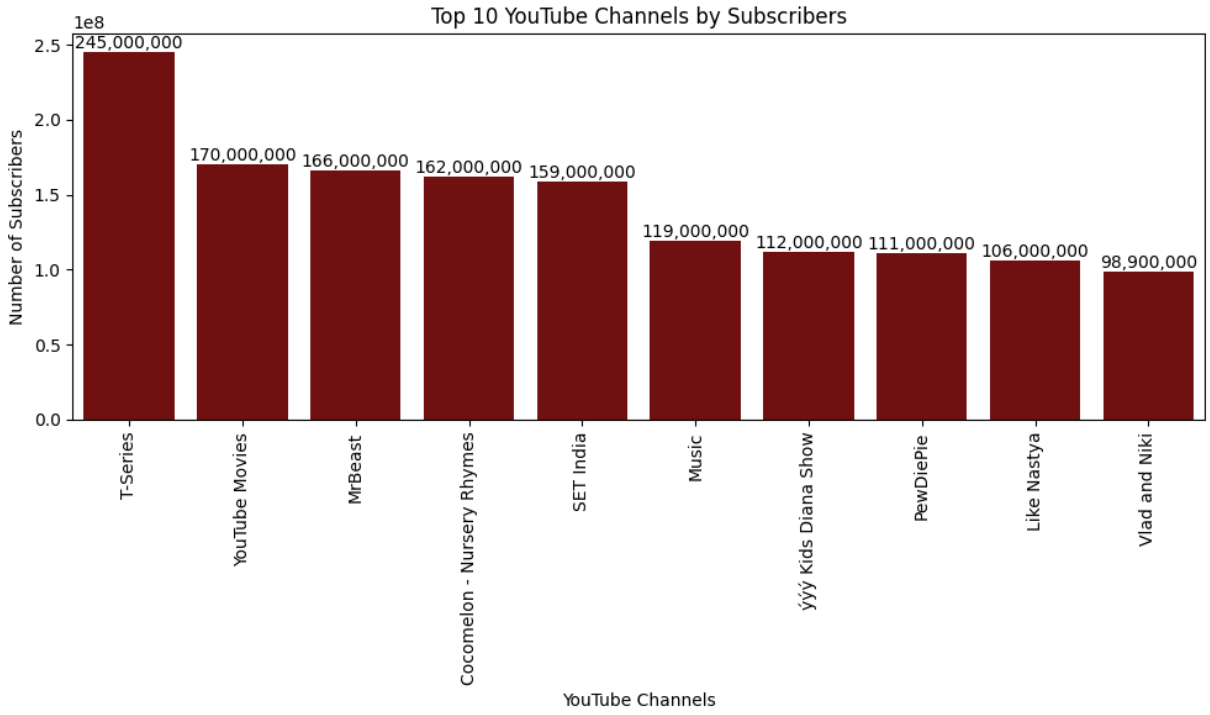


Fig Top 10 Channels by Subscribers

```
Top 10 Youtube Channels by Comparing the Subscribers:
                    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          ýýý Kids Diana Show  112000000.0
7                    PewDiePie  111000000.0
8                  Like Nastya  106000000.0
9                Vlad and Niki   98900000.0
```

Code:

```
top_10_channels = data.sort_values('subscribers' , ascending = False ).head(10)
print("Top 10 Youtube Channels by Comparing the Subscribers:")
print(top_10_channels[['Youtuber','subscribers']])
```

```
print("\n")


# Bar plot for question 1
plt.figure(figsize=(10, 6))
#sns.barplot(x='Youtuber', y='subscribers', data=top_10_channels, color='maroon')
ax = sns.barplot(x='Youtuber', y='subscribers', data=top_10_channels, color='maroon')

# Rotate x-axis labels
plt.xticks(rotation=90)

# Values written on top
bars = ax.patches
for bar in bars:
    value = int(bar.get_height())
    ax.text(bar.get_x() + bar.get_width() / 2, value, f'{value:,.0f}', ha='center', va='bottom')



# Add labels and title
plt.xlabel('YouTube Channels')
plt.ylabel('Number of Subscribers')
plt.title('Top 10 YouTube Channels by Subscribers')

# Show the plot
plt.tight_layout()
plt.show()
```

2. Which category has the highest average number of subscribers?

The category having highest average number of subscriber is : Shows

Code:

```
avg_subs_by_category=data.groupby('category')['subscribers'].mean().sort_values(ascending
=False)

highest_avg_subs_category = avg_subs_by_category.index[0]

print(f"Category with highest average subscribers: {highest_avg_subs_category}")

print("\n")


print("Different types of categories")

categories_list = data.groupby(['category']).size().reset_index(name='count')

print(categories_list,"\n")


# Bar plot for question 2

plt.figure(figsize=(12, 8))

 #sns.barplot(x='category', y='count', data=categories_list,palette = "Set2" )

ax1 = sns.barplot(x='category', y='count', data=categories_list,palette = "Set2")


# Rotate x-axis labels

plt.xticks(rotation=90)


bars = ax1.patches

for bar in bars:

    value = int(bar.get_height())

    ax1.text(bar.get_x() + bar.get_width() / 2, value, f'{value:,.0f}', ha='center', va='bottom')
```

```
# Add labels and title
plt.xlabel('YouTube Categories')
plt.ylabel('Number of channels in the categories')
plt.title('YouTube Channels Categories')

# Show the plot
plt.tight_layout()
plt.show()
```

Output:

```
Category with highest average subscribers: Shows


Different types of categories
                  category  count
0                        0     53
1        Autos & Vehicles      2
2                  Comedy     69
3               Education     45
4           Entertainment    238
5        Film & Animation     45
6                  Gaming     94
7           Howto & Style     39
8                  Movies      2
9                   Music    201
10        News & Politics     26
11  Nonprofits & Activism      2
12         People & Blogs    131
13         Pets & Animals      4
14   Science & Technology     17
15                  Shows     13
16                 Sports     11
17               Trailers      2
18        Travel & Events      1
```

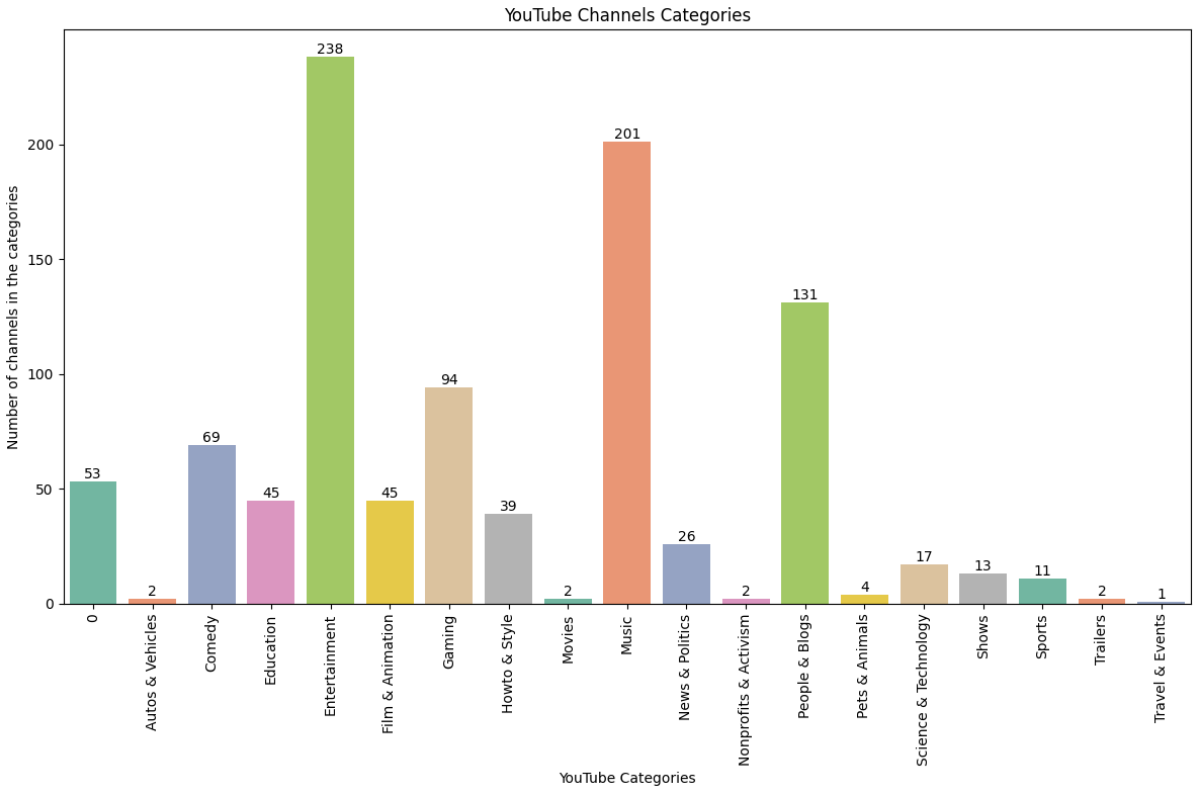


Fig Number of YouTube Channels on the basis ‘category’

Code:

```
print("Different types of channel types")
channel_list = data.groupby(['channel_type']).size().reset_index(name='count')
print(channel_list,"\n")


# Bar plot for question 2
plt.figure(figsize=(12, 8))
ax2 = sns.barplot(x='channel_type', y='count', data=channel_list,palette = "Set2" )


# Rotate x-axis labels
plt.xticks(rotation=90)
```

```
bars = ax2.patches
for bar in bars:
    value = int(bar.get_height())
    ax2.text(bar.get_x() + bar.get_width() / 2, value, f'{value:,.0f}', ha='center', va='bottom')



# Add labels and title
plt.xlabel('YouTube Channels')
plt.ylabel('Number of channels in Youtube')
plt.title('YouTube Channels')


# Show the plot
plt.tight_layout()
plt.show()
```

Output:

```
Different types of channel types
     channel_type  count
0               0     30
1         Animals      3
2           Autos      3
3          Comedy     51
4       Education     49
5   Entertainment    304
6            Film     42
7           Games     98
8           Howto     36
9           Music    216
10           News     30
11      Nonprofit      2
12         People    101
13         Sports     13
14           Tech     17
```

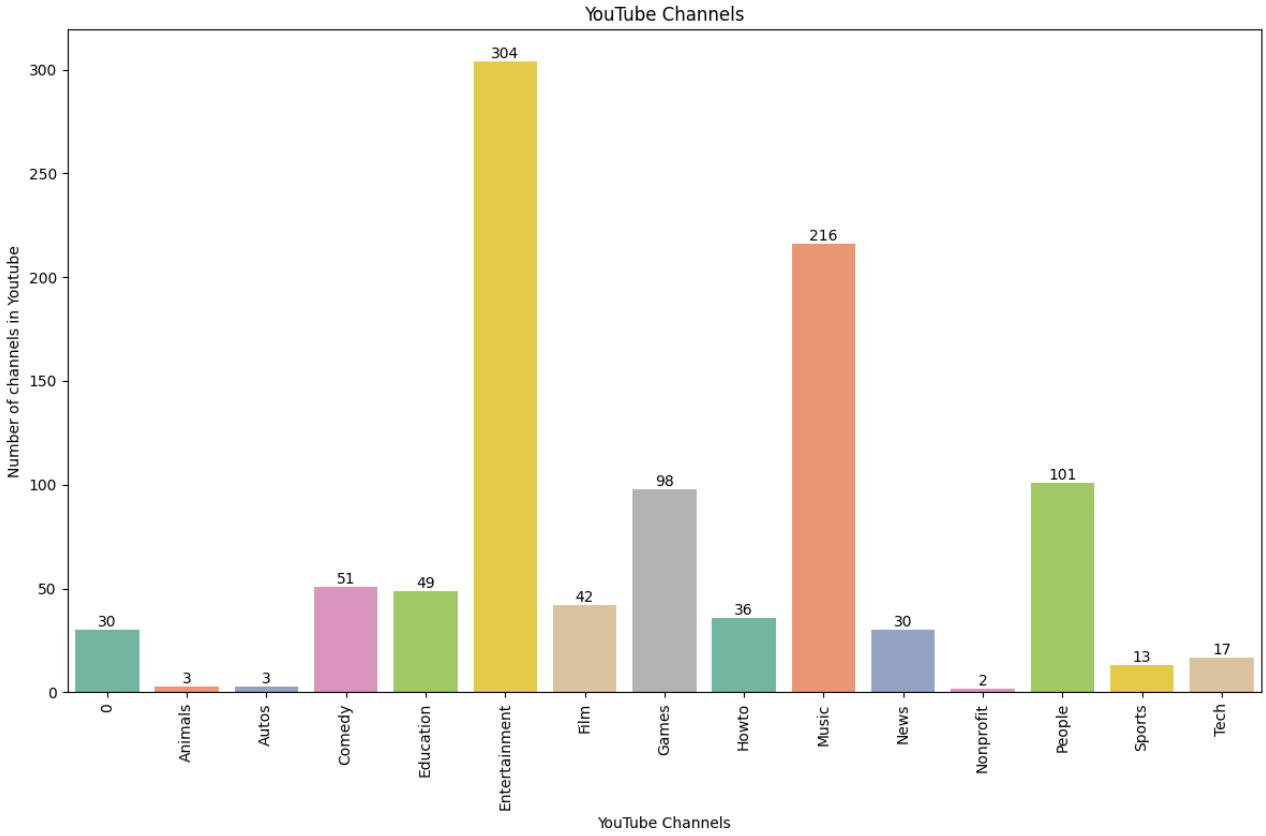


Fig Number of Youtube Channels on the basis 'channel_type'

3. How many videos, on average, are uploaded by YouTube channels in each category?

Code:

```
avg_uploads_by_category =
data.groupby('category')['uploads'].mean().reset_index(name="avg_upload")

print("Average uploads by category:")

print(avg_uploads_by_category)

print("\n")


# Bar plot for question 3

plt.figure(figsize=(12, 8))

ax = sns.barplot(x='category', y='avg_upload', data=avg_uploads_by_category,palette =
"husl")


# Rotate x-axis labels
```

```
plt.xticks(rotation=90)


bars = ax.patches
for bar in bars:
    value = int(bar.get_height())
    ax.text(bar.get_x() + bar.get_width() / 2, value, f'{value:,.0f}', ha='center', va='bottom')



# Add labels and title
plt.xlabel('YouTube Categories')
plt.ylabel('Average number of Uploads')
plt.title('Average Number of Uploads for different Categories of YouTube Channels')


# Show the plot
plt.tight_layout()
plt.show()
```

Output:

```
Average uploads by category:
                  category     avg_upload
0                        0     817.339623
1         Autos & Vehicles    1898.500000
2                   Comedy    1119.753623
3                Education    3142.866667
4            Entertainment   12052.445378
5         Film & Animation    2861.844444
6                   Gaming    4313.414894
7            Howto & Style    1700.794872
8                   Movies    3553.000000
9                    Music    2347.129353
10         News & Politics  112484.384615
11  Nonprofits & Activism  102912.000000
12          People & Blogs    9256.793893
13          Pets & Animals    4451.500000
14   Science & Technology    2114.058824
15                  Shows   27443.692308
16                 Sports   14493.727273
17               Trailers    6839.000000
18         Travel & Events     766.000000
```

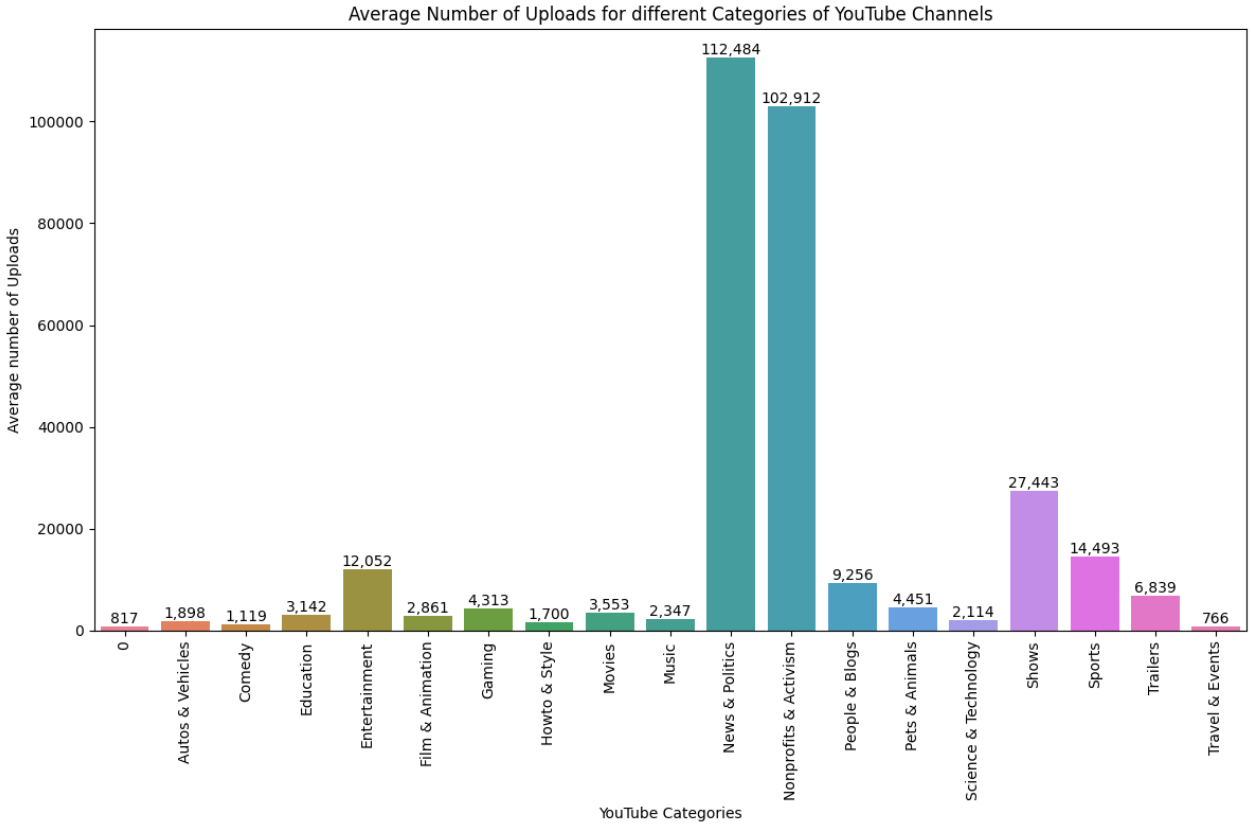


Fig Average Number of Uploads for different Youtube Categories

4. What are the top 5 countries with the highest number of YouTube channels?

The below code shows the list of Countries

Code:

```
top_5_countries = data['Country'].value_counts().head(5).reset_index(name='top_5_country')
top_countries = data['Country'].value_counts().reset_index(name='top_country')


print("Countries by Number of YouTube Channels:")
print(top_countries)
print("\n")


print("Top 5 Countries by Number of YouTube Channels:")
print(top_5_countries)
print("\n")
```

```
# Bar plot for question 4
plt.figure(figsize=(12, 8))
sns.barplot(x='top_country', y='Country', data=top_countries,palette = "husl")

# Add labels and title
plt.xlabel('Number of YouTube Channels')
plt.ylabel('Countries')
plt.title('Number of Youtube Channel in Countries')

# Show the plot
plt.tight_layout()
plt.show()

# Bar plot for question 4
plt.figure(figsize=(12, 8))
sns.barplot(x='top_5_country', y='Country', data=top_5_countries,palette = "husl")

# Add labels and title
plt.xlabel('Number of YouTube Channels')
plt.ylabel('Countries')
plt.title('Top 5 Countries with number of Youtube Channel')

# Show the plot
plt.tight_layout()
plt.show()
```

Output: List of the countries, here the ‘0’ is Nan ,not specified.

```
Countries by Number of YouTube Channels:
                Country  top_country
0         United States          313
1                 India          167
2                     0          122
3                Brazil           62
4        United Kingdom           43
5                Mexico           33
6             Indonesia           28
7                 Spain           22
8              Thailand           18
9           South Korea           17
10               Russia           16
11               Canada           15
12            Argentina           13
13          Philippines           12
14             Colombia           11
15         Saudi Arabia            9
16            Australia            9
17              Ukraine            8
18  United Arab Emirates            7
19              Germany            6
20             Pakistan            6
21                Japan            5
22               France            5
23               Turkey            4
24               Sweden            4
25              Vietnam            3
26                Chile            3
27               Jordan            3
28          Netherlands            3
29            Singapore            3
30              Ecuador            2
31                Egypt            2
32                 Iraq            2
33                Italy            2
34              Morocco            1
35                China            1
36                 Peru            1
37              Andorra            1
38           Bangladesh            1
39              Finland            1
40                india            1
41            Venezuela            1
42             Malaysia            1
43          Switzerland            1
44               Latvia            1

45          Afghanistan            1
46               Kuwait            1
47             Barbados            1
48          El Salvador            1
49                 Cuba            1
50                Samoa            1
```

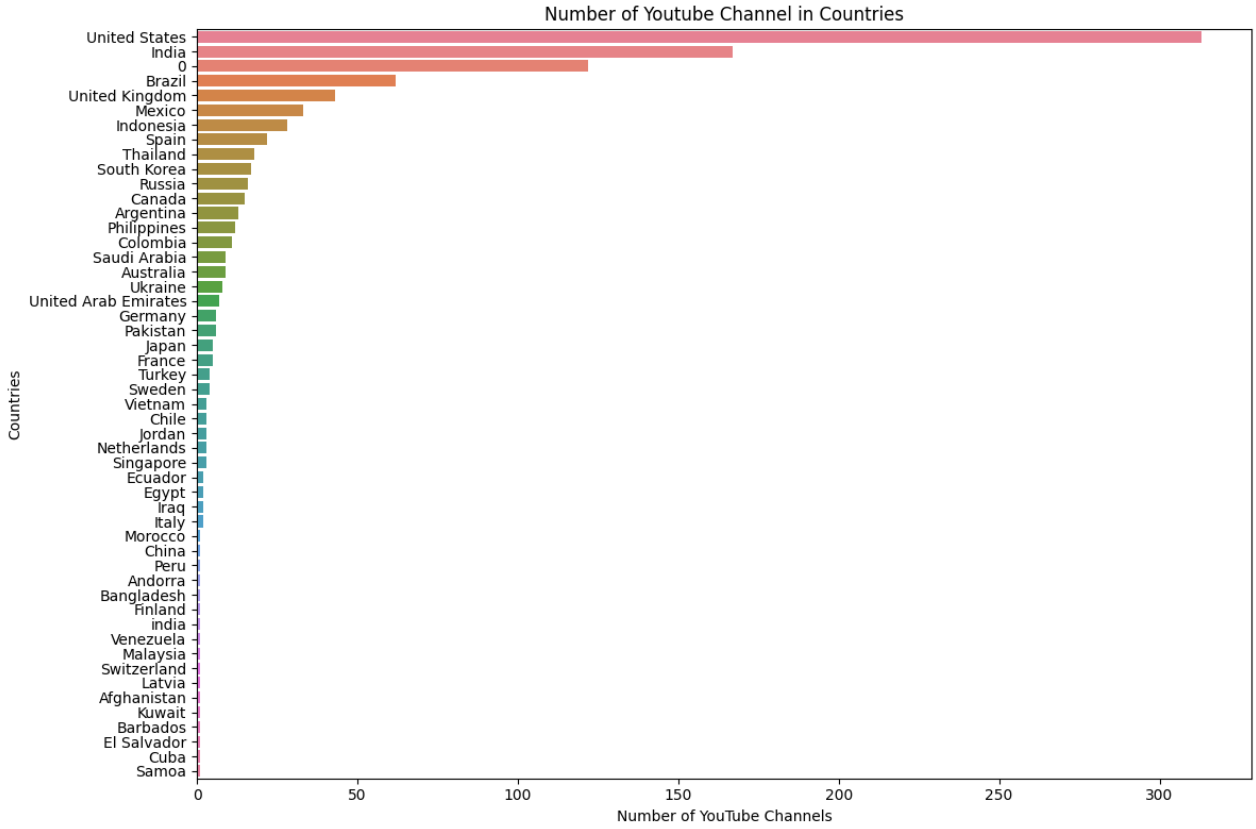
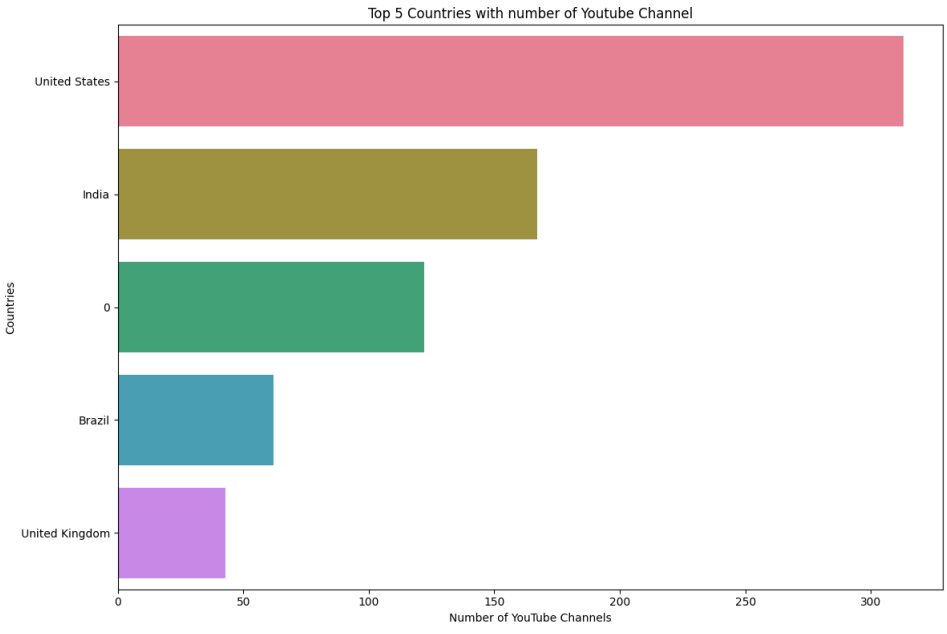


Fig Number of Youtube Channel in Countries (all)

List of top 5:

```
Top 5 Countries by Number of YouTube Channels:
          Country  top_5_country
0   United States            313
1           India            167
2               0            122
3          Brazil             62
4  United Kingdom             43
```



5. What is the distribution of channel types across different categories?

Code:

```
channel_type_dist = data.groupby(['category',
'channel_type']).size().reset_index(name='group_count')

print("Distribution of Channel Types across Categories:")

print(channel_type_dist)

print("\n")

# Create a pivot table

pivot_table = channel_type_dist.pivot(index='category', columns='channel_type',
values='group_count')



# Fill missing values with 0

pivot_table = pivot_table.fillna(0)

# Plot the correlation heatmap

plt.figure(figsize=(12, 8))

sns.heatmap(pivot_table, annot=True, cmap='YlGnBu', linewidths=0.5)

plt.title('Distribution of Channel Types across Categories')

plt.show()
```

Output:

```
Distribution of Channel Types across Categories:
           category   channel_type  group_count
0                 0              0            4
1                 0      Education            4
2                 0  Entertainment           16
3                 0           Film            2
4                 0          Games            6
..              ...            ...          ...
94           Sports  Entertainment            1
95           Sports         Sports           10
96         Trailers  Entertainment            1
97         Trailers          Music            1
98  Travel & Events  Entertainment            1

[99 rows x 3 columns]
```

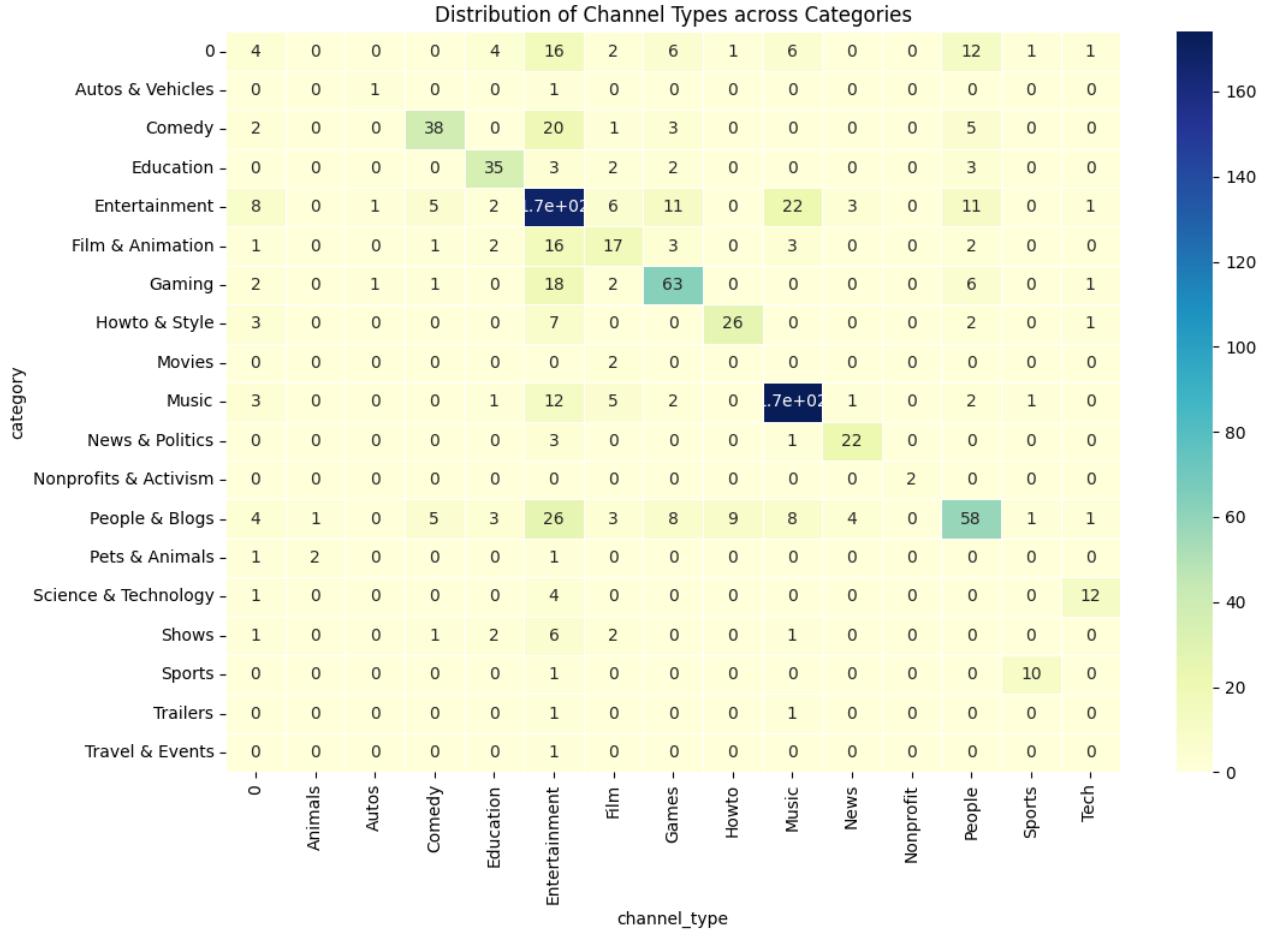


Fig Distribution of Channel Types across Categories

6. Is there a correlation between the number of subscribers and total video views for YouTube channels?

Most of the time the views on the videos are more than the subscribers ,and same trend can be seen is followed here. Many people watch the video but do not follow the youtubers .

Code:

```
# Sorting the data to get the top 10 YouTubers

top_10_youtubers = data.sort_values(by='subscribers', ascending=False).head(10)

print(top_10_youtubers[['Youtuber','subscribers','video views']])


# Create a figure and axis

fig, ax1 = plt.subplots(figsize=(12, 8))
```

```
# Create a bar plot for video views
bar_width = 0.4
index = range(len(top_10_youtubers['Youtuber']))
bars = ax1.bar(index, top_10_youtubers['video views'], bar_width, color='b', label='Video Views')

# Add a secondary y-axis for the line plot
ax2 = ax1.twinx()
ax2.plot(index, top_10_youtubers['subscribers'], color='r', marker='o', linestyle='-', linewidth=2, label='Subscribers')

# Set x-axis ticks and labels
ax1.set_xticks(index)
ax1.set_xticklabels(top_10_youtubers['Youtuber'], rotation=90)

# Add labels and title
ax1.set_xlabel('YouTuber')
ax1.set_ylabel('Video Views', color='b')
ax2.set_ylabel('Subscribers', color='r')
plt.title('Subscribers and Video Views of Top 10 YouTubers')

# Adding legends
#bars[0].set_label('Video Views')
ax1.legend(loc='upper left')
ax2.legend(loc='upper right')

# Adjust spacing and display the plot
plt.subplots_adjust()
plt.show()
```

Output:

	Youtuber	subscribers	video views
0	T-Series	245000000.0	2.280000e+11
1	YouTube Movies	170000000.0	0.000000e+00
2	MrBeast	166000000.0	2.836884e+10
3	Cocomelon - Nursery Rhymes	162000000.0	1.640000e+11
4	SET India	159000000.0	1.480000e+11
5	Music	119000000.0	0.000000e+00
6	ýýý Kids Diana Show	112000000.0	9.324704e+10
7	PewDiePie	111000000.0	2.905804e+10
8	Like Nastya	106000000.0	9.047906e+10
9	Vlad and Niki	98900000.0	7.718017e+10

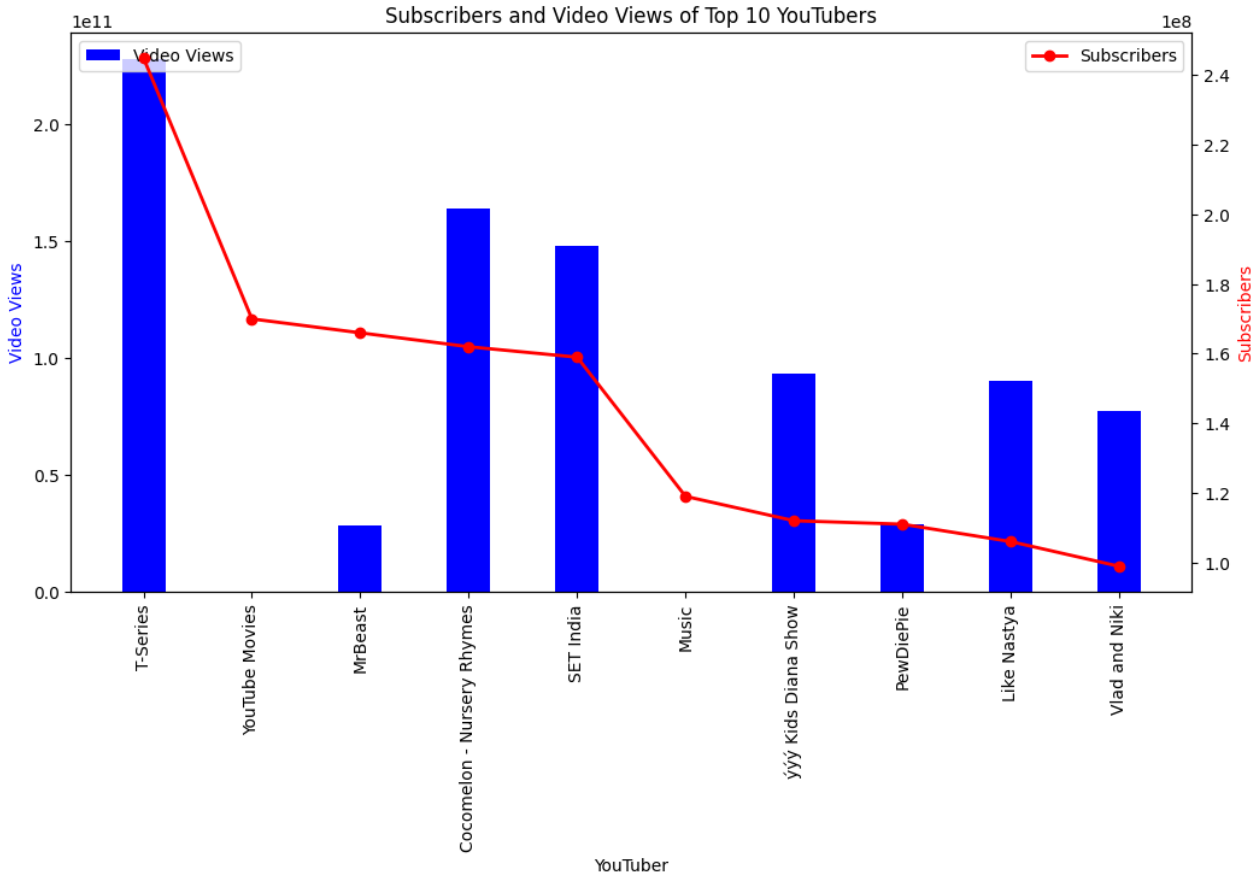


Fig Subscribers and Video Views of Top 10 Youtubers

7. How do the monthly earnings vary throughout different categories?

Code:

```
# Average Monthly earnings by Category

earnings_by_category_low = data.groupby('category')[['lowest_monthly_earnings']].mean()

earnings_by_category_high = data.groupby('category')[[ 'highest_monthly_earnings']].mean()

print("Monthly Earnings by Category:")

print(earnings_by_category_low ,'\n')

print(earnings_by_category_high,'\n')
```

Output:

```
Monthly Earnings by Category:
                        lowest_monthly_earnings
category
0                                  56316.078302
Autos & Vehicles                   68300.000000
Comedy                             41867.536232
Education                          43371.311333
Entertainment                      39393.724370
Film & Animation                   46802.533556
Gaming                             16957.979574
Howto & Style                      12053.692821
Movies                             28400.000000
Music                              34764.449801
News & Politics                    40192.625000
Nonprofits & Activism              24400.000000
People & Blogs                     33485.993969
Pets & Animals                     49975.500000
Science & Technology               12635.411765
Shows                             126961.538462
Sports                             50063.636364
Trailers                           22600.000000
Travel & Events                     7800.000000
```

category	highest_monthly_earnings
0	9.010938e+05
Autos & Vehicles	1.086350e+06
Comedy	6.683122e+05
Education	6.951778e+05
Entertainment	6.293549e+05
Film & Animation	7.489841e+05
Gaming	2.719054e+05
Howto & Style	1.925452e+05
Movies	4.547000e+05
Music	5.550474e+05
News & Politics	6.426320e+05
Nonprofits & Activism	3.904000e+05
People & Blogs	5.357493e+05
Pets & Animals	7.944322e+05
Science & Technology	2.020432e+05
Shows	2.037662e+06
Sports	8.069364e+05
Trailers	3.619000e+05
Travel & Events	1.240000e+05

Code: The average of the monthly lowest and highest earnings for different categories,

```
# Group by 'category' and calculate descriptive statistics

earnings_by_category = data.groupby('category')[['lowest_monthly_earnings',
'highest_monthly_earnings']].mean().reset_index()



# Create the plot

plt.figure(figsize=(12, 8))



# Plot lowest monthly earnings

ax = sns.barplot(x='category', y='lowest_monthly_earnings', data=earnings_by_category,
color='blue', label='Lowest Monthly Earnings')



# Plot highest monthly earnings

ax = sns.barplot(x='category', y='highest_monthly_earnings', data=earnings_by_category,
color='red', label='Highest Monthly Earnings', alpha=0.7)
```

```
# Rotate x-axis labels
plt.xticks(rotation=90)

# Add labels and title
plt.xlabel('YouTube Categories')
plt.ylabel('Average Monthly Earnings')
plt.title('Average Monthly Earnings for Different Categories of YouTube Channels')

# Add values on top of the bars
bars = ax.patches
for i, bar in enumerate(bars):
    value = int(bar.get_height())
    if i < len(bars) // 2:
        ax.text(bar.get_x() + bar.get_width() / 2, value, f'{value:,.0f}', ha='center', va='bottom', color='blue')
    else:
        ax.text(bar.get_x() + bar.get_width() / 2, value, f'{value:,.0f}', ha='center', va='bottom', color='red')

# Add a legend
plt.legend()
# Show the plot
plt.tight_layout()
plt.show()
```

Output:

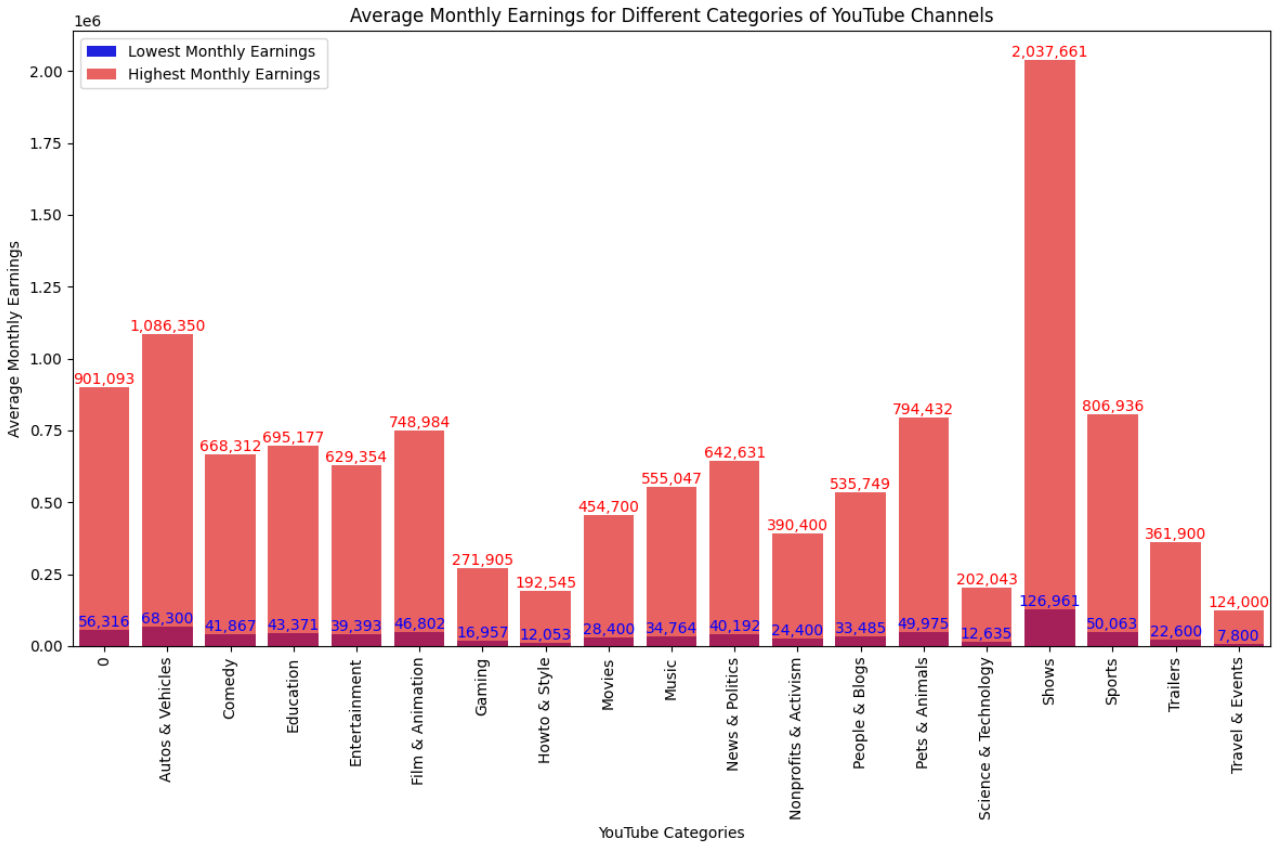


Fig Average Monthly earnings for different categories of Youtube Channels

8. What is the overall trend in subscribers gained in the last 30 days across all channels?

Code:

```
subscribers_gained_last_30_days = data['subscribers_for_last_30_days'].sum()

subs_gained =
data.groupby(['channel_type','subscribers_for_last_30_days']).size().reset_index(name='sub_
gained')

print("Subscribers gained in the last 30 days across all channels:")

print(subs_gained)



# Bar plot for question 8

plt.figure(figsize=(12, 8))

ax = sns.barplot(x='channel_type', y='subscribers_for_last_30_days',
data=subs_gained,palette = "husl")
```

```
# Rotate x-axis labels
plt.xticks(rotation=90)


bars = ax.patches
for bar in bars:
    value = int(bar.get_height())
    ax.text(bar.get_x() + bar.get_width() / 2, value, f'{value:,.0f}', ha='center', va='bottom')



# Add labels and title
plt.xlabel('YouTube Channel Types')
plt.ylabel('Subscribers Gained in Last 30 Days')
plt.title('Subscribers Gained in the Last 30 Days for Different YouTube Channel Types')


# Show the plot
plt.tight_layout()
plt.show()
```

Output:

```
Subscribers gained in the last 30 days across all channels:
    channel_type  subscribers_for_last_30_days  sub_gained
0              0                           0.0          17
1              0                           1.0           3
2              0                           2.0           3
3              0                           5.0           2
4              0                           6.0           1
..           ...                           ...         ...
170         Tech                      100000.0          10
171         Tech                      200000.0           1
172         Tech                      400000.0           1
173         Tech                      500000.0           1
174         Tech                     1100000.0           1

[175 rows x 3 columns]
```

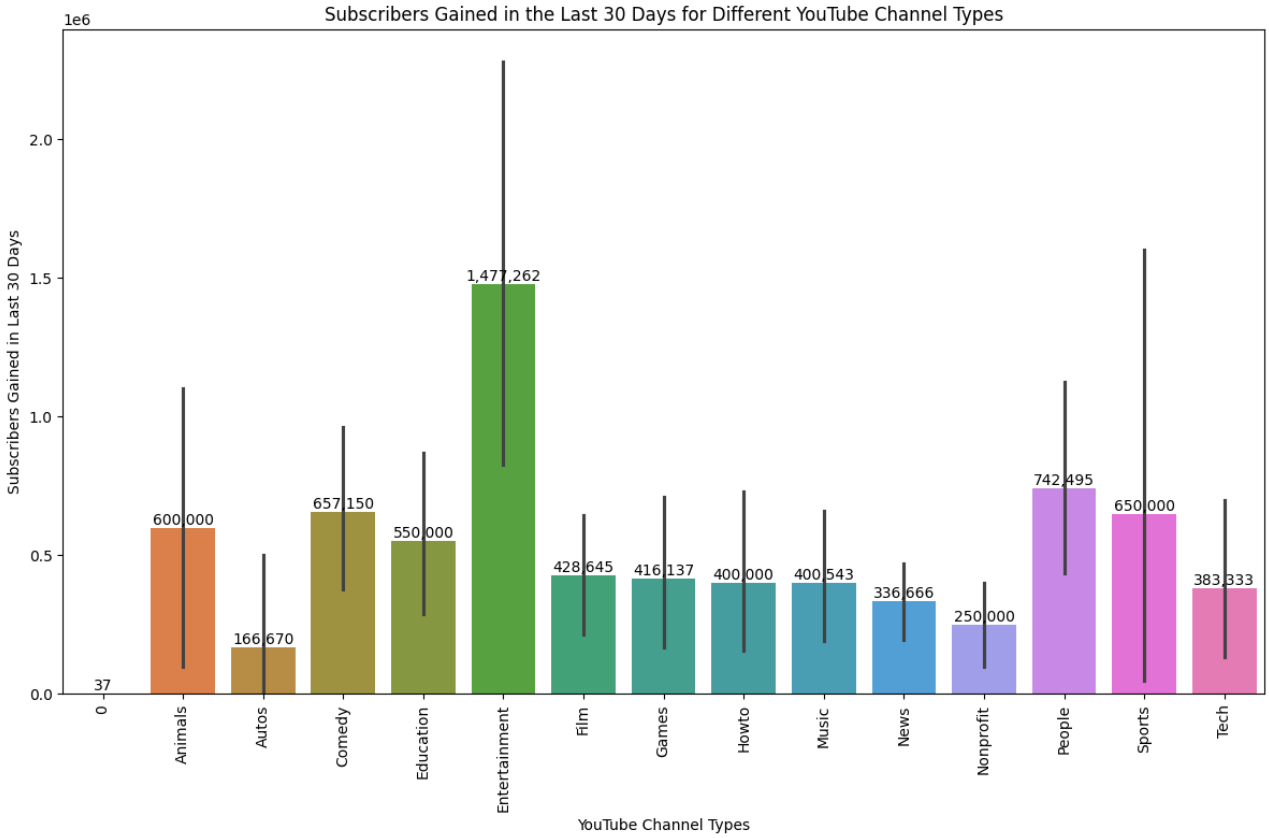


Fig Subscribers gained in the last 30 days for different YouTube channel Types

9. Are there any outliers in terms of yearly earnings from YouTube channels?

The scatterplot shows the distribution of the highest yearly earning in blue, and the outliers are highlighted in red. Using IQR method , outliers were found.

The IQR method uses Q1 , Q3 where Q1 is the 25 percentile of the highest yearly earnings and Q3 is the 75 percentile of the highest yearly earnings of the Youtube channel types . Here, IQR is the difference between the value of Q3 and Q1 , from this the lower and upper bound for outliers is decided.The values which which fall below the lower bound or above the upper bound are outliers.

Code:

```
# Calculate Q1 (25th percentile) and Q3 (75th percentile)

Q1 = data['highest_yearly_earnings'].quantile(0.25)

Q3 = data['highest_yearly_earnings'].quantile(0.75)

IQR = Q3 - Q1
```

```
# Determine the outlier thresholds

lower_bound = Q1 - 1.5 * IQR

upper_bound = Q3 + 1.5 * IQR



# Identify the outliers

outliers = data[(data['highest_yearly_earnings'] < lower_bound) |
(data['highest_yearly_earnings'] > upper_bound)]



# Add an index for plotting

data['index'] = data.index



# Scatter plot using seaborn

plt.figure(figsize=(10, 6))

sns.scatterplot(x='index', y='highest_yearly_earnings', data=data, color='blue', alpha=0.6,
label='Data Points')



# Highlight the outliers

sns.scatterplot(x='index', y='highest_yearly_earnings', data=outliers, color='red', s=100,
label='Outliers')



# Add labels and title

plt.xlabel('Index')

plt.ylabel('Highest Yearly Earnings (USD)')

plt.title('Scatter Plot of Highest Yearly Earnings with Outliers Highlighted')



# Show the plot

plt.tight_layout()
```

```
plt.legend()

plt.show()



print("The scatter plot shows the distribution of the highest yearly earnings with outliers highlighted in red.")



# Identify the outliers

outliers = data[(data['highest_yearly_earnings'] < lower_bound) |
(data['highest_yearly_earnings'] > upper_bound)]



# Extract YouTube channel names of outliers

outlier_channel_names = outliers['Youtuber'].tolist()

outliers = pd.DataFrame(outlier_channel_names)



# Print the outlier YouTube channel names

print("Outlier YouTube Channel Names:")

#print(outlier_channel_names)

print(outliers)
```

Output: The Outliers Channels Names

```
                      0
0                 T-Series
1                  MrBeast
2   Cocomelon - Nursery Rhymes
3                SET India
4          ýýý Kids Diana Show
..                      ...
94            Peet Montzingo
95                    GH'S
96               Susy Mouriz
97                  _vector_
98             Natan por Aï¿

[99 rows x 1 columns]
```

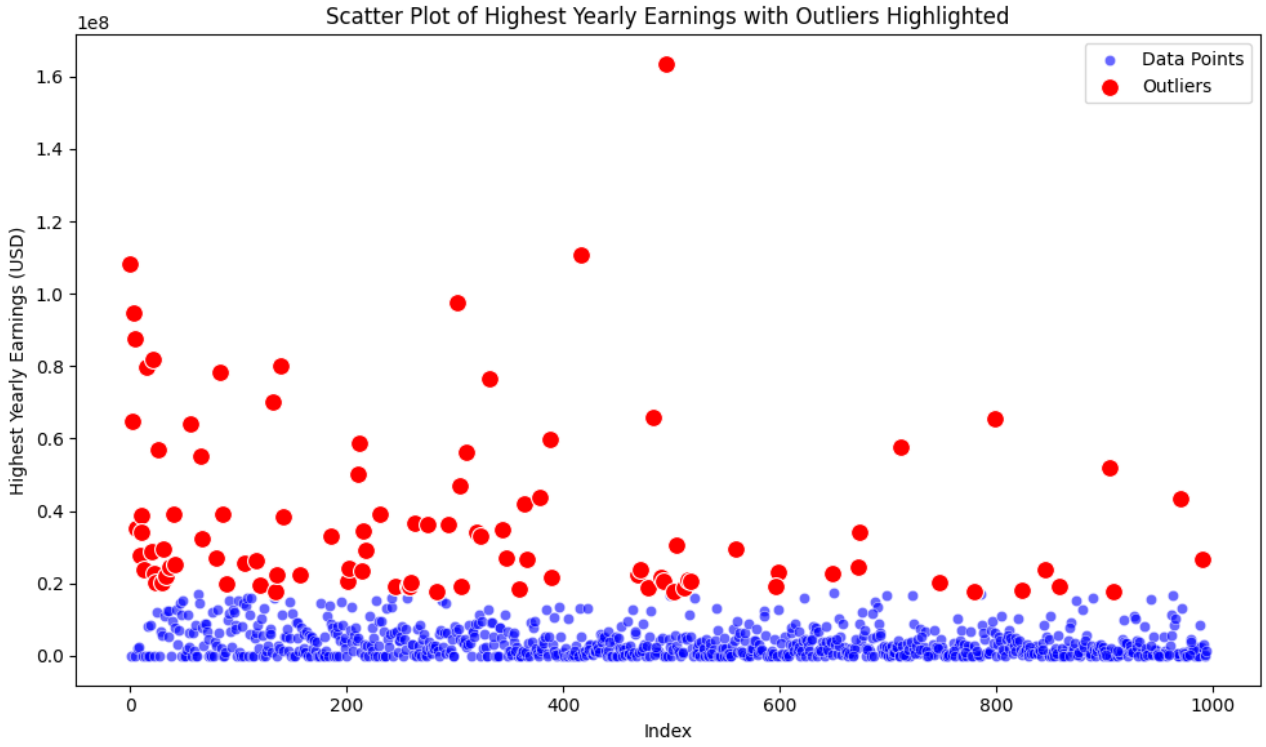


Fig Scatter plot of Highest Yearly earnings with Outliers Highlighted

10. What is the distribution of channel creation dates? Is there any trend over time?

From the given dataset , the number of channels created every year is given in the below plot. We can surge in the number channels created in the 2006 ,then in 2011 , and then 2014 , this shows there has been fluctuation in the number of the channels created over the time.

Code :

```
# Group by 'created_year' and count the number of channels

yearly_channel_count = data['created_year'].value_counts().reset_index()

yearly_channel_count.columns = ['created_year', 'channel_count']




# Create a DataFrame with all years from 1980 to the maximum year in the dataset

all_years = pd.DataFrame({'created_year': range(2000,
yearly_channel_count['created_year'].max() + 1)})



# Merge to include years with zero channels
```

```
yearly_channel_count = pd.merge(all_years, yearly_channel_count, on='created_year', how='left')

yearly_channel_count['channel_count'] = yearly_channel_count['channel_count'].fillna(0)



# Sort by 'created_year'

yearly_channel_count = yearly_channel_count.sort_values(by='created_year')



# Plotting the line plot

plt.figure(figsize=(10, 6))

sns.lineplot(x='created_year', y='channel_count', data=yearly_channel_count, marker='o', color='blue')



# Adding labels and title

plt.title('Number of YouTube Channels Created by Year')

plt.xlabel('Year')

plt.ylabel('Number of Channels')



# Set x-axis range from 1980 to maximum year

plt.xlim(2000, yearly_channel_count['created_year'].max())



# Set integer ticks with an increment of 1 for the year axis

plt.xticks(yearly_channel_count['created_year'], rotation=45)



# Show the plot

plt.tight_layout()

plt.show()
```

Output:

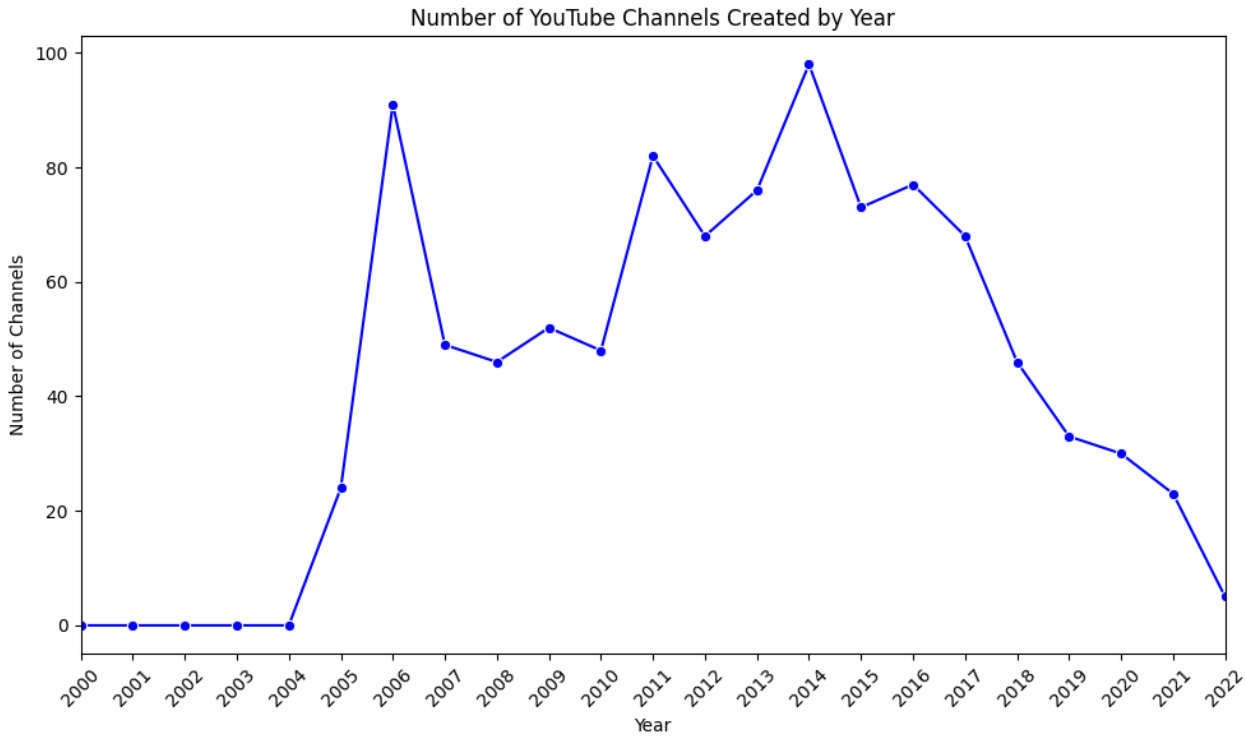


Fig Number of Youtube channels Created by Year

11. Is there a relationship between gross tertiary education enrollment and the number of YouTube channels in a country?

Gross tertiary education enrolment (%) is the percentage of population enrolled in tertiary education in the country, which includes universities, colleges, and vocational schools.

It can happen that more viewers with the educational background to engage with complex or specialized content. Here in plot we can see channel type of music is more used by the individuals.

Code:

```
gross = data.groupby(['channel_type','Gross tertiary education enrollment (%)']).size().reset_index(name='average_gross_of Channels')

print(gross)



# Create a pivot table

pivot_table = gross.pivot(index='Gross tertiary education enrollment (%)', columns='channel_type', values='average_gross_of Channels')
```

```
# Fill missing values with 0
pivot_table = pivot_table.fillna(0)


# Plot stacked bar plot
pivot_table.plot(kind='bar', stacked=True, figsize=(14, 8), colormap='tab20')
plt.title('Distribution of Channel Types across Gross Tertiary Education Enrollment')
plt.xlabel('Gross Tertiary Education Enrollment (%)')
plt.ylabel('Number of Channels')
plt.legend(title='Channel Type', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.tight_layout()
plt.show()
```

Output:

	channel_type	Gross tertiary education enrollment (%)
0	0	0.0
1	0	9.0
2	0	28.1
3	0	70.2
4	0	84.8
..	...	...
183	Tech	51.3
184	Tech	60.0
185	Tech	68.9
186	Tech	70.2
187	Tech	88.2

	average_gross_of Channels
0	23
1	1
2	2
3	1
4	1
..	...
183	1
184	1
185	2
186	1
187	4

[188 rows x 3 columns]

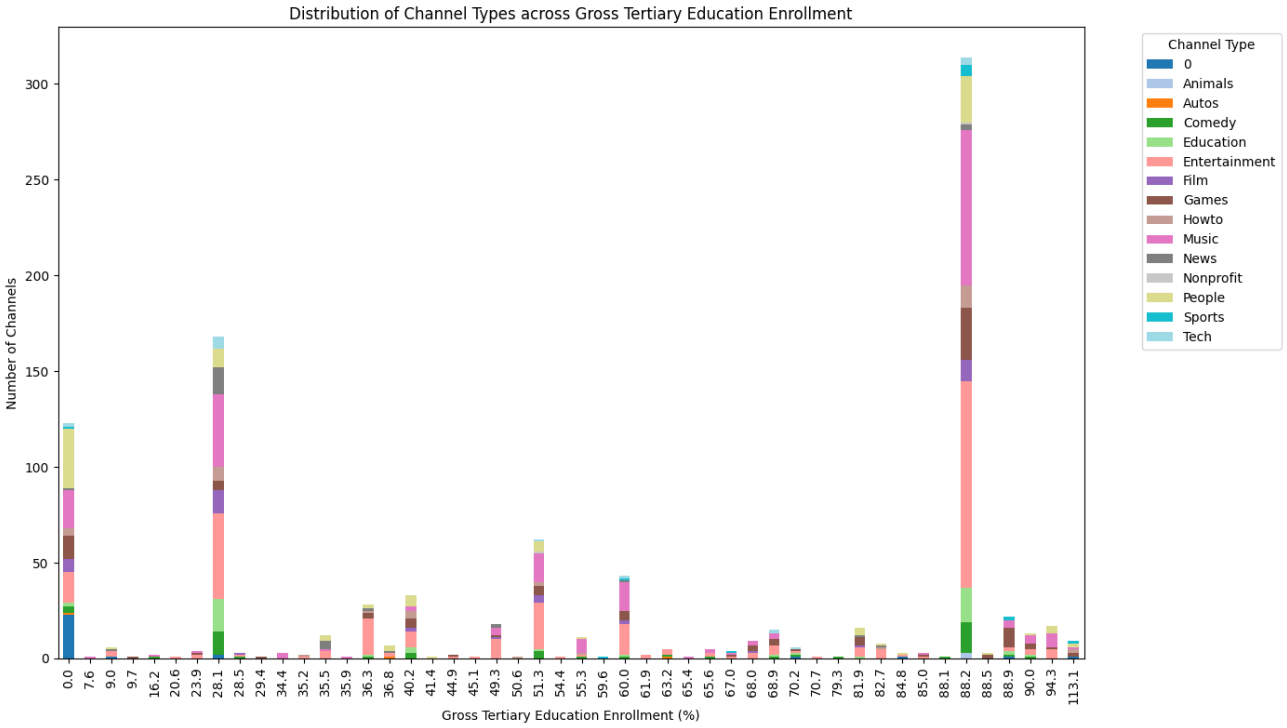


Fig Distribution of Channel Type across Different GTE Enrollment

12. How does the unemployment rate vary among the top 10 countries with the highest number of YouTube channels?

Code:

```
# Find the top 10 countries with the most YouTube channels

top_10_countries = data['Country'].value_counts().head(10).index

print("Top 10 countries:")

print(top_10_countries)



# Extract the unemployment rates for these top 10 countries

unemployment_rates = data[data['Country'].isin(top_10_countries)][['Country',
'Unemployment rate']].dropna()

print("Unemployment rates:")

print(unemployment_rates)



# Plot the unemployment rate with respect to country
```

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

sns.barplot(x='Country', y='Unemployment rate', data=unemployment_rates,
palette='viridis')



# Add labels and title

plt.xlabel('Country')

plt.ylabel('Unemployment Rate (%)')

plt.title('Unemployment Rate by Country for Top 10 Countries with Most YouTube
Channels')



# Display the plot

plt.tight_layout()

plt.show()
```

Output:

```
Top 10 countries:
Index([ 'United States',          'India',                0,         'Brazil',
       'United Kingdom',         'Mexico',      'Indonesia',          'Spain',
             'Thailand',    'South Korea'],
      dtype='object', name='Country')
Unemployment rates:
            Country  Unemployment rate
1     United States              14.70
2     United States              14.70
3     United States              14.70
4             India               5.36
5                 0               0.00
..              ...                ...
989   United States              14.70
990          Brazil              12.08
991           India               5.36
992  United Kingdom               3.85
994           India               5.36
```

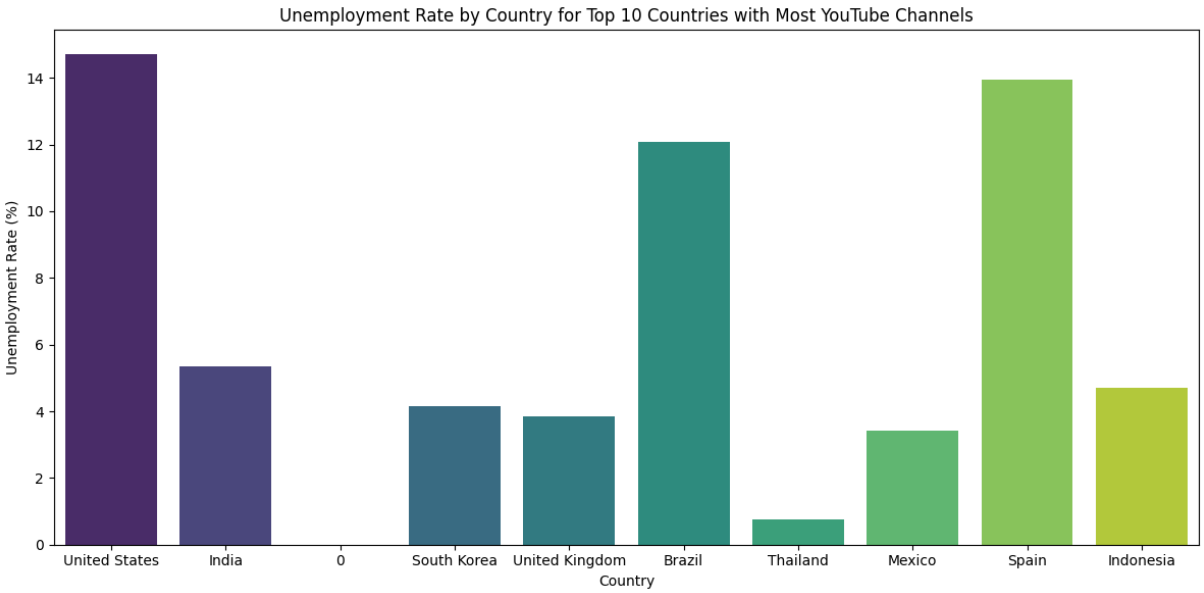


Fig Unemployment Rate by Country for Top 10 Countries with most Youtube channels

13. What is the average urban population percentage in countries with YouTube channels?

Code:

```
# Ensure no duplicates are present for unique country entries
relevant_data = data.drop_duplicates(subset=['Country'])


# Convert 'Urban_population' and 'Population' columns to numeric if they are not already
relevant_data['Urban_population'] = pd.to_numeric(relevant_data['Urban_population'], errors='coerce')
relevant_data['Population'] = pd.to_numeric(relevant_data['Population'], errors='coerce')


# Filter out rows where 'Urban_population' or 'Population' is 0
relevant_data = relevant_data[(relevant_data['Urban_population'] != 0) & (relevant_data['Population'] != 0)]


# Calculate urban population percentage, handling potential division by zero or NaN
relevant_data['Urban population percentage'] = (relevant_data['Urban_population'] / relevant_data['Population']) * 100
```

```
# Calculate the average urban population for each row
relevant_data['Average Urban Population'] = relevant_data.apply(lambda row: row['Urban_population'] / row['Population'], axis=1)


# Calculate the average urban population percentage, excluding NaN values
average_urban_population_percentage = relevant_data['Urban population percentage'].mean()


# Plotting the urban population percentage and average urban population for each country
plt.figure(figsize=(14, 10))


# Plot 1: Bar plot for Urban Population Percentage
plt.subplot(2, 1, 1)
ax1 = sns.barplot(x='Country', y='Urban population percentage', data=relevant_data, palette='viridis')
plt.axhline(average_urban_population_percentage, color='red', linestyle='--', label=f'Average: {average_urban_population_percentage:.2f}%')
for p in ax1.patches:
    ax1.annotate(f'{p.get_height():.2f}%', (p.get_x() + p.get_width() / 2., p.get_height()), ha='center', va='center', xytext=(0, 10), textcoords='offset points')
plt.xlabel('Country')
plt.ylabel('Urban Population Percentage (%)')
plt.title('Urban Population Percentage by Country')
plt.legend()
plt.xticks(rotation=90)


# Plot 2: Line plot for Average Urban Population
plt.subplot(2, 1, 2)
sns.lineplot(x='Country', y='Average Urban Population', data=relevant_data, marker='o', color='b')
plt.xlabel('Country')
```

```
plt.ylabel('Average Urban Population')
plt.title('Average Urban Population by Country')
plt.xticks(rotation=90)

plt.tight_layout()
plt.show()

print(f"The average urban population percentage in countries with YouTube channels is: {average_urban_population_percentage:.2f}%")
```

Output:

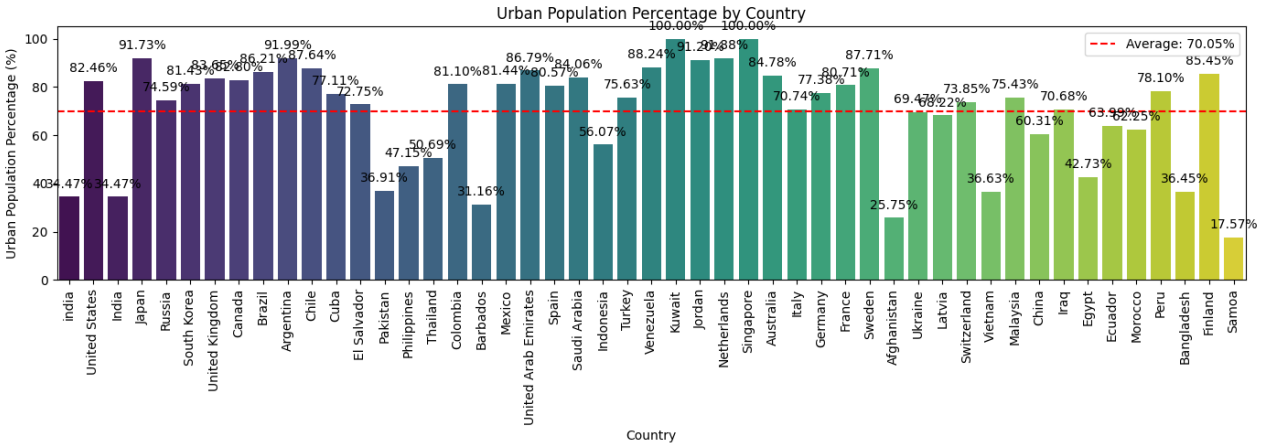


Fig Urban population percentage by country

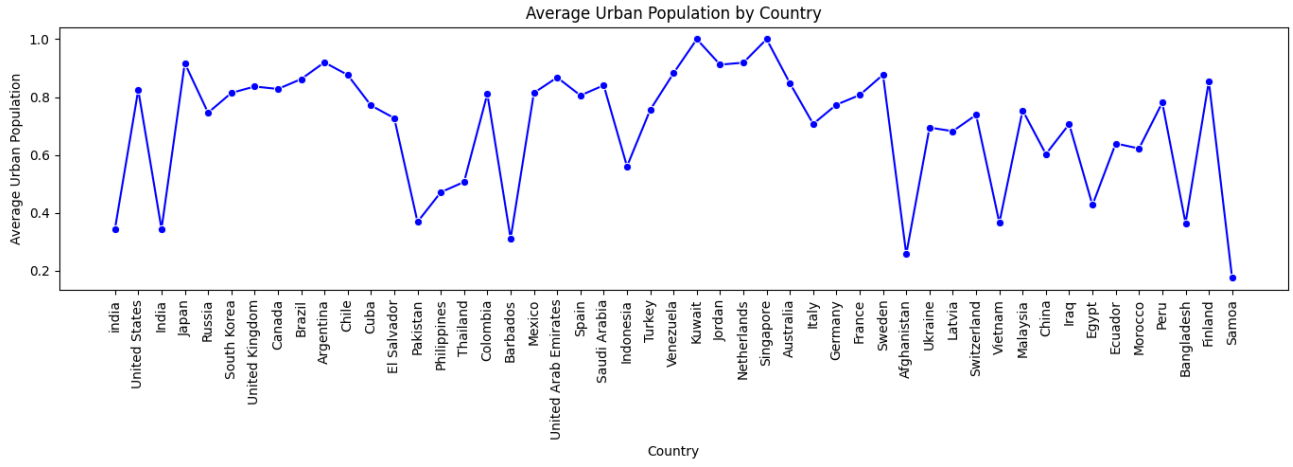


Fig Average urban population by country

```
The average urban population percentage in countries with YouTube channels is: 70.05%
```

14. Are there any patterns in the distribution of YouTube channels based on latitude and longitude coordinates?

Code:

```
# Scatter plot comparing Latitude and Longitude with Channel Type

plt.figure(figsize=(10, 8))

sns.scatterplot(x='Longitude', y='Latitude', hue='channel_type', data=data, s=100, palette='viridis', legend='full')

# Add labels and title

plt.xlabel('Longitude')

plt.ylabel('Latitude')

plt.title('Distribution of YouTube Channels by Latitude and Longitude')


# Show legend

plt.legend(loc='lower left', title='Channel Type')

plt.tight_layout()

plt.show()
```

Output:

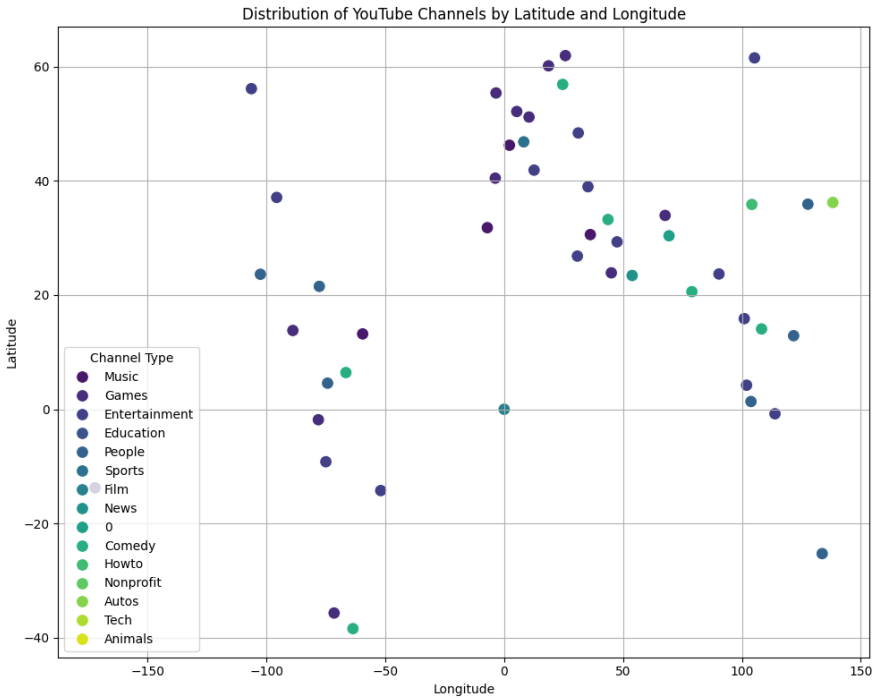


Fig Distribution of YouTube channel by latitude and longitude

15. What is the correlation between the number of subscribers and the population of a country?

Code:

```
# Convert 'Country' to categorical data

data['Country'] = data['Country'].astype(str)



# Convert 'Country' to categorical data

data['Country'] = pd.Categorical(data['Country'], categories=data['Country'].unique(), ordered=True)



# Plotting the relationship between Subscribers and Population

plt.figure(figsize=(10, 6))

sns.lineplot(x='Country', y='subscribers', data=data, marker='o', color='b', label='subscribers')

sns.lineplot(x='Country', y='Population', data=data, marker='o', color='r', label='Population')



# Add labels and title

plt.xlabel('Country')

plt.ylabel('Count')

plt.title('Subscribers and Population by Country')



# Show the plot

plt.legend()

plt.xticks(rotation=90)

plt.tight_layout()



plt.show()
```

Output:

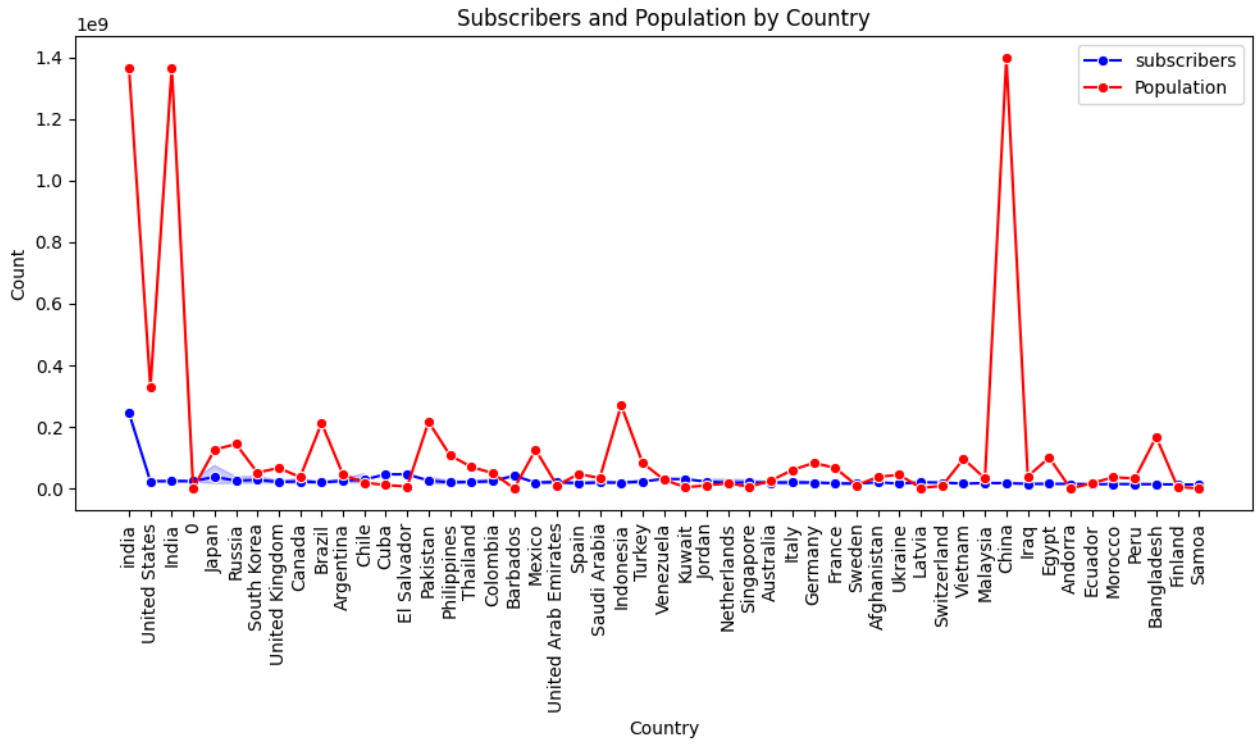


Fig Subscribers and Population by countries

16. How do the top 10 countries with the highest number of YouTube channels compare in terms of their total population?

Code:

```
top_10_countries = data['Country'].value_counts().head(10).index


# Select data for top 10 countries

top_10_data = data[data['Country'].isin(top_10_countries)]


# Plotting the comparison of Channel Types and Population for top 10 countries

plt.figure(figsize=(12, 8))


# Plotting Channel Types

sns.countplot(y='Country', hue='channel_type', data=top_10_data, palette='viridis')


# Adding labels and title
```

```
plt.xlabel('Count')
plt.ylabel('Country')
plt.title('Comparison of Channel Types for Top 10 Countries')

# Displaying legends
plt.legend(title='Channel Type')

# Adjusting layout
plt.tight_layout()

# Showing the plot
plt.grid()
plt.show()
```

Output:

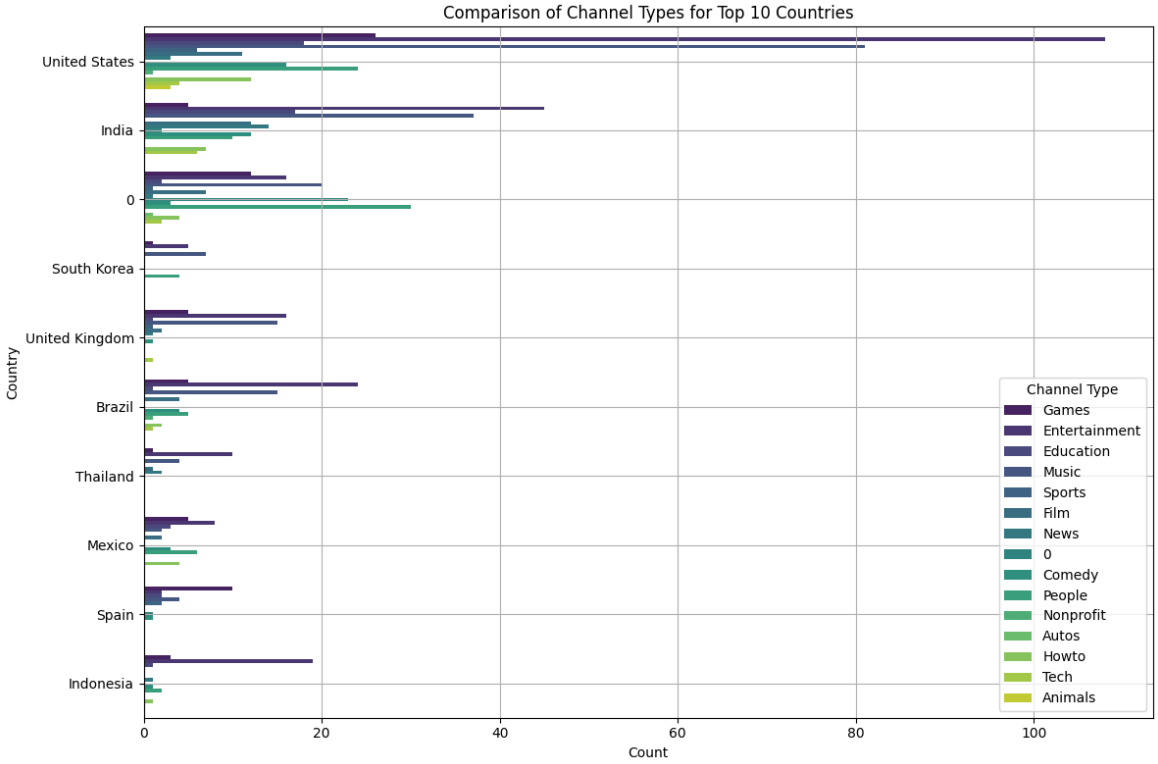


Fig Comparison of channel types for top 10 countries

17. Is there a correlation between the number of subscribers gained in the last 30 days and the unemployment rate in a country?

Code:

```
# Calculate the average subscribers gained in the last 30 days for each channel type
average_subs = data.groupby('channel_type')['subscribers_for_last_30_days'].mean().reset_index()

# Plotting the bar graph
plt.figure(figsize=(10, 6))
sns.barplot(x='channel_type', y='subscribers_for_last_30_days', data=data, ci=None, palette='viridis')
plt.title('Average Subscribers Gained Last 30 Days by Channel Type')
plt.xlabel('Channel Type')
plt.ylabel('Average Subscribers Gained Last 30 Days')

# Adding the average values as text on the bars
bars = sns.barplot(x='channel_type', y='subscribers_for_last_30_days', data=average_subs, ci=None, palette='viridis')
for bar in bars.patches:
    bars.annotate(format(bar.get_height(), '.1f'), (bar.get_x() + bar.get_width() / 2, bar.get_height()), ha='center', va='center', size=12, xytext=(0, 8),textcoords='offset points')

# Rotating x-axis labels for better readability
plt.xticks(rotation=45)

# Showing plot
plt.tight_layout()
plt.show()
```

Output:

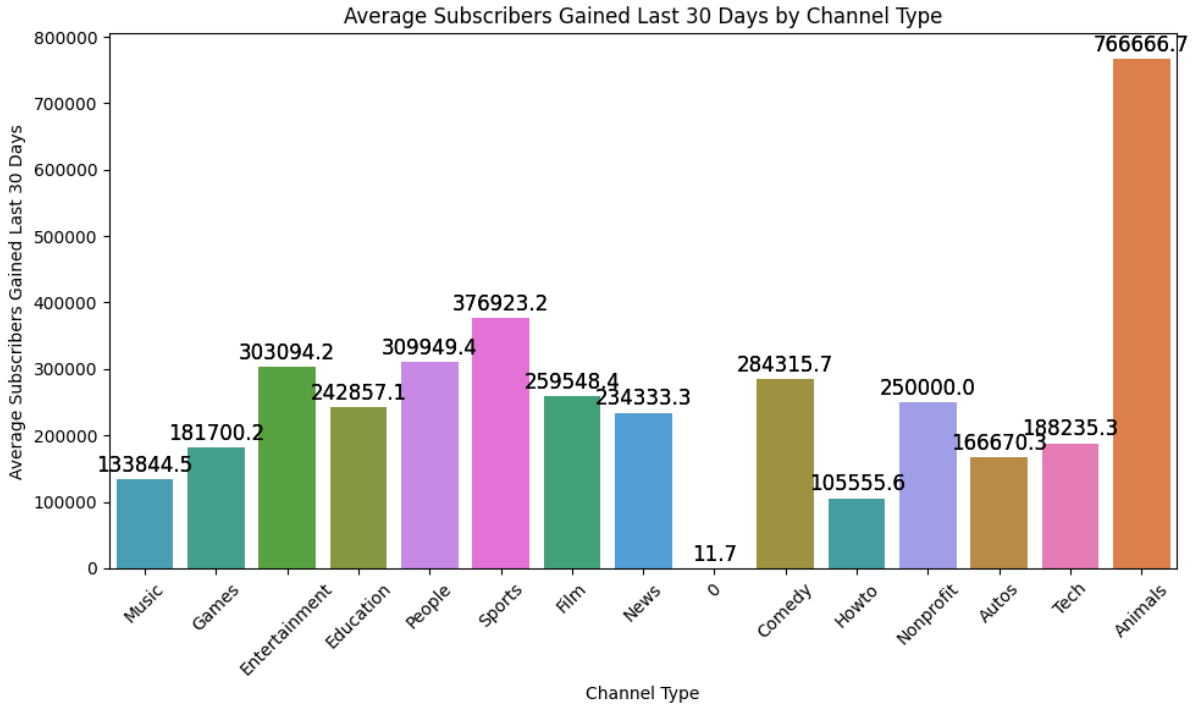


Fig Average subscribers gained last 30 days by Channel Type

18. How does the distribution of video views for the last 30 days vary across different channel types?

Code:

```
# Calculate the average video views for the last 30 days for each channel type

average_views =
data.groupby('channel_type')['video_views_for_the_last_30_days'].mean().reset_index()



# Plotting the bar graph

plt.figure(figsize=(10, 6))

sns.barplot(x='channel_type', y='video_views_for_the_last_30_days', data=data, ci=None,
palette='viridis')

plt.title('Average Video Views for the Last 30 Days by Channel Type')

plt.xlabel('Channel Type')

plt.ylabel('Average Video Views for the Last 30 Days')
```

```
# Adding the average values as text on the bars (vertical orientation)

bars = sns.barplot(x='channel_type', y='video_views_for_the_last_30_days',
data=average_views, ci=None, palette='husl')

for bar in bars.patches:

    bars.annotate(format(bar.get_height(), '.1f'),

                  (bar.get_x() + bar.get_width() / 2,

                   bar.get_height()),

                  ha='center',

                  va='bottom',  # Write the text below the horizontal line

                  size=12, xytext=(0, 10),

                  textcoords='offset points',

                  rotation=90)  # Rotate the text to vertical


# Rotating x-axis labels for better readability

plt.xticks(rotation=45)


# Showing plot

plt.tight_layout()

plt.show()
```

Output:

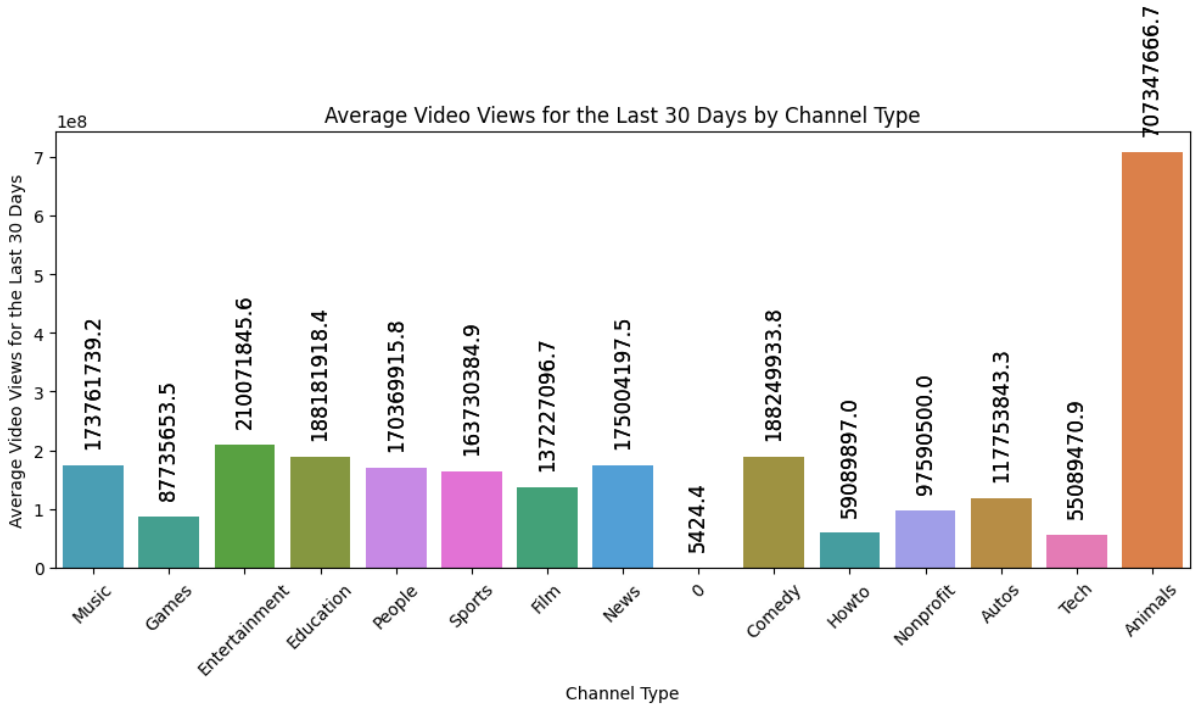


Fig Average video views for the last 30 days by channel type