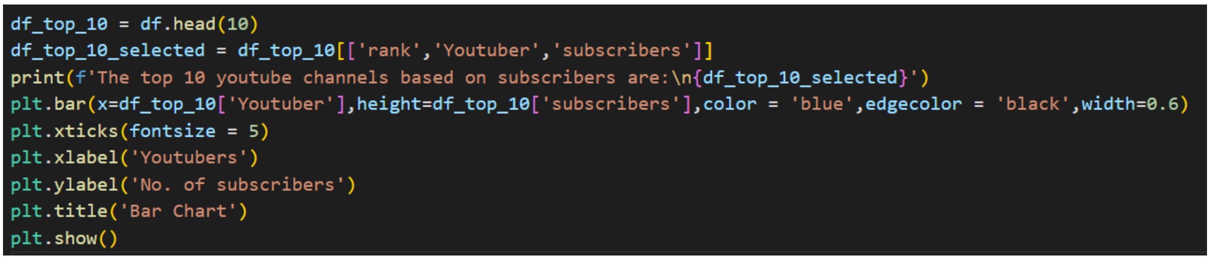
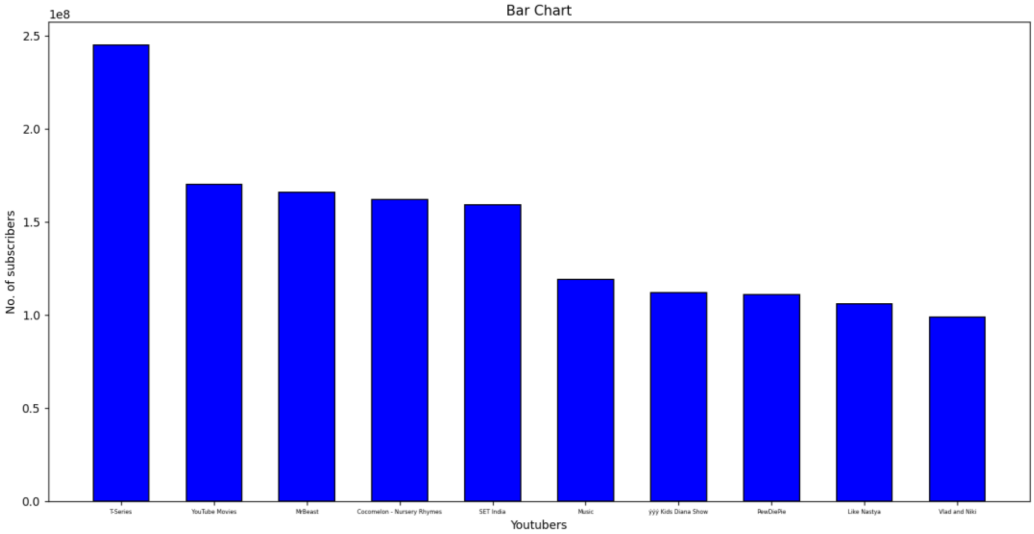
CASE PROJECT

Media and Technology

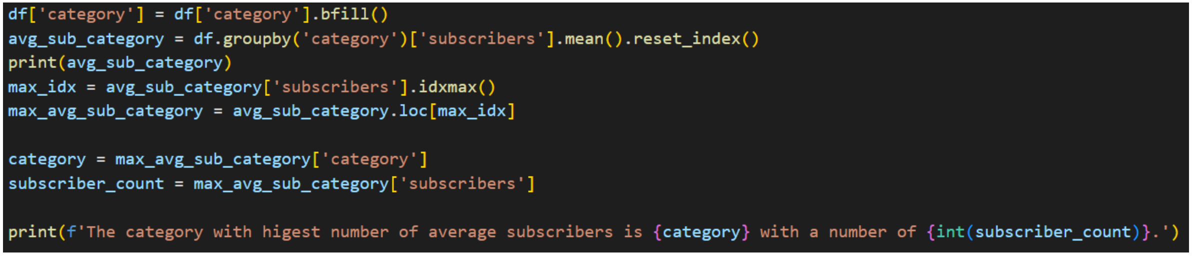
1) What are the top 10 YouTube channels based on the number of subscribers?



```
df_top_10 = df.head(10)
df_top_10_selected = df_top_10[['rank','Youtuber','subscribers']]
print(f'The top 10 youtube channels based on subscribers are:\n{df_top_10_selected}')
plt.bar(x=df_top_10['Youtuber'],height=df_top_10['subscribers'],color = 'blue',edgecolor = 'black',width=0.6)
plt.xticks(fontsize = 5)
plt.xlabel('Youtubers')
plt.ylabel('No. of subscribers')
plt.title('Bar Chart')
plt.show()
```

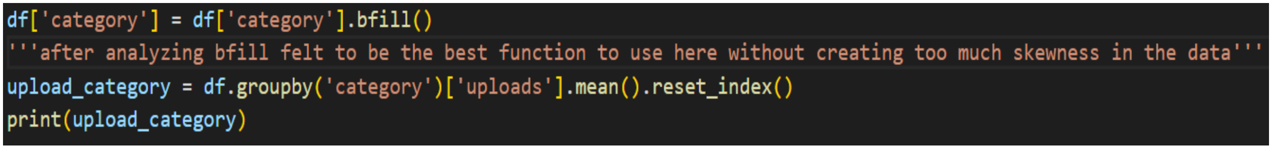


2) Which category has the highest average number of subscribers?



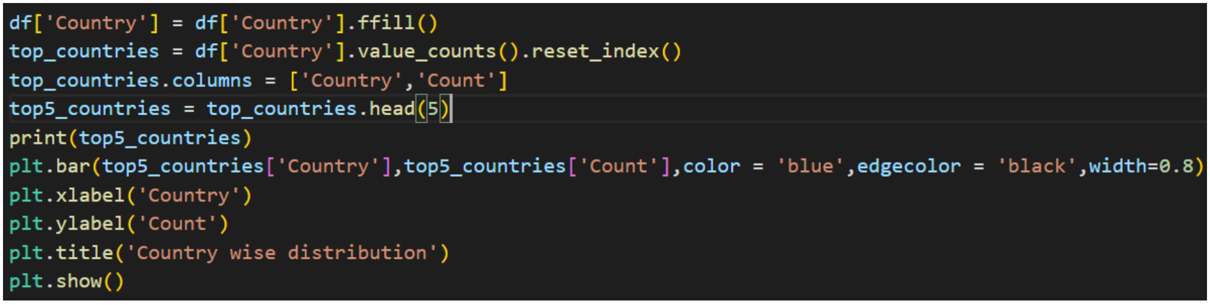
```
df['category'] = df['category'].bfill()
avg_sub_category = df.groupby('category')['subscribers'].mean().reset_index()
print(avg_sub_category)
max_idx = avg_sub_category['subscribers'].idxmax()
max_avg_sub_category = avg_sub_category.loc[max_idx]

category = max_avg_sub_category['category']
subscriber_count = max_avg_sub_category['subscribers']

print(f'The category with higest number of average subscribers is {category} with a number of {int(subscriber_count)}.')
```

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

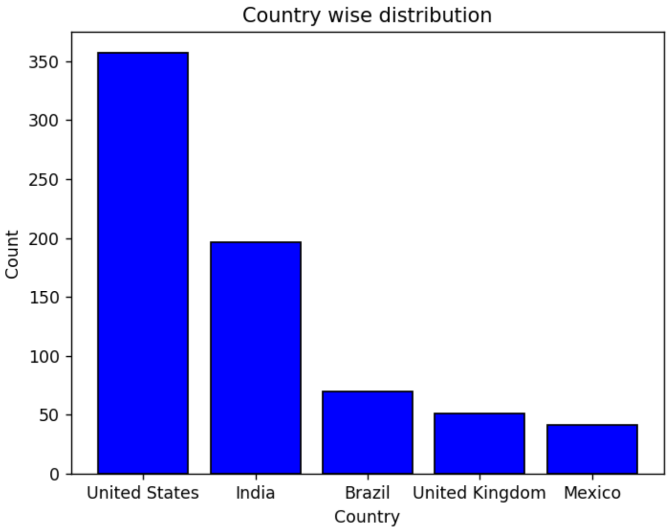


```
df['category'] = df['category'].bfill()
'''after analyzing bfill felt to be the best function to use here without creating too much skewness in the data'''
upload_category = df.groupby('category')['uploads'].mean().reset_index()
print(upload_category)
```

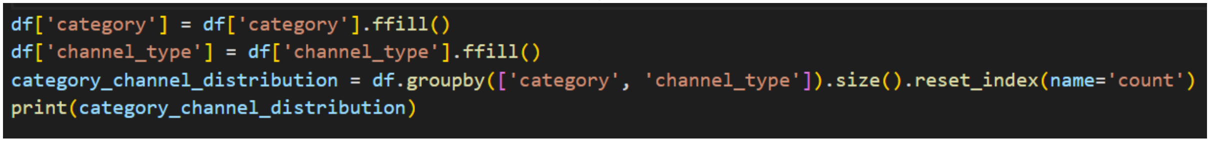
4) What are the top 5 countries with the most YouTube channels?



```
df['Country'] = df['Country'].ffill()
top_countries = df['Country'].value_counts().reset_index()
top_countries.columns = ['Country','Count']
top5_countries = top_countries.head(5)
print(top5_countries)
plt.bar(top5_countries['Country'],top5_countries['Count'],color = 'blue',edgecolor = 'black',width=0.8)
plt.xlabel('Country')
plt.ylabel('Count')
plt.title('Country wise distribution')
plt.show()
```

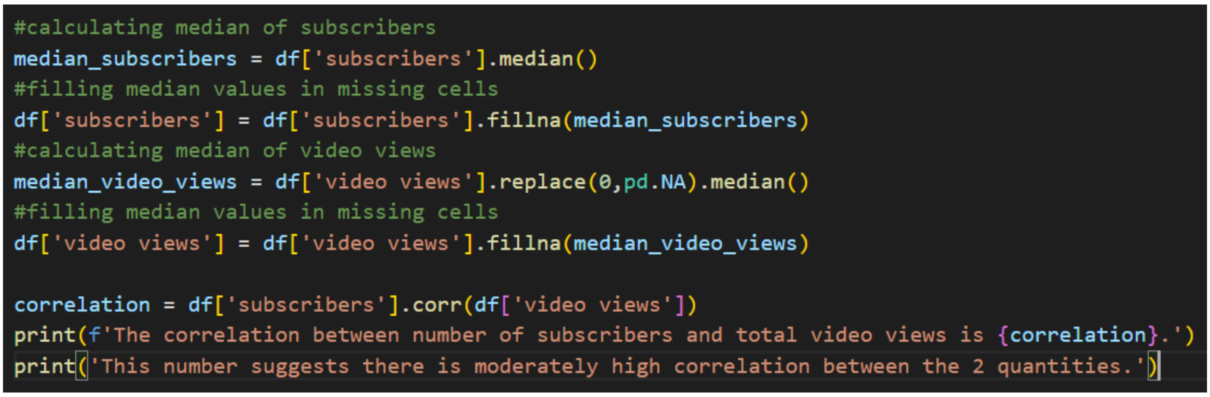


5) What is the distribution of channel types across different categories?



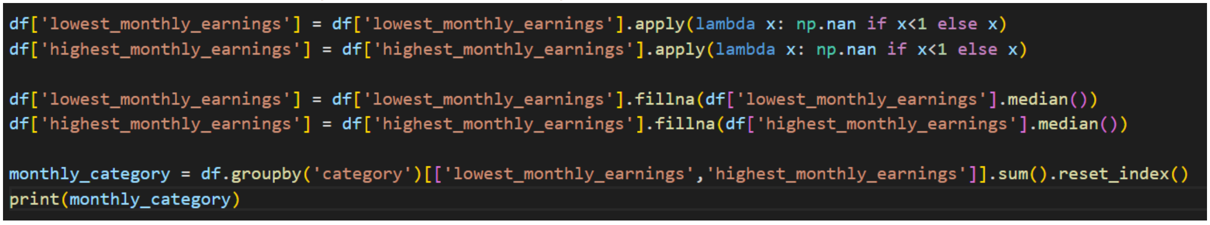
```
df['category'] = df['category'].ffill()
df['channel_type'] = df['channel_type'].ffill()
category_channel_distribution = df.groupby(['category', 'channel_type']).size().reset_index(name='count')
print(category_channel_distribution)
```

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



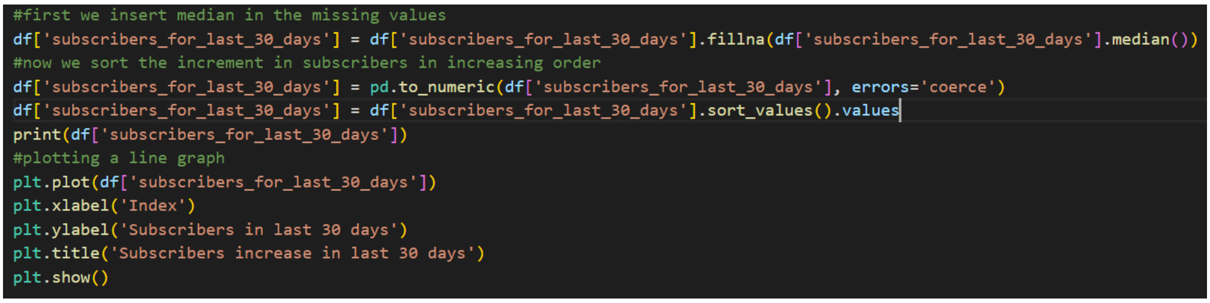
```
#calculating median of subscribers
median_subscribers = df['subscribers'].median()
#filling median values in missing cells
df['subscribers'] = df['subscribers'].fillna(median_subscribers)
#calculating median of video views
median_video_views = df['video views'].replace(0,pd.NA).median()
#filling median values in missing cells
df['video views'] = df['video views'].fillna(median_video_views)

correlation = df['subscribers'].corr(df['video views'])
print(f'The correlation between number of subscribers and total video views is {correlation}.')
print('This number suggests there is moderately high correlation between the 2 quantities.')
```

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

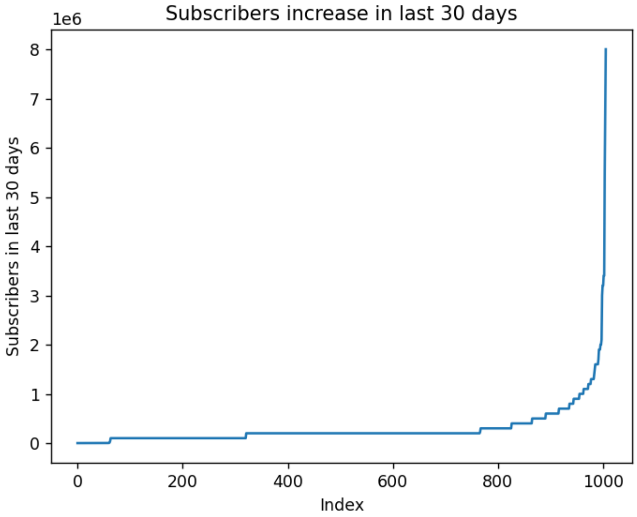


```
df['lowest_monthly_earnings'] = df['lowest_monthly_earnings'].apply(lambda x: np.nan if x<1 else x)
df['highest_monthly_earnings'] = df['highest_monthly_earnings'].apply(lambda x: np.nan if x<1 else x)

df['lowest_monthly_earnings'] = df['lowest_monthly_earnings'].fillna(df['lowest_monthly_earnings'].median())
df['highest_monthly_earnings'] = df['highest_monthly_earnings'].fillna(df['highest_monthly_earnings'].median())

monthly_category = df.groupby('category')[['lowest_monthly_earnings','highest_monthly_earnings']].sum().reset_index()
print(monthly_category)
```

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

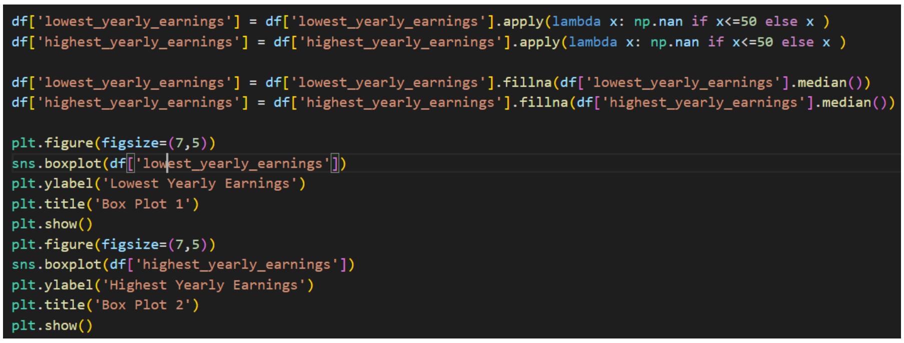


```
#first we insert median in the missing values
df['subscribers_for_last_30_days'] = df['subscribers_for_last_30_days'].fillna(df['subscribers_for_last_30_days'].median())
#now we sort the increment in subscribers in increasing order
df['subscribers_for_last_30_days'] = pd.to_numeric(df['subscribers_for_last_30_days'], errors='coerce')
df['subscribers_for_last_30_days'] = df['subscribers_for_last_30_days'].sort_values().values
print(df['subscribers_for_last_30_days'])
#plotting a line graph
plt.plot(df['subscribers_for_last_30_days'])
plt.xlabel('Index')
plt.ylabel('Subscribers in last 30 days')
plt.title('Subscribers increase in last 30 days')
plt.show()
```

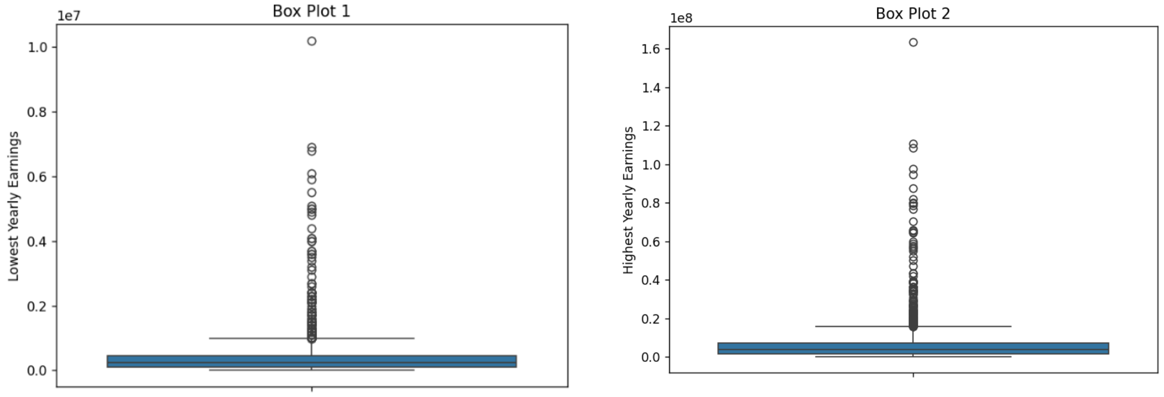


As we can see every Youtube channel showed an increase in the number of subscribers in the last 30 days. Although most of them show an increase and their values are around the central value but there were some which grew immensely in this period which has been shown towards the end of the graph.

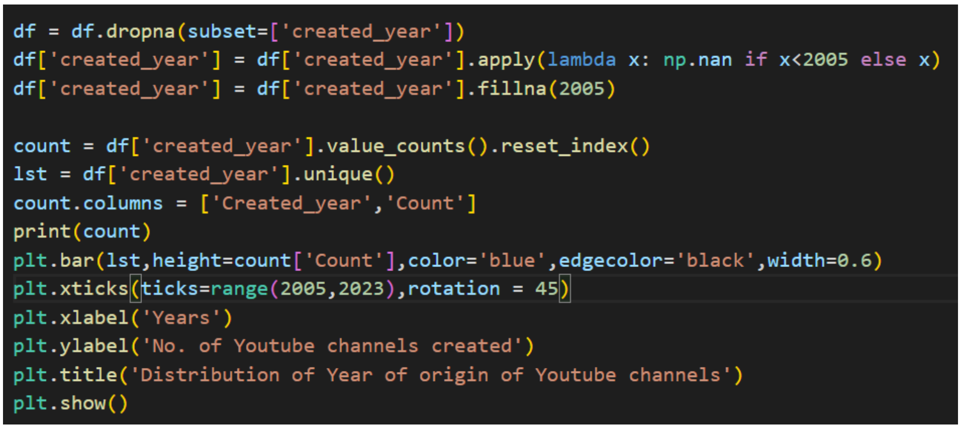
9) Are there any outliers in terms of yearly earnings from YouTube channels?



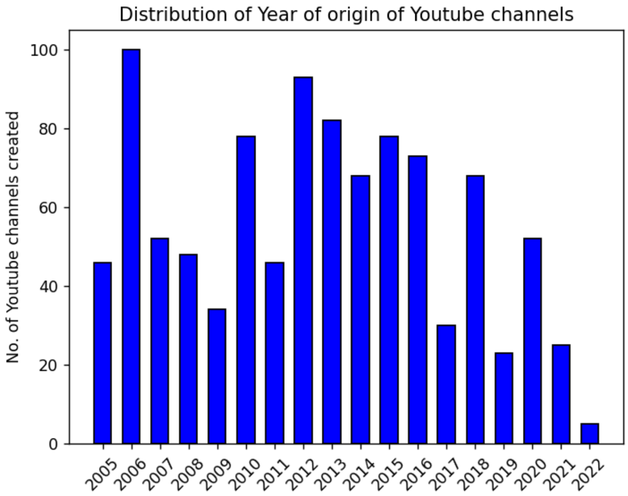
```
df['lowest_yearly_earnings'] = df['lowest_yearly_earnings'].apply(lambda x: np.nan if x<=50 else x )
df['highest_yearly_earnings'] = df['highest_yearly_earnings'].apply(lambda x: np.nan if x<=50 else x )

df['lowest_yearly_earnings'] = df['lowest_yearly_earnings'].fillna(df['lowest_yearly_earnings'].median())
df['highest_yearly_earnings'] = df['highest_yearly_earnings'].fillna(df['highest_yearly_earnings'].median())

plt.figure(figsize=(7,5))
sns.boxplot(df['lowest_yearly_earnings'])
plt.ylabel('Lowest Yearly Earnings')
plt.title('Box Plot 1')
plt.show()
plt.figure(figsize=(7,5))
sns.boxplot(df['highest_yearly_earnings'])
plt.ylabel('Highest Yearly Earnings')
plt.title('Box Plot 2')
plt.show()
```



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

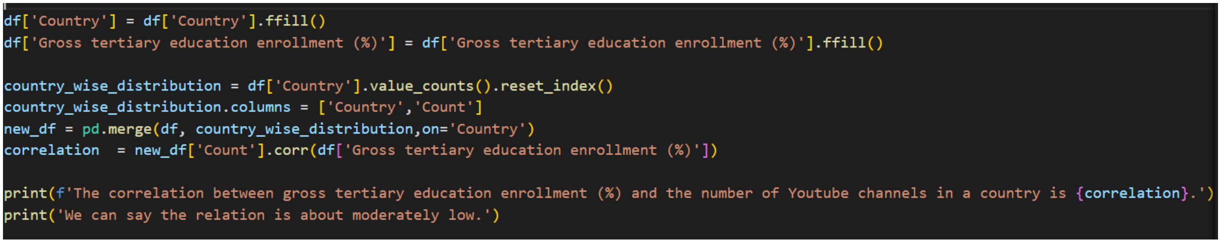


```
df = df.dropna(subset=['created_year'])
df['created_year'] = df['created_year'].apply(lambda x: np.nan if x<2005 else x)
df['created_year'] = df['created_year'].fillna(2005)

count = df['created_year'].value_counts().reset_index()
lst = df['created_year'].unique()
count.columns = ['Created_year','Count']
print(count)
plt.bar(lst,height=count['Count'],color='blue',edgecolor='black',width=0.6)
plt.xticks(ticks=range(2005,2023),rotation = 45)
plt.xlabel('Years')
plt.ylabel('No. of Youtube channels created')
plt.title('Distribution of Year of origin of Youtube channels')
plt.show()
```



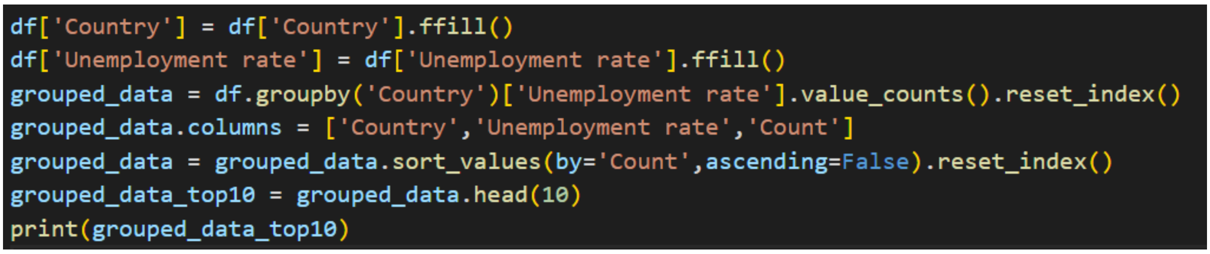
As we can see there is no clear trend in the data as such but we can point out that after 2012 the number of new Youtube channels originating can be seen on a decline. The year where the maximum number of new Youtube channels were created is 2006 followed by 2012.

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



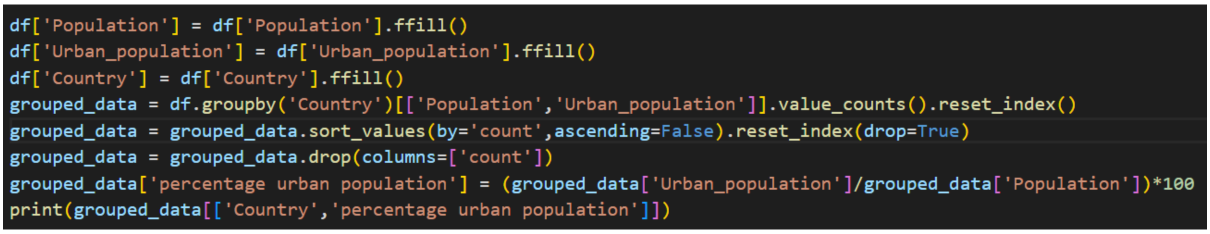
```
df['Country'] = df['Country'].ffill()
df['Gross tertiary education enrollment (%)'] = df['Gross tertiary education enrollment (%)'].ffill()

country_wise_distribution = df['Country'].value_counts().reset_index()
country_wise_distribution.columns = ['Country','Count']
new_df = pd.merge(df, country_wise_distribution,on='Country')
correlation  = new_df['Count'].corr(df['Gross tertiary education enrollment (%)'])

print(f'The correlation between gross tertiary education enrollment (%) and the number of Youtube channels in a country is {correlation}.')
print('We can say the relation is about moderately low.')
```

12) How does unemployment vary among the top 10 countries with the most YouTube channels?



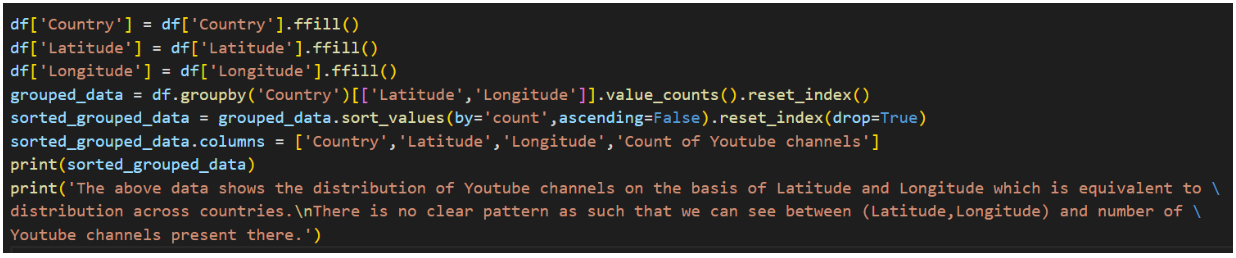
```
df['Country'] = df['Country'].ffill()
df['Unemployment rate'] = df['Unemployment rate'].ffill()
grouped_data = df.groupby('Country')['Unemployment rate'].value_counts().reset_index()
grouped_data.columns = ['Country','Unemployment rate','Count']
grouped_data = grouped_data.sort_values(by='Count',ascending=False).reset_index()
grouped_data_top10 = grouped_data.head(10)
print(grouped_data_top10)
```

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



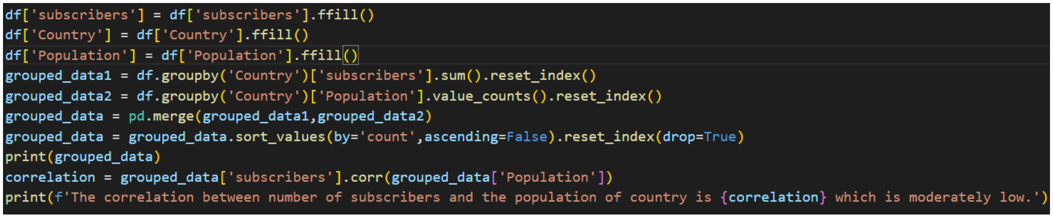
```
df['Population'] = df['Population'].ffill()
df['Urban_population'] = df['Urban_population'].ffill()
df['Country'] = df['Country'].ffill()
grouped_data = df.groupby('Country')[['Population','Urban_population']].value_counts().reset_index()
grouped_data = grouped_data.sort_values(by='count',ascending=False).reset_index(drop=True)
grouped_data = grouped_data.drop(columns=['count'])
grouped_data['percentage urban population'] = (grouped_data['Urban_population']/grouped_data['Population'])*100
print(grouped_data[['Country','percentage urban population']])
```

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



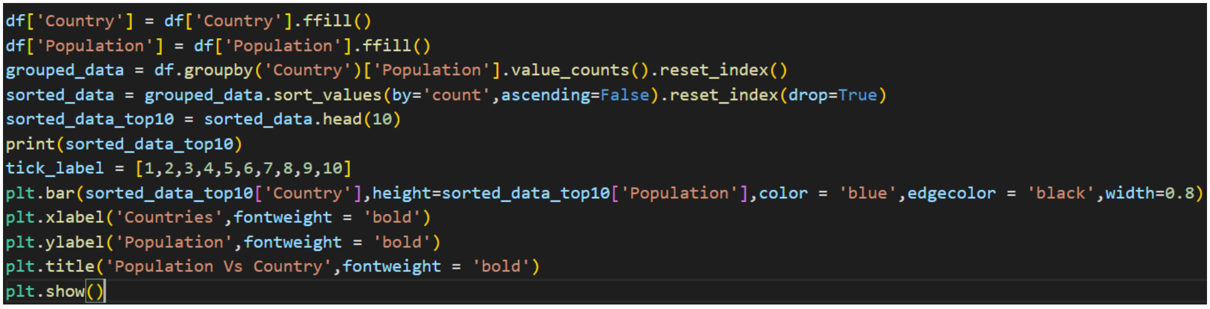
```
df['Country'] = df['Country'].ffill()
df['Latitude'] = df['Latitude'].ffill()
df['Longitude'] = df['Longitude'].ffill()
grouped_data = df.groupby('Country')[['Latitude','Longitude']].value_counts().reset_index()
sorted_grouped_data = grouped_data.sort_values(by='count',ascending=False).reset_index(drop=True)
sorted_grouped_data.columns = ['Country','Latitude','Longitude','Count of Youtube channels']
print(sorted_grouped_data)
print('The above data shows the distribution of Youtube channels on the basis of Latitude and Longitude which is equivalent to \
distribution across countries.\nThere is no clear pattern as such that we can see between (Latitude,Longitude) and number of \
Youtube channels present there.')
```

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

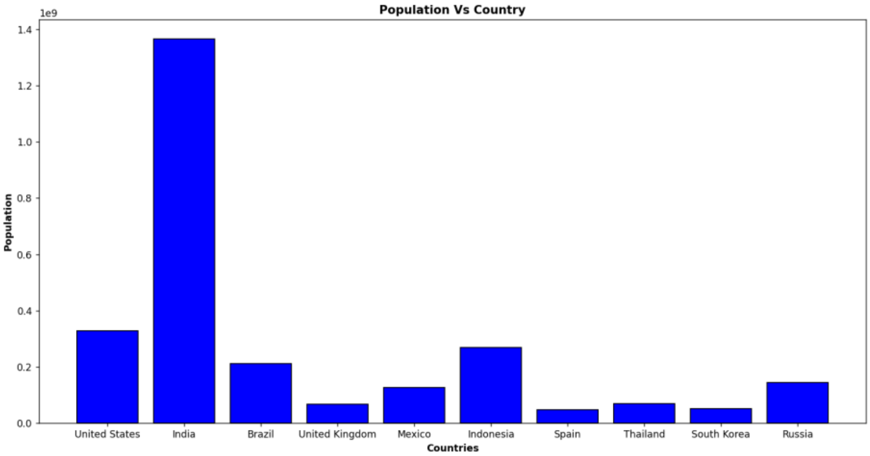


```
df['subscribers'] = df['subscribers'].ffill()
df['Country'] = df['Country'].ffill()
df['Population'] = df['Population'].ffill()
grouped_data1 = df.groupby('Country')['subscribers'].sum().reset_index()
grouped_data2 = df.groupby('Country')['Population'].value_counts().reset_index()
grouped_data = pd.merge(grouped_data1,grouped_data2)
grouped_data = grouped_data.sort_values(by='count',ascending=False).reset_index(drop=True)
print(grouped_data)
correlation = grouped_data['subscribers'].corr(grouped_data['Population'])
print(f'The correlation between number of subscribers and the population of country is {correlation} which is moderately low.')
```

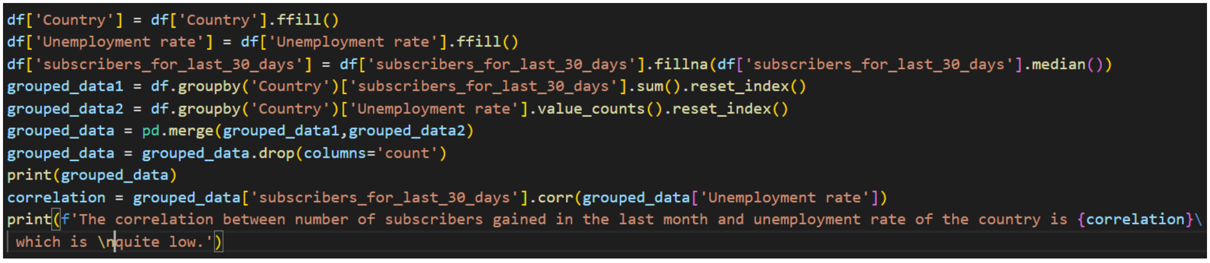
16) How do the top 10 countries with the highest number of YouTube channels compare in terms of their total population?



```
df['Country'] = df['Country'].ffill()
df['Population'] = df['Population'].ffill()
grouped_data = df.groupby('Country')['Population'].value_counts().reset_index()
sorted_data = grouped_data.sort_values(by='count',ascending=False).reset_index(drop=True)
sorted_data_top10 = sorted_data.head(10)
print(sorted_data_top10)
tick_label = [1,2,3,4,5,6,7,8,9,10]
plt.bar(sorted_data_top10['Country'],height=sorted_data_top10['Population'],color = 'blue',edgecolor = 'black',width=0.8)
plt.xlabel('Countries',fontweight = 'bold')
plt.ylabel('Population',fontweight = 'bold')
plt.title('Population Vs Country',fontweight = 'bold')
plt.show()
```

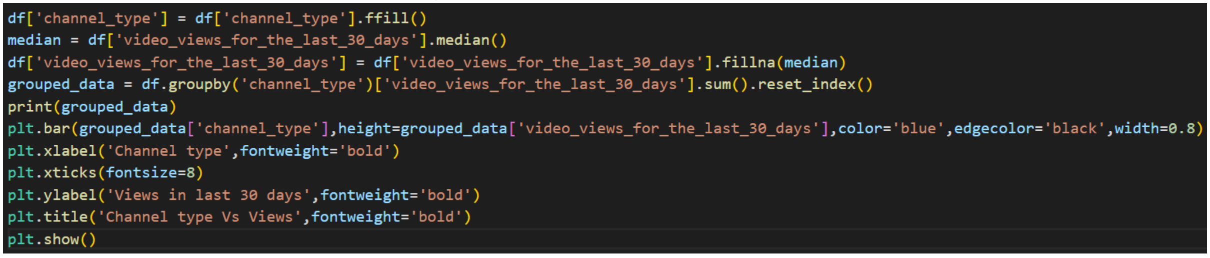


17) Is there a correlation between the number of subscribers gained in the last 30 days and the unemployment rate in a country?

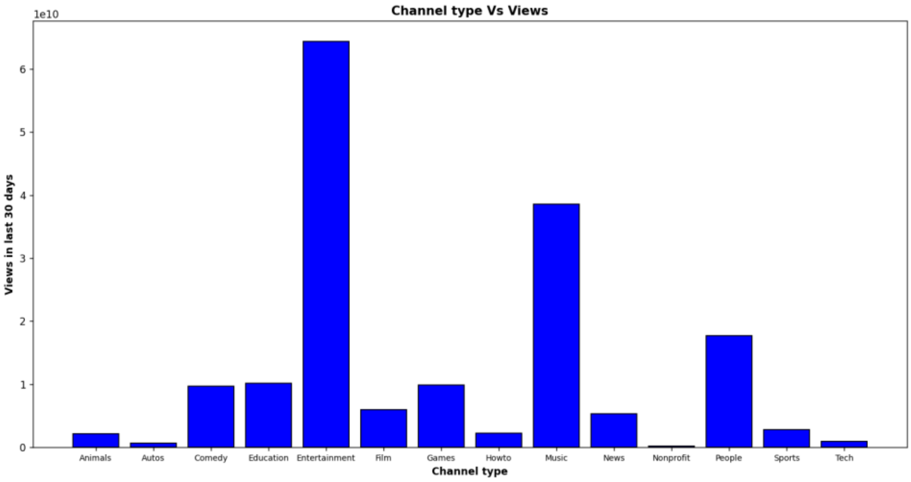


```
df['Country'] = df['Country'].ffill()
df['Unemployment rate'] = df['Unemployment rate'].ffill()
df['subscribers_for_last_30_days'] = df['subscribers_for_last_30_days'].fillna(df['subscribers_for_last_30_days'].median())
grouped_data1 = df.groupby('Country')['subscribers_for_last_30_days'].sum().reset_index()
grouped_data2 = df.groupby('Country')['Unemployment rate'].value_counts().reset_index()
grouped_data = pd.merge(grouped_data1,grouped_data2)
grouped_data = grouped_data.drop(columns='count')
print(grouped_data)
correlation = grouped_data['subscribers_for_last_30_days'].corr(grouped_data['Unemployment rate'])
print(f'The correlation between number of subscribers gained in the last month and unemployment rate of the country is {correlation}\
 which is \nquite low.')
```

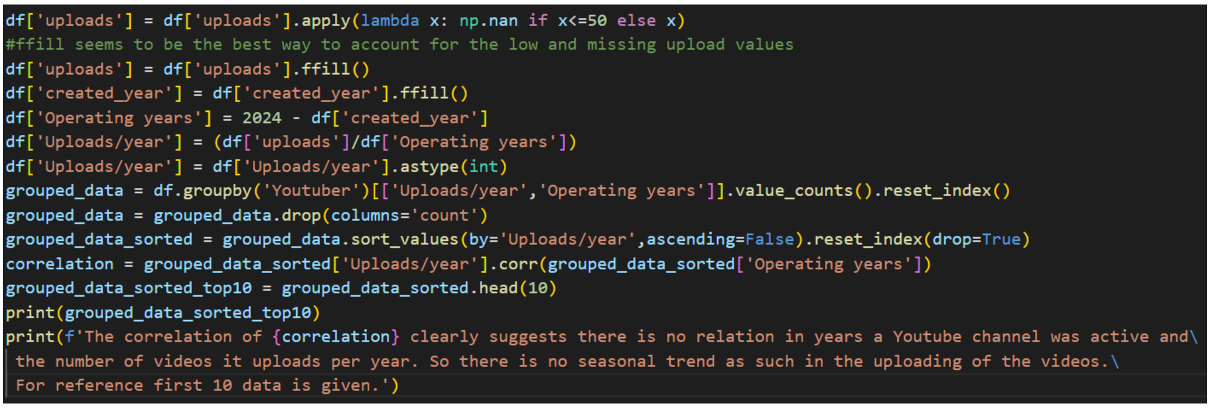
18) How does the distribution of video views for the last 30 days vary across different channel types?



```
df['channel_type'] = df['channel_type'].ffill()
median = df['video_views_for_the_last_30_days'].median()
df['video_views_for_the_last_30_days'] = df['video_views_for_the_last_30_days'].fillna(median)
grouped_data = df.groupby('channel_type')['video_views_for_the_last_30_days'].sum().reset_index()
print(grouped_data)
plt.bar(grouped_data['channel_type'],height=grouped_data['video_views_for_the_last_30_days'],color='blue',edgecolor='black',width=0.8)
plt.xlabel('Channel type',fontweight='bold')
plt.xticks(fontsize=8)
plt.ylabel('Views in last 30 days',fontweight='bold')
plt.title('Channel type Vs Views',fontweight='bold')
plt.show()
```

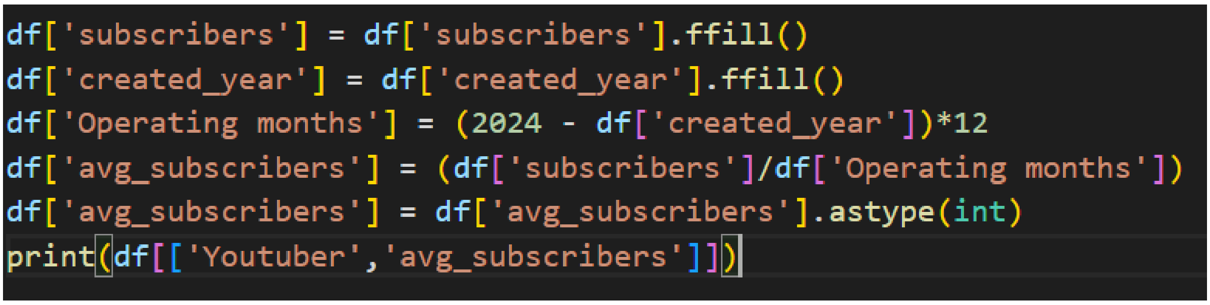


19) Are there any seasonal trends in the number of videos uploaded by YouTube channels?



```
df['uploads'] = df['uploads'].apply(lambda x: np.nan if x<=50 else x)
#ffill seems to be the best way to account for the low and missing upload values
df['uploads'] = df['uploads'].ffill()
df['created_year'] = df['created_year'].ffill()
df['Operating years'] = 2024 - df['created_year']
df['Uploads/year'] = (df['uploads']/df['Operating years'])
df['Uploads/year'] = df['Uploads/year'].astype(int)
grouped_data = df.groupby('Youtuber')[['Uploads/year','Operating years']].value_counts().reset_index()
grouped_data = grouped_data.drop(columns='count')
grouped_data_sorted = grouped_data.sort_values(by='Uploads/year',ascending=False).reset_index(drop=True)
correlation = grouped_data_sorted['Uploads/year'].corr(grouped_data_sorted['Operating years'])
grouped_data_sorted_top10 = grouped_data_sorted.head(10)
print(grouped_data_sorted_top10)
print(f'The correlation of {correlation} clearly suggests there is no relation in years a Youtube channel was active and\
 the number of videos it uploads per year. So there is no seasonal trend as such in the uploading of the videos.\
 For reference first 10 data is given.')
```

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



```
df['subscribers'] = df['subscribers'].ffill()
df['created_year'] = df['created_year'].ffill()
df['Operating months'] = (2024 - df['created_year'])*12
df['avg_subscribers'] = (df['subscribers']/df['Operating months'])
df['avg_subscribers'] = df['avg_subscribers'].astype(int)
print(df[['Youtuber','avg_subscribers']])
```