Importing libraries :

In [1]:
```
import pandas as pd
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
```

Data read :

In [2]:
```
df = pd.read_csv(r'/Users/priyankaagarwal/Downloads/finlatics/DsResearch/
```

In [3]:
```
pd.set_option('display.max_rows',None)
pd.set_option('display.max_columns',None)
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1006 entries, 0 to 1005
Data columns (total 29 columns):
 #   Column                                  Non-Null Count  Dtype  
---  ------                                  --------------  -----  
 0   rank                                    1006 non-null   int64  
 1   Youtuber                                1006 non-null   object 
 2   subscribers                             1003 non-null   float64
 3   video views                             1006 non-null   float64
 4   category                                951 non-null    object 
 5   Title                                   1006 non-null   object 
 6   uploads                                 1006 non-null   int64  
 7   Country of origin                       881 non-null    object 
 8   Country                                 881 non-null    object 
 9   Abbreviation                            881 non-null    object 
 10  channel_type                            974 non-null    object 
 11  video_views_rank                        1005 non-null   float64
 12  country_rank                            887 non-null    float64
 13  channel_type_rank                       971 non-null    float64
 14  video_views_for_the_last_30_days        949 non-null    float64
 15  lowest_monthly_earnings                 1006 non-null   float64
 16  highest_monthly_earnings                1006 non-null   float64
 17  lowest_yearly_earnings                  1006 non-null   float64
 18  highest_yearly_earnings                 1006 non-null   float64
 19  subscribers_for_last_30_days            666 non-null    float64
 20  created_year                            1001 non-null   float64
 21  created_month                           994 non-null    object 
 22  created_date                            1001 non-null   float64
 23  Gross tertiary education enrollment (%) 880 non-null    float64
 24  Population                              880 non-null    float64
 25  Unemployment rate                       880 non-null    float64
 26  Urban_population                        880 non-null    float64
 27  Latitude                                880 non-null    float64
 28  Longitude                               880 non-null    float64
dtypes: float64(19), int64(2), object(8)
memory usage: 228.0+ KB
None
```

Clearing data:

In [7]:
```
# print(df['Country'].describe())
# print(df['Abbreviation'].describe())

df.drop(columns=['Abbreviation'], inplace=True)

# print(df['Youtuber'].describe())
# print(df['Title'].describe())

df.drop(columns=['Title'], inplace=True)

df['subscribers']=df['subscribers'].fillna(df['subscribers'].median())
df['category']=df['category'].fillna('other')

# print(df['Country'].describe())
# print(df['Country of origin'].describe())

df.drop(columns=['Country of origin'], inplace=True)

df_country_analysis = df.dropna(subset=['Country','Population'])

columns_to_analyze = ['Country', 'Gross tertiary education enrollment (%)
                      'Population', 'Unemployment rate', 'Urban_population
                      'Latitude', 'Longitude']

for col in columns_to_analyze:
    print(f"\n=== {col} ===")
    print(f"Unique values: {df[col].nunique()}")
    print(f"Most frequent value: {df[col].mode()[0] if len(df[col].mode()
    print("Top 5 most frequent:")
    print(df[col].value_counts().head())

# print(df_country_analysis.isnull().sum())

df = df.dropna(subset=['channel_type','channel_type_rank'])

cols = ['video_views_for_the_last_30_days','subscribers_for_last_30_days'
for c in cols:
    df[c]=df[c].fillna(df[c].mean())

changed = ['created_year','created_month','created_date']
for c in changed:
    df[c]=df[c].fillna(df[c].mode()[0])

# print(df.isnull().sum())
# print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 971 entries, 0 to 1005
Data columns (total 26 columns):
 #   Column                                  Non-Null Count  Dtype  
---  ------                                  --------------  -----  
 0   rank                                    971 non-null    int64  
 1   Youtuber                                971 non-null    object 
 2   subscribers                             971 non-null    float64
 3   video views                             971 non-null    float64
 4   category                                971 non-null    object 
 5   uploads                                 971 non-null    int64  
 6   Country                                 872 non-null    object 
 7   channel_type                            971 non-null    object 
 8   video_views_rank                        971 non-null    float64
 9   country_rank                            878 non-null    float64
 10  channel_type_rank                       971 non-null    float64
 11  video_views_for_the_last_30_days        971 non-null    float64
 12  lowest_monthly_earnings                 971 non-null    float64
 13  highest_monthly_earnings                971 non-null    float64
 14  lowest_yearly_earnings                  971 non-null    float64
 15  highest_yearly_earnings                 971 non-null    float64
 16  subscribers_for_last_30_days            971 non-null    float64
 17  created_year                            971 non-null    float64
 18  created_month                           971 non-null    object 
 19  created_date                            971 non-null    float64
 20  Gross tertiary education enrollment (%) 871 non-null    float64
 21  Population                              871 non-null    float64
 22  Unemployment rate                       871 non-null    float64
 23  Urban_population                        871 non-null    float64
 24  Latitude                                871 non-null    float64
 25  Longitude                               871 non-null    float64
dtypes: float64(19), int64(2), object(5)
memory usage: 204.8+ KB
None
```

QUESTIONS :

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

In [5]:
```
print(df[['rank', 'Youtuber','subscribers']].head(10))
```

```
   rank                    Youtuber  subscribers
0     1                    T-Series  245000000.0
1     2               YouTube Movies  170000000.0
2     3                     MrBeast  166000000.0
3     4  Cocomelon - Nursery Rhymes  162000000.0
4     5                    SET India  159000000.0
6     7          ��? Kids Diana Show  112000000.0
7     8                    PewDiePie  111000000.0
8     9                  Like Nastya  106000000.0
9    10                Vlad and Niki   98900000.0
10   11            Zee Music Company   96700000.0
```

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

In [8]:
```
grped_data = df.groupby(['category'])['subscribers'].mean().sort_values(a
print(grped_data.head(1))
```

```
category
Shows    4.350833e+07
Name: subscribers, dtype: float64
```

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

In [9]:
```
grouped_data = df.groupby(['category'])['uploads'].mean()
print(grouped_data)
```

```
category
Autos & Vehicles           1550.666667
Comedy                     1255.776119
Education                  3087.086957
Entertainment             12471.365217
Film & Animation           2926.750000
Gaming                     4377.430108
Howto & Style              1827.135135
Movies                     3553.000000
Music                      2372.587940
News & Politics          112484.384615
Nonprofits & Activism    102912.000000
People & Blogs             9548.275591
Pets & Animals             5932.666667
Science & Technology       2232.250000
Shows                     29730.666667
Sports                    19129.833333
Trailers                   6839.000000
Travel & Events             766.000000
other                       886.918367
Name: uploads, dtype: float64
```

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

In [10]:
```
grouped_data_country = df_country_analysis.groupby(['Country'])['Youtuber
print(grouped_data_country.head(5))
```

```
Country
United States     315
India             169
Brazil             62
United Kingdom     44
Mexico             33
Name: Youtuber, dtype: int64
```

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

In [11]:
```
grouped_data_categories = df.groupby(['category','channel_type'])['channe
print(grouped_data_categories)
```

category	channel_type	
Autos & Vehicles	Autos	2
	Entertainment	1
Comedy	Comedy	38
	Entertainment	20
	Film	1
	Games	3
	People	5
Education	Education	36
	Entertainment	3
	Film	2
	Games	2
	People	3
Entertainment	Autos	1
	Comedy	5
	Education	2
	Entertainment	168
	Film	6
	Games	11
	Music	22
	News	3
	People	11
	Tech	1
Film & Animation	Comedy	1
	Education	2
	Entertainment	16
	Film	17
	Games	3
	Music	3
	People	2
Gaming	Autos	1
	Comedy	1
	Entertainment	18
	Film	2
	Games	64
	People	6
	Tech	1
Howto & Style	Entertainment	7
	Howto	27
	People	2
	Tech	1
Movies	Film	2
Music	Education	1
	Entertainment	12
	Film	5
	Games	2
	Music	175
	News	1
	People	2
	Sports	1
News & Politics	Entertainment	3
	Music	1
	News	22
Nonprofits & Activism	Nonprofit	2
People & Blogs	Animals	1
	Comedy	5
	Education	3
	Entertainment	26
	Film	3
	Games	8

```
                        Howto              9
                        Music              8
                        News               4
                        People            58
                        Sports             1
                        Tech               1
Pets & Animals          Animals            2
                        Entertainment      1
Science & Technology    Entertainment      4
                        Tech              12
Shows                   Comedy             1
                        Education          2
                        Entertainment      6
                        Film               2
                        Music              1
Sports                  Entertainment      1
                        Sports            11
Trailers                Entertainment      1
                        Music              1
Travel & Events         Entertainment      1
other                   Education          4
                        Entertainment     16
                        Film               2
                        Games              7
                        Howto              1
                        Music              4
                        People            13
                        Sports             1
                        Tech               1
Name: channel_type, dtype: int64
```

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

In [12]:
```
correlation = df['subscribers'].corr(df['video views'])
print("correlation: ",correlation)

if correlation > 0.9:
    print("Very high positive correlation")
elif correlation > 0.7:
    print("High positive correlation")
elif correlation > 0.5:
    print("Moderate high positive correlation")
elif correlation > 0.3:
    print("Low positive correlation")
elif correlation > 0.0:
    print("negligible correlation")
elif correlation == 0:
    print("No correlation")
elif correlation > -0.3:
    print("Negligible correlation")
elif correlation > -0.5:
    print("Low negative correlation")
elif correlation > -0.7:
    print("Moderate negative correlation")
elif correlation > -0.9:
    print("High negative correlation")
else:
    print("Very high negative correlation.")
```

```
correlation:  0.7651380938648109
High positive correlation
```

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

In [13]:
```
earnings_stats = df.groupby('category')[['lowest_monthly_earnings', 'high
print(earnings_stats)
```

	lowest_monthly_earnings			highest_monthly_earnings		
	mean	median	std	min	max	mean
category						
Autos & Vehicles	74966.666667	88300.0	23094.010768	48300.0	88300.0	1.190900e+06
Comedy	43182.761194	8800.0	69071.785805	0.0	311200.0	6.893015e+05
Education	46863.239348	22800.0	82734.102384	0.0	493800.0	7.518043e+05
Entertainment	40763.919130	12250.0	74668.899721	0.0	508100.0	6.512451e+05
Film & Animation	47866.227500	16100.0	100978.973331	0.0	576000.0	7.660064e+05
Gaming	17330.646022	9900.0	31904.693141	0.0	270300.0	2.778700e+05
Howto & Style	12996.703243	6200.0	22064.398282	0.0	125700.0	2.076081e+05
Movies	28400.000000	28400.0	40163.665171	0.0	56800.0	4.547000e+05
Music	35368.614925	22200.0	56872.430781	0.0	564600.0	5.647006e+05
News & Politics	40192.625000	30850.0	35280.065373	0.0	115400.0	6.426320e+05
Nonprofits & Activism	24400.000000	24400.0	18384.776311	11400.0	37400.0	3.904000e+05
People & Blogs	34540.670945	12000.0	56692.095622	0.0	340900.0	5.526233e+05
Pets & Animals	66633.333333	11100.0	105944.907067	0.0	188800.0	1.059233e+06
Science & Technology	13425.000000	11550.0	11464.641294	0.0	41900.0	2.146688e+05
Shows	137541.666667	51200.0	161800.997066	14100.0	455900.0	2.207467e+06
Sports	60783.333333	27250.0	68489.492669	2600.0	178700.0	9.813583e+05
Trailers	22600.000000	22600.0	31961.226510			
Travel & Events	7800.000000	7800.0	NaN			
other	60918.105102	11700.0	144427.900107			

	highest_monthly_earnings
	median
category	
Autos & Vehicles	140000.0
Comedy	14050.0
Education	36495.0
Entertainment	19565.0
Film & Animation	25725.0
Gaming	15910.0
Howto & Style	9880.0
Movies	45470.0
Music	35580.0
News & Politics	49330.0
Nonprofits & Activism	39040.0
People & Blogs	19270.0
Pets & Animals	17770.0
Science & Technology	18455.0
Shows	82005.0
Sports	43615.0

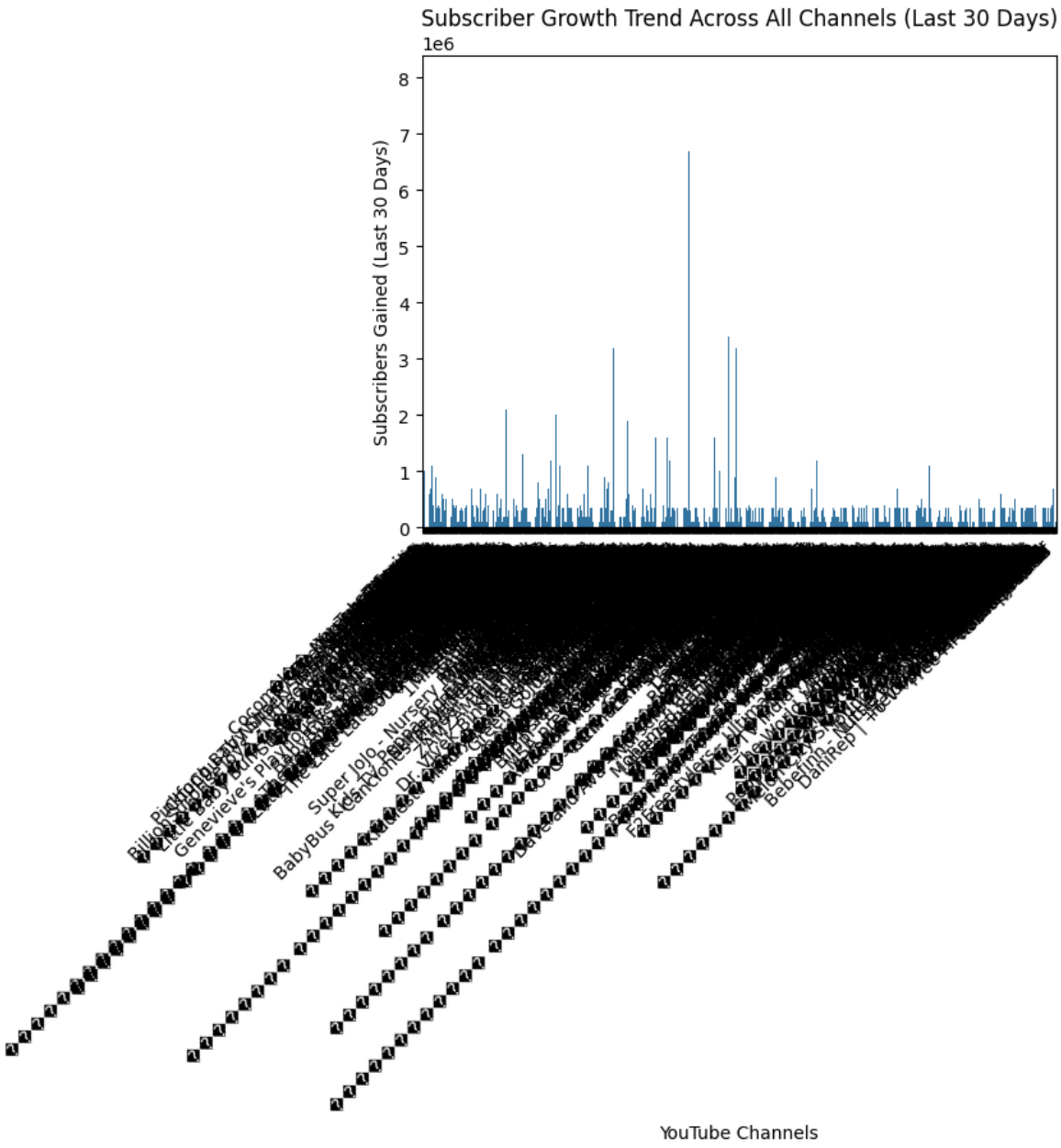
```
Trailers                    0.0   45200.0             3.619000e+05   36190
0.0
Travel & Events          7800.0    7800.0             1.240000e+05   12400
0.0
other                       0.0  850900.0             9.747300e+05   18640
0.0
```

category	std	min	max
Autos & Vehicles	3.621718e+05	772700.0	1400000.0
Comedy	1.104281e+06	0.0	5000000.0
Education	1.326222e+06	0.0	7900000.0
Entertainment	1.192167e+06	0.0	8100000.0
Film & Animation	1.614060e+06	0.0	9200000.0
Gaming	5.103550e+05	0.0	4300000.0
Howto & Style	3.514417e+05	0.0	2000000.0
Movies	6.430429e+05	0.0	909400.0
Music	9.068679e+05	0.0	9000000.0
News & Politics	5.610108e+05	0.0	1800000.0
Nonprofits & Activism	2.938736e+05	182600.0	598200.0
People & Blogs	9.071410e+05	0.0	5500000.0
Pets & Animals	1.683100e+06	0.0	3000000.0
Science & Technology	1.834049e+05	0.0	670800.0
Shows	2.589495e+06	226100.0	7300000.0
Sports	1.109512e+06	40900.0	2900000.0
Trailers	5.118039e+05	0.0	723800.0
Travel & Events	NaN	124000.0	124000.0
other	2.310611e+06	0.0	13600000.0

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

In [14]:
```
print(df['subscribers_for_last_30_days'].describe())
sns.barplot(x='Youtuber',y='subscribers_for_last_30_days',data=df)
plt.xticks(rotation=45, ha='right')
plt.xlabel('YouTube Channels')
plt.ylabel('Subscribers Gained (Last 30 Days)')
plt.title('Subscriber Growth Trend Across All Channels (Last 30 Days)')
plt.show()
```

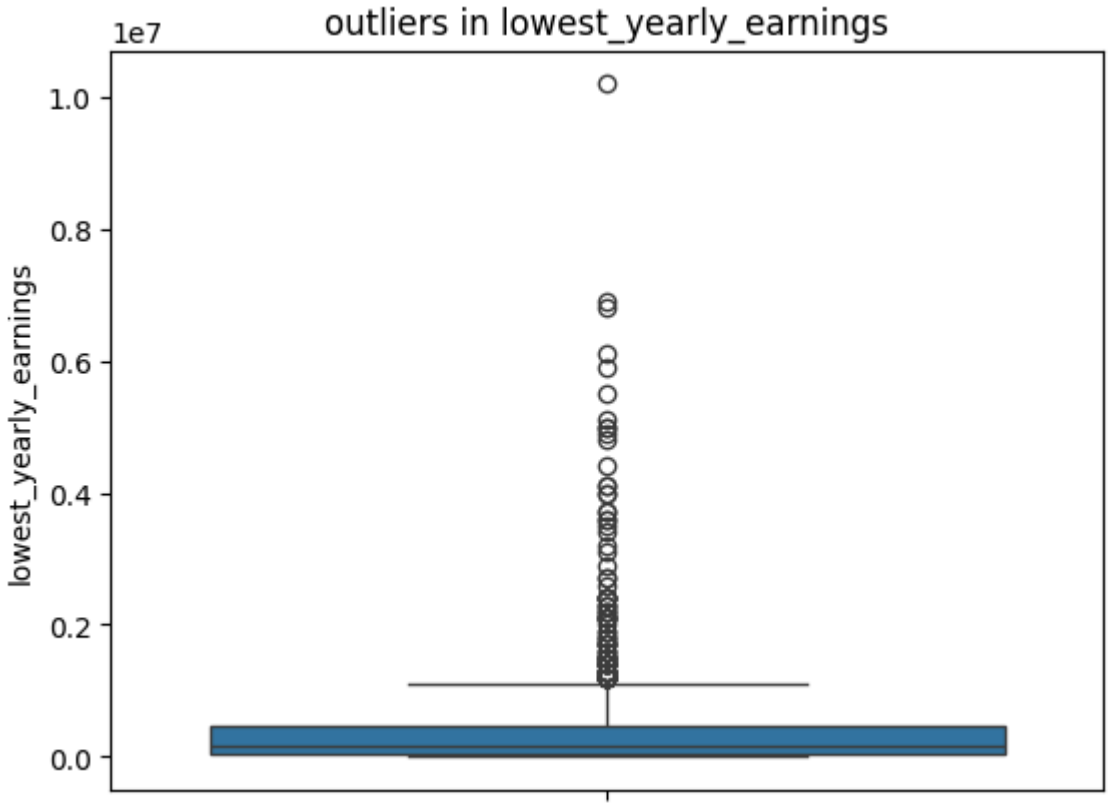
```
count    9.710000e+02
mean     3.570468e+05
std      5.059095e+05
min      1.000000e+00
25%      1.000000e+05
50%      3.570468e+05
75%      3.570468e+05
max      8.000000e+06
Name: subscribers_for_last_30_days, dtype: float64
```

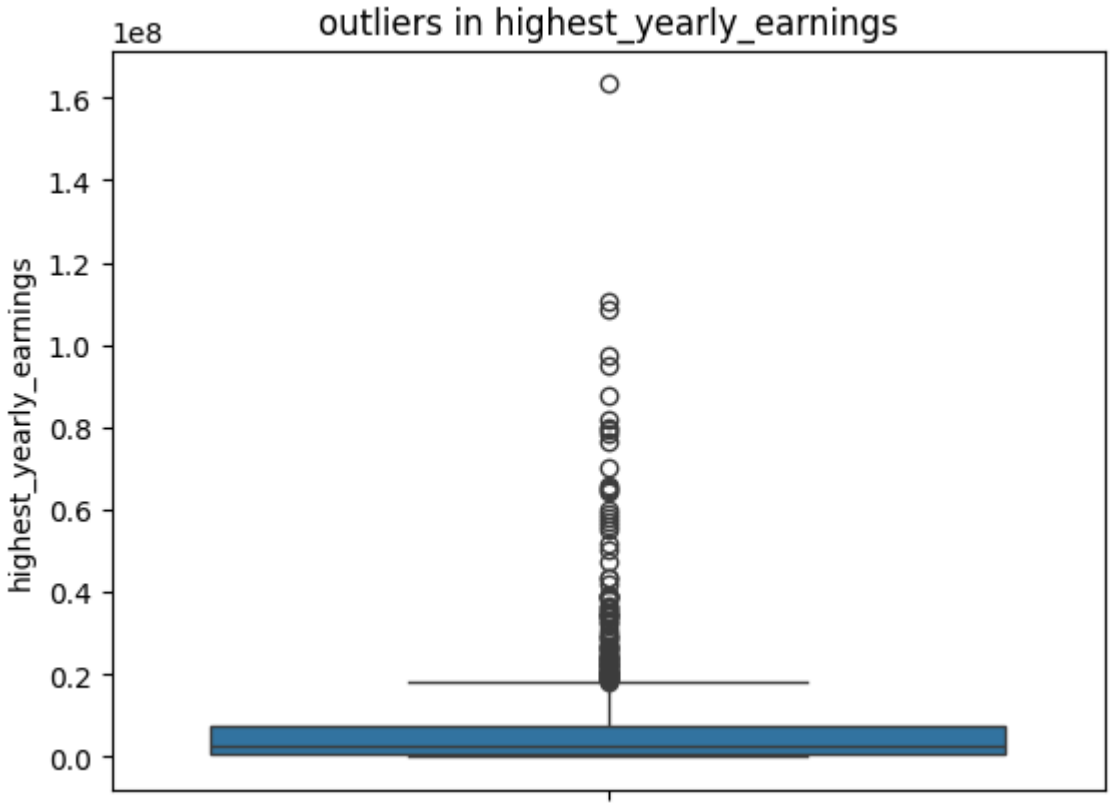


9. Are there any outliers in terms of yearly earnings from YouTube channels?

In [15]:
```
sns.boxplot(y='lowest_yearly_earnings',data=df)
plt.title("outliers in lowest_yearly_earnings")
plt.show()

sns.boxplot(y='highest_yearly_earnings',data=df)
plt.title("outliers in highest_yearly_earnings")
plt.show()

q1_low = df['lowest_yearly_earnings'].quantile(0.25)
q3_low = df['lowest_yearly_earnings'].quantile(0.75)
iqr_low = q3_low - q1_low

q1_high = df['highest_yearly_earnings'].quantile(0.25)
q3_high = df['highest_yearly_earnings'].quantile(0.75)
iqr_high = q3_high - q1_high

outliers_low = df[(df['lowest_yearly_earnings']<q1_low-1.5*iqr_low) | (df
outliers_high = df[(df['highest_yearly_earnings']<q1_high-1.5*iqr_high) |
```

```
print(f"Outliers in lowest yearly earnings: {len(outliers_low)}")
print(f"Outliers in highest yearly earnings: {len(outliers_high)}")
```






```
Outliers in lowest yearly earnings: 95
Outliers in highest yearly earnings: 95
```

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

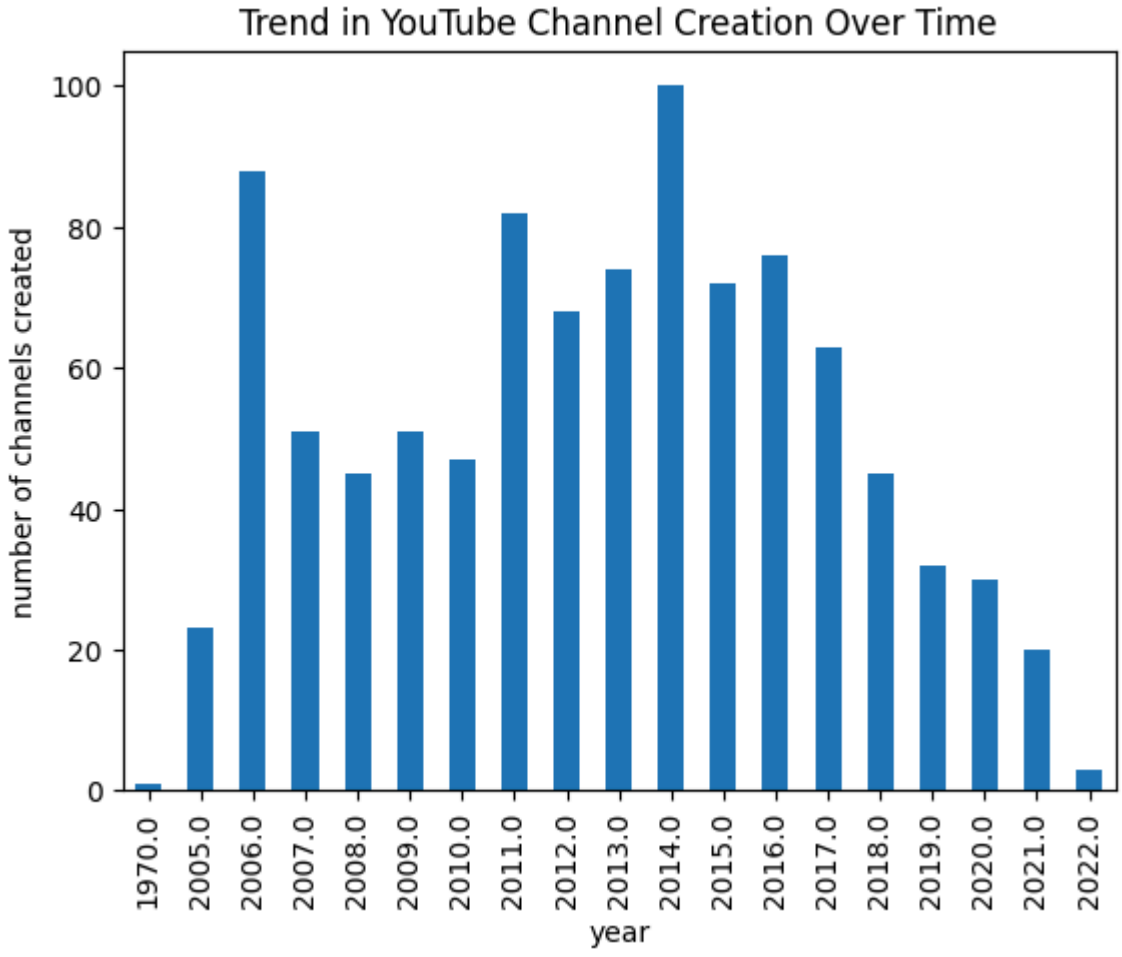
In [16]:
```
print(df['created_date'].describe())

grouped_data_year = df.groupby(['created_year','created_month'])['Youtube
print(grouped_data_year)

yearly_counts = df.groupby('created_year')['Youtuber'].count()
yearly_counts.plot(kind = 'bar')
plt.xlabel('year')
plt.ylabel('number of channels created')
plt.title('Trend in YouTube Channel Creation Over Time')
plt.show()
```

```
count    971.000000
mean      15.675592
std        8.769136
min        1.000000
25%        8.000000
50%       16.000000
75%       23.000000
max       31.000000
Name: created_date, dtype: float64
created_year  created_month
1970.0        Jan              1
2005.0        Dec              3
              Jun              2
              Nov              8
              Oct              6
              Sep              4
2006.0        Apr              6
              Aug              7
              Dec              7
              Feb              6
              Jan             10
              Jul              5
              Jun              4
              Mar             13
              May             10
              Nov              7
              Oct              3
              Sep             10
2007.0        Apr              2
              Aug              1
              Dec              1
              Feb              8
              Jan             11
              Jul              4
              Jun              2
              Mar              2
              May              6
              Nov              4
              Oct              5
              Sep              5
2008.0        Apr              8
              Aug              5
              Feb              2
              Jan              5
              Jul              4
              Jun              9
              Mar              3
              May              3
              Nov              4
              Sep              2
2009.0        Apr              1
              Aug              7
              Dec              3
              Feb              3
              Jul              5
              Jun              3
              Mar              4
              May              6
              Nov              3
              Oct              5
```

	Sep	11
2010.0	Apr	5
	Aug	3
	Dec	8
	Feb	3
	Jan	3
	Jul	3
	Jun	2
	Mar	2
	May	5
	Nov	3
	Oct	4
	Sep	6
2011.0	Apr	5
	Aug	9
	Dec	5
	Feb	7
	Jan	10
	Jul	3
	Jun	5
	Mar	4
	May	10
	Nov	8
	Oct	9
	Sep	7
2012.0	Apr	4
	Aug	3
	Dec	9
	Feb	6
	Jan	6
	Jul	9
	Jun	5
	Mar	7
	May	2
	Nov	9
	Oct	3
	Sep	5
2013.0	Apr	4
	Aug	9
	Dec	6
	Feb	4
	Jan	11
	Jul	3
	Jun	3
	Mar	10
	May	5
	Nov	6
	Oct	6
	Sep	7
2014.0	Apr	3
	Aug	12
	Dec	8
	Feb	5
	Jan	12
	Jul	13
	Jun	8
	Mar	10
	May	8
	Nov	3
	Oct	8

	Sep	10
2015.0	Apr	5
	Aug	6
	Dec	5
	Feb	2
	Jan	7
	Jul	5
	Jun	4
	Mar	8
	May	8
	Nov	7
	Oct	8
	Sep	7
2016.0	Apr	6
	Aug	6
	Dec	3
	Feb	3
	Jan	11
	Jul	11
	Jun	7
	Mar	6
	May	7
	Nov	4
	Oct	7
	Sep	5
2017.0	Apr	6
	Aug	4
	Dec	3
	Feb	6
	Jan	7
	Jul	5
	Jun	6
	Mar	3
	May	4
	Nov	5
	Oct	6
	Sep	8
2018.0	Apr	7
	Aug	2
	Dec	3
	Feb	3
	Jan	6
	Jul	4
	Jun	3
	Mar	2
	May	3
	Nov	8
	Oct	2
	Sep	2
2019.0	Apr	2
	Dec	3
	Feb	2
	Jan	4
	Jul	4
	Jun	2
	Mar	3
	May	4
	Nov	5
	Oct	2
	Sep	1

```
2020.0        Apr             2
              Aug             3
              Dec             3
              Feb             1
              Jan             2
              Jul             7
              Jun             1
              Mar             1
              May             2
              Nov             2
              Oct             3
              Sep             3
2021.0        Apr             2
              Aug             2
              Feb             4
              Jul             1
              Jun             2
              Mar             5
              May             1
              Nov             1
              Sep             2
2022.0        Jun             2
              Mar             1
Name: Youtuber, dtype: int64
```



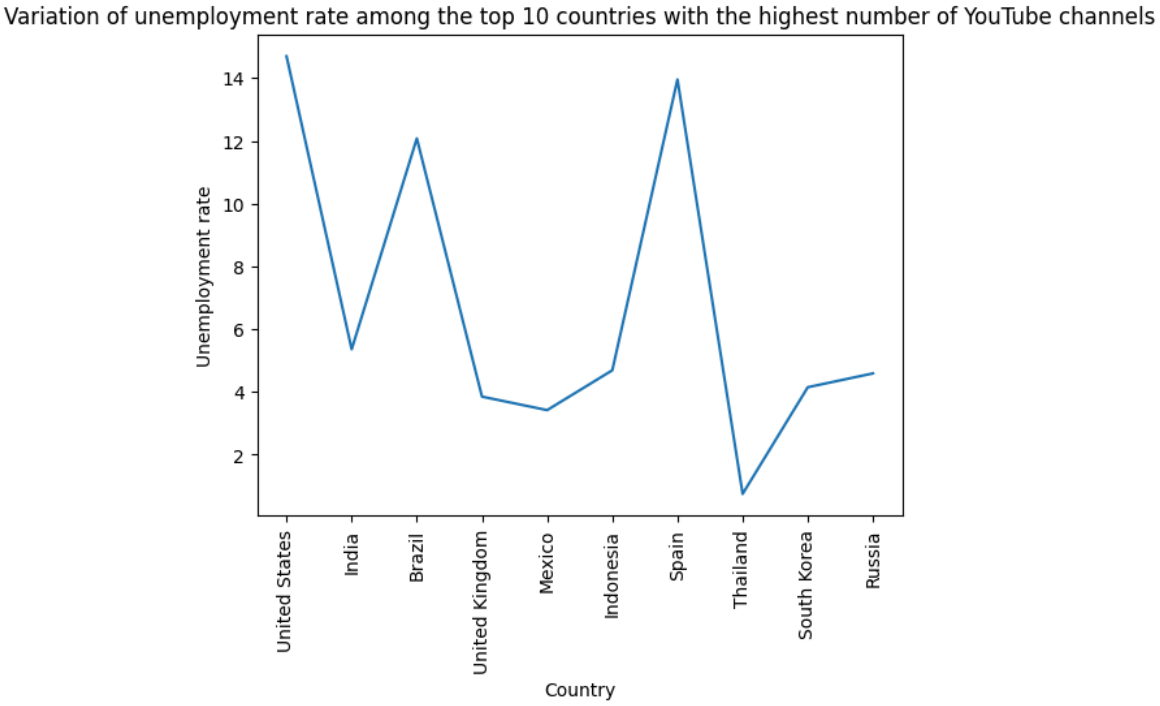
11. Is there a relationship between gross tertiary education enrollment and the number of YouTube channels in a country?

In [27]:
```
data_country = df.groupby(['Country']).agg({'Youtuber': 'count','Gross te
relation = data_country['Youtuber'].corr(data_country['Gross tertiary edu
print(relation)

if relation > 0.9:
    print("Very high positive correlation")
elif relation > 0.7:
    print("High positive correlation")
elif relation > 0.5:
    print("Moderate high positive correlation")
elif relation > 0.3:
    print("Low positive correlation")
elif relation > 0.0:
    print("negligible correlation")
elif relation == 0:
    print("No correlation")
elif relation > -0.3:
    print("Negligible correlation")
elif relation > -0.5:
    print("Low negative correlation")
elif relation > -0.7:
    print("Moderate negative correlation")
elif relation > -0.9:
    print("High negative correlation")
else:
    print("Very high negative correlation.")
```

```
0.11226801275185411
negligible correlation
```

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

In [28]:
```
top10 = data_country.sort_values('Youtuber', ascending = False).head(10)
plt.plot(top10['Country'], top10['Unemployment rate'])
plt.xlabel('Country')
plt.ylabel('Unemployment rate')
plt.xticks(rotation=90)
plt.title('Variation of unemployment rate among the top 10 countries with
plt.show()
```



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

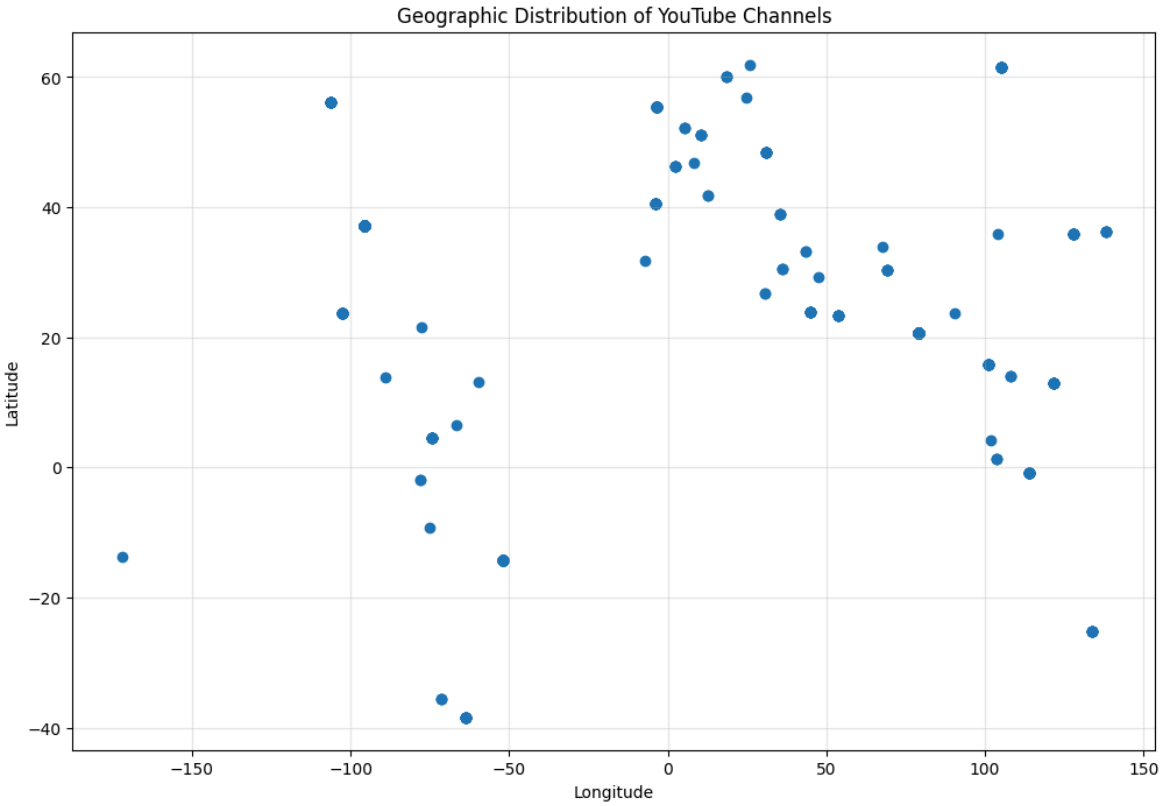
In [23]:
```
df_country_analysis['Urban_ratio'] = (df_country_analysis['Urban_populati

country_avg_urban = df_country_analysis.groupby('Country')['Urban_ratio']

overall_avg = country_avg_urban.mean()
print(country_avg_urban)
print("Average urban population percentage in countries with YouTube chan
```

```
Country
Afghanistan              25.753999
Argentina                91.991001
Australia                84.779334
Bangladesh               36.451564
Barbados                 31.157913
Brazil                   86.207256
Canada                   82.797626
Chile                    87.643002
China                    60.308000
Colombia                 81.104000
Cuba                     77.108996
Ecuador                  63.985998
Egypt                    42.730000
El Salvador              72.746005
Finland                  85.446009
France                   80.709000
Germany                  77.376001
India                    34.472000
Indonesia                56.072364
Iraq                     70.677999
Italy                    70.736000
Japan                    91.725869
Jordan                   91.203000
Kuwait                  100.000000
Latvia                   68.222005
Malaysia                 75.432168
Mexico                   81.440824
Morocco                  62.245130
Netherlands              91.875998
Pakistan                 36.907000
Peru                     78.099001
Philippines              47.149000
Russia                   74.587000
Samoa                    17.573800
Saudi Arabia             84.065000
Singapore               100.000000
South Korea              81.430001
Spain                    80.565001
Sweden                   87.707999
Switzerland              73.849004
Thailand                 50.692000
Turkey                   75.630000
Ukraine                  69.473001
United Arab Emirates     86.788996
United Kingdom           83.651999
United States            82.459000
Venezuela                88.240002
Vietnam                  36.628000
india                    34.472000
Name: Urban_ratio, dtype: float64
Average urban population percentage in countries with YouTube channels: 7
0.04830333449618
```

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

In [42]:
```
plt.figure(figsize=(12, 8))
plt.scatter(df['Longitude'], df['Latitude'])
```

```
plt.xlabel('Longitude')
plt.ylabel('Latitude')
plt.title('Geographic Distribution of YouTube Channels')
plt.grid(True, alpha=0.3)
plt.show()

x = df['Longitude'].corr(df['Latitude'])
print('Relation between latitudes and longitudes: ',x)
```




```
Relation between latitudes and longitudes:  -0.1996084848408054
```

This concludes that there is negligible correlation in distribution of you tube channels based on latitude and longitude.

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

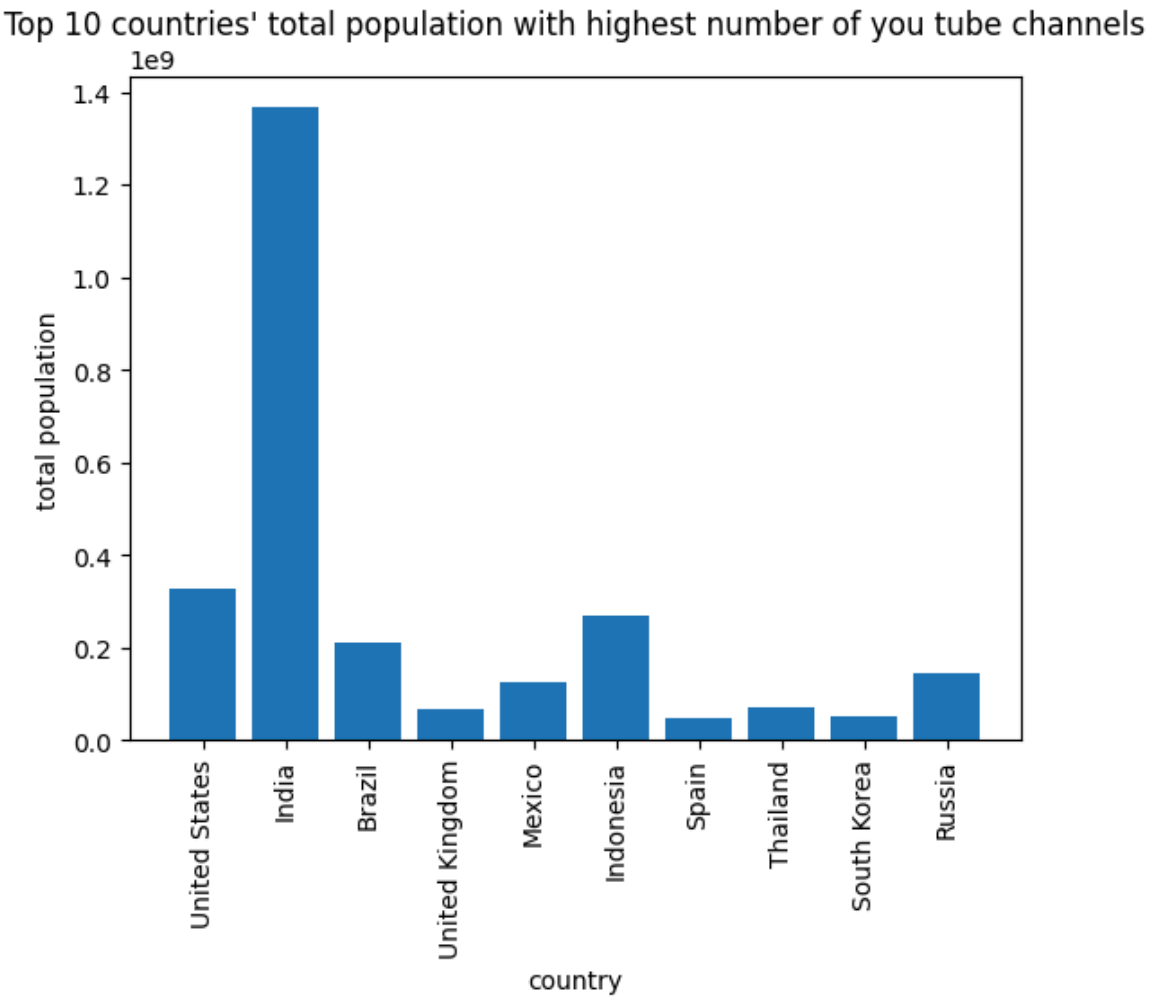
In [25]:
```
correlation_subscribers_population = df['subscribers'].corr(df['Populatio
print(correlation_subscribers_population)

if correlation_subscribers_population > 0.9:
    print("Very high positive correlation")
elif correlation_subscribers_population > 0.7:
    print("High positive correlation")
elif correlation_subscribers_population > 0.5:
    print("Moderate high positive correlation")
elif correlation_subscribers_population > 0.3:
    print("Low positive correlation")
elif correlation_subscribers_population > 0.0:
    print("negligible correlation")
elif correlation_subscribers_population == 0:
    print("No correlation")
elif correlation_subscribers_population > -0.3:
    print("Negligible correlation")
```

```
elif correlation_subscribers_population > -0.5:
    print("Low negative correlation")
elif correlation_subscribers_population > -0.7:
    print("Moderate negative correlation")
elif correlation_subscribers_population > -0.9:
    print("High negative correlation")
else:
    print("Very high negative correlation.")
```

```
0.0852845885697223
negligible correlation
```

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

In [31]:
```
plt.bar(top10['Country'], top10['Population'])
plt.xlabel('country')
plt.ylabel('total population')
plt.title("Top 10 countries' total population with highest number of you
plt.xticks(rotation = 90)
plt.show()
```



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

In [32]:
```
result_correlation = df['subscribers_for_last_30_days'].corr(df['Unemploy
print(result_correlation)

if result_correlation > 0.9:
    print("Very high positive correlation")
elif result_correlation > 0.7:
    print("High positive correlation")
elif result_correlation > 0.5:
    print("Moderate high positive correlation")
elif result_correlation > 0.3:
    print("Low positive correlation")
elif result_correlation > 0.0:
    print("negligible correlation")
elif result_correlation == 0:
    print("No correlation")
elif result_correlation > -0.3:
    print("Negligible correlation")
elif result_correlation > -0.5:
    print("Low negative correlation")
elif result_correlation > -0.7:
    print("Moderate negative correlation")
elif result_correlation > -0.9:
    print("High negative correlation")
else:
    print("Very high negative correlation.")
```

```
-0.02045636616100732
Negligible correlation
```

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

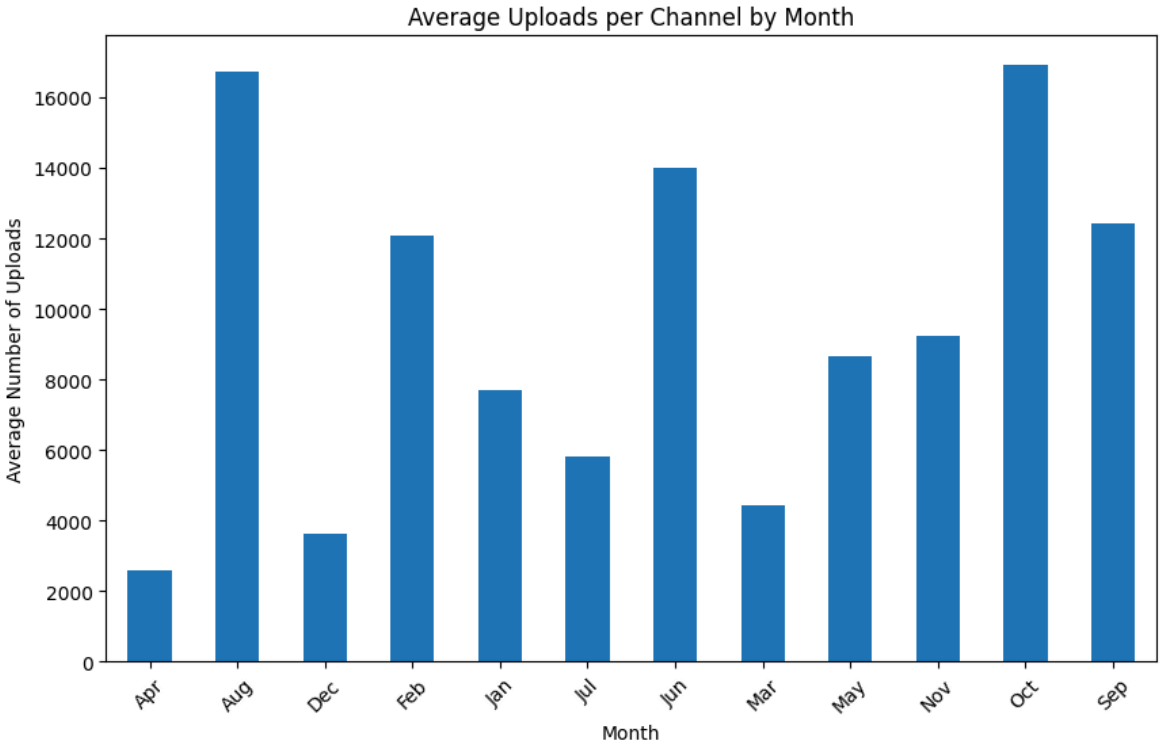
In [33]:
```
data_channel_type = df.groupby(['channel_type'])['video_views_for_the_las
print(data_channel_type)
```

```
                count          mean           std         min          25%
\
channel_type
Animals           3.0  7.073477e+08  6.817585e+08   2989000.0  379021500.0
Autos             4.0  1.766301e+08  2.039534e+08        53.0       1871.0
Comedy           51.0  1.920896e+08  2.984321e+08         2.0   19989500.0
Education        50.0  2.007373e+08  3.136122e+08         1.0   52362000.0
Entertainment   304.0  2.124186e+08  5.116646e+08         1.0   19954500.0
Film             42.0  1.414736e+08  1.691649e+08         2.0   28398750.0
Games           100.0  1.170081e+08  2.004310e+08         2.0   20896250.0
Howto            37.0  5.865822e+07  9.878407e+07    336291.0   18045000.0
Music           215.0  1.788310e+08  4.672408e+08         1.0   44263000.0
News             30.0  1.809493e+08  1.362948e+08       998.0   69759500.0
Nonprofit         2.0  9.759050e+07  7.347193e+07  45638000.0   71614250.0
People          102.0  1.792002e+08  4.141207e+08         1.0     939647.0
Sports           14.0  2.030792e+08  2.603170e+08         1.0   16664000.0
Tech             17.0  5.508947e+07  4.557638e+07         5.0   18518000.0

                      50%           75%           max
channel_type
Animals        7.550540e+08  1.059527e+09  1.364000e+09
Autos          1.766307e+08  3.532590e+08  3.532590e+08
Comedy         4.986100e+07  2.556615e+08  1.245000e+09
Education      1.112980e+08  2.054065e+08  1.975000e+09
Entertainment  6.125600e+07  1.807348e+08  6.589000e+09
Film           8.587800e+07  1.795155e+08  7.577890e+08
Games          5.613150e+07  1.783529e+08  1.463000e+09
Howto          2.738200e+07  4.386800e+07  5.027790e+08
Music          9.728400e+07  1.773395e+08  6.148000e+09
News           1.707415e+08  2.592932e+08  4.614720e+08
Nonprofit      9.759050e+07  1.235668e+08  1.495430e+08
People         4.291050e+07  1.783529e+08  3.404000e+09
Sports         7.894600e+07  3.287735e+08  7.146140e+08
Tech           4.648400e+07  8.822400e+07  1.676970e+08
```

19. Are there any seasonal trends in the number of videos uploaded by YouTube channels?

In [45]:
```
monthly_avg_uploads = df.groupby('created_month')['uploads'].mean()

print(monthly_avg_uploads)

plt.figure(figsize=(10,6))
monthly_avg_uploads.plot(kind='bar')
plt.title('Average Uploads per Channel by Month')
plt.xlabel('Month')
plt.ylabel('Average Number of Uploads')
plt.xticks(rotation=45)
plt.show()
```

```
created_month
Apr     2603.764706
Aug    16717.012658
Dec     3628.228571
Feb    12093.584615
Jan     7711.018868
Jul     5799.220930
Jun    13992.000000
Mar     4442.452381
May     8670.690476
Nov     9247.839080
Oct    16905.233766
Sep    12438.578947
Name: uploads, dtype: float64
```



20. What is the average number of subscribers gained per month since the creation of YouTube channels till now?

In [38]:
```
current_year = 2025
current_month = 6

month_num = {'Jan':1 , 'Feb':2, 'Mar':3, 'Apr' : 4, 'May': 5, "Jun": 6, '
df['months_age'] = (current_year - df['created_year']) * 12 + (current_mo

df['subscribers_per_month'] = df['subscribers']/df['months_age']

avg_subscribers = df['subscribers_per_month'].mean()
print("Average subscribers gained per month since creation: ",avg_subscri
```

```
Average subscribers gained per month since creation:  173797.09372864303
```