**A PROJECT REPORT**

**ON**

# "*YouTube Pulse: Tracking Trends & Engagement*"

**Submitted to**

**THE GEORGE WASHINGTON UNIVERSITY**

**BY**

***SURYA VAMSI PATIBALLA (G40559527)***

NSTRUCTOR :- Dr.REZA JAFARI

COURSE: VISUALIZATION OF COMPLEX DATA (**CRN: 92824**)

DATE: 04/25/2024

## A Quick glimpse

Currently, in the phase of the evolution of Data modelling and Visualization, the ability to analyze and visualize complex datasets is a must for getting business insights. My Final Term Project (FTP) showcases a comprehensive application of theoretical knowledge to a practical scenario, focusing on the in-depth visualization of the 'youtube.csv' dataset, a rich compilation of YouTube trend data that provides a wide range of variables for detailed examination. Throughout this course, the curriculum has systematically developed my capabilities in data manipulation, exploratory analysis, and the creation of complex visualizations. This project enabled me to apply the skills I have honed throughout the semester to perform data cleaning, transformation, and finally, the development of a beautiful interactive dashboard. I used the Dash framework to build this dashboard with the support of Python, and I deployed it on Google Cloud Platform (GCP), offering a neat and clear visual representation of what is going on in YouTube's trends.

## Target

The ultimate agenda of the FTP is to utilize Python's visualization libraries to gain insights into the 'youtube.csv' dataset. By implementing libraries such as Matplotlib, Seaborn, Plotly, and Dash, the project aims to :-

- Conduct an exhaustive exploratory data analysis (EDA) to uncover underlying patterns, correlations, and dependencies particularly focusing on how various features link with the 'views' variable.
- Visualize the distribution and dynamics of key features like views, likes, and dislikes, which represent the pulse of current trends on YouTube.
- Examine relationships between different YouTube video attributes to understand their impact on viewership and user engagement, reflecting the mini human world of YouTube where individuals interact and express their interests.

## Report Summary

1. **Introduction:** This section overviews the methodologies and procedures used to achieve the FTP's objectives. It also outlines the report's structure and sets the context for the subsequent analysis.
2. **Description of the Dataset:** This section comprehensively describes the TMDB Movies dataset, discussing how it meets the selection criteria. The variables within the dataset will be categorized into dependent and independent ones.
3. **Pre-processing of the Dataset:** The processes used to clean the dataset, including handling missing samples and NaN values, will be explained. The initial observations of the cleaned dataset and relevant statistics will be displayed.
4. **Outlier Detection & Removal:** The method chosen for detecting and removing outliers from the raw dataset will be documented, along with observations.
5. **Principal Component Analysis (PCA):** A complete PCA for feature dimension reduction will be conducted. The report will fully explain this analysis and review the condition number and the singular values of the reduced dimension features.
6. **Normality Test:** Histograms will be plotted to show normality, and the plots will be explained. A K-S statistic test will also be applied to check whether the variables are normally distributed.
7. **Data Transformation:** This section covers in detail the methods for converting non-Gaussian distributions to Gaussian distributions.
8. **Heatmap & Pearson Correlation Coefficient Matrix:** This section will showcase the Pearson correlation coefficient between variables through a heatmap and a scatter plot matrix. Observations and interpretations of these visualizations will be discussed and tabulated.
9. **Statistics:** The dataset will be analysed using a t-test, and insights and observations will be documented. Inferences from the multivariate kernel density estimate will also be included.
10. **Data Visualization:** The dataset will be visualized using static plots. Each plot will be accompanied by an observation section detailing the insights gained from the visualization.
11. **Subplots**: Subplots will present data in a comparative format and inferences gained from these subplots.
12. **Tables:** Tabulated data will be presented with insights from their analysis.
13. **Dashboard:** This section will present an interactive dashboard using Dash and detail its features and functionalities.
14. **Conclusion:** This crucial section will reflect on the project's learning outcomes. It will discuss the insights derived from the visualizations, the utility of the Python dashboard, user-friendliness, and the app's functionality based on user feedback gathered through social media platforms.
15. **Appendix:** A compilation of the Python code developed for the project will be included in a separate appendix section, serving as a technical reference.

16. **References:** A list of all the scholarly and technical sources referenced throughout the report, formatted according to the appropriate citation style.

In the subsequent sections, I will cover each section in detail.

# DATASET DISCREPTION

*This dataset consists of almost 1,60,000 rows describing the likes,dislikes,views and count of comments given by users to all videos.It has a total of 4 Numerical,10 Categorical and 3 Boolean feature.*

## Variables in the Dataset

- **Dependent Variable (Target):** From this dataset, I considered 'views' as my target variable deciding to predict the views dependency on other features.
- **Independent Variables (Features):** The dataset includes various independent variables given in the table below:

| Variable | Type | Category | Description |
|---|---|---|---|
| views | int | Numerical | Views of video |
| likes | int | Numerical | Likes of video |
| dislikes | int | Numerical | Dislikes of video |
| comment_count | int | Numerical | comment count of video |
| Video_id | str | Categorical | video id of youtube video |
| title | str | Categorical | title of released video |
| channel_title | str | Categorical | Youtube channel name |
| category_id | int | Categorical | category id of video |
| publish_date | str | Categorical | Published date in Youtube |
| time_frame | str | Categorical | time gap in day |
| published_day_of_week | str | Categorical | published weekday |
| publish_country | str | Categorical | Published Country |
| tags | str | Categorical | hash tags of videos |
| comments_disabled | Boolean | Categorical | comments disabled/not |
| ratings_disabled | Boolean | Categorical | ratings disabled/not |
| video_error_or_removed | Boolean | Categorical | video error or removed |

Table 1 Description of Dataset

## Satisfying the Dataset Criteria

1. **Multivariate with sufficient observations:** The dataset contains over 1,60,000 observations, well above the 50,000 minimum requirement. It includes a wide range of both numerical and categorical data.
2. **Numerical & Categorical Data:** There are several numerical data points (e.g., likes, views) and categorical data points (e.g., channel_title,video_id).
3. **Public Database:** Youtube.csv is a publicly accessible dataset from Kaggle. So the dataset meets the criteria for non-classified data.

## Importance of Dataset in Industry

The dataset is purely based on the views,likes,dislikes and the comments count of all the youtube videos timed in between the year 2017 and 2018.The thorough visualization of the data will be easily giving the insights about the highest number of views,dislikes etc.This will be elevating the users interaction and their ideology at a particular phase.This gives a straightaway visual about the kind of the topics/genres users are highly interested/not interested in.Youtube can also take the highest disliked video/content personally and decide to keep or remove it for maintaining dignity which some people don't have!!

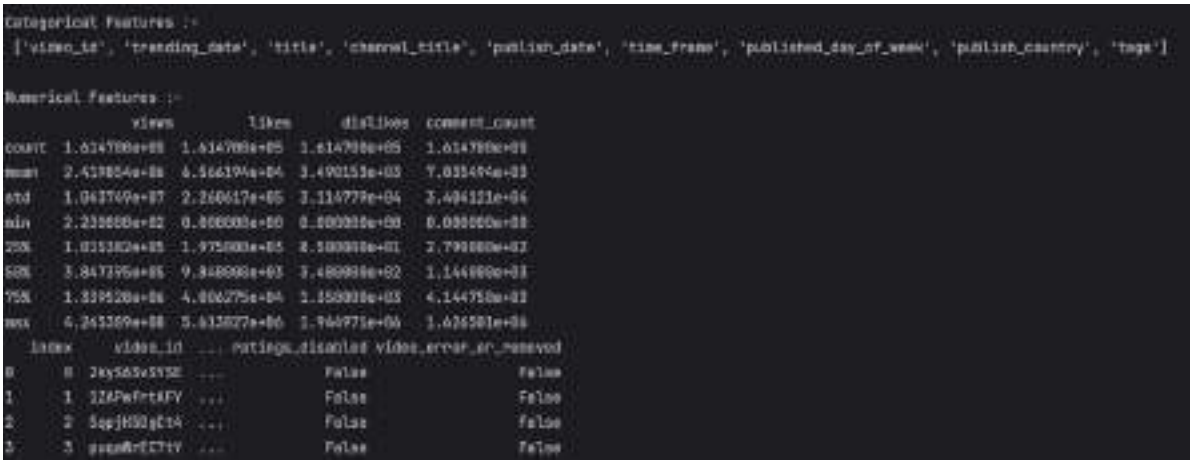
# DATA PRE-PROCESSING

In the data pre-processing stage of our analysis, a rigorous approach was used to ensure the feasibility and quality of the Youtube videos dataset. The following steps were performed to clean the dataset :-

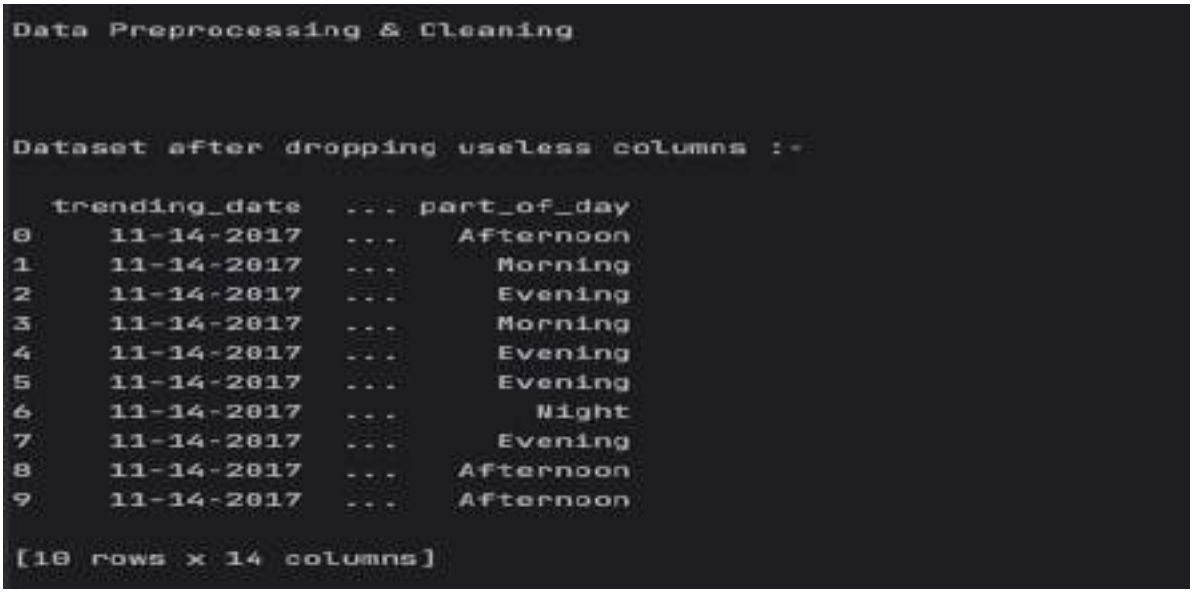
1. **Dropping Unnecessary**

**Columns**: index,video_id,tags,video_error_or_removed,comments_disabled,ratings_disabled are dropped because these fields, predominantly containing textual and URL data, were extraneous for the data visualizations ahead.

**Before Dropping the Columns :-**



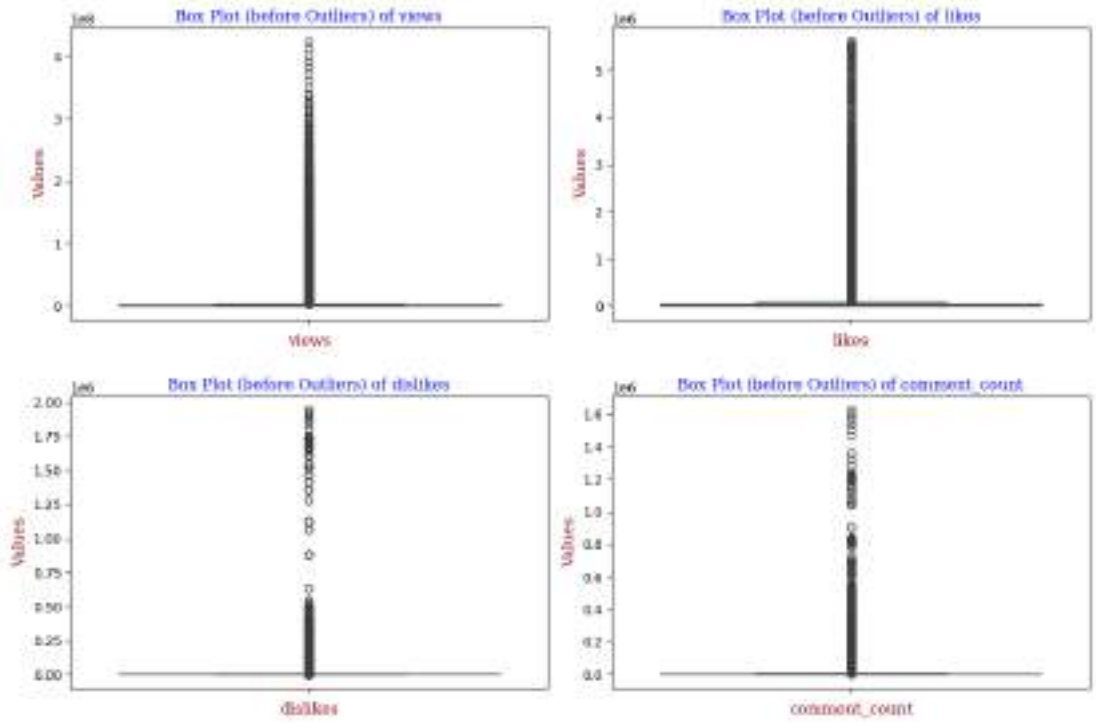

**After Dropping the Columns** :-




2. **Handling Missing Values**: I verified for NULL and NaN values across the dataset and identified the columns containing NULL values. I removed rows where 'views' and 'likes' were missing since these are crucial for any dataset analysis.

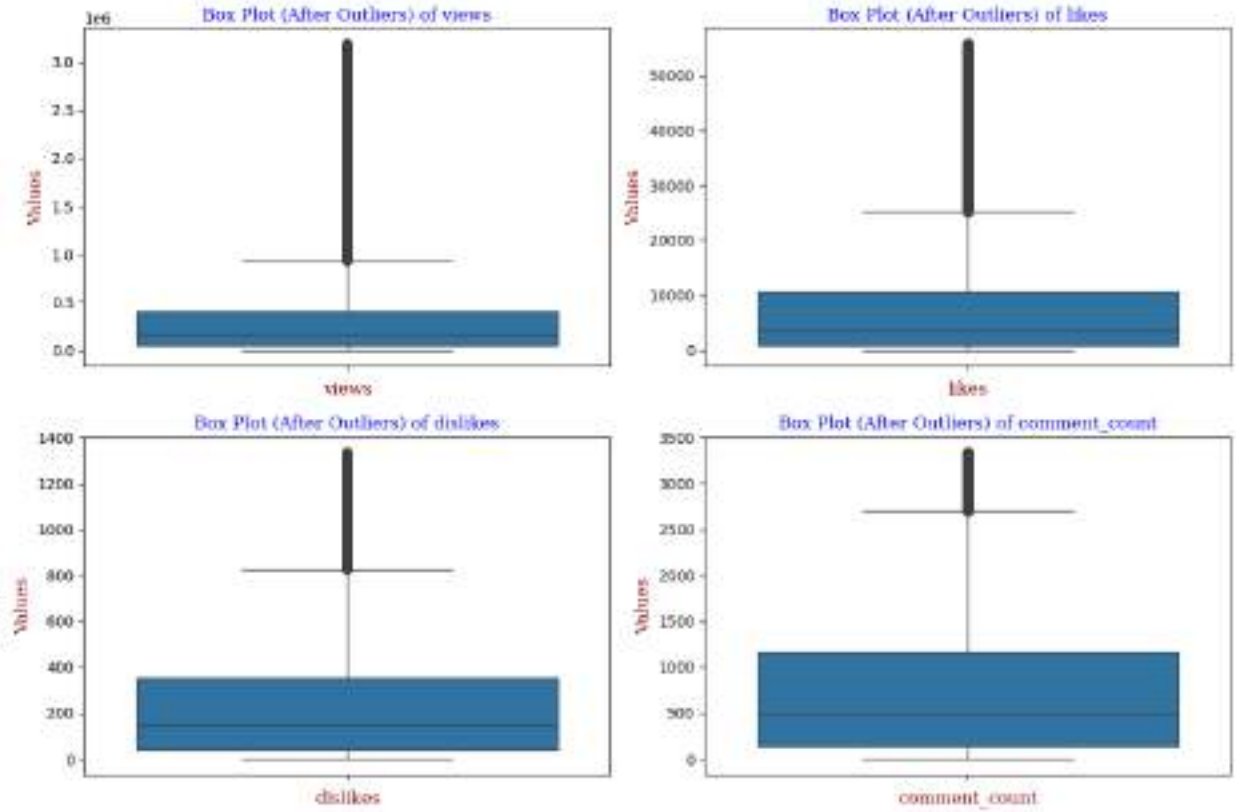
# OUTLIER DETECTION AND REMOVAL

Data points that substantially depart from the general pattern of data in a dataset are called outliers. Machine learning models that are trained using outliers may be skewed and misled, producing models that are less accurate. Therefore, one of the most important steps in pre-processing data for statistical analysis and machine learning is outlier detection and removal. I created a subplot of boxplots using the numerical columns views,likes,dislikes and comment_count in order to look for outliers.




Here's an interpretation of each plot:

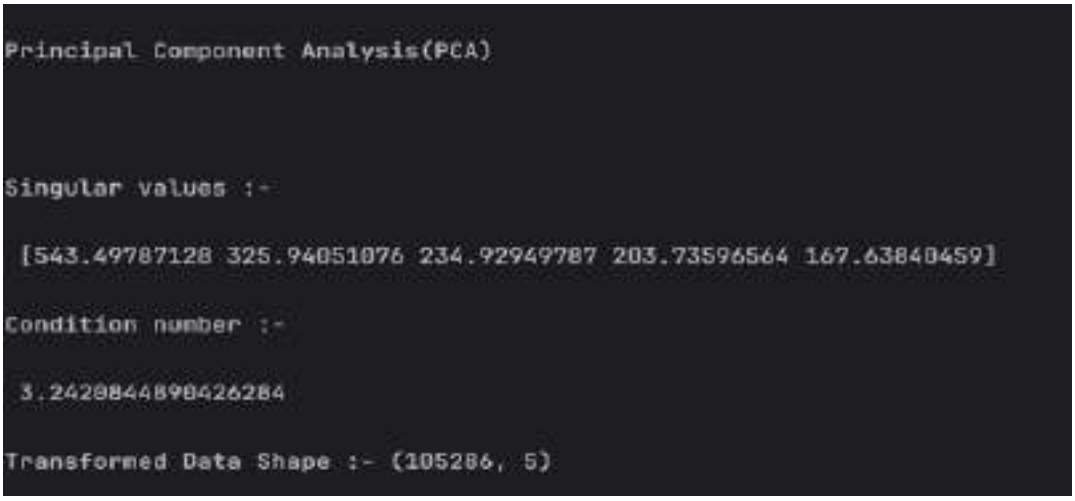
- There are many outliers on the upper end, indicating that some videos have exceptionally high view counts compared to the rest.
- The median likes seem to be quite low relative to the maximum, suggesting that the average video receives a modest number of likes.
- The median is closer to the bottom of the range, indicating that most videos receive fewer dislikes.
- The median comment count is low relative to the range, indicating that most videos do not receive a large number of comments.




- All the outliers are cleared and now my data is OUTLIER FREE!!

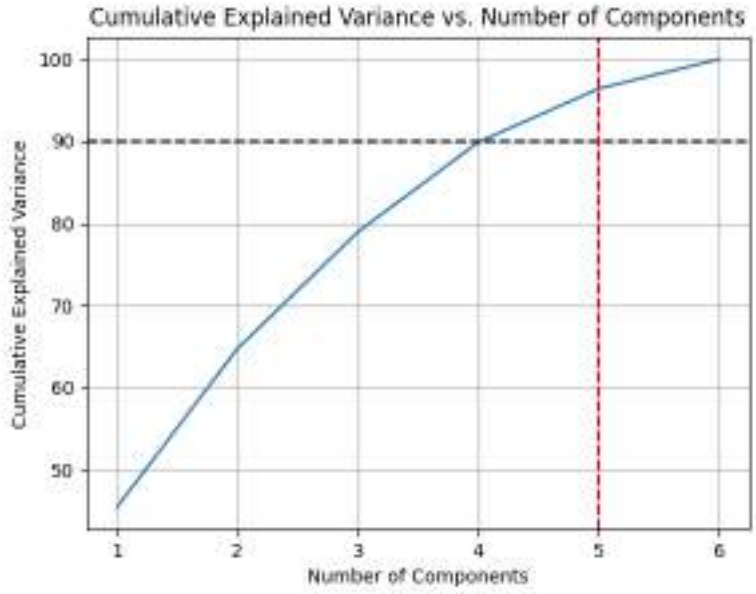
# PRINCIPAL COMPONENT ANALYSIS

After thoroughly cleaning the dataset, I performed Principal Component Analysis (PCA) for dimensionality reduction. The condition number and singular values obtained after performing PCA are shown in the figure below.



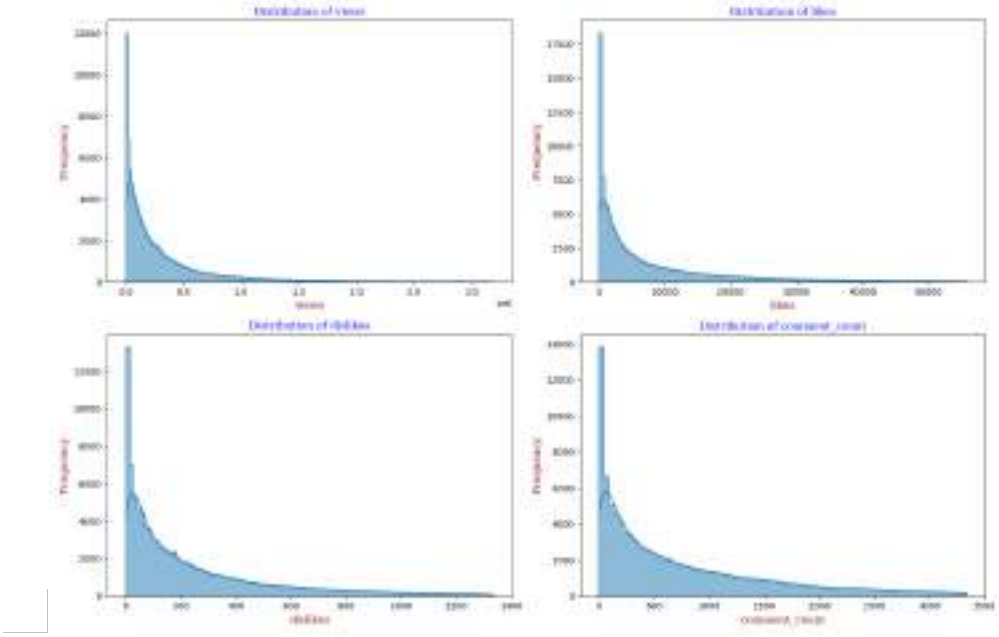

- After the PCA transformation, the data shape is (105286, 5). This indicates that the original dataset with 105,286 observations (rows) has been reduced to 5 principal components (columns). This reduction implies that the original dataset had more than 5 features, and PCA has been used to reduce its dimensionality.
- The condition number of the PCA is relatively low, which is good. A low condition number indicates that the dataset is well-conditioned and the PCA has numerically stable results. It suggests that the data is not excessively collinear.

The Cumulative Explained Variance graph is as shown below:


Cumulative Explained Variance vs. Number of Components
Cumulative Explained Variance
100
90
80
70
60
50
1
2
3
4
5
6
Number of Components

# NORMALITY TESTS

The distributions of the dataset's primary numerical features—views,likes,dislikes,comment_count—were analyzed in this section. These metrics' distributions play a crucial role in both summarizing the central tendencies and dispersions and directing the methodological strategy for additional statistical analysis. Each metric's histogram was plotted to give the data distributions a visual representation. I started my analysis with these histograms, which let me look at each feature's symmetry, skewness, and potential outlier presence.



Below are some observations about the distributions that can inform the interpretation of normality tests :-

1. **Distribution of Views:**
    - The distribution is highly right-skewed, with a peak at the lower end, indicating that most videos have a relatively low number of views, with a few videos having a very high number of views.
2. **Distribution of Likes:**
    - Similarly to views, the distribution of likes is also right-skewed. This suggests that most videos have fewer likes, while a small number have a high number of likes.
3. **Distribution of Dislikes:**
    - The distribution of dislikes shows an even more pronounced right skew. The frequency of dislikes is heavily concentrated at the lower numbers, indicating that most videos receive a small number of dislikes.
4. **Distribution of Comment Count:**

- The comment count distribution mirrors the pattern of the other metrics, being right-skewed. This implies that most videos have fewer comments, with the frequency dropping off sharply as the number of comments increases.

To verify these results further, I performed a statistical test, the Kolmogorov–Smirnov test, to assess the normality of the runtime variable.
**Null Hypothesis (H0):** The data follows a normal distribution.
**Alternative Hypothesis(H1):** The data does not follow a normal distribution.
I assumed the significance level(alpha) to be 0.05. If the p-value is greater than or equal to the significance level, we accept the null hypothesis and assume the variable to be normally distributed.
The below image shows the result of the test.

```
Kolmogorov-Smirnov Test

Column :- transformed_views
K-S Statistic :- 0.05462722666489872, P-value :- 3.493173396028142e-263
Not Normally distributed

Column :- transformed_likes
K-S Statistic :- 0.05180620210451366, P-value :- 9.880557144850730e-237
Not Normally distributed

Column :- transformed_dislikes
K-S Statistic :- 0.06898116236242502, P-value :- 0.0
Not Normally distributed

Column :- transformed_comment_count
K-S Statistic :- 0.07320401665762633, P-value :- 0.0
Not Normally distributed
```

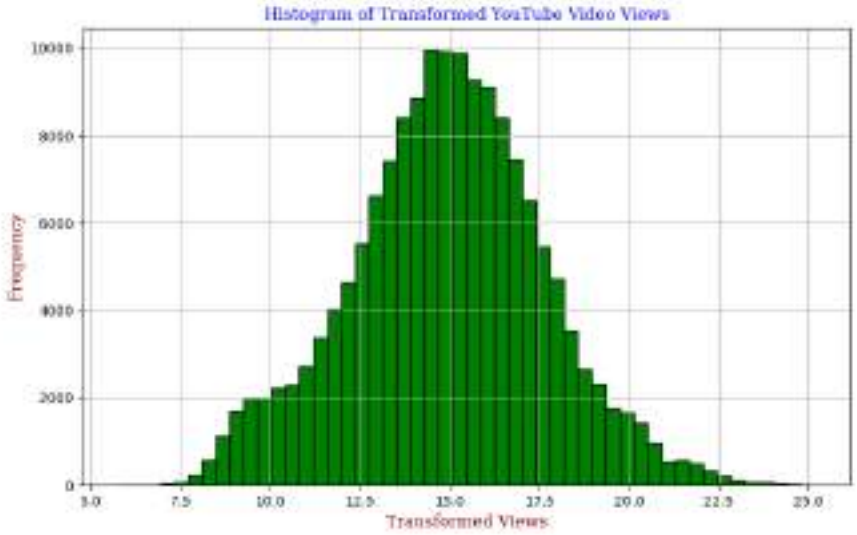
Interpretation :-

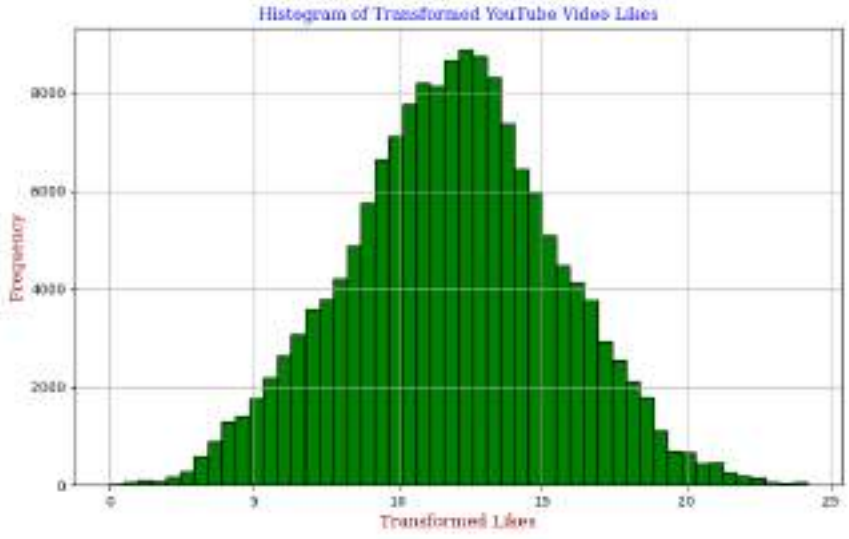
1. **Statistic = 0.07:** The K-S statistic measures the maximum distance between the data's observed cumulative distribution and the theoretical normal distribution's cumulative distribution. A value close to 1 indicates a large discrepancy between the two distributions.
2. **p-value = 0.0:** A p-value of 0 indicates strong evidence against the null hypothesis.

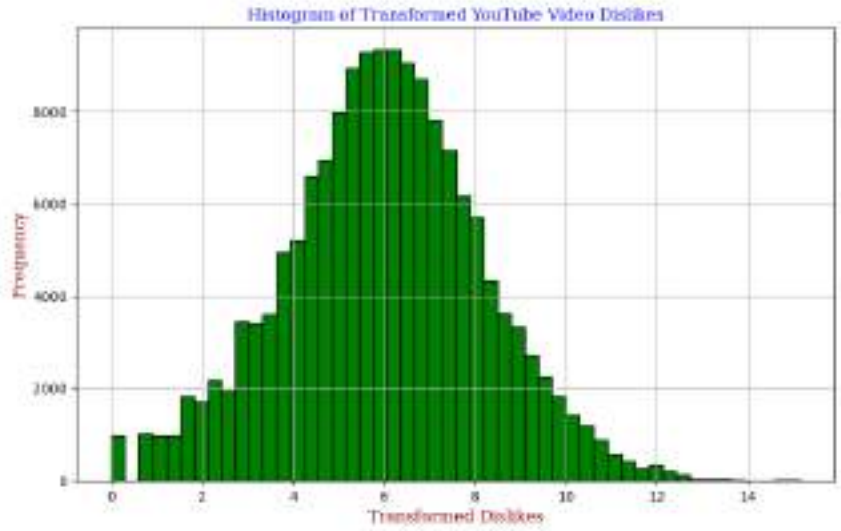
Given these results, we reject the null hypothesis and conclude that the runtime data do not come from a normal distribution. The outcome is consistent with the histogram, where the data appeared to be bimodal, and it supports the observation that the runtime data are not normally distributed.

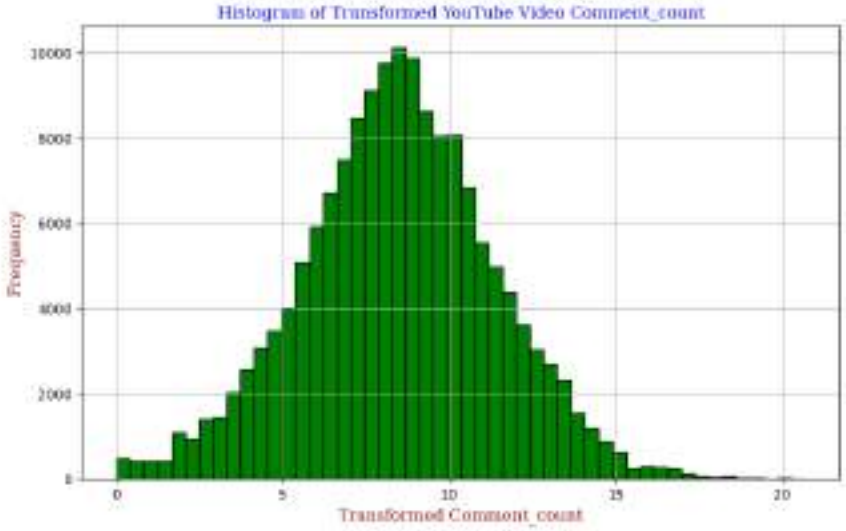
# DATA TRANSFORMATION

The variables views,likes,dislikes and comment_count were evident from the preceding sections. In order to address skewness and non-normality, data transformation was therefore required for these variables.The Box-Cox transformation was selected for these because it can identify the ideal transformation parameter (lambda) that reduces skewness and normalizes the distribution of the data.





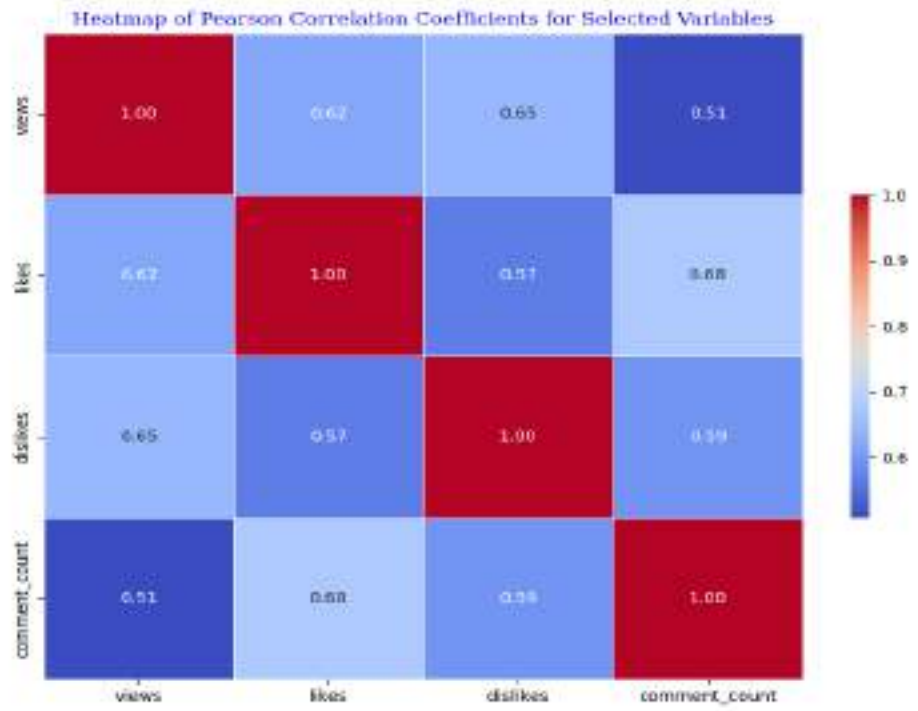

- Hence,the normalized data is obtained here.

# HEATMAP AND PEARSON CORRELATION COEFFICIENT MATRIX

Exploratory data analysis requires an understanding of the relationships between various variables. Two visual tools that can help with this investigation are the scatter plot matrix and the Pearson correlation coefficient heatmap. While the latter provides a more detailed, visual understanding of the shape of the relationship between variable pairs, the former summarizes the strength of the relationship between pairs of variables.

**Heatmap**

The heatmap displays Pearson correlation coefficients ranging from -1 to 1, where 1 indicates a perfect positive correlation, -1 indicates a perfect negative correlation, and 0 indicates no correlation.



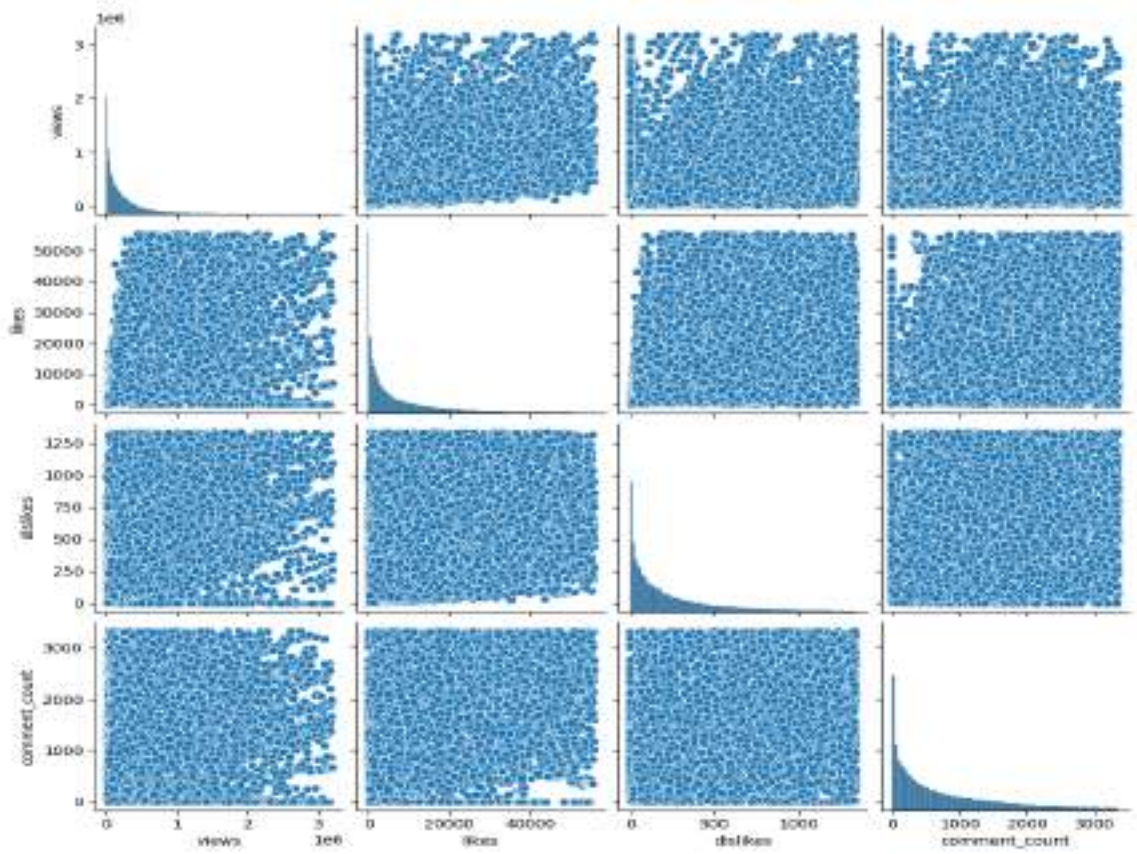
Observations :-

- There is a strong positive correlation between all the pairs of variables. Likes, dislikes, and comment counts are all positively correlated with views, suggesting that as the number of views increases, so does the number of likes, dislikes, and comments. This indicates that higher visibility leads to higher engagement.

- Likes and comment counts have a relatively high correlation coefficient (0.68), suggesting that videos that receive a lot of likes also tend to receive a lot of comments. This could imply that users who engage with content by liking it are also likely to engage in the comments section.

## Scatter Plot Matrix

The scatter plot matrix complements these findings by providing a visual representation of the data points and their distribution across two variables.



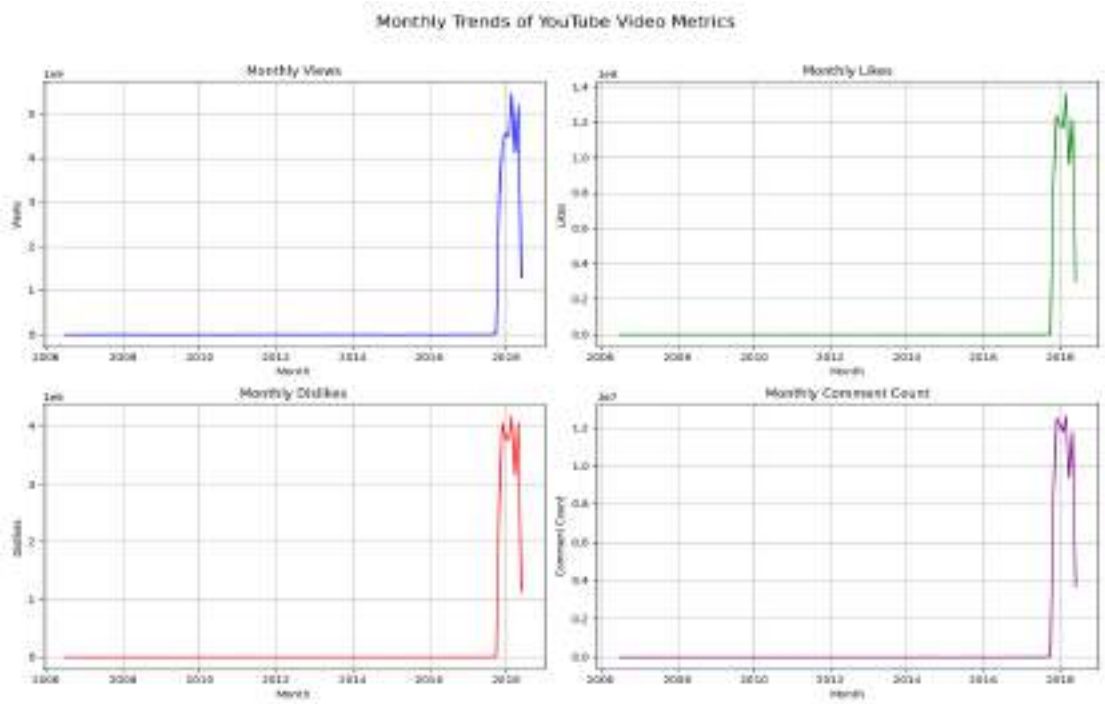
INFERENCES :-

- For views, likes, dislikes, and comment counts, there is a heavy concentration of data points near the origin. This suggests that most videos have low engagement metrics, which is consistent with the presence of a long tail distribution in each individual histogram along the diagonal.

- The scatter plots show that higher values in one metric correspond to higher values in another, but the concentration of points near the lower end of the axes indicates that this relationship is stronger among less popular videos.

# DATA VISUALIZATION

Understanding data's underlying patterns, relationships, and insights requires visualizing the data. We can identify patterns, outliers, and correlations that may be hidden in the data by using static plots to obtain a thorough overview of the dataset. In this section, I used static plots to draw conclusions and provide relevant answers to the questions at hand. I've included a set of questions and corresponding plots below, along with a thorough explanation of the observations I made from each plot.

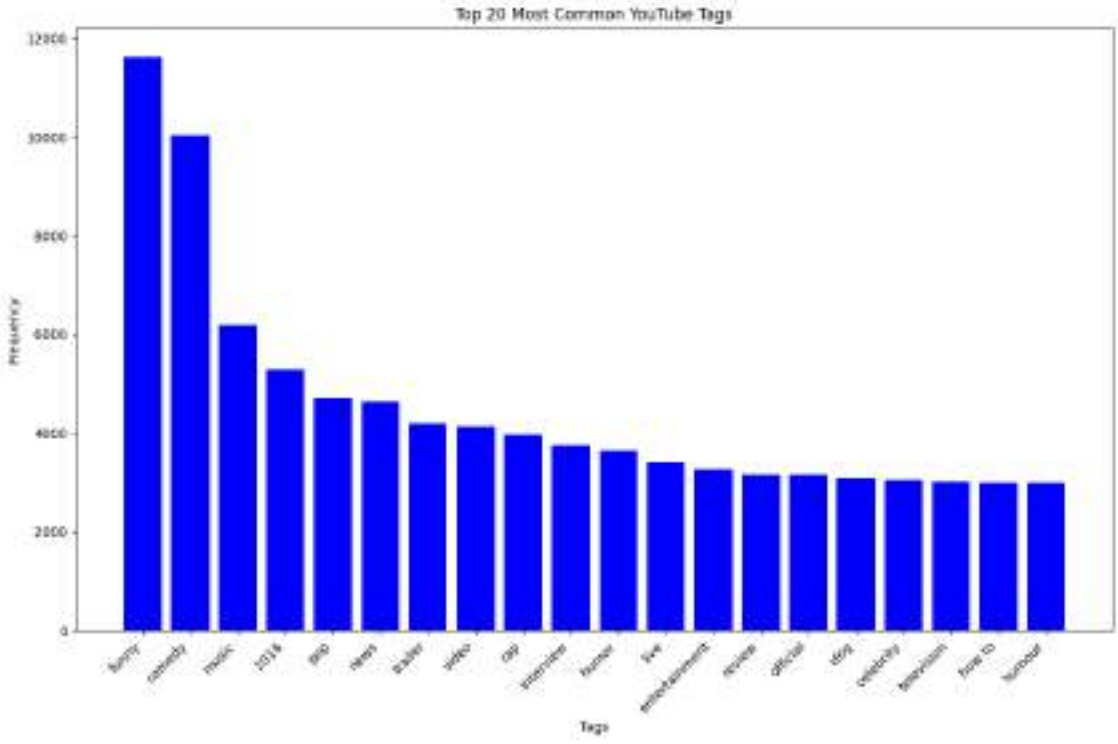



INFERENCES :-

- All four metrics—views, likes, dislikes, and comment count—show a significant increase in recent years, particularly around 2017 and 2018. This suggests that engagement with YouTube content has surged during this period, which could be attributed to a number of factors including increased internet accessibility, more content creators etc.

- The spikes in the graphs for views, likes, and comments appear to be correlated, as they increase around the same time. The dislikes graph also shows a spike, but it's less aligned with the other metrics, which may indicate different viewer behavior with regard to content that is not well-received.

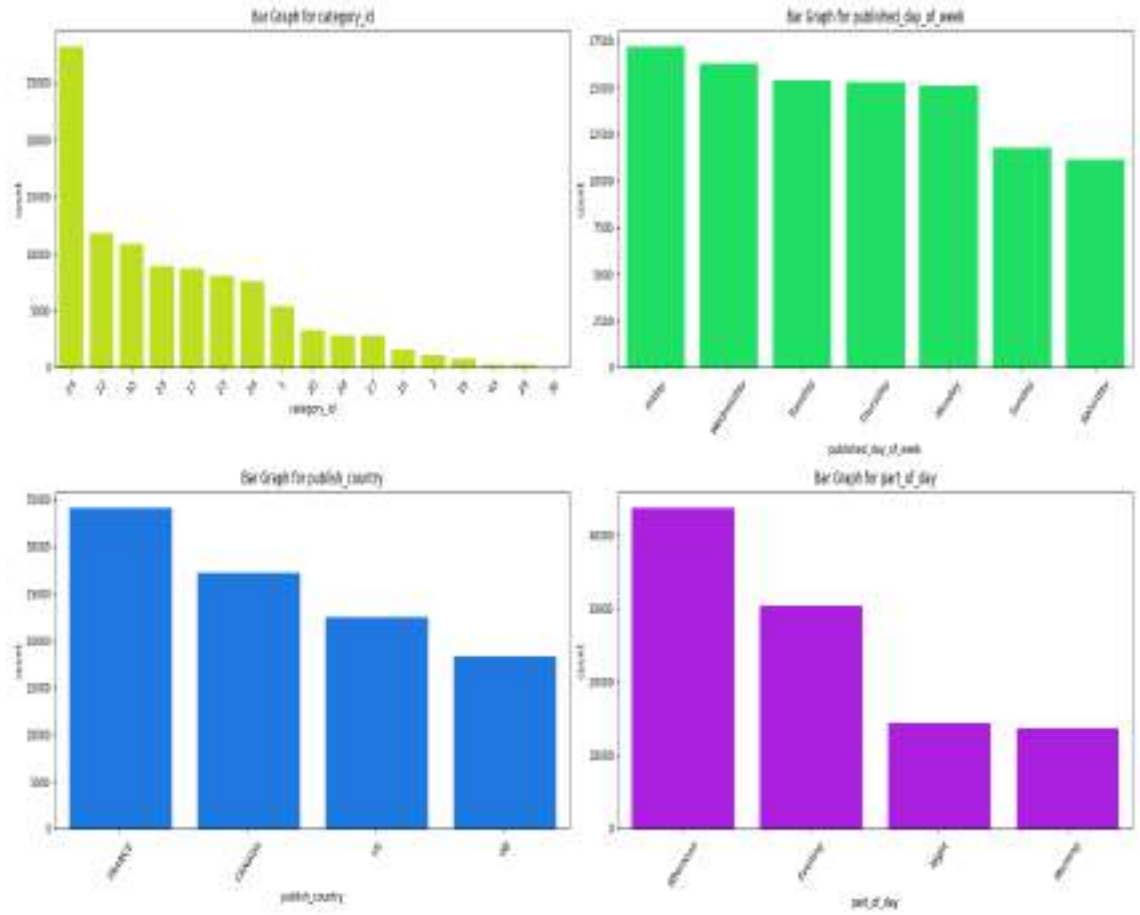
- The sharp peaks, particularly visible in the views and likes graphs, suggest the presence of outlier events, likely viral videos that have captured the attention of a large audience in a short span of time.




INFERENCES :-

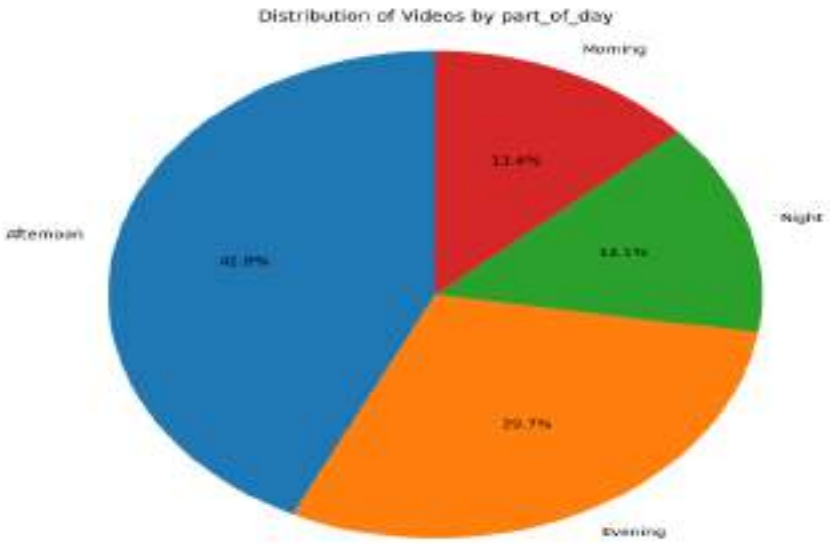
- The tags 'funny' and 'comedy' are the most frequent, indicating that humorous content is highly popular on YouTube.

- The variety of tags, including 'music', '2018', 'pop', 'news', 'trailer', 'video', 'interview', 'humor', 'live', 'entertainment', 'review', 'official', 'vlog', 'celebrity', 'television', and 'how-to' represent a broad range of interests among YouTube users.



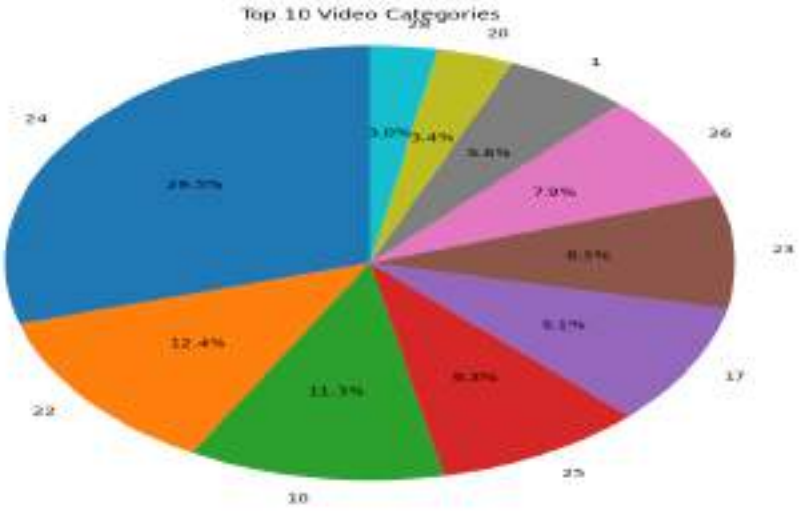

INFERENCES :-

- The first graph shows that some categories (notably the one with category ID 24) are much more common than others.
- The second graph indicates that video publishing is relatively consistent across weekdays, with a slight decrease on Saturday and a more notable drop on Sunday.
- The third graph shows that certain countries (possibly 'RU' and 'FR' based on the visible labels) have higher counts of published videos than others.
- The fourth graph displays the count of videos published during different parts of the day, with 'afternoon' having the highest count followed by 'evening'.



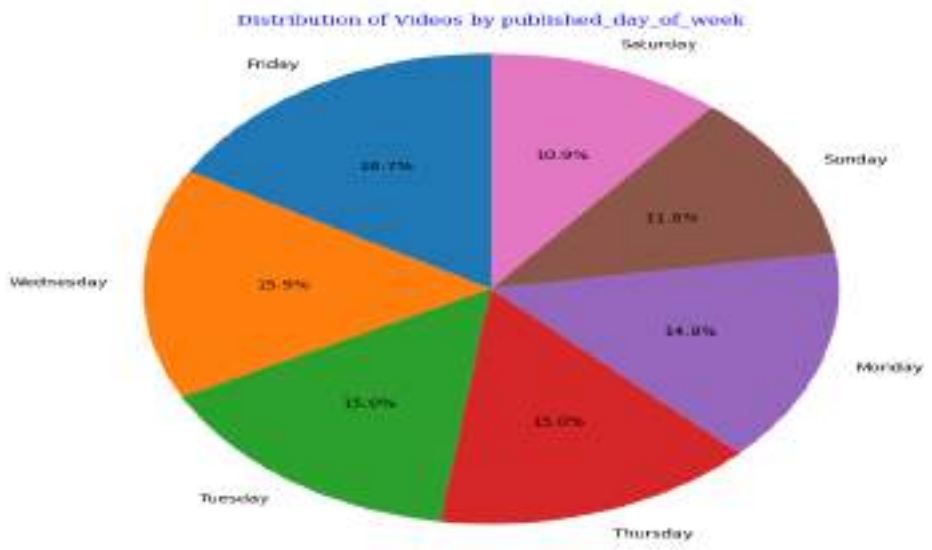

INFERENCES :-

The main inference from the pie chart, which illustrates the distribution of videos by part of the day, is that the afternoon is the most popular time for publishing videos, accounting for approximately 42.8% of the total. This is followed by the evening at 29.7%, suggesting that the latter half of the day is the preferred time for video uploads.



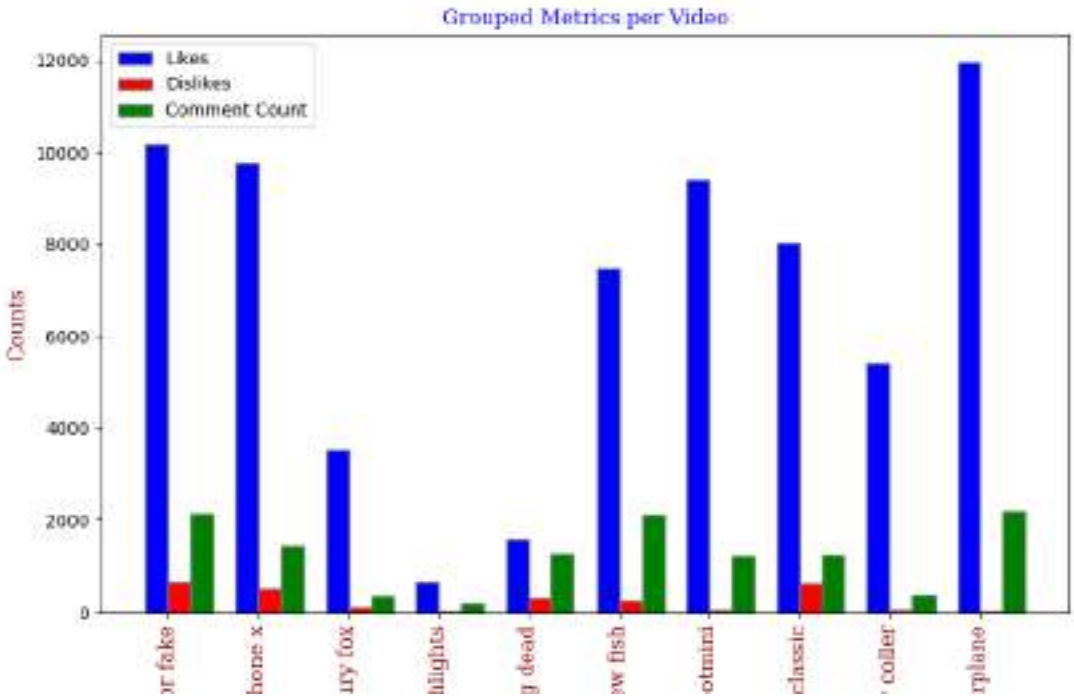
INFERENCES :-

The main inference here is that Category 24 dominates the chart, constituting 29.5% of the videos, which suggests that it is the most popular or most frequently uploaded category on YouTube within this dataset. Other categories represent a smaller share of the total, with Categories 22 and 1 also having significant portions at 12.4% and 11.3%, respectively.



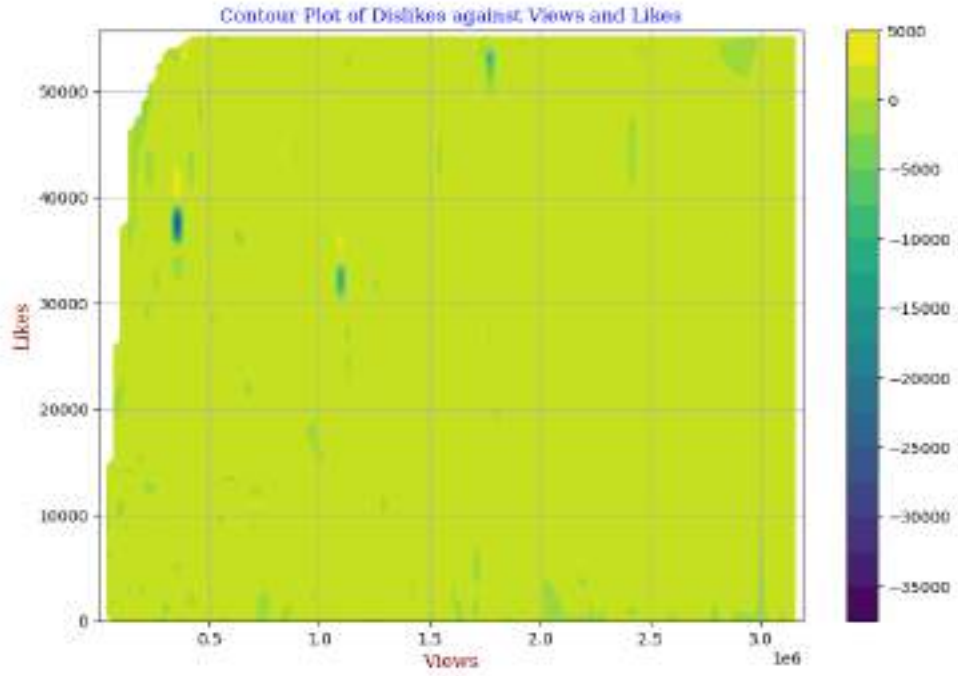

INFERENCES :-

1. **Friday** is the most common day for publishing videos, accounting for 16.7% of the videos in the dataset. This suggests that content creators may favor publishing going into the weekend, possibly to capture viewers when they might have more free time.
2. **Wednesday and Tuesday** follow closely, each with a 15.9% and 15.0% share, respectively. It appears that midweek is also a popular time for video releases.
3. **Monday and Thursday** see a slightly lower but comparable share of video publications, with 14.8% and 15.0%, respectively.
4. **The weekend days**, Saturday and Sunday, have the lowest percentages of video publications, with 10.9% and 11.6% respectively. This could indicate that fewer videos are published on weekends, or it might reflect a strategy of uploading content before the weekend begins.




INFERENCES :-

1. **Likes** dominate the engagement metrics for all video categories, with the counts significantly higher than dislikes and comment counts in every case.
2. **Dislikes** are the least frequent form of engagement across all videos or categories shown.
3. **Comment Counts** are generally higher than dislikes but still much lower than likes. Comments represent a more significant investment of time and effort from a viewer compared to likes or dislikes, which could account for the lower counts.



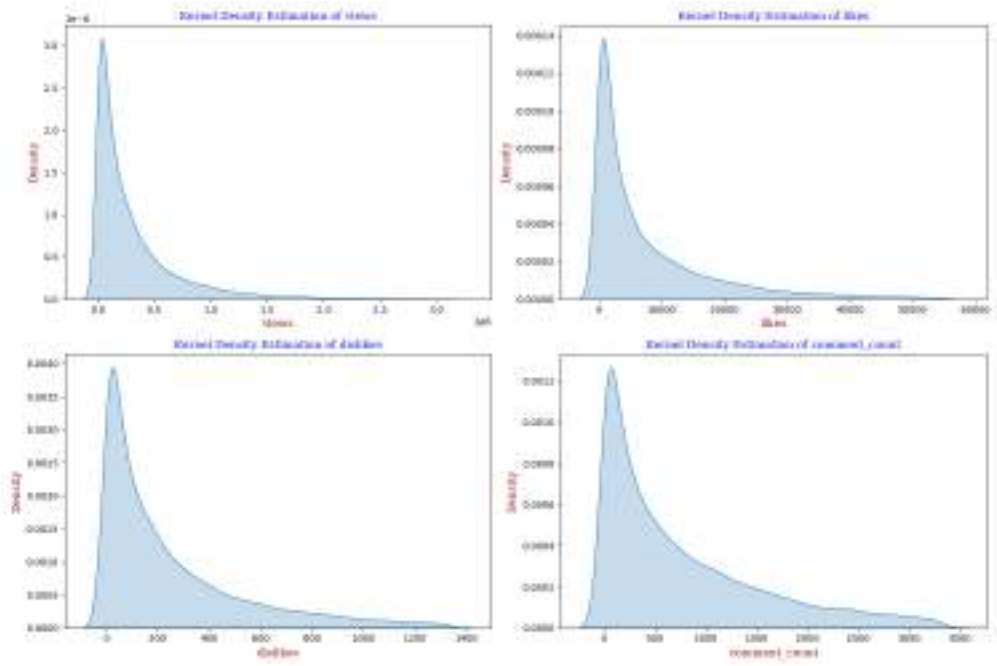

INFERENCES :-

1. **Density of Points**: The concentration of the contours indicates the density of points or the frequency of videos with those specific likes and views. Areas with more contours are denser, meaning more videos have that combination of views and likes.
2. **Likes and Views Range**: The plot shows a wide range of views (up to 3 million) and likes (up to 50,000). However, the majority of the data seems to be concentrated in the lower-left corner, indicating that most videos have fewer likes and views.
3. **Dislikes Representation**: The dislikes are not explicitly plotted on the axes but are likely represented by the contour lines themselves. The color bar on the right suggests a range of values that are presumably associated with the number of dislikes.
4. **Anomalies**: There are distinct spots or areas with a higher concentration of contour lines (the blue and purple spots), suggesting clusters of videos with a higher ratio of

dislikes to likes and views. This could represent videos that were particularly unpopular or controversial.

5. **General Trend**: The majority of the plotting area is a uniform color, indicating a relatively consistent ratio of dislikes to likes and views across most of the dataset.

6. **Potential Issues**: The color scale includes negative values, which doesn't make sense in the context of counts of likes, dislikes, or views. This could indicate an issue with the data, the plotting code, or the scale of the color bar.



INFERENCES :-

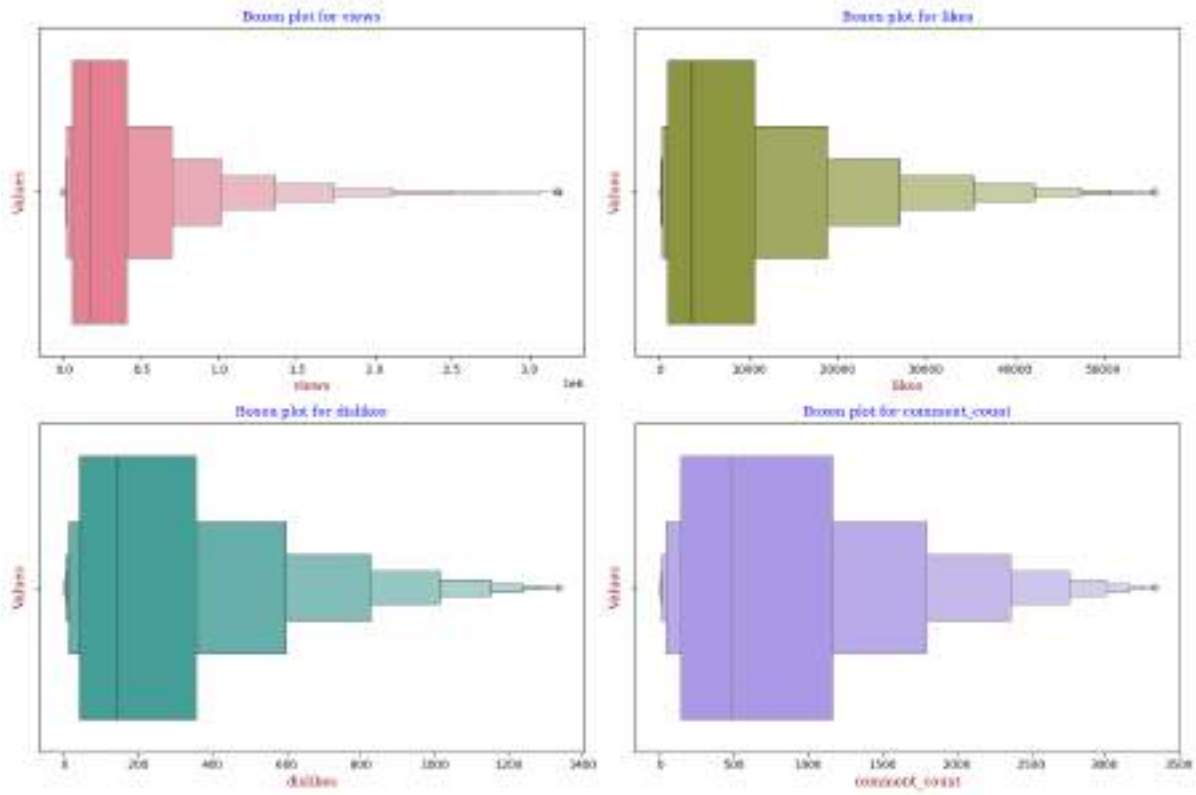
1. **Right-Skewed Distributions**: All four plots show distributions that are heavily right-skewed, with a peak at the lower end of the scale. This indicates that a large number of observations have low values for views, likes, dislikes, and comment counts, with fewer observations having very high values.

2. **Long Tails**: Each plot has a long tail stretching to the right, suggesting the presence of outliers with very high counts for views, likes, dislikes, and comment counts. This is characteristic of data where a few items (such as viral videos) have very high engagement metrics.

3. **Views**:
    - The distribution of views has the highest peak, which suggests that most videos have a relatively low number of views.
4. **Likes**:
    - The likes distribution also has a significant peak near zero, indicating that many videos receive a small number of likes.

5. **Dislikes**:
    - The dislikes distribution has the lowest height among the peaks, suggesting a lower overall occurrence of dislikes compared to likes and views. It also indicates that the majority of videos receive very few dislikes.

6. **Comment Count**:
    - The comment count distribution is similar to that of dislikes but with a slightly higher peak, suggesting a slightly greater frequency of videos with a low to moderate number of comments.



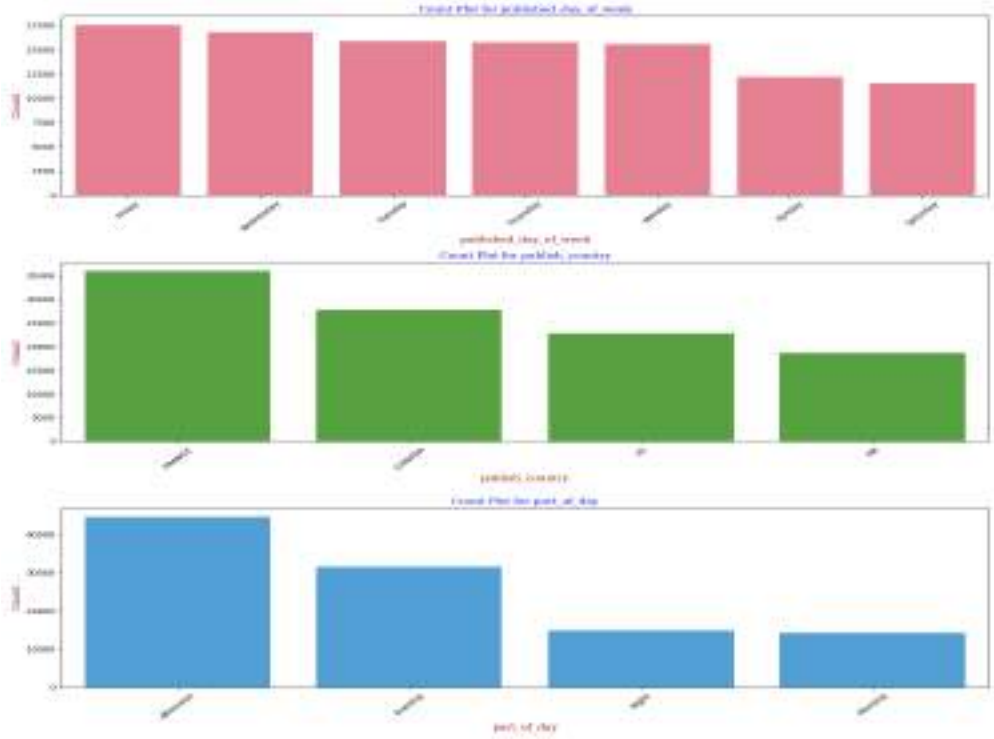
INFERENCES :-

1. **Views:**
    - The distribution of views is right-skewed with a large number of outliers, which are indicated by the points beyond the whiskers.
    - The plot shows that the median is closer to the lower quartile, indicating a concentration of lower values.

2. **Likes:**
    - The likes distribution is also right-skewed with many outliers.
    - The median of likes is closer to the lower end of the interquartile range, suggesting that most videos have fewer likes.

3. **Dislikes:**
    - The dislikes have a slightly less skewed distribution compared to views and likes but still show right skewness with outliers present.
    - The median dislikes are low, as indicated by the position within the lower half of the box.

4. **Comment Count:**
    - The distribution of comment counts is right-skewed with numerous outliers, similar to the distributions of views and likes.
    - The median comment count is also on the lower side of the box, indicating that most videos have fewer comments.

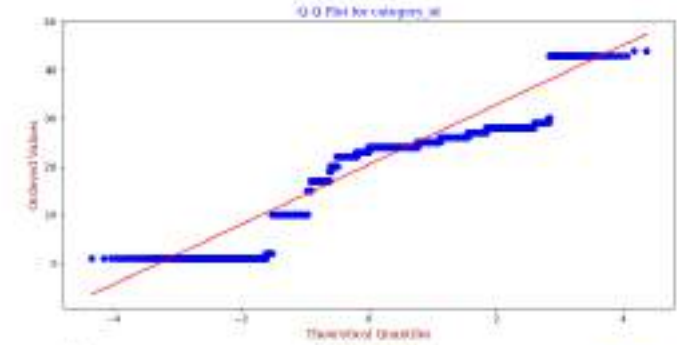


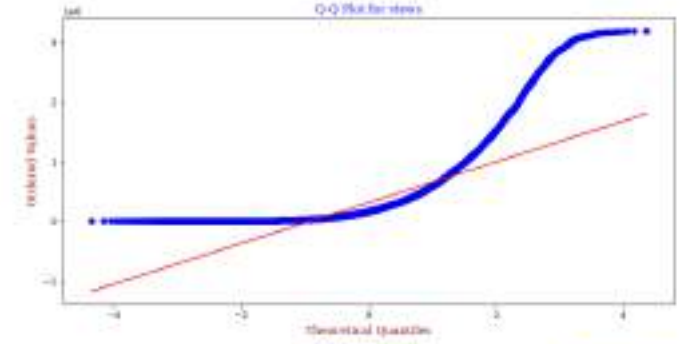
INFERENCES :-

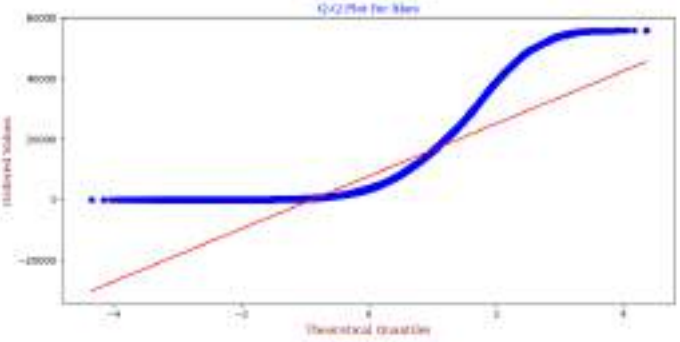
1. **Published Day of the Week:**
    - Videos are published fairly evenly across the weekdays, with slight variations.
    - Saturday and Sunday have a noticeably lower number of video publications compared to weekdays, indicating that fewer videos are published on weekends.

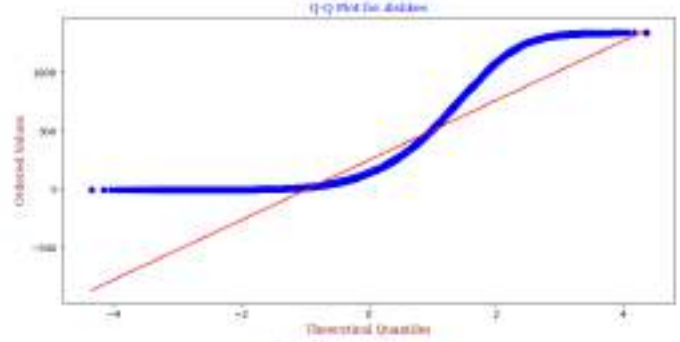
2. **Publish Country:**
    - The count plot for publish country shows that the dataset contains videos from four different countries, with the label 'US' likely referring to the United States.
    - The country represented by 'FRANCE' has the highest number of video publications in this dataset, followed by 'CA' (presumably Canada) and 'US'.
    - The country represented by 'DE' (possibly Germany) has the fewest video publications among the listed countries.

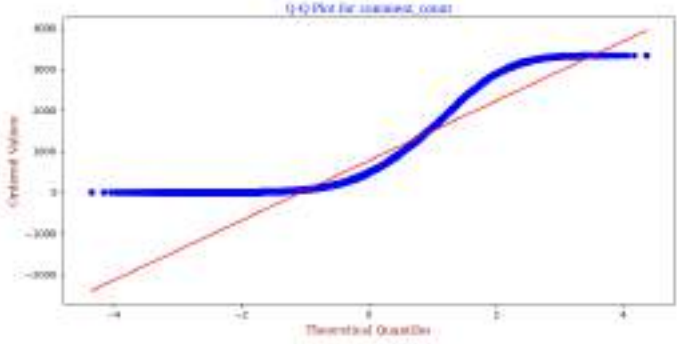
3. **Part of Day:**
    - Videos seem to be published in two parts of the day, 'morning' and 'afternoon'.
    - The morning sees the most video publications, followed by the afternoon, with the evening and the night having significantly fewer publications.


Q-Q Plot for category_id
Ordered Values
Theoretical Quantiles


Q-Q Plot for views
Ordered Values
Theoretical Quantiles


Q-Q Plot for likes
Ordered Values
Theoretical Quantiles


Q-Q Plot for dislikes
Ordered Values
Theoretical Quantiles


Q-Q Plot for comment_count
Ordered Values
Theoretical Quantiles

INFERENCES :-

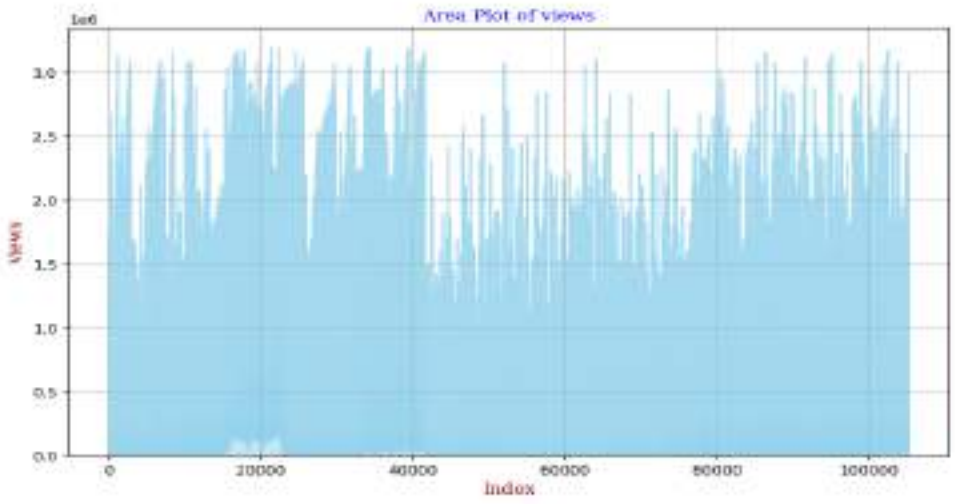
1. **Category ID:**
    - The Q-Q plot for category_id shows that the data points lie relatively close to the line, especially in the middle quantiles. However, since category IDs are typically nominal data, their distribution on a Q-Q plot is not particularly meaningful for assessing normality.

2. **Views:**
    - The Q-Q plot for views shows a pronounced deviation from the line, especially at the higher quantiles, indicating that the distribution of views is heavily right-skewed and not normally distributed.

3. **Likes:**
    - Similarly to views, the distribution of likes is not following the line in the Q-Q plot, particularly at the higher quantiles, which suggests that the likes data is also not normally distributed and is right-skewed.

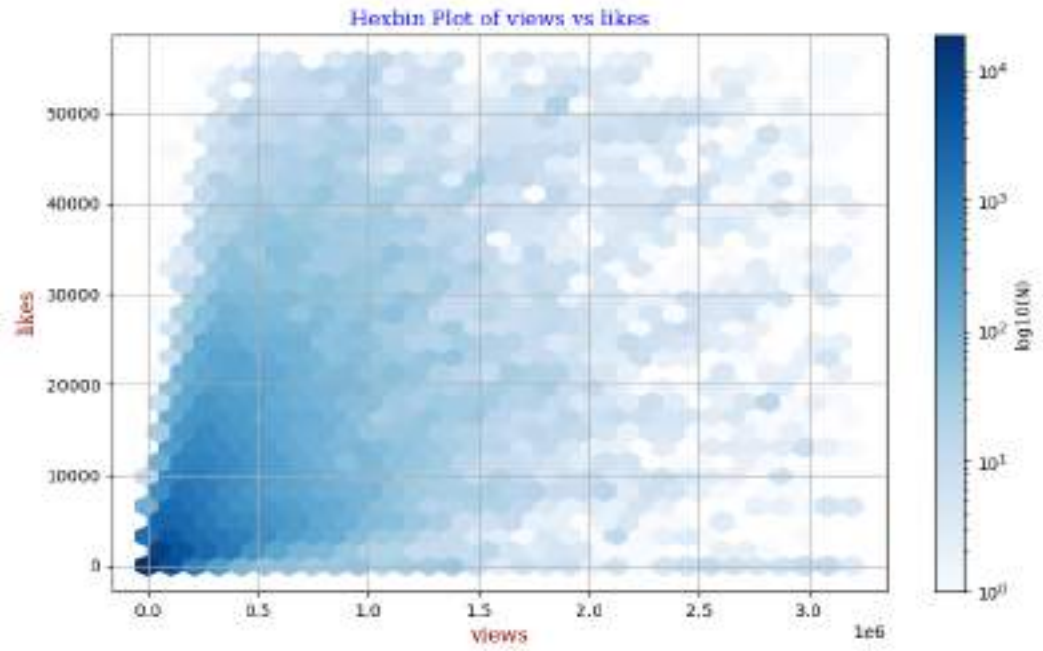
4. **Dislikes:**
    - The dislikes Q-Q plot also diverges from the line at the upper quantiles, indicating a right-skewed distribution.

5. **Comment Count:**
    - The comment count has a similar pattern to likes and dislikes, with the data points deviating from the theoretical line, especially at higher quantiles, suggesting a right-skewed distribution.



INFERENCES :-

1. **High Variability**: The views show high variability, with many spikes reaching up to 3 million views. The variability indicates that the views per video or content piece are not consistent across the dataset.
2. **Peak Patterns**: There are several peaks which suggest that certain videos or pieces of content have significantly more views than others. These could be outliers, viral content, or content from popular creators or topics.
3. **Data Volume**: The index on the x-axis suggests a large volume of data points, which implies a large dataset. The plot appears to extend beyond 100,000 on the index, which could correspond to a substantial number of videos or a long timeframe over which the data was collected.
4. **Distribution**: The distribution of views is not uniform, and the darker area at the lower part of the plot indicates a concentration of videos with fewer views, while the lighter area extending upwards reflects the presence of videos with a wide range of higher view counts.



INFERENCES :-

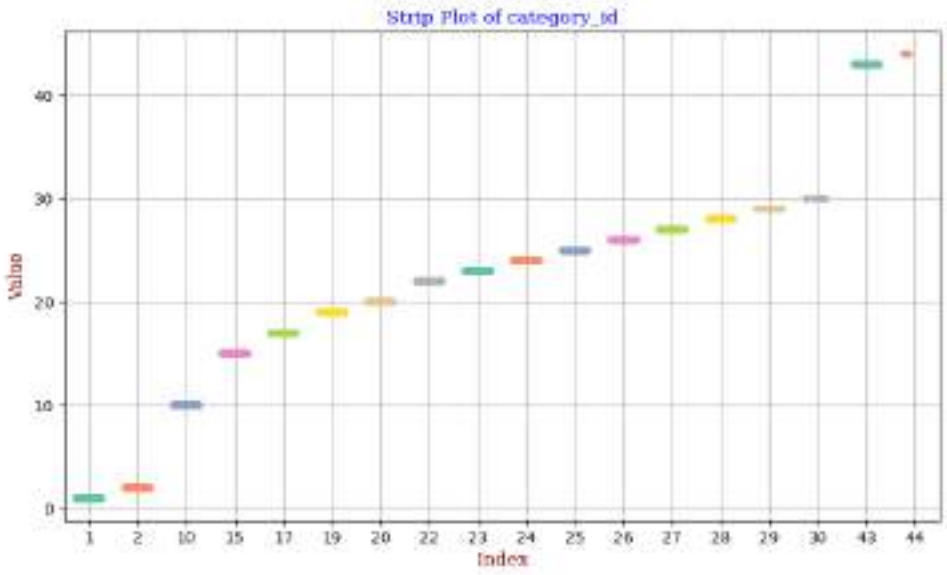
1. **Concentration of Data Points**: There's a high concentration of data points in the lower left corner of the plot. This indicates that most videos have a relatively low count of both views and likes.

2. **Positive Relationship**: The distribution of the hexagons suggests a positive relationship between views and likes — as the number of views increases, the number of likes also tends to increase.

3. **Density and Outliers**: The color intensity represents the density of points within the hexagons, with darker colors indicating higher densities. The darkest area is near the origin, showing that the majority of the videos have few views and likes. Lighter hexagons scattered towards the top right indicate fewer occurrences of videos with extremely high views and likes.

4. **Long Tail Distributions**: Both views and likes display long tail distributions, evidenced by the fact that the density of points decreases as you move up and to the right, but there are still occurrences in these higher ranges.

5. **Scaling**: The color bar on the right, which shows the log count, confirms that the density of points varies over several orders of magnitude. The logarithmic scale helps visualize a wide range of densities on the plot.



INFERENCES :-

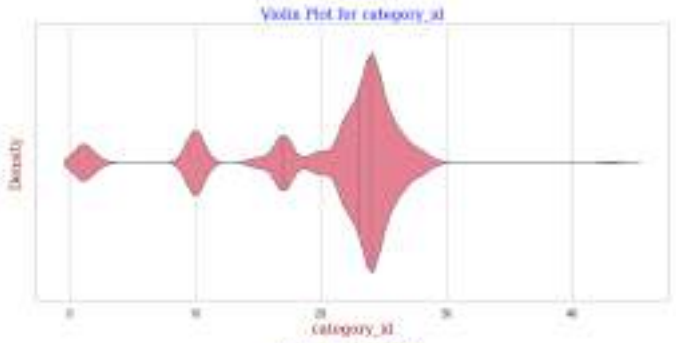
1. **Distinct Categories**: The x-axis, labeled 'Index', appears to represent different categories by their IDs. Each dot corresponds to a unique value in the 'category_id' variable.

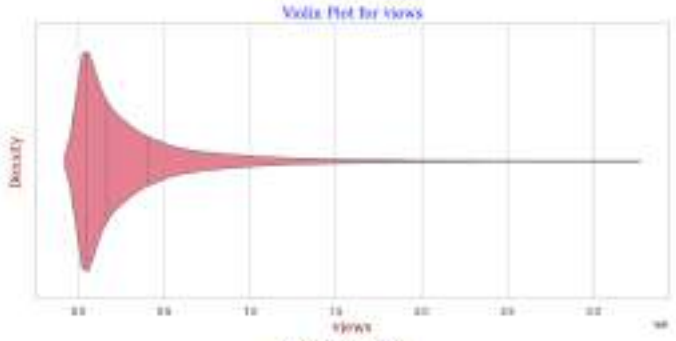
2. **Non-Numerical Category Values**: The 'category_id' values are treated as categorical despite being numerical because they represent distinct, non-ordered groups.

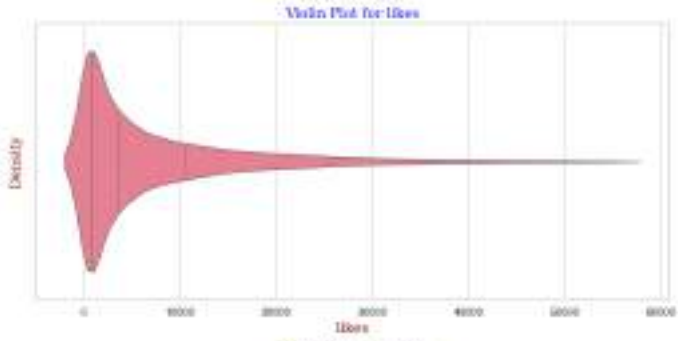
3. **Sparse Categories**: Some categories, such as those corresponding to index numbers 1, 2, and 43, have only one or two videos, suggesting that they are less common or less populated within this dataset.

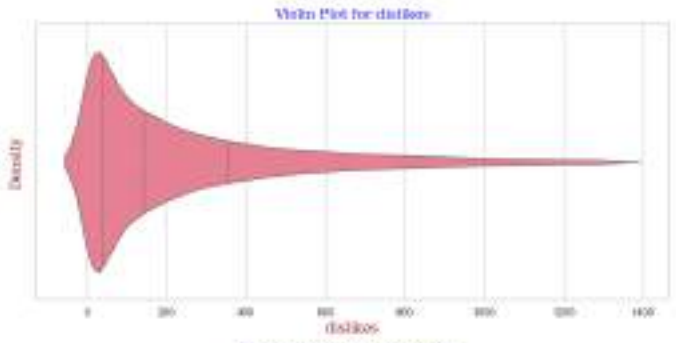
4. **Most Populated Categories**: The most populated categories seem to be those indexed at 10, 15, 17, 22, 23, 24, 25, and 26, indicated by the presence of more dots. These categories could represent more popular or common video types.
5. **Outliers or Unique Cases**: There are some categories with values significantly higher than others (as indicated by the dots far above the x-axis line), such as the one at index 44 reaching up to 40. These could be outliers or unique cases within the dataset.

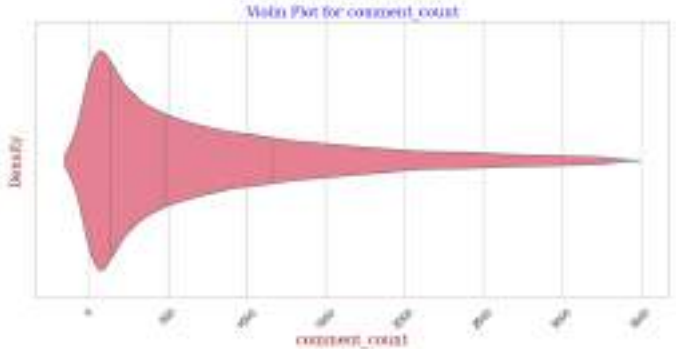
6. **Uniformity Across Categories**: The plot does not show any significant variance in values within most categories, which could imply that each category may only have one value associated with it in this dataset, or the plot is not detailed enough to show the spread within each category.


Violin Plot for category_id
Density
category_id


Violin Plot for views
Density
views


Violin Plot for likes
Density
likes


Violin Plot for dislikes
Density
dislikes


Violin Plot for comment_count
Density
comment_count

INFERENCES :-

1. **Category ID:**

    - The violin plot for category_id shows multiple peaks, suggesting that there are several categories with a high number of videos. This indicates that the data is not uniformly distributed across categories.

2. **Views:**
    - The plot for views is highly skewed to the right, with a long tail stretching toward higher view counts. This indicates that while most videos have a relatively low number of views, there is a significant number of videos with very high views.

3. **Likes:**
    - The distribution of likes is also right-skewed, similar to views, but the density is thinner for higher counts of likes, which suggests that high numbers of likes are less common than high numbers of views.

4. **Dislikes:**
    - Dislikes show a right-skewed distribution with a less pronounced tail than likes or views. This suggests that videos typically receive fewer dislikes compared to likes and views.

5. **Comment Count:**
    - The comment count has a right-skewed distribution with a long tail, indicating that most videos have fewer comments, but a small number have a very high comment count.

# TABLES

Heat Map,Scatter plot and Table

| | views | likes | dislikes | comment_count |
|---|---|---|---|---|
| views | 1.0 | 0.6187788331863184 | 0.6455818983226698 | 0.5080672252852084 |
| likes | 0.6187788331863184 | 1.0 | 0.5650129197704388 | 0.6754405001311277 |
| dislikes | 0.6455818983226698 | 0.5650129197704388 | 1.0 | 0.5932920617193137 |
| comment_count | 0.5080672252852084 | 0.6754405001311277 | 0.5932920617193137 | 1.0 |
| count | 105286.0 | 105286.0 | 105286.0 | 105286.0 |
| mean | 310203.8285649374 | 7710.254117356533 | 250.26147825921774 | 763.981696331896 |
| std | 404720.9518257565 | 10034.24170660183 | 284.44128517500013 | 791.3162923725997 |
| min | 223.0 | 0.0 | 0.0 | 0.0 |
| 25% | 50901.0 | 841.0 | 41.0 | 137.0 |
| 50% | 164477.5 | 3596.5 | 143.0 | 482.0 |
| 75% | 403588.5 | 10654.0 | 356.0 | 1161.75 |
| max | 3190447.0 | 55707.0 | 1336.0 | 3337.0 |

INFERENCES :-

- Likes have a high positive correlation with views (0.62), suggesting that videos with more views tend to have more likes.
- Dislikes also have a positive correlation with views (0.65) and likes (0.56), indicating that videos with more views or likes tend to have more dislikes as well, which could be due to increased visibility and broader audience.
- Comment_count is positively correlated with views (0.51) and likes (0.67), meaning videos that are viewed or liked more also tend to have more comments.
- The correlations suggest that all these engagement metrics tend to increase together; as one goes up, the others do as well.
- The count for all variables is the same (105286.0), which means there are no missing values for these metrics in the dataset.
- Views have a very wide range, as indicated by the minimum (223.0) and maximum (3190447.0) values.

- The standard deviation (std) is relatively high for all variables, especially for views (404728.95), indicating a large variability in the number of views, likes, dislikes, and comments across different videos.

- The median (50% quantile) for views is 16477.5, likes is 3596.5, dislikes is 145.0, and comment_count is 482.0, which are all significantly lower than their respective mean values, showing a right-skewed distribution (confirmed by the previous histograms and box plots you shared). This means most videos have engagement metrics lower than the average, skewed by a small number of videos with exceptionally high engagement.

- The 75% quantile for each metric shows that the top 25% of the videos have at least 40358 views, 10654 likes, 356 dislikes, and 1161 comments, further highlighting the skewness in the data.

## DASHBOARD

## CODE:-

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import dash
from dash import dcc, html, Input, Output
import plotly.express as px
from plotly.subplots import make_subplots

#%%
pd.set_option('display.max_columns', None)

videos_df = pd.read_csv('/Users/surya/Documents/Class files/Data
visualization/FTP/FTP(Surya)/finaldata.csv')


videos_df['publish_date'] = pd.to_datetime(videos_df['publish_date'])
videos_df['year'] = videos_df['publish_date'].dt.year

#%%

# ===============Phase 2 -  Initializing the app ================
app = dash.Dash(__name__)

server = app.server

# ===============Phase 3 -  Defining the layout of the app ================

# Define the layout of the app
app.layout = html.Div([
    html.H1("Youtube Videos Analysis", style={'textAlign': 'center', 'color':
'orange', 'font-size': 30}),
    dcc.Tabs(id='tabs', value='tab-1', children=[
        dcc.Tab(label = 'Channel Analysis', value='tab-1',
style={'backgroundColor': '#AED6F1', 'color': 'black'},
selected_style={'backgroundColor': '#5499C7', 'color': 'white'}),
        dcc.Tab(label='Time Analysis', value='tab-2',
style={'backgroundColor': '#AED6F1', 'color': 'black'},
selected_style={'backgroundColor': '#5499C7', 'color': 'white'}),
        dcc.Tab(label='Countries Analysis', value='tab-3',
style={'backgroundColor': '#AED6F1', 'color': 'black'},
selected_style={'backgroundColor': '#5499C7', 'color': 'white'})
    ]),
    html.Div(id='layout', style={'backgroundColor': '#F0F8FF'})
])

# Defining layout for tab 1
tab1_layout = html.Div([
            html.H3('Select Channel:'),
            dcc.Dropdown(
```

```
                id='channel-dropdown',
                options=[{'label': i, 'value': i} for i in
videos_df['channel_title'].unique()],
                value=videos_df['channel_title'].unique()[0],
                multi= True,
                clearable=False
            ),
            html.Div(id='channel-graphs')
        ])

## Defining layout for tab 2

# Define the order for sorting days of the week and times of day
day_order = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday',
'Saturday', 'Sunday']
time_order = ['Morning', 'Afternoon', 'Evening', 'Night']

# Define layout for tab 2
tab2_layout = html.Div([
    html.H3('Select Day of Week:'),
    dcc.Checklist(
        id='day-of-week-checklist',
        options=[{'label': day, 'value': day} for day in day_order],
        value=['Monday'],  # Default value
        inline=True
    ),
    html.H3('Select Time of Day:'),
    dcc.Checklist(
        id='time-of-day-checklist',
        options=[{'label': time, 'value': time} for time in time_order],
        value=['Morning'],  # Default value
        inline=True
    ),
    html.H3('Select Year Range:'),
    dcc.RangeSlider(
        id='year-range-slider',
        min=videos_df['year'].min(),
        max=videos_df['year'].max(),
        value=[videos_df['year'].min(), videos_df['year'].max()],
        marks={str(year): str(year) for year in
sorted(videos_df['year'].unique())},
        step=1
    ),
    html.Div(id='day-time-bar-chart'),
    html.Div(id='year-line-chart')
])

# Define layout for tab 3
tab3_layout = html.Div([
    html.H3('Select Country:'),
    dcc.Checklist(
        id='country-checklist',
        options=[{'label': country, 'value': country} for country in
videos_df['publish_country'].unique()],
        value=videos_df['publish_country'].unique()[:3],  # Default to first
3 countries as an example
        inline=True
```

```
    ),
    html.Div(id='country-bar-chart'),
    html.Div(id='country-pie-chart'),
    html.Div(id='country-subplots')
])

# ===============Phase 4 -  Defining the callback functions ================

# Define callback for updating tabs
@app.callback(Output('layout', 'children'), Input('tabs', 'value'))
def update_tabs(tab):
    if tab == 'tab-1':
        return tab1_layout
    elif tab == 'tab-2':
        return tab2_layout
    elif tab == 'tab-3':
        return tab3_layout
    else:
        return html.H1('Tab not implemented', style={'color': 'red'})


@app.callback(
    Output('channel-graphs', 'children'),
    [Input('channel-dropdown', 'value')])
def update_graph(selected_channels):
    # Ensure selected_channels is a list even if it is a single value
    if not isinstance(selected_channels, list):
        selected_channels = [selected_channels]

    # Group by 'channel_title' and sum the numeric columns
    channel_totals = videos_df.groupby('channel_title')[['views', 'likes',
'dislikes', 'comment_count']].sum().reset_index()

    # Filter the summarized data for selected channels
    filtered_totals =
channel_totals[channel_totals['channel_title'].isin(selected_channels)]

    # Now create a bar chart using Plotly Express for the filtered totals
    fig = make_subplots(
        rows=2, cols=2,
        subplot_titles=('Total Views', 'Total Likes', 'Total Dislikes',
'Total Comment Count'),
        vertical_spacing=0.15
    )

    views_fig = px.bar(filtered_totals, x='channel_title', y='views',
labels={'x': 'Channel', 'y': 'Total Views'})
    likes_fig = px.bar(filtered_totals, x='channel_title', y='likes',
labels={'x': 'Channel', 'y': 'Total Likes'})
    dislikes_fig = px.bar(filtered_totals, x='channel_title', y='dislikes',
labels={'x': 'Channel', 'y': 'Total Dislikes'})
    comments_fig = px.bar(filtered_totals, x='channel_title',
y='comment_count', labels={'x': 'Channel', 'y': 'Total Comment Count'})

    # Add traces from each figure to the corresponding subplot
    for trace in views_fig.data:
        fig.add_trace(trace, row=1, col=1)
```

```
    for trace in likes_fig.data:
        fig.add_trace(trace, row=1, col=2)
    for trace in dislikes_fig.data:
        fig.add_trace(trace, row=2, col=1)
    for trace in comments_fig.data:
        fig.add_trace(trace, row=2, col=2)

    # Update layout
    fig.update_layout(showlegend=False, height=800)

    return dcc.Graph(figure=fig)

# Callback for updating graphs on Tab 2
@app.callback(
    [Output('day-time-bar-chart', 'children'),
     Output('year-line-chart', 'children')],
    [Input('day-of-week-checklist', 'value'),
     Input('time-of-day-checklist', 'value'),
     Input('year-range-slider', 'value')])
def update_day_time_graphs(selected_days, selected_times, selected_years):
    # If no day or time is selected, use the defaults
    selected_days = selected_days or ['Monday']
    selected_times = selected_times or ['Morning']

    # Filter dataframe based on selections
    filtered_df =
videos_df[videos_df['published_day_of_week'].isin(selected_days) &
                          videos_df['part_of_day'].isin(selected_times) &
                          videos_df['year'].between(selected_years[0],
selected_years[1])]

    # Line chart for total videos by year
    line_fig_data =
filtered_df.groupby('year').size().reset_index(name='total_videos')
    line_fig = px.line(line_fig_data, x='year', y='total_videos',
                       labels={'total_videos': 'Total Videos', 'year':
'Year'},
                       title='Total YouTube Videos by Year')

    # Bar chart for total videos by day and time
    bar_fig_data = filtered_df.groupby(['published_day_of_week',
'part_of_day']).size().reset_index(name='total_videos')
    bar_fig_data['published_day_of_week'] =
pd.Categorical(bar_fig_data['published_day_of_week'], categories=day_order,
ordered=True)
    bar_fig_data['part_of_day'] = pd.Categorical(bar_fig_data['part_of_day'],
categories=time_order, ordered=True)
    bar_fig_data = bar_fig_data.sort_values('published_day_of_week')

    bar_fig = px.bar(bar_fig_data, x='published_day_of_week',
y='total_videos', color='part_of_day',
                     labels={'total_videos': 'Total Videos',
'published_day_of_week': 'Day of Week'},
                     title='Total YouTube Videos by Day of Week and Time of
Day')

    return dcc.Graph(figure=bar_fig), dcc.Graph(figure=line_fig)
```

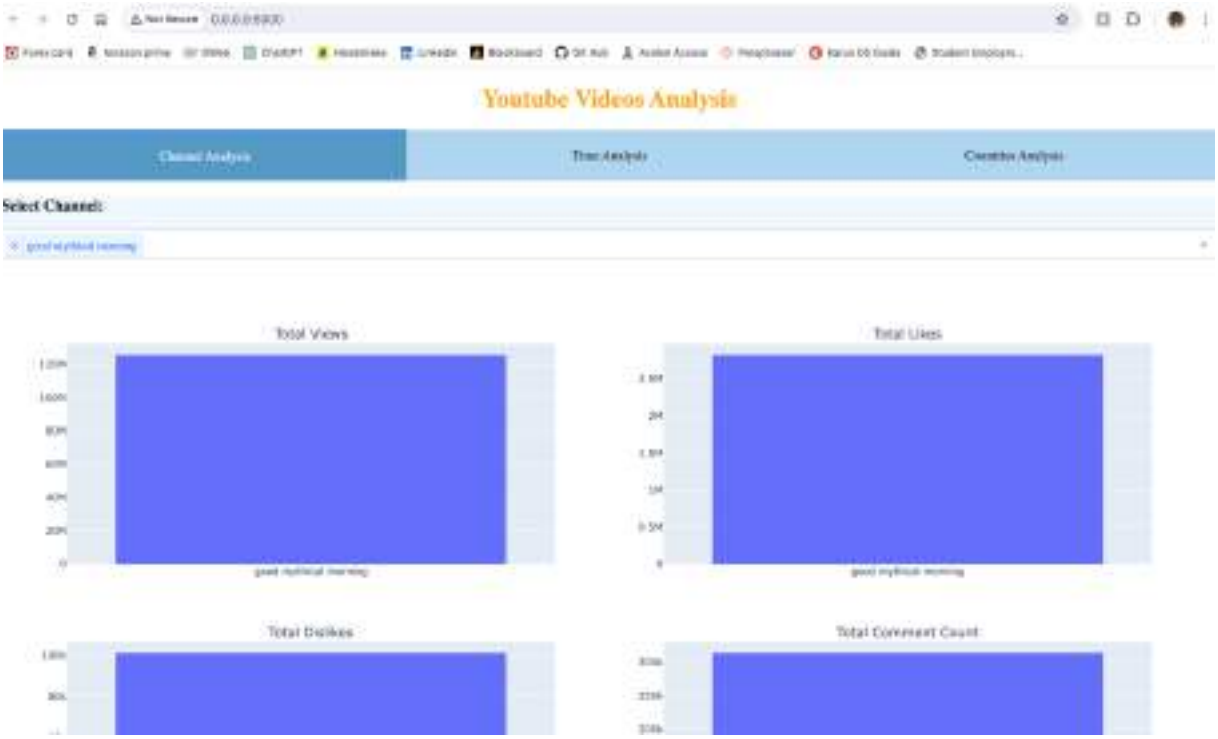
```
# Callback for updating graphs on Tab 3
@app.callback(
    [Output('country-bar-chart', 'children'),
     Output('country-pie-chart', 'children'),
     Output('country-subplots', 'children')],
    [Input('country-checklist', 'value')])
def update_country_graphs(selected_countries):
    # Filter dataframe based on selections
    filtered_df =
videos_df[videos_df['publish_country'].isin(selected_countries)]

    # Bar chart for total videos by country with a legend
    bar_fig_data =
filtered_df.groupby('publish_country').size().reset_index(name='total_videos'
)
    bar_fig = px.bar(bar_fig_data, x='publish_country', y='total_videos',
                     title='Total YouTube Videos by Country',
color='publish_country')
    # Ensure that legend is displayed for the bar chart
    bar_fig.update_layout(showlegend=True)

    # Pie chart for proportion of total videos by country
    pie_fig = px.pie(bar_fig_data, names='publish_country',
values='total_videos',
                     title='Proportion of Total YouTube Videos by Country')

    # Subplots for metrics by day of week
    subplots_fig = make_subplots(rows=2, cols=2,
                                 subplot_titles=['Total Views', 'Total
Likes', 'Total Dislikes', 'Total Comment Count'],
                                 horizontal_spacing=0.1)  # Adjust spacing as
needed

    # Define colors for countries
    country_colors = px.colors.qualitative.Plotly

    # Add line plots to subplots
    for i, metric in enumerate(['views', 'likes', 'dislikes',
'comment_count']):
        for j, country in enumerate(selected_countries):
            country_metrics = filtered_df[filtered_df['publish_country'] ==
country]
            country_metrics =
country_metrics.groupby(['published_day_of_week',
'publish_country'])[metric].sum().reset_index()
            country_metrics['published_day_of_week'] = pd.Categorical(
                country_metrics['published_day_of_week'],
categories=day_order, ordered=True)
            country_metrics =
country_metrics.sort_values('published_day_of_week')
            line_fig = px.line(country_metrics, x='published_day_of_week',
y=metric, color='publish_country',
                               labels={metric: f'Total {metric.title()}'})
            line_fig.update_traces(line=dict(color=country_colors[j %
len(country_colors)]))
```

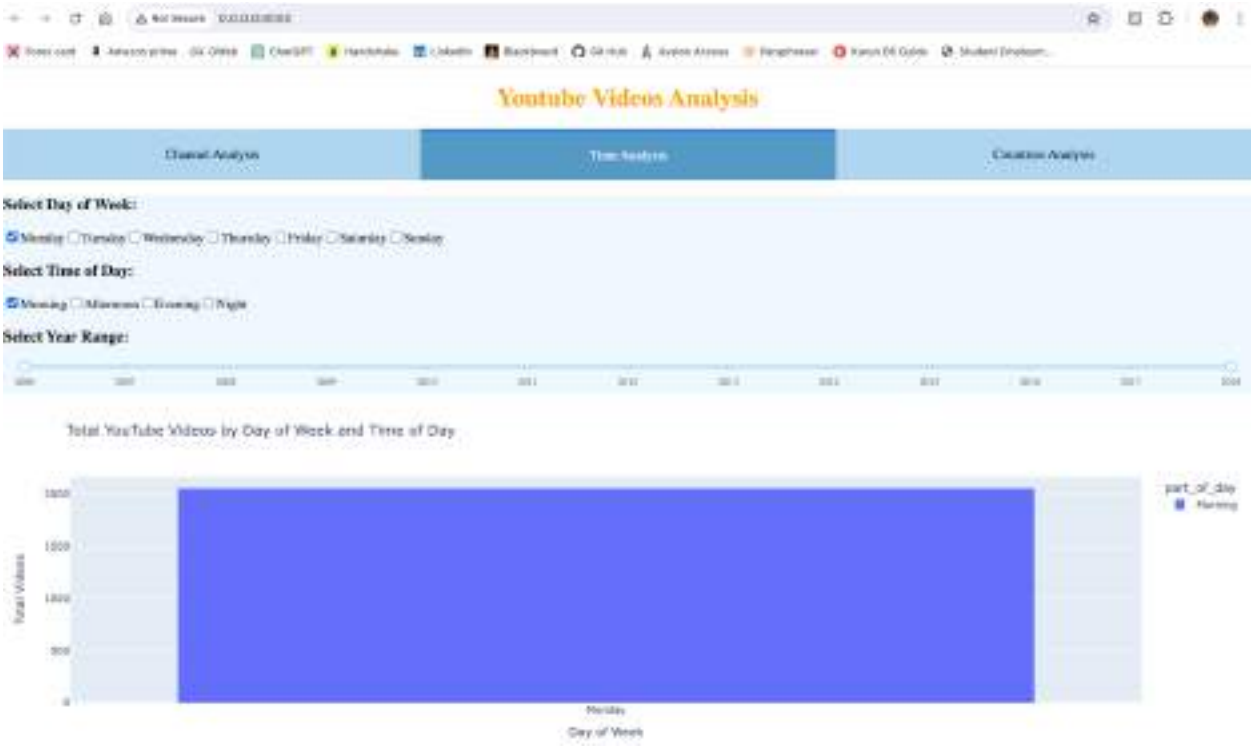
```
            for trace in line_fig.data:
                subplots_fig.add_trace(trace, row=i // 2 + 1, col=i % 2 + 1)

    # Adjust subplots layout for legend positioning
    subplots_fig.update_layout(
        height=800,
        title_text='Metrics by Day of the Week for Selected Countries',
        legend=dict(
            orientation="v",
            yanchor="middle",
            y=0.5,
            xanchor="left",
            x=1.05
        )
    )

    return dcc.Graph(figure=bar_fig), dcc.Graph(figure=pie_fig),
dcc.Graph(figure=subplots_fig)


# ===============Phase 5 -  Running the app ================
if __name__ == '__main__':
    app.run_server(debug=False,
                   port=8000,
                   host='0.0.0.0')
```

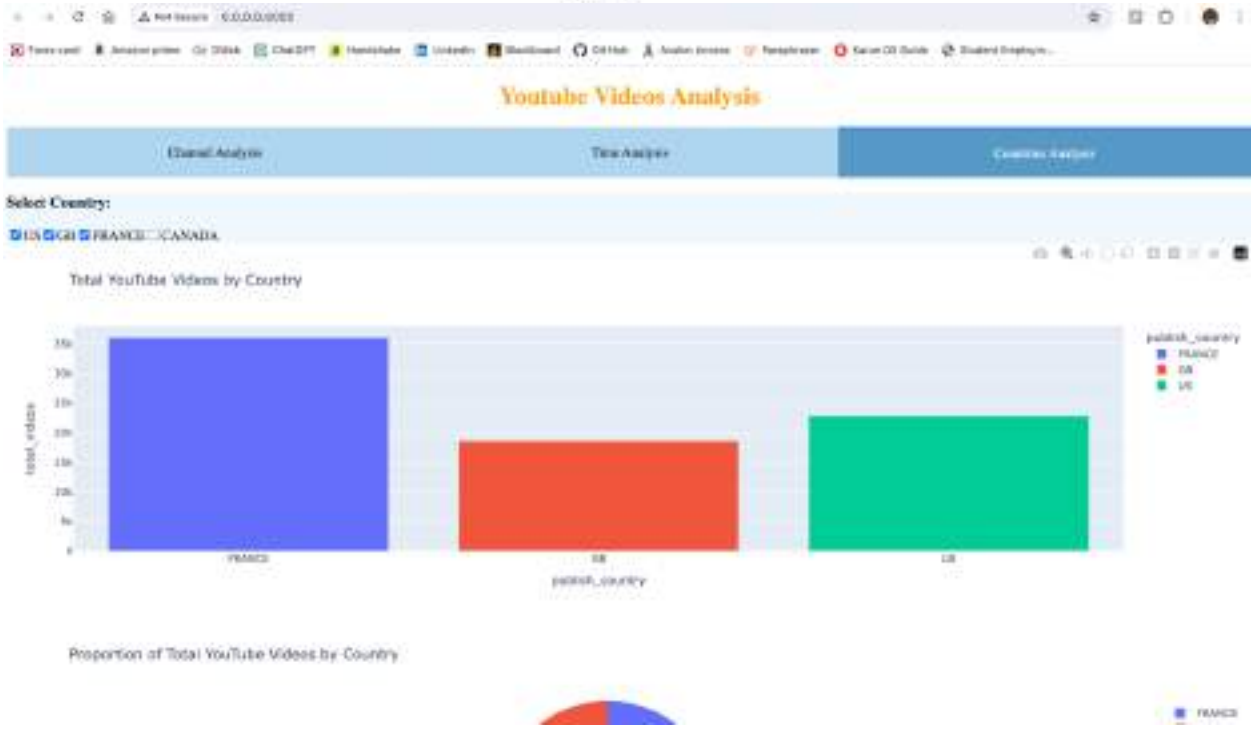
## DASH BOARD LINK :-

https://dashapp-zuxpawv4nq-ue.a.run.app

- I built this dash board using **DASH** framework in python.
- There are total 3 tabs in my dashboard named Channel analysis,Countries analysis and Time Analysis.
- I deployed this YouTube Analysis Dashboard using **Google Cloud Platform(GCP).**


Youtube Videos Analysis
Channel Analysis
Time Analysis
Countries Analysis
Select Channel:
Total Views
Total Likes
Total Dislikes
Total Comment Count


Youtube Videos Analysis
Channel Analysis
Time Analysis
Creators Analysis
Select Day of Week:
Monday Tuesday Wednesday Thursday Friday Saturday Sunday
Select Time of Day:
Morning Afternoon Evening Night
Select Year Range:
Total YouTube Videos by Day of Week and Time of Day
Total Videos
Monday
Day of Week
part_of_day
Morning


Youtube Videos Analysis
Channel Analysis
Time Analysis
Creators Analysis
Select Country:
US GB FRANCE CANADA
Total YouTube Videos by Country
total_videos
FRANCE
GB
US
publish_country
publish_country
FRANCE
GB
US
Proportion of Total YouTube Videos by Country
FRANCE