Mini Project

CSE2015 DATA ANALYSIS AND VISUALIZATION

YOUTUBE ANALYSIS

Submitted By

20211CAI0062: PRANAV GANESH

20211CAI0057: SUHAS K

20211CAI0203: VATHSALA B S

GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING
PRESIDENCY UNIVERSITY

BENGALURU

DECEMBER 2023

ABSTRACT

YouTube is one of the leading data streaming platforms where a lot of people share and watch content. This platform has a lot of content creators who share videos of various kinds in various formats and generate a huge amount of data every single second. A platform with such a high user base and generating huge amounts of data will have some data missing and a lot of hidden insights that may be useful for decision-making. In our project, we have applied some of the data processing techniques to get rid of the missing values and have performed data visualization to discover some insights about the platform. The insights obtained from the visualization are useful for discovering what topics are trending and what people are watching more often so that the recommendation system of YouTube can be improved. Our project aims to discover new insights about the platform and implement them to improve the user experience so that the users get the best experience when compared to the rival platforms.

PROBLEM STATEMENT

We are addressing the problem of increasing the viewer base and the recommendation system of YouTube. By performing Data Analysis and Visualization we are discovering insights that will help us understand the user's behavior so that we can gain more viewers and improve the recommendation system of YouTube.

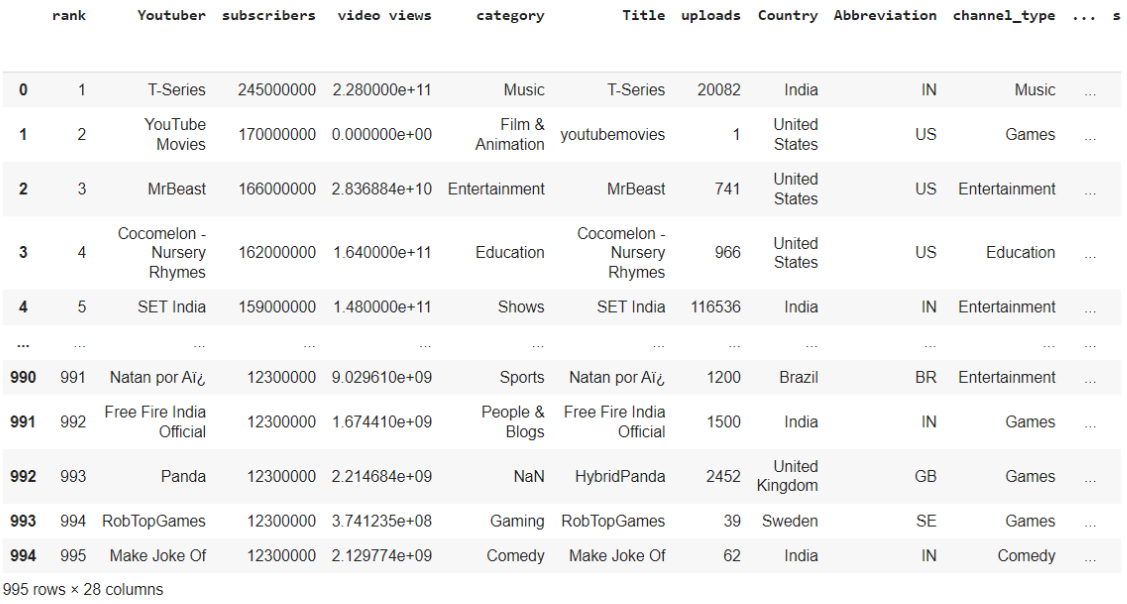
DATA

Data collection:

We have collected the YouTube dataset from Kaggle and performed data analysis on it. The data initially had 995 rows and 28 columns. The data is about what kind of videos are being watched more frequently, the subscriber count of YouTube channels, the views it is accumulating and much more.

Loading the dataset:

```
data = pd.read_csv('/content/Global YouTube Statistics.csv’)
```

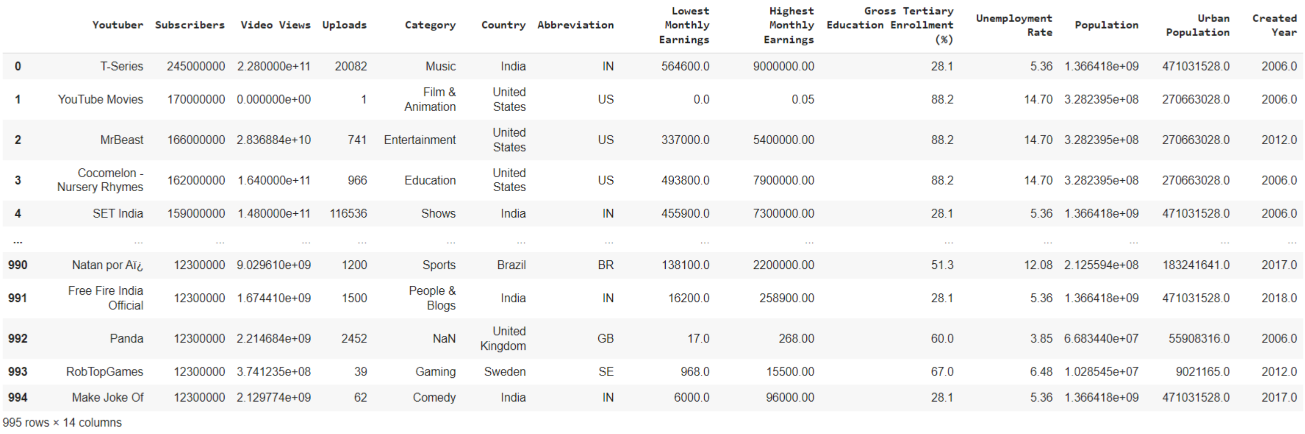


	rank	Youtuber	subscribers	video views	category	Title	uploads	Country	Abbreviation	channel_type	...	s
0	1	T-Series	245000000	2.280000e+11	Music	T-Series	20082	India	IN	Music	...	
1	2	YouTube Movies	170000000	0.000000e+00	Film & Animation	youtubemovies	1	United States	US	Games	...	
2	3	MrBeast	166000000	2.836884e+10	Entertainment	MrBeast	741	United States	US	Entertainment	...	
3	4	Cocomelon - Nursery Rhymes	162000000	1.640000e+11	Education	Cocomelon - Nursery Rhymes	966	United States	US	Education	...	
4	5	SET India	159000000	1.480000e+11	Shows	SET India	116536	India	IN	Entertainment	...	
...	...	...	...	...	...	...	...	...	...	...	...	
990	991	Natan por Aï¿	12300000	9.029610e+09	Sports	Natan por Aï¿	1200	Brazil	BR	Entertainment	...	
991	992	Free Fire India Official	12300000	1.674410e+09	People & Blogs	Free Fire India Official	1500	India	IN	Games	...	
992	993	Panda	12300000	2.214684e+09	NaN	HybridPanda	2452	United Kingdom	GB	Games	...	
993	994	RobTopGames	12300000	3.741235e+08	Gaming	RobTopGames	39	Sweden	SE	Games	...	
994	995	Make Joke Of	12300000	2.129774e+09	Comedy	Make Joke Of	62	India	IN	Comedy	...	

995 rows × 28 columns

Selecting the variables: We are selecting the variables which are necessary for analysis and leaving the unwanted ones.

```
youtube = data[['Youtuber', 'Subscribers', 'Video Views', 'Uploads', 'Category', 'Country', 'Abbreviation', 'Lowest Monthly Earnings', 'Highest Monthly Earnings', 'Gross Tertiary Education Enrollment (%)', 'Unemployment Rate', 'Population', 'Urban Population', 'Created Year']]
youtube
```



	Youtuber	Subscribers	Video Views	Uploads	Category	Country	Abbreviation	Lowest Monthly Earnings	Highest Monthly Earnings	Gross Tertiary Education Enrollment (%)	Unemployment Rate	Population	Urban Population	Created Year
0	T-Series	245000000	2.280000e+11	20082	Music	India	IN	564600.0	9000000.00	28.1	5.36	1.366418e+09	471031528.0	2006.0
1	YouTube Movies	170000000	0.000000e+00	1	Film & Animation	United States	US	0.0	0.05	88.2	14.70	3.282395e+08	270663028.0	2006.0
2	MrBeast	166000000	2.836884e+10	741	Entertainment	United States	US	337000.0	5400000.00	88.2	14.70	3.282395e+08	270663028.0	2012.0
3	Cocomelon - Nursery Rhymes	162000000	1.640000e+11	966	Education	United States	US	493800.0	7900000.00	88.2	14.70	3.282395e+08	270663028.0	2006.0
4	SET India	159000000	1.480000e+11	116536	Shows	India	IN	455900.0	7300000.00	28.1	5.36	1.366418e+09	471031528.0	2006.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
990	Natan por Aï¿	12300000	9.029610e+09	1200	Sports	Brazil	BR	138100.0	2200000.00	51.3	12.08	2.125594e+08	183241641.0	2017.0
991	Free Fire India Official	12300000	1.674410e+09	1500	People & Blogs	India	IN	16200.0	258900.00	28.1	5.36	1.366418e+09	471031528.0	2018.0
992	Panda	12300000	2.214684e+09	2452	NaN	United Kingdom	GB	17.0	268.00	60.0	3.85	6.683440e+07	55908316.0	2006.0
993	RobTopGames	12300000	3.741235e+08	39	Gaming	Sweden	SE	968.0	15500.00	67.0	6.48	1.028545e+07	9021165.0	2012.0
994	Make Joke Of	12300000	2.129774e+09	62	Comedy	India	IN	6000.0	96000.00	28.1	5.36	1.366418e+09	471031528.0	2017.0

995 rows × 14 columns

Getting rid of null values in categorical variables:

```
categorical_variables = youtube.select_dtypes(include= 'O').columns
youtube[categorical_variables] = youtube[categorical_variables].fillna('Other')
youtube_not_null = youtube.dropna(subset= ['Gross Tertiary Education Enrollment (%)','Unemployment Rate', 'Population', 'Urban Population'])
```

Calculating the mean of Numerical Variables:

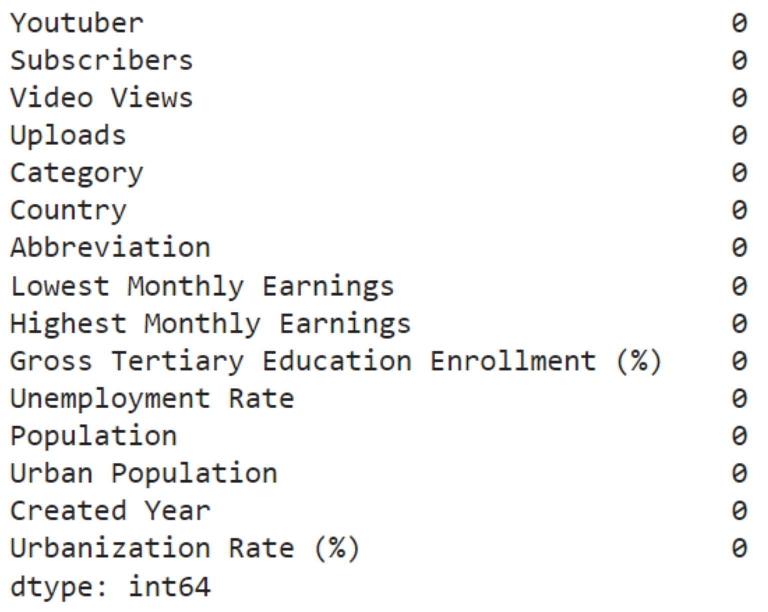
```
mean_edu = round(youtube_not_null['Gross Tertiary Education Enrollment (%)'].mean(),1)
print('Mean Gross Tertiary Education Enrollment:', mean_edu)
mean_unemp = round(youtube_not_null['Unemployment Rate'].mean(),1)
print('Mean Unemployment Rate:', mean_unemp)
mean_popu = round(youtube_not_null['Population'].mean(),0)
print('Mean Population:', mean_popu)
mean_urban = round(youtube_not_null['Urban Population'].mean(),0)
print('Mean Urban Population:', mean_urban)
```

Replacing the null values in Numerical Variables with the mean:

```
youtube['Gross Tertiary Education Enrollment (%)'] = youtube['Gross Tertiary Education Enrollment (%)'].fillna(mean_edu)
youtube['Unemployment Rate'] = youtube['Unemployment Rate'].fillna(mean_unemp)
youtube['Population'] = youtube['Population'].fillna(mean_popu)
youtube['Urban Population'] = youtube['Urban Population'].fillna(mean_urban)
```

Verify that there are no more null values:

```
youtube.isnull().sum()
```



```
Youtuber                                   0
Subscribers                                0
Video Views                                0
Uploads                                    0
Category                                   0
Country                                    0
Abbreviation                               0
Lowest Monthly Earnings                    0
Highest Monthly Earnings                   0
Gross Tertiary Education Enrollment (%)    0
Unemployment Rate                          0
Population                                 0
Urban Population                           0
Created Year                               0
Urbanization Rate (%)                      0
dtype: int64
```

After processing the data had 965 rows and 14 columns which we have used for data visualization.

TOOLS USED

- Software:

  Google Colab: Colab or Colaboratory allows us to write and execute Python codes in our browser.

- Libraries:

  1. NumPy: It is a library in Python that provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on arrays.
  2. Pandas: It is a data manipulation and analysis library in Python. It provides data tools to work on various types of data.
  3. Seaborn: It is a Python data visualization library based on Matplotlib. It provides an interface for creating attractive and informative statistical graphs.
  4. Matplotlib: It is a comprehensive 2D plotting library for Python. It enables the creation of a wide variety of static, animated, and interactive plots in Python.
  5. SciPy: It is an open-source library for mathematics, science, and engineering. It builds on NumPy and provides additional functionality that extends the capabilities of NumPy.
  6. Wordcloud: The wordcloud library is a popular Python library used to generate word clouds from text data.
  7. Re: it is a built-in Python library that provides support for regular expressions. Regular expressions are powerful tools for pattern matching and string manipulation.

- Hardware:

  Dell Inspiron 15 Laptop: Processor 11th Gen Intel(R) Core(TM) i5-11320H @ 3.20GHz ,16 GB RAM ,Windows 11 Home

- Database: Global YouTube Statistics

OUTCOME

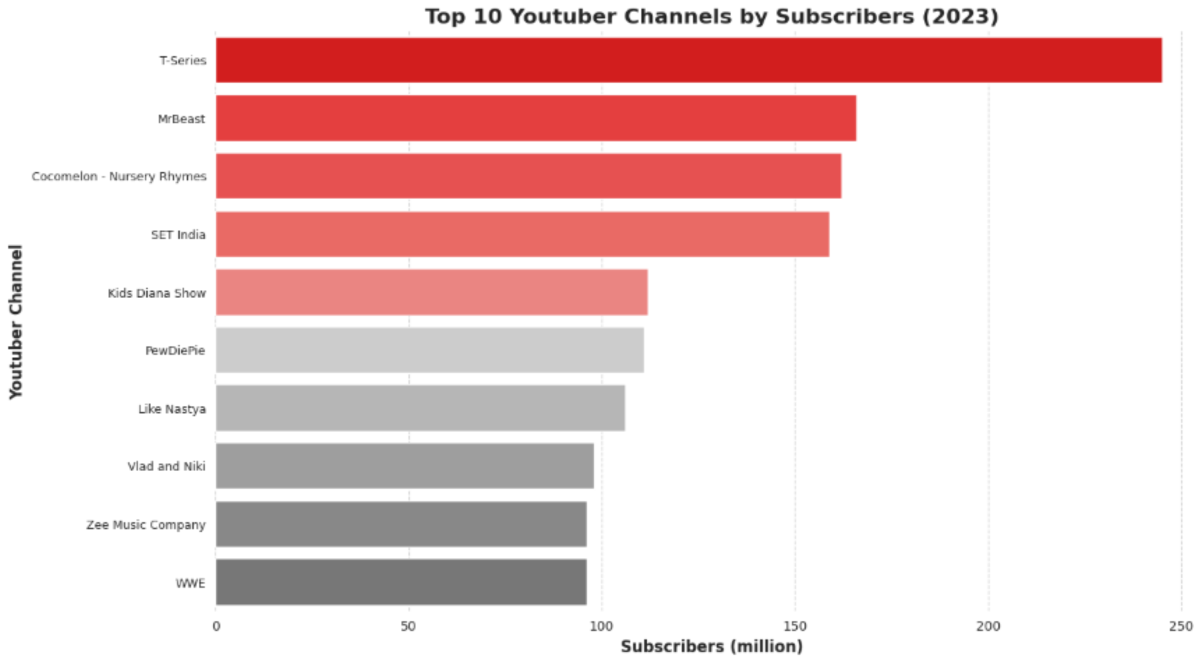


Figure 1a: Bar Chart Showing the Top 10 YouTube Channels

This chart shows which are the Top 10 YouTube channels by number of subscribers. We can infer that T-Series is the having the highest number of subscribers.

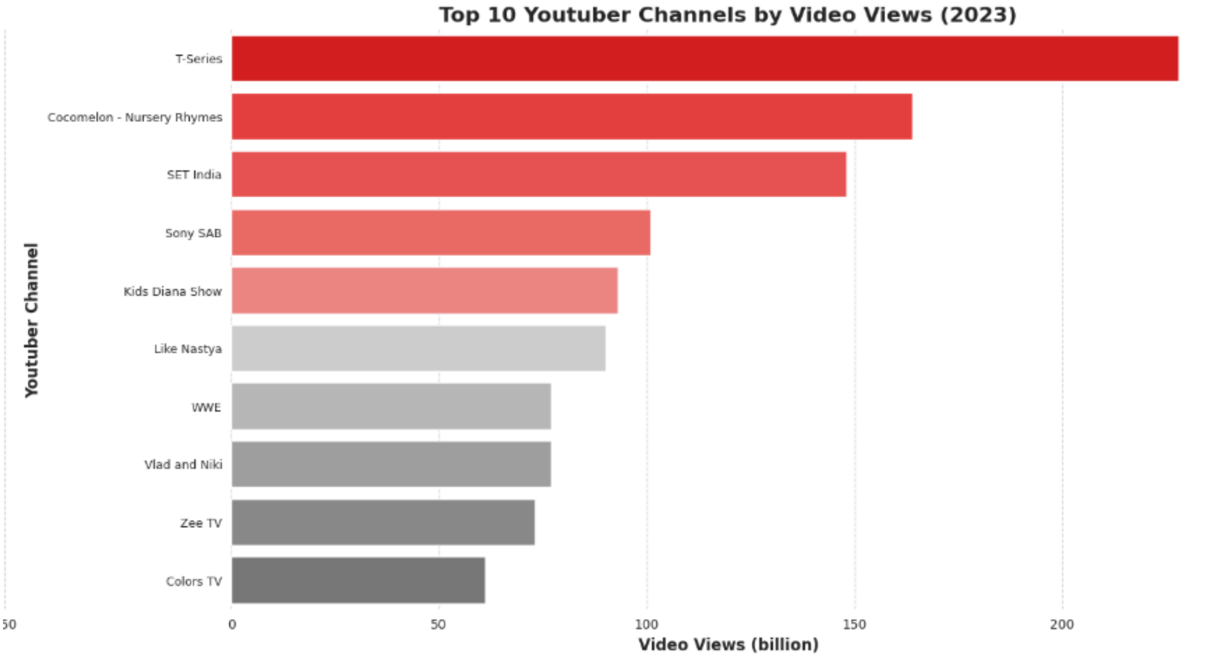
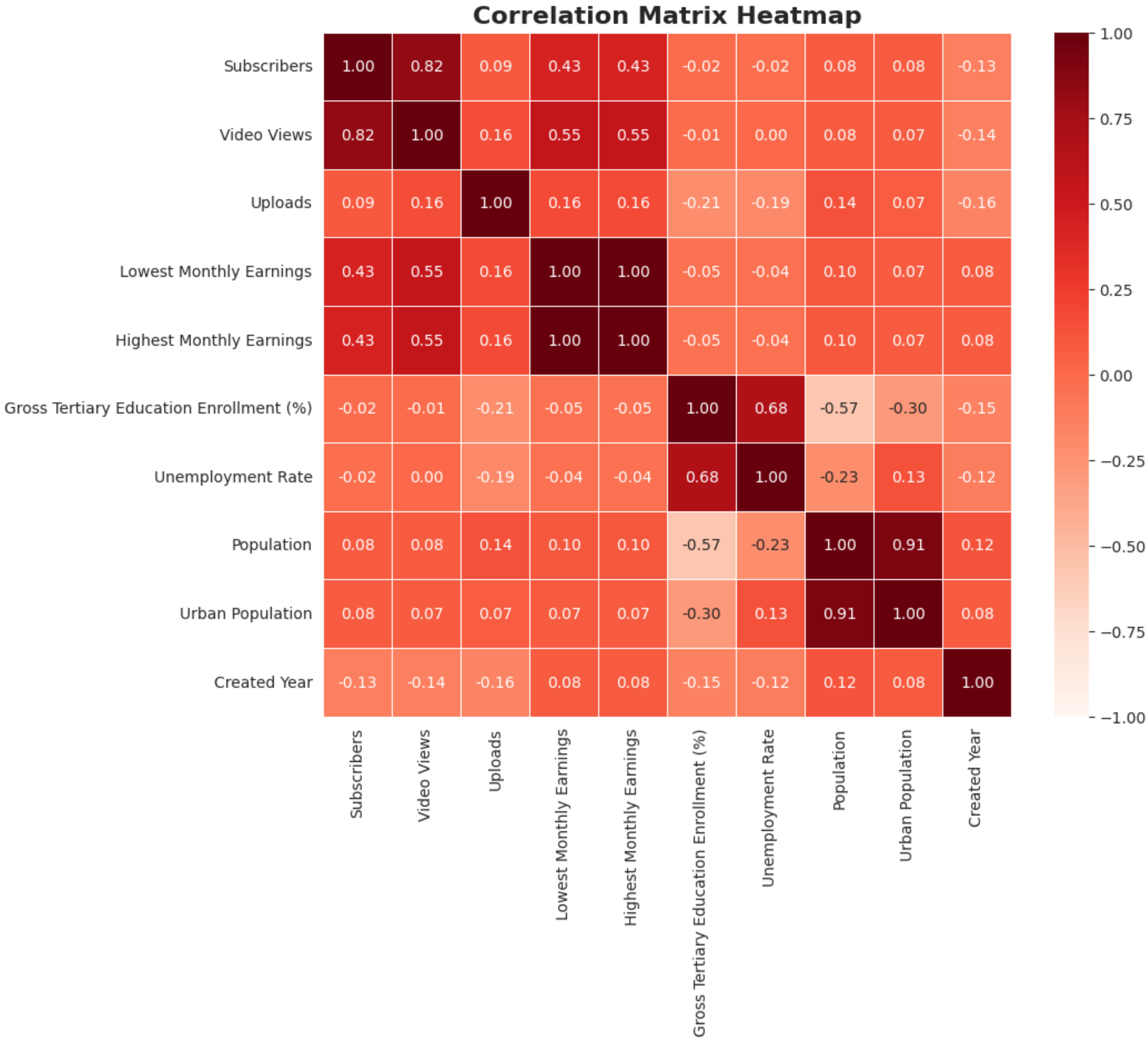


Figure 1b: Bar Chart Showing the Top 10 YouTube Channels

This chart shows which are the Top 10 YouTube channels by number of views. We can infer that T-Series is the having the highest number of views for its videos.



	Subscribers	Video Views	Uploads	Lowest Monthly Earnings	Highest Monthly Earnings	Gross Tertiary Education Enrollment (%)	Unemployment Rate	Population	Urban Population	Created Year
Subscribers	1.00	0.82	0.09	0.43	0.43	-0.02	-0.02	0.08	0.08	-0.13
Video Views	0.82	1.00	0.16	0.55	0.55	-0.01	0.00	0.08	0.07	-0.14
Uploads	0.09	0.16	1.00	0.16	0.16	-0.21	-0.19	0.14	0.07	-0.16
Lowest Monthly Earnings	0.43	0.55	0.16	1.00	1.00	-0.05	-0.04	0.10	0.07	0.08
Highest Monthly Earnings	0.43	0.55	0.16	1.00	1.00	-0.05	-0.04	0.10	0.07	0.08
Gross Tertiary Education Enrollment (%)	-0.02	-0.01	-0.21	-0.05	-0.05	1.00	0.68	-0.57	-0.30	-0.15
Unemployment Rate	-0.02	0.00	-0.19	-0.04	-0.04	0.68	1.00	-0.23	0.13	-0.12
Population	0.08	0.08	0.14	0.10	0.10	-0.57	-0.23	1.00	0.91	0.12
Urban Population	0.08	0.07	0.07	0.07	0.07	-0.30	0.13	0.91	1.00	0.08
Created Year	-0.13	-0.14	-0.16	0.08	0.08	-0.15	-0.12	0.12	0.08	1.00

Figure 2: Heat Map showing the correlation between the Variables

From this heat map we can learn that there is a strong correlation between Subscribers, Urban Population and Views. We can infer that as the number of subscribers grow and the urban population increases the number of views will increase.

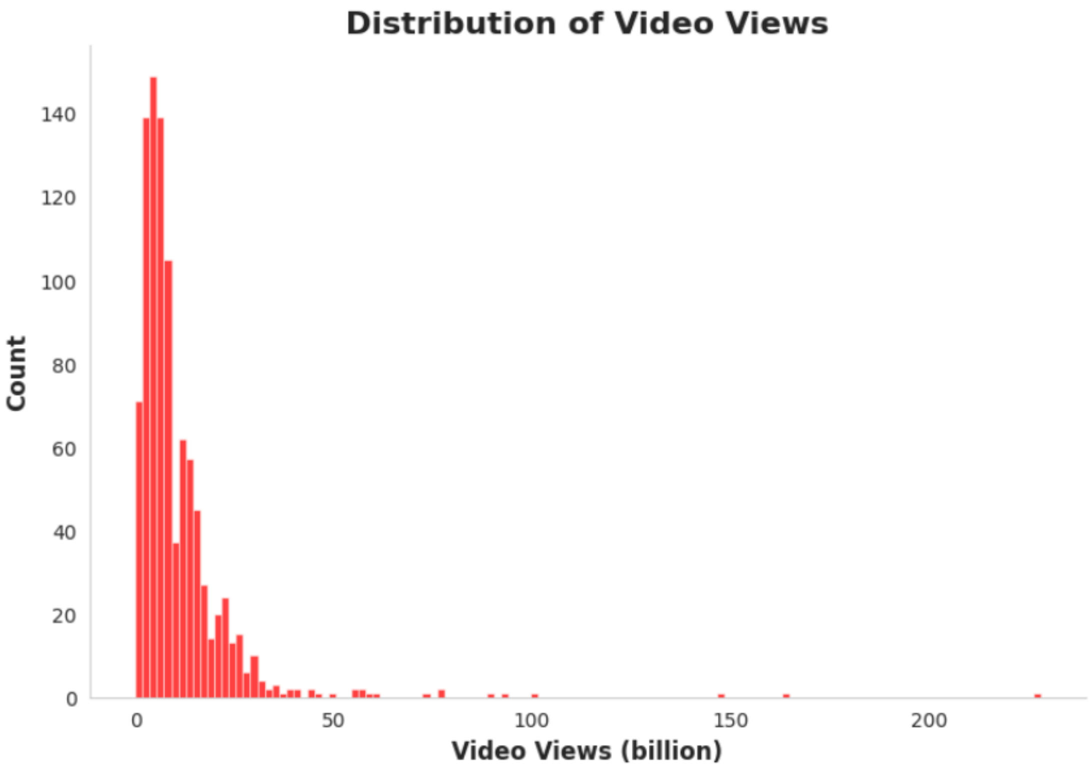


Figure 3: Histogram of Channel Views

This histogram shows how many views the channels are accumulating video views. We can see that the no of channels crossing 50 billion views is really less.

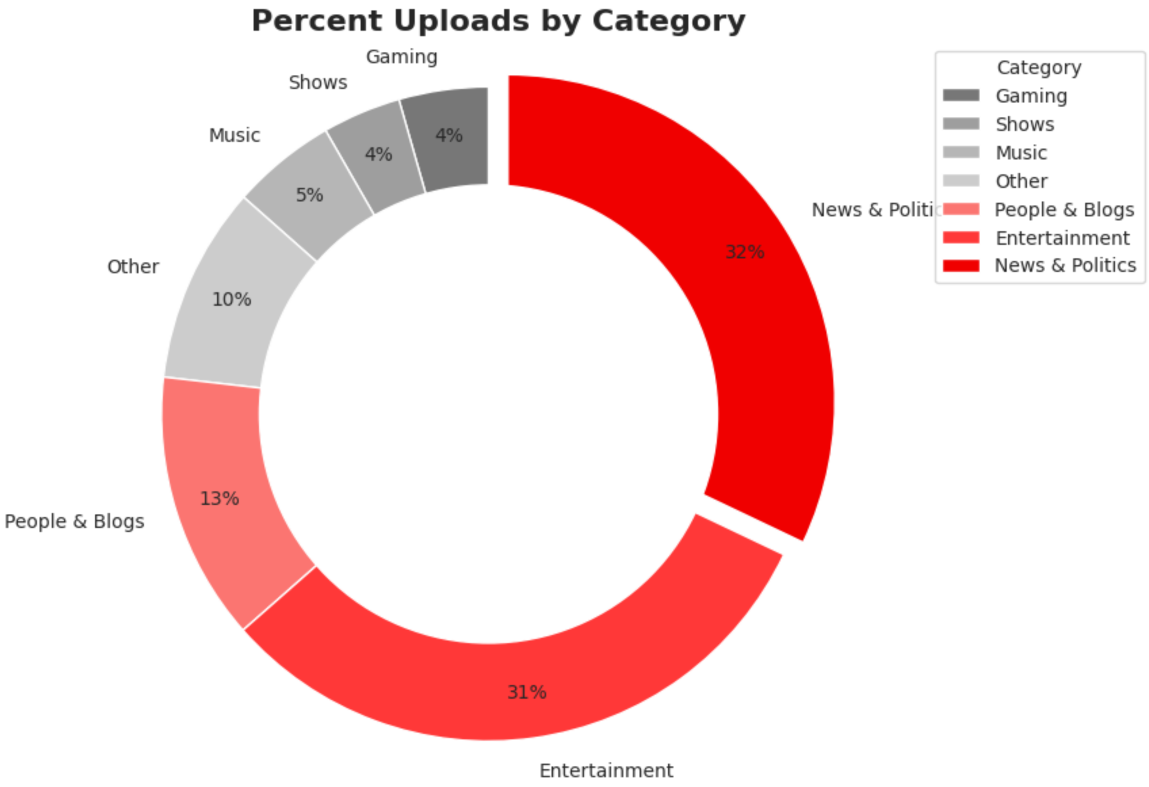


Figure 4: Pie Chart Showing Category Wise Video Uploads

This pie chart shows in which category of videos the highest number of videos are being uploaded. We can see that more number of videos are being uploaded in the category of News & Politics

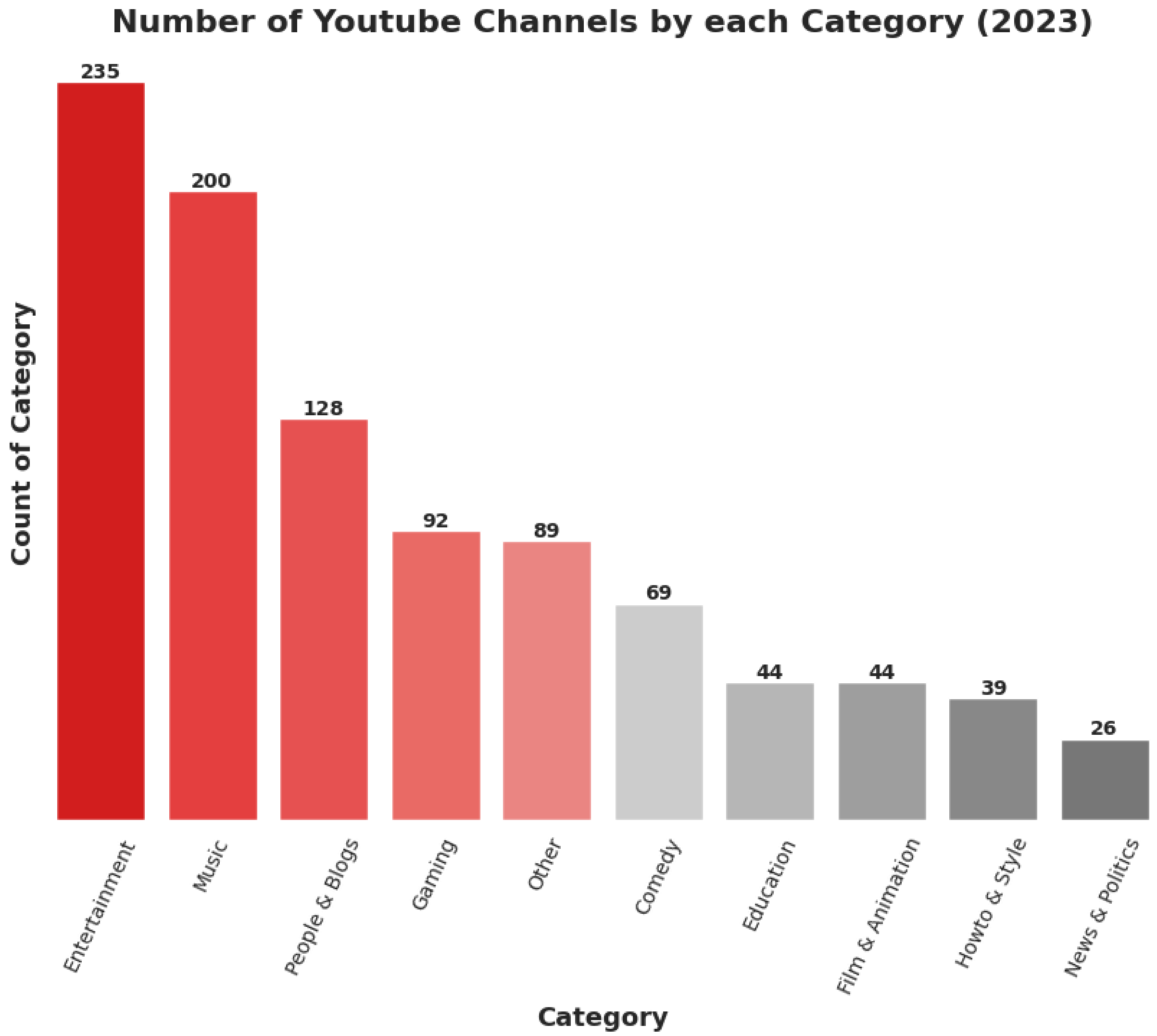


Figure 5: Bar Chart showing the Categories with highest no channels in 2023

From this chart, we can see that the Category of entertainment has the most number of YouTube Channels in 2023.

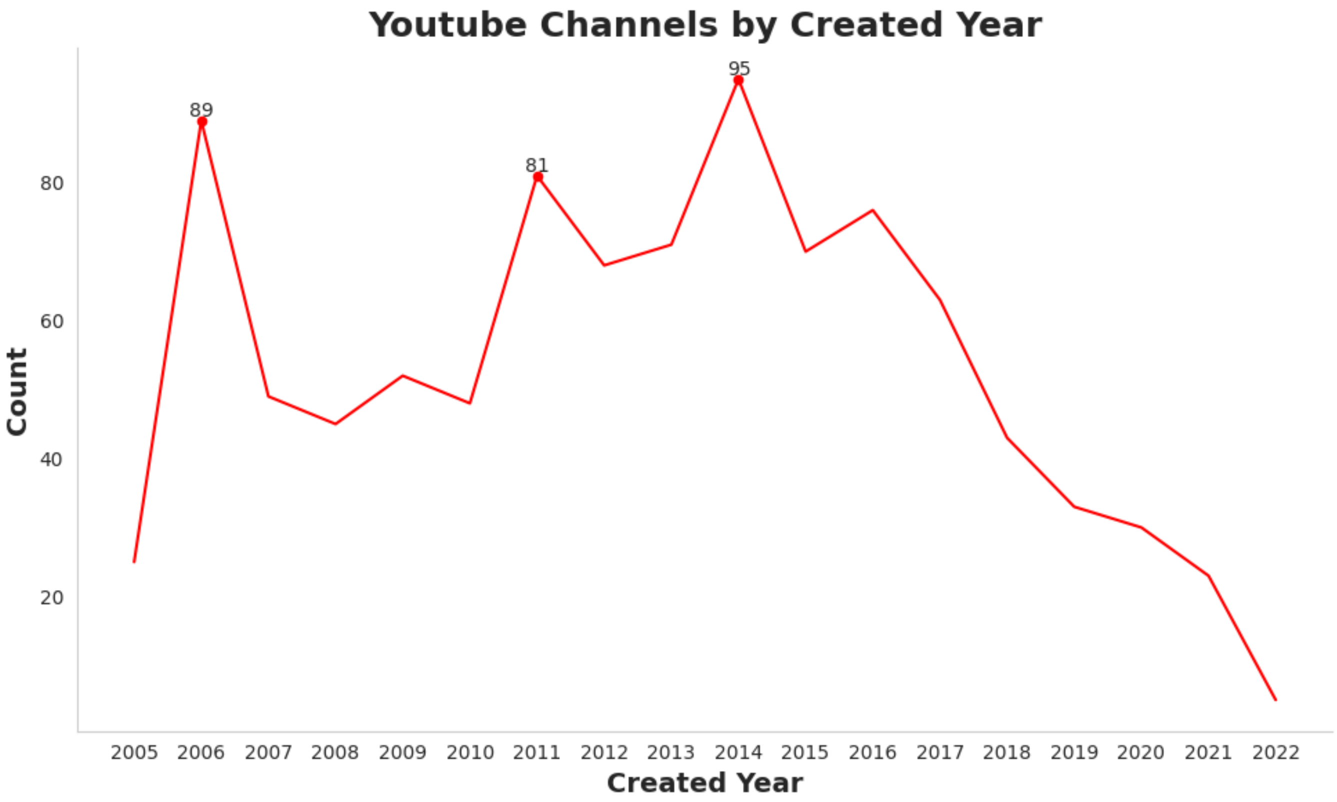


Figure 6: Line Chart Showing the number of channels created annually

This line chart shows the number of YouTube channels created each year. We can see that the most number of channels were created in the year 2014.

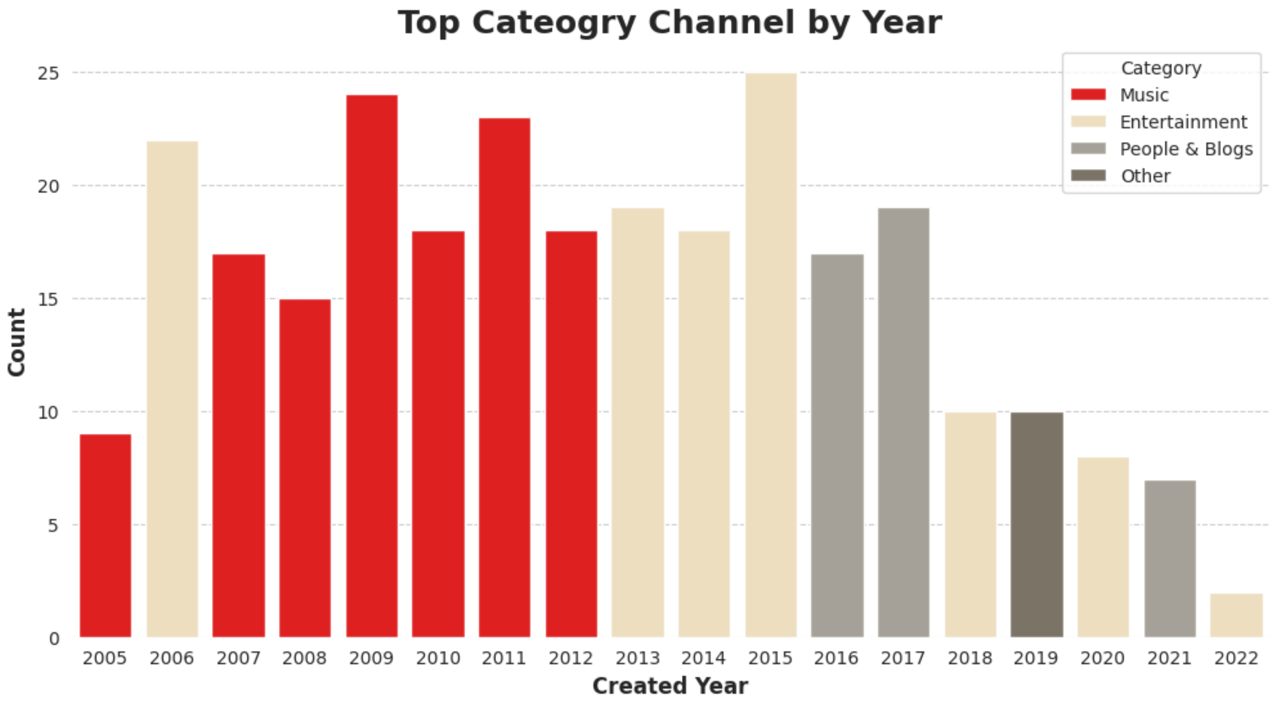


Figure 7: Bar Chart showing Category Wise Highest Creation of YouTube Channels

This chart shows the category in which highest number of channels were created in each year. We can see that more number of channels were created in the category of Music in many years.

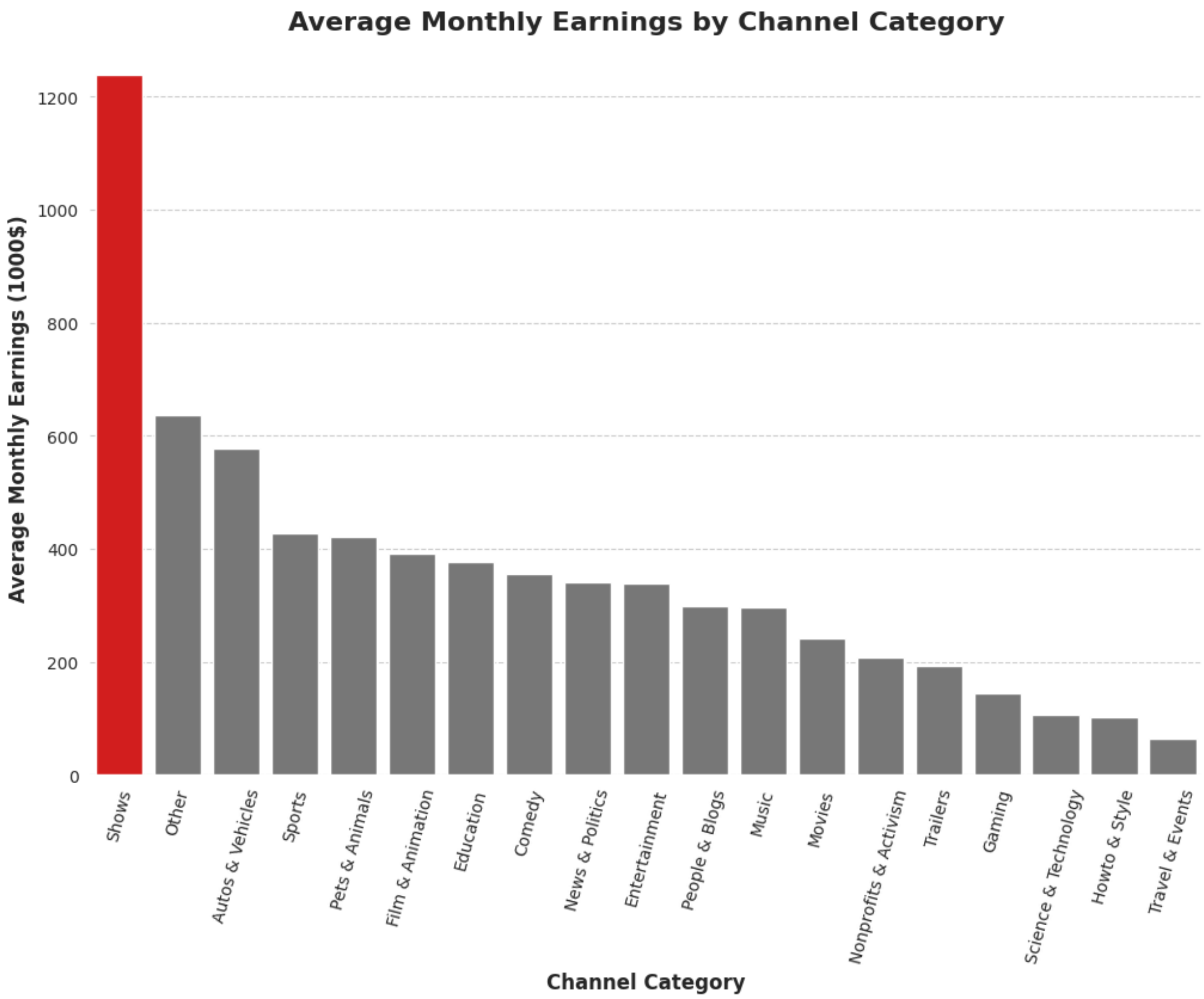


Figure 8: Bar Chart Showing the revenue of each category

This chart shows the earnings of YouTube channels category wise. The channels which are making and uploading the videos related to shows are earning the highest amount of money.

CODE

```
sb.set_style("whitegrid",{'axes.grid' : True})

fig, ax = plt.subplots(1, 2, figsize = (30,8))

sb.barplot(top_10_subs,x = 'Subscribers (million)' ,y = 'Youtuber', palette=
youtube_color_barchart_10, ax = ax[0])

ax[0].set_title('Top 10 Youtuber Channels by Subscribers (2023)',fontweight = 'heavy',
fontsize = 16)

ax[0].set_xlabel('Subscribers (million)',fontweight = 'heavy', fontsize = 12)

ax[0].set_ylabel('Youtuber Channel',fontweight = 'heavy', fontsize = 12)

ax[0].grid(axis = 'x', linestyle ='--')

ax[0].spines['top'].set_visible(False)

ax[0].spines['right'].set_visible(False)

ax[0].spines['bottom'].set_visible(False)

ax[0].spines['left'].set_visible(False)

plt.sca(ax[0])

plt.yticks(size = 9)

sb.barplot(top_10_views,x = 'Video Views (billion)' ,y = 'Youtuber', palette=
youtube_color_barchart_10, ax = ax[1])

ax[1].set_title('Top 10 Youtuber Channels by Video Views (2023)',fontweight = 'heavy',
fontsize = 16)

ax[1].set_xlabel('Video Views (billion)',fontweight = 'heavy', fontsize = 12)

ax[1].set_ylabel('Youtuber Channel',fontweight = 'heavy', fontsize = 12)

ax[1].grid(axis = 'x', linestyle ='--')

ax[1].spines['top'].set_visible(False)

ax[1].spines['right'].set_visible(False)

ax[1].spines['bottom'].set_visible(False)

ax[1].spines['left'].set_visible(False)

plt.sca(ax[1])

plt.yticks(size = 9)

plt.show();


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

plt.title('Correlation Matrix Heatmap', fontweight = 'heavy', fontsize = 16)
```

```
sb.heatmap(correlation, annot= True,cmap= 'Reds',vmin= -1, vmax= 1,linecolor = 'white', linewidths = 0.5,fmt = '.2f')
plt.show()

sb.histplot(youtube_corr_statistics['Video Views (billion)'], color= 'red', ax = ax2)
ax2.set_title('Distribution of Video Views', fontweight = 'heavy', fontsize = 16)
ax2.set_xlabel('Video Views (billion)', fontweight = 'heavy', fontsize = 12)
ax2.set_ylabel('Count', fontweight = 'heavy', fontsize = 12)
ax2.spines['top'].set_visible(False)
ax2.spines['right'].set_visible(False)
ax2.spines['bottom'].set_visible(True)
ax2.spines['left'].set_visible(True)

labels = uploads_category['Category']
sizes = uploads_category['Uploads (k.)']
explode = [0, 0, 0, 0, 0, 0, 0.07]
plt.figure(figsize= (9,6))
plt.pie(sizes, labels= labels,autopct='%1.0f%%',startangle=90,explode= explode,colors= youtube_color_donutchart,pctdistance=0.86)
plt.legend(title = 'Category',bbox_to_anchor=(1.1, 1), labels = labels)
plt.axis('equal')
plt.title('Percent Uploads by Category', fontweight = 'heavy', fontsize = 16)
plt.tight_layout()
circle = plt.Circle(xy= (0,0), radius= .70, facecolor = 'white')
plt.gca().add_artist(circle)
plt.show()

sb.set_style("whitegrid",{'axes.grid' : False})
plt.figure(figsize=(10,7))
# Select data for categoies and values feature
cnt_category = count_category['Count of Category'].to_list()
plt.title('Number of Youtube Channels by each Category (2023)', fontsize = 16, fontweight = 'heavy')
```

```
sb.barplot(data= count_category, x = 'Category', y = 'Count of Category', palette= youtube_color_barchart_10)
for i, value in enumerate(cnt_category):
    plt.text(i, value + 1, cnt_category[i],  ha = 'center', fontsize=10, fontweight = 'heavy')
sb.despine(left=True, bottom=True)
plt.grid(axis = 'y', linestyle ='--')
plt.xlabel('Category', fontsize = 13, fontweight = 'heavy')
plt.ylabel('Count of Category', fontweight = 'heavy', fontsize = 13)
plt.xticks(rotation = 65)
plt.yticks([]) #loại bỏ các tham số trên trục y
plt.show();

plt.figure(figsize= (12,6))
sb.barplot(data = top_cnt_channel_by_year, x = 'Created Year', y = 'Youtuber', hue = 'Category', dodge=False, palette= youtube_color_barchart_freestyle2)
plt.title('Top Cateogry Channel by Year', fontweight = 'heavy', fontsize = 18)
plt.xlabel('Created Year', fontweight = 'heavy', fontsize = 12)
plt.ylabel('Count', fontweight = 'heavy', fontsize = 12)
plt.xticks(rotation = 0, ha = 'center')
sb.despine(left=True, bottom=True)
plt.grid(axis = 'y', linestyle ='--')
plt.show()
plt.figure(figsize=(12,8))
sb.barplot(data= category_avg_earnings, x = 'Category', y = 'Average Monthly Earnings (1000$)', palette= youtube_color_barchart_freestyle3)
plt.title('Average Monthly Earnings by Channel Category', fontweight = 'heavy', fontsize = 16)
sb.despine(left=True, bottom=True)
plt.grid(axis = 'y', linestyle ='--')
plt.xlabel('Channel Category', fontweight = 'heavy', fontsize = 12)
plt.ylabel('Average Monthly Earnings (1000$)', fontweight = 'heavy', fontsize = 12)
plt.xticks(rotation = 75)
plt.show()
```

LIMITATIONS AND FUTURE SCOPE

The analysis and visualization of the entire data have been performed from which all null values have been cleared and new insights regarding the data have been discovered. However, the inference of the insights found has not been implemented. Implementing them in the future will make sure that the user base of YouTube will be improved, content creators will be able to earn more money from their channels, and the recommendation of videos can be improved leading to improved user satisfaction. Using ML Models to include these improvements will help YouTube become a much better platform and remain as the market leader in the segment of video streaming. This process can be an iterative process where we can perform data analysis and visualization on the data of the improvised version of YouTube to find out some more new insights that can be implemented in future versions of YouTube.

CONCLUSION

In this project, we achieved a cleaned dataset of the YouTube data which can be used for visualization. By performing visualization we can to know about many insights about YouTube which can be used to improve YouTube as a platform. We discovered the topics that are trending, the categories with the highest demand, the channel with the most number of subscribers, the video with the most number of views, the category with the highest revenue, the share of videos of each category, the number of channels that are being created each year and much more. All these insights are useful in making decisions for the stakeholders of YouTube and improving YouTube. Additionally, these insights can also be used by content creators to analyze and improve their content. We would like to conclude by saying that there is always room for improvement and by implementing the findings of this project YouTube can be improved further to improve the user experience and remain as the market leader.