Video Popularity Analysis:

Problem Statement: Determine the factors that influence a video's popularity.

Questions to Answer: What are the trends in view counts, likes, dislikes, and comments? Are there correlations between these metrics and the video's category or the publishing date?

Importing Libraries:

In [1]:
```
# Importing Libraries:

import numpy as np
import pandas as pd

import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import plotly.express as px
```

Loading Dataset:

In [2]:
```
# Loading Dataset:
import pandas as pd

# Load the CSV file with the 'latin1' encoding
india_df = pd.read_csv('IN_youtube_trending_data.csv', encoding='latin1')

# Load the first few rows of a file
india_df.head()
```

Out[2]:

	video_id	title	publishedAt	channelId	channelTitle	c
0	Iot0eF6EoNA	Sadak 2 | Official Trailer | Sanjay | Pooja | ...	2020-08-12T04:31:41Z	UCGqvJPRcv7aVFun-eTsatcA	FoxStarHindi	
1	x-KbnJ9fvJc	Kya Baat Aa : Karan Aujla (Official Video) Tan...	2020-08-11T09:00:11Z	UCm9SZAI03Rev9sFwloCdz1g	Rehaan Records	
2	KX06ksuS6Xo	Diljit Dosanjh: CLASH (Official) Music Video |...	2020-08-11T07:30:02Z	UCZRdNleCgW-BGUJf-bbjzQg	Diljit Dosanjh	
		Dil Ko				

Exploratory Data Analysis (EDA)

In [3]:
```
# general information
india_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 220921 entries, 0 to 220920
Data columns (total 16 columns):
 #   Column             Non-Null Count   Dtype 
---  ------             --------------   ----- 
 0   video_id           220921 non-null  object
 1   title              220921 non-null  object
 2   publishedAt        220921 non-null  object
 3   channelId          220921 non-null  object
 4   channelTitle       220920 non-null  object
 5   categoryId         220921 non-null  int64 
 6   trending_date      220921 non-null  object
 7   tags               220921 non-null  object
 8   view_count         220921 non-null  int64 
 9   likes              220921 non-null  int64 
 10  dislikes           220921 non-null  int64 
 11  comment_count      220921 non-null  int64 
 12  thumbnail_link     220921 non-null  object
 13  comments_disabled  220921 non-null  bool  
 14  ratings_disabled   220921 non-null  bool  
 15  description        202549 non-null  object
dtypes: bool(2), int64(5), object(9)
memory usage: 24.0+ MB
```

In [4]:
```
 # Summary statistics
india_df.describe()
```

Out[4]:

	categoryId	view_count	likes	dislikes	comment_count
count	220921.000000	2.209210e+05	2.209210e+05	2.209210e+05	2.209210e+05
mean	20.849544	2.895213e+06	1.468311e+05	2.653852e+03	8.784114e+03
std	6.044239	7.089427e+06	4.049589e+05	7.678115e+04	7.442354e+04
min	1.000000	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
25%	20.000000	4.012340e+05	1.347600e+04	0.000000e+00	3.660000e+02
50%	24.000000	9.959170e+05	4.049800e+04	0.000000e+00	1.198000e+03
75%	24.000000	2.535156e+06	1.243660e+05	9.810000e+02	4.197000e+03
max	29.000000	2.644074e+08	1.611524e+07	1.234147e+07	6.738565e+06

In [5]:
```
# give the number of rows and columns
india_df.shape
```

Out[5]: (220921, 16)

In [6]:
```
# extract all columns of the dataset
india_df.columns
```

Out[6]:
```
Index(['video_id', 'title', 'publishedAt', 'channelId', 'channelTitle',
       'categoryId', 'trending_date', 'tags', 'view_count', 'likes',
       'dislikes', 'comment_count', 'thumbnail_link', 'comments_disabled',
       'ratings_disabled', 'description'],
      dtype='object')
```

In [7]:
```
# check for null values
india_df.isna().sum()
```

Out[7]:
```
video_id                0
title                   0
publishedAt             0
channelId               0
channelTitle            1
categoryId              0
trending_date           0
tags                    0
view_count              0
likes                   0
dislikes                0
comment_count           0
thumbnail_link          0
comments_disabled       0
ratings_disabled        0
description         18372
dtype: int64
```

In [8]:
```
 # Fill missing values with a specific value
india_df.fillna("not known")
```

Out[8]:

	video_id	title	publishedAt	channelId	chanr
0	Iot0eF6EoNA	Sadak 2 | Official Trailer | Sanjay | Pooja | ...	2020-08-12T04:31:41Z	UCGqvJPRcv7aVFun-eTsatcA	FoxSt
1	x-KbnJ9fvJc	Kya Baat Aa : Karan Aujla (Official Video) Tan...	2020-08-11T09:00:11Z	UCm9SZAl03Rev9sFwloCdz1g	F R
2	KX06ksuS6Xo	Diljit Dosanjh: CLASH (Official) Music Video |...	2020-08-11T07:30:02Z	UCZRdNleCgW-BGUJf-bbjzQg	Diljit D
3	UsMRgnTcchY	Dil Ko Maine Di Kasam Video | Amaal M Ft.Ariji...	2020-08-10T05:30:49Z	UCq-Fj5jknLsUf-MWSy4_brA	T
4	WNSEXJJhKTU	Baarish (Official Video) Payal Dev,Stebin Ben ...	2020-08-11T05:30:13Z	UCye6Oz0mg46S362LwARGVcA	VYRLO
...	...	...	...	...	
220916	Zl8alBdIfpg	NEW! Barsatein - Mausam Pyar Ka - Ep 74 | 19 O...	2023-10-19T15:00:33Z	UCpEhnqL0y41EpW2TvWAHD7Q	SE
220917	LIsfMO5Jd_w	NAPOLEON - Official Trailer #2 (HD)	2023-10-18T12:59:40Z	UCz97F7dMxBNOfGYu3rx8aCw	Sony F Enterta
220918	RYI12J1nz4A	KING - NEW LIFE | Full Album	2023-10-17T18:30:25Z	UCrtOnzd9dWH9lXTAB-64Hfg	
220919	fhf7lDNrUus	Ghost | Second OGM | Dr.Shivarajkumar | Anupam...	2023-10-17T13:30:02Z	UCovxnbWKPCA5iJDxa9zbBew	T Ka
220920	K5oI7trwdOw	Sukoon Episode 2 - 19 Oct 2023 (Eng Sub) | San...	2023-10-19T16:24:01Z	UC4JCksJF76g_MdzPVBJoC3Q	ARY

220921 rows × 16 columns

In [9]:
```
# To check skewness of the views
india_df["view_count"].skew()
```

Out[9]:
```
9.180066861968834
```

In [10]:
```
# Check unique values of channel Title & tags
india_df["channelTitle"].unique()
```

Out[10]:
```
array(['FoxStarHindi', 'Rehaan Records', 'Diljit Dosanjh', ...,
       'Ajith Vinayaka Films', 'Malik Vlogs', 'Dante Hindustani Shorts'],
      dtype=object)
```

In [11]:
```
india_df["tags"].unique()
```

Out[11]:
```
array(['sadak|sadak 2|mahesh bhatt|vishesh films|pooja bhatt|alia bhatt|san
jay dutt|aditya roy kapur|alia bhatt movies|alia bhatt new movies|aditya ro
y kapur new movies|aditya roy kapur movies|sanjay dutt sadak 2|sanjay dutt
sadak|sanjay dutt new movies|fox star studios|fox star hindi|disney plus ho
tstar|disney plus movie|bollywood|cinema|movie|hindi cinema|upcoming bollyw
ood movie|love story|action|thriller|suspense',
       '[None]',
       'clash diljit dosanjh|diljit dosanjh|diljit dosanjh goat album|dilji
t dosanjh new album|punjabi songs 2020|punjabi new song|new song 2020|goat
diljit dosanjh|the kidd punjabi music|the kidd music|raj ranjodh songs|goat
diljit dosanjh full album|diljit dosanjh karan aujla song|Diljit dosanjh ne
w songs|diljit dosanjh songs|goat diljit dosanjh 2020|goat 2020|latest punj
abi songs 2020|punajbi 2020 latest songs|punjabi songs|punjabi|new songs pu
njabi|clash',
       ...,
       'monkey magic|monkey magic new series|melodies of india|monkey magic
travel india|monkey magic melodies of india',
       'Hindi Love song|Latest love song|Love song|New Hindi song|Hindi son
g 2023',
       'dewaangi ost|sahir ali bagga|geo tv drama|hum tv dramas|sangeet pk|
sahir ali bagga tum nahi ho|sahir ali bagga latest song|Har pal geo|geo dra
mas|latest pakistani drama|top pakistani dramas|best pakistani dramas|lates
t pakistani dramas|drama 2019|sahir ali bagga songs|Kahin Deep Jalay | Full
OST|kahin deep jale ost|kahin deep jale|kahin deep jale ep 2|kahin deep jal
e OST Official|kahin deep jale full song|Kahin Deep Jalay|mahi|maahi|maahi
queen'],
      dtype=object)
```

In [12]:
```
# Replace the null values
india_df["channelTitle"].fillna("unknown", inplace = True)
india_df["tags"].fillna("none", inplace = True)
```

In [13]:
```
# check for null values
india_df.isna().sum()
```

Out[13]:
```
video_id                0
title                   0
publishedAt             0
channelId               0
channelTitle            0
categoryId              0
trending_date           0
tags                    0
view_count              0
likes                   0
dislikes                0
comment_count           0
thumbnail_link          0
comments_disabled       0
ratings_disabled        0
description         18372
dtype: int64
```

In [14]:
```
# Check for duplicate values
india_df.duplicated().sum()
```

Out[14]:
```
75
```

In [15]:
```
# Remove duplicate rows
india_df.drop_duplicates()
```

Out[15]:

	video_id	title	publishedAt	channelId	chanr
0	Iot0eF6EoNA	Sadak 2 | Official Trailer | Sanjay | Pooja | ...	2020-08-12T04:31:41Z	UCGqvJPRcv7aVFun-eTsatcA	FoxSt
1	x-KbnJ9fvJc	Kya Baat Aa : Karan Aujla (Official Video) Tan...	2020-08-11T09:00:11Z	UCm9SZAl03Rev9sFwloCdz1g	R
2	KX06ksuS6Xo	Diljit Dosanjh: CLASH (Official) Music Video |...	2020-08-11T07:30:02Z	UCZRdNleCgW-BGUJf-bbjzQg	Diljit D
3	UsMRgnTcchY	Dil Ko Maine Di Kasam Video | Amaal M Ft.Ariji...	2020-08-10T05:30:49Z	UCq-Fj5jknLsUf-MWSy4_brA	T
4	WNSEXJJhKTU	Baarish (Official Video) Payal Dev,Stebin Ben ...	2020-08-11T05:30:13Z	UCye6Oz0mg46S362LwARGVcA	VYRLO
...	...	...	...	...	
220916	Zl8alBdIfpg	NEW! Barsatein - Mausam Pyar Ka - Ep 74 | 19 O...	2023-10-19T15:00:33Z	UCpEhnqL0y41EpW2TvWAHD7Q	SE
220917	LIsfMO5Jd_w	NAPOLEON - Official Trailer #2 (HD)	2023-10-18T12:59:40Z	UCz97F7dMxBNOfGYu3rx8aCw	Sony F Enterta
220918	RYI12J1nz4A	KING - NEW LIFE | Full Album	2023-10-17T18:30:25Z	UCrtOnzd9dWH9IXTAB-64Hfg	
220919	fhf7IDNrUus	Ghost | Second OGM | Dr.Shivarajkumar | Anupam...	2023-10-17T13:30:02Z	UCovxnbWKPCA5iJDxa9zbBew	T Ka
220920	K5oI7trwdOw	Sukoon Episode 2 - 19 Oct 2023 (Eng Sub) | San...	2023-10-19T16:24:01Z	UC4JCksJF76g_MdzPVBJoC3Q	ARY

220846 rows × 16 columns

In [16]:
```
# Renaming the columns
india_df.rename(columns={'view_count': 'views'}, inplace=True)
india_df.columns # to check the columns names
```

Out[16]:
```
Index(['video_id', 'title', 'publishedAt', 'channelId', 'channelTitle',
       'categoryId', 'trending_date', 'tags', 'views', 'likes', 'dislikes',
       'comment_count', 'thumbnail_link', 'comments_disabled',
       'ratings_disabled', 'description'],
      dtype='object')
```

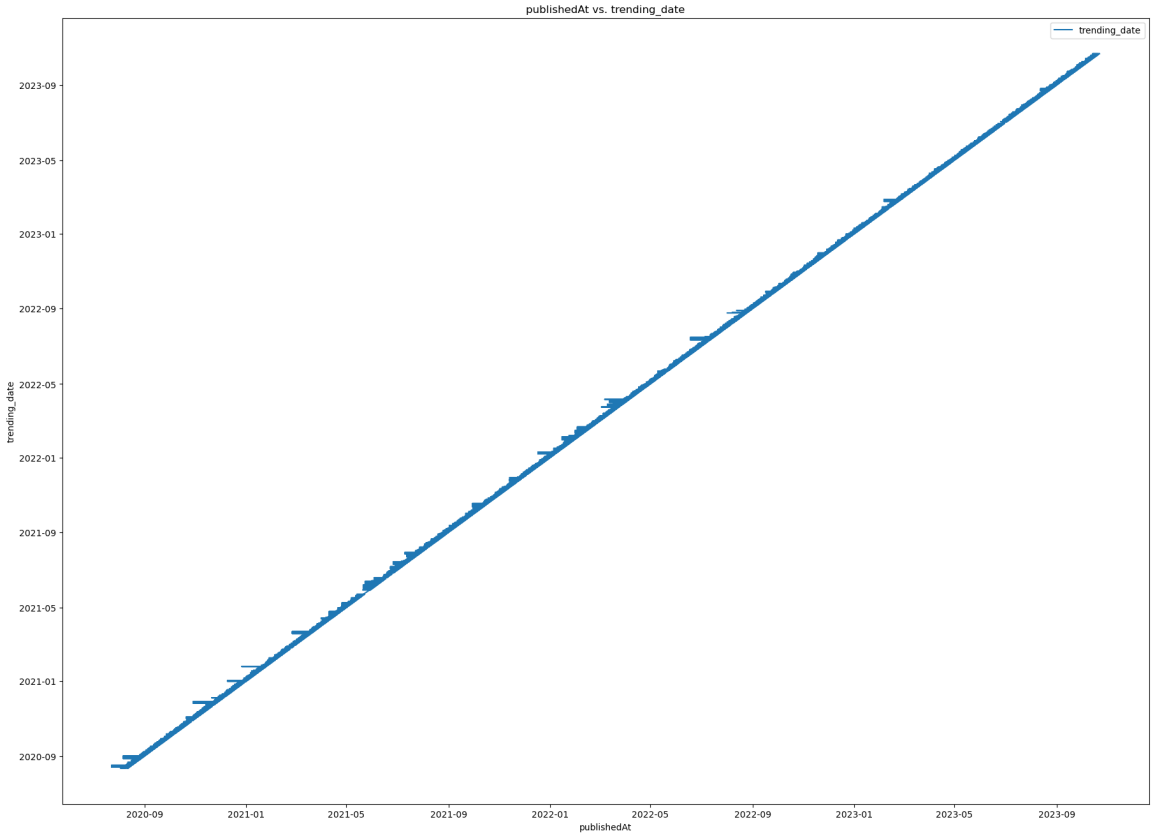
In [17]:
```
#Saving the cleaned Data
india_df.to_csv('cleaned_data.csv', index=False)
```

Time Series Analysis

In [18]:
```
# Import Necessary libraries:

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import statsmodels.api as sm
```

In [19]:
```
# Load your data:

# Replace 'your_data.csv' with the actual file path
india_df = pd.read_csv('IN_youtube_trending_data.csv', encoding='latin1')

# Ensure that the date columns are in datetime format
india_df['publishedAt'] = pd.to_datetime(india_df['publishedAt'])
india_df['trending_date'] = pd.to_datetime(india_df['trending_date'])

# Set the date column as the index, which is important for time series analy
india_df.set_index('publishedAt', inplace=True)
```

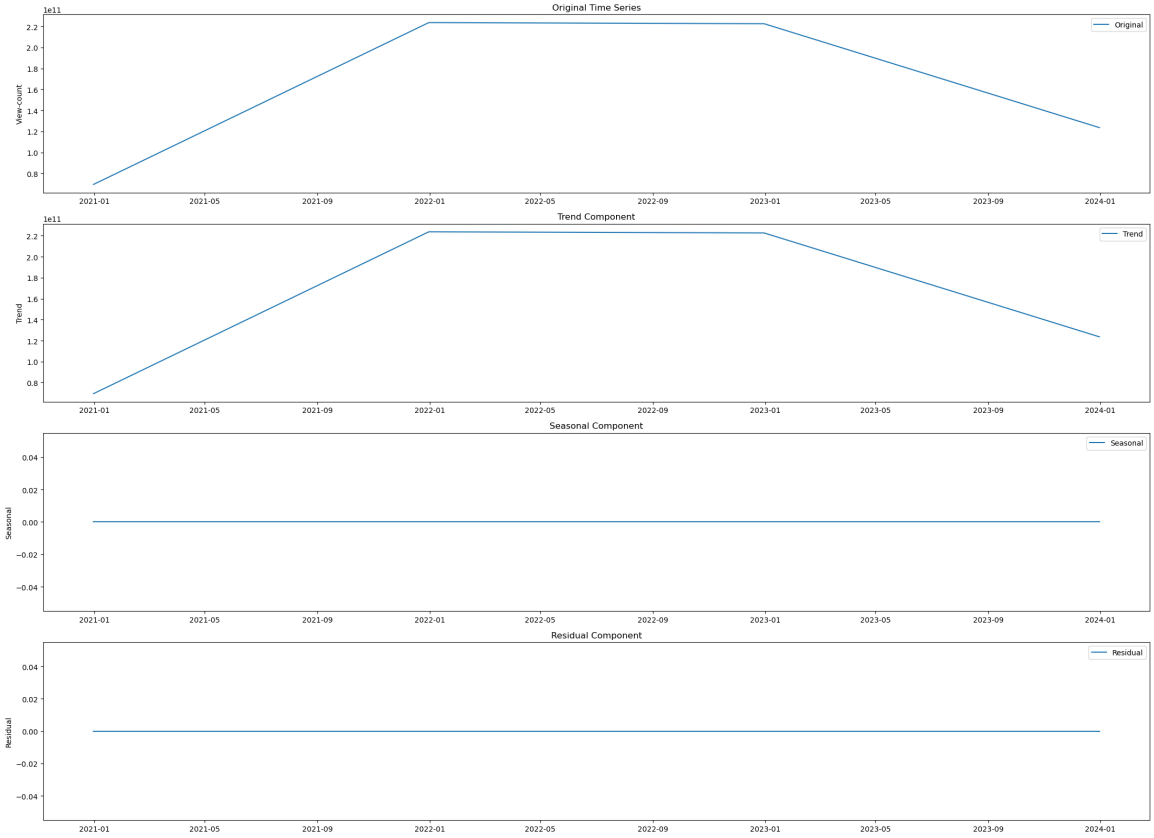
In [20]:
```
# Explore the data
plt.figure(figsize=(22, 16))
plt.plot(india_df['trending_date'], label='trending_date')
plt.xlabel('publishedAt')
plt.ylabel('trending_date')
plt.title('publishedAt vs. trending_date')
plt.legend()
plt.show()
```



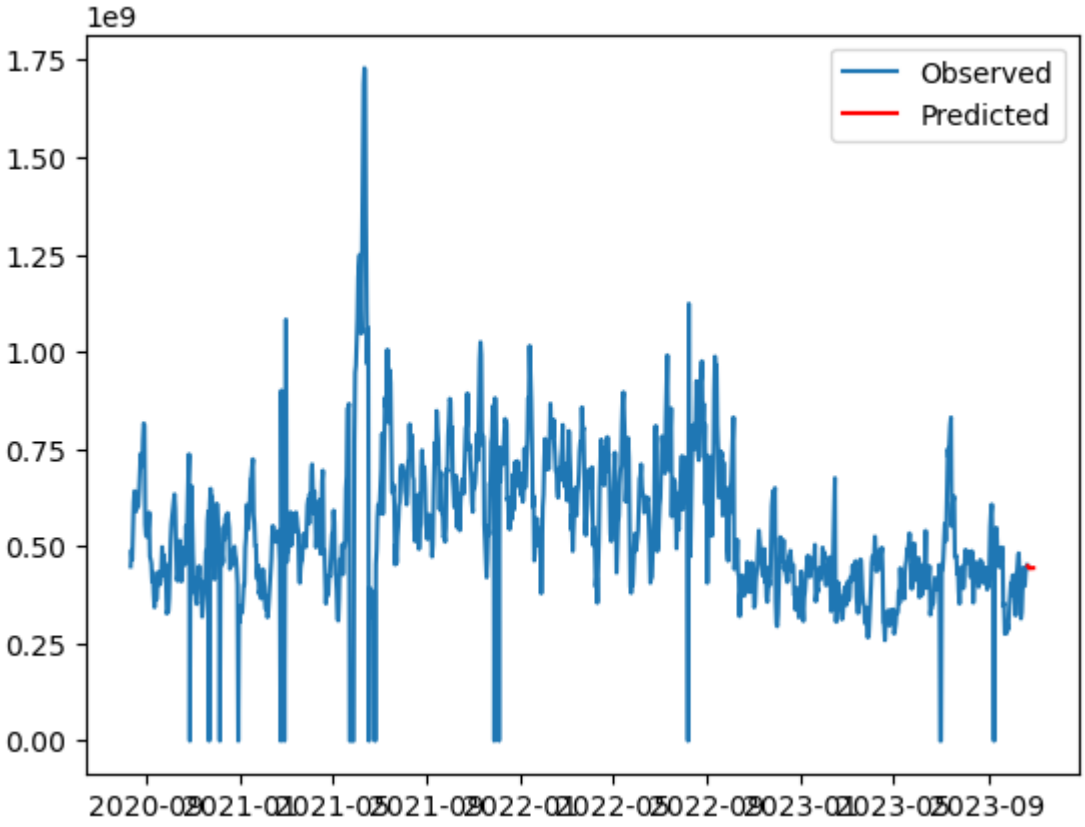
In [9]:
```
import pandas as pd

# Assuming 'trending_date' is a datetime column in your DataFrame
# and 'views' is the column you want to resample

# Example:
india_df = pd.read_csv('IN_youtube_trending_data.csv', encoding='latin1')
india_df['trending_date'] = pd.to_datetime(india_df['trending_date'])
india_df.set_index('trending_date', inplace=True)

# Resample the data to a yearly frequency and count the number of observatio
india_df_yearly = india_df['view_count'].resample('Y').count()

# Print or visualize the result
print(india_df_yearly.head())

# Resample the data to a yearly frequency

india_df_yearly = india_df.resample('Y').count()
```

```
trending_date
2020-12-31 00:00:00+00:00    26121
2021-12-31 00:00:00+00:00    70439
2022-12-31 00:00:00+00:00    71056
2023-12-31 00:00:00+00:00    53305
Freq: A-DEC, Name: view_count, dtype: int64
```

In [10]:
```
import statsmodels.api as sm
import matplotlib.pyplot as plt

# Assuming 'trending_date' is a datetime column in your DataFrame
# and 'views' is the column you want to resample

# Example:
india_df = pd.read_csv('IN_youtube_trending_data.csv', encoding='latin1')
india_df['trending_date'] = pd.to_datetime(india_df['trending_date'])
india_df.set_index('trending_date', inplace=True)

# Resample the data to a specific frequency (e.g., 'Y' for yearly)
india_df_resampled = india_df['view_count'].resample('Y').sum()

# Perform seasonal decomposition
decomposition = sm.tsa.seasonal_decompose(india_df_resampled, model='additiv
trend = decomposition.trend
seasonal = decomposition.seasonal
residual = decomposition.resid

# Plot the results
plt.figure(figsize=(22, 16))
plt.subplot(411)
plt.plot(india_df_resampled, label='Original')
plt.legend()
plt.ylabel('View-count')
plt.title('Original Time Series')

plt.subplot(412)
plt.plot(trend, label='Trend')
plt.legend()
plt.ylabel('Trend')
plt.title('Trend Component')

plt.subplot(413)
plt.plot(seasonal, label='Seasonal')
plt.legend()
plt.ylabel('Seasonal')
plt.title('Seasonal Component')

plt.subplot(414)
plt.plot(residual, label='Residual')
plt.legend()
plt.ylabel('Residual')
plt.title('Residual Component')

plt.tight_layout()
plt.show()
```


Original Time Series
1e11
Original
View-count
Trend Component
Trend
Seasonal Component
Seasonal
Residual Component
Residual
2021-01
2021-05
2021-09
2022-01
2022-05
2022-09
2023-01
2023-05
2023-09
2024-01

In [16]:
```
import pandas as pd
from statsmodels.tsa.arima.model import ARIMA
import matplotlib.pyplot as plt

# Load or define your dataset
india_df = pd.read_csv('IN_youtube_trending_data.csv', encoding='latin1')
india_df['trending_date'] = pd.to_datetime(india_df['trending_date'])
india_df.set_index('trending_date', inplace=True)

# Assuming 'views' is your time series variable
india_df_resampled = india_df['view_count'].resample('D').sum()  # Adjust th

# Fit an ARIMA model to the data
model = ARIMA(india_df_resampled, order=(5, 1, 0))
model_fit = model.fit()

# Make predictions
predictions = model_fit.predict(start=len(india_df_resampled), end=len(india

# Plot the predictions
plt.plot(india_df_resampled, label='Observed')
plt.plot(predictions, label='Predicted', color='red')
plt.legend()
plt.show()
```



In [ ]:

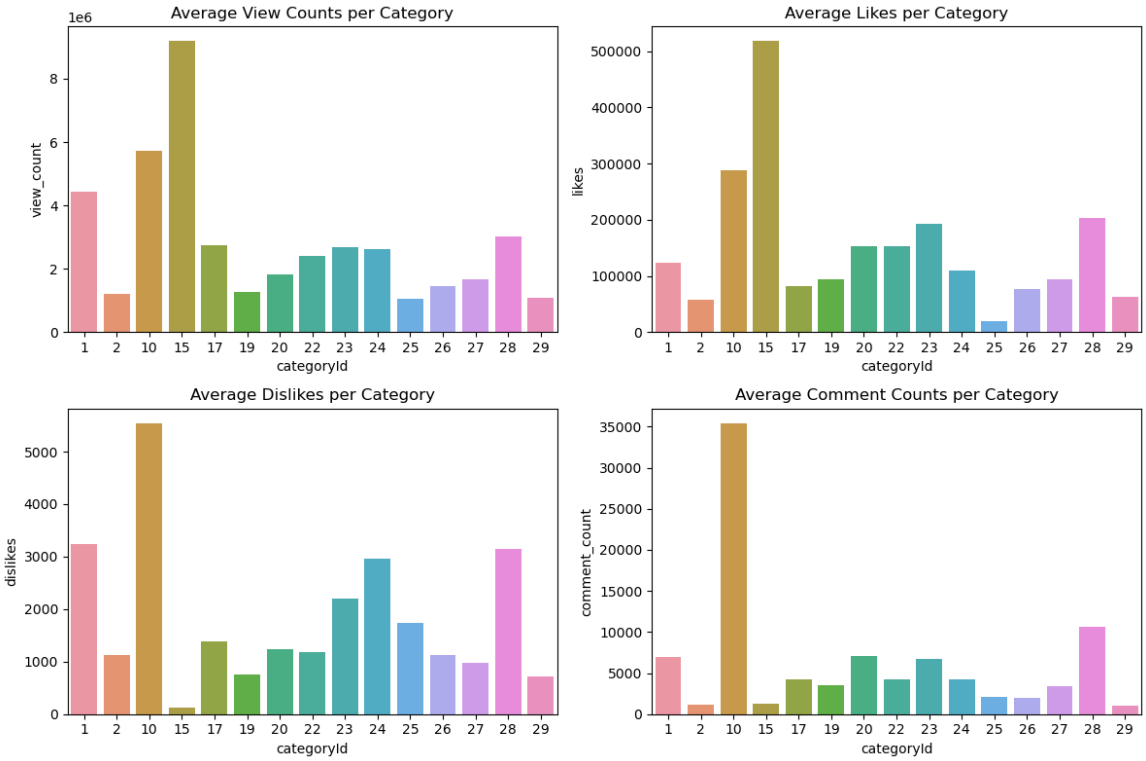
Correlation Analysis

In [50]:
```
# import pandas as pd

# Sample data with columns: views, likes, dislikes, comment_count
data = {
    'views': [100, 200, 300, 400, 500],
    'likes': [10, 20, 30, 40, 50],
    'dislikes': [5, 10, 15, 20, 25],
    'comment_count': [2, 5, 8, 11, 14]
}

india_df = pd.DataFrame(data)

# Calculate the correlation matrix
correlation_matrix = india_df[['views', 'likes', 'dislikes', 'comment_count'

# Print the correlation matrix
print(correlation_matrix)
```

```
               views  likes  dislikes  comment_count
views            1.0    1.0       1.0            1.0
likes            1.0    1.0       1.0            1.0
dislikes         1.0    1.0       1.0            1.0
comment_count    1.0    1.0       1.0            1.0
```

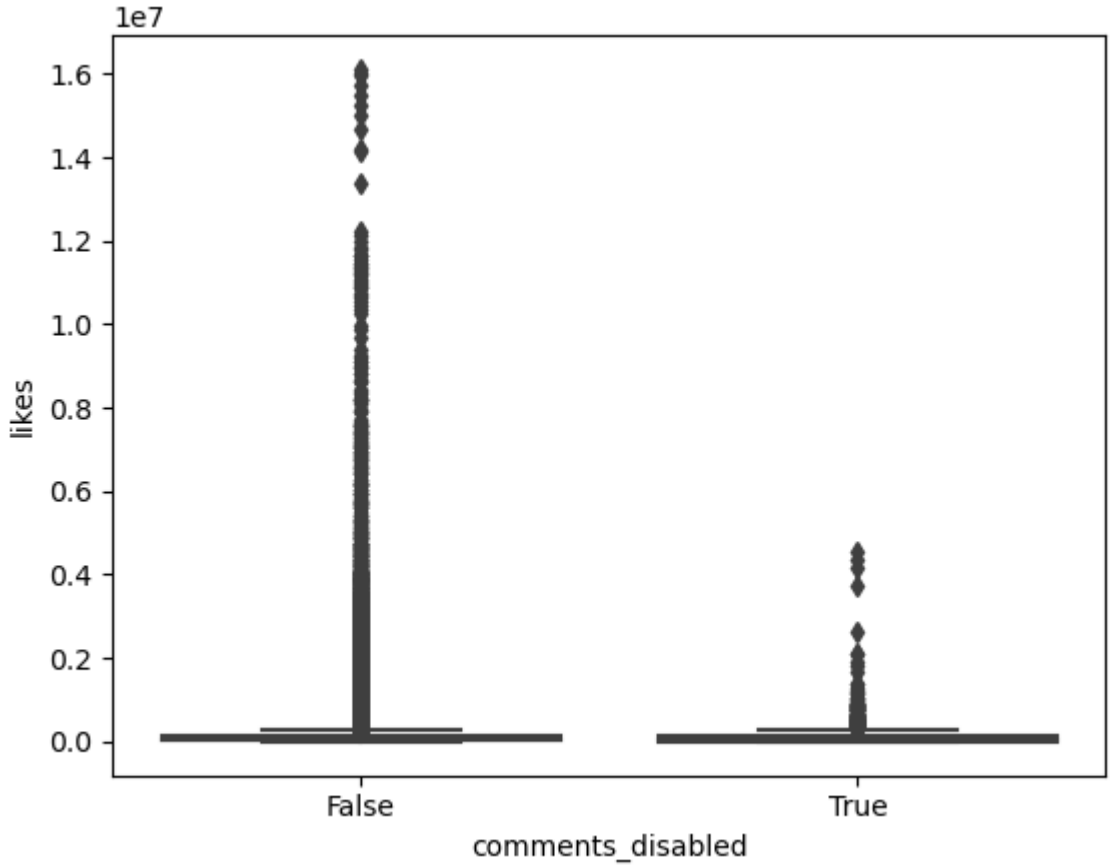
Category Analysis

In [27]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load your data
# Load the CSV file with the 'latin1' encoding
india_df = pd.read_csv('IN_youtube_trending_data.csv', encoding='latin1')

# Assuming your DataFrame has columns like 'views', 'likes', 'dislikes', 'co
# Adjust column names based on your actual DataFrame

# Group data by categoryId and calculate mean values
category_stats = india_df.groupby('categoryId').agg({
    'view_count': 'mean',
    'likes': 'mean',
    'dislikes': 'mean',
    'comment_count': 'mean'
}).reset_index()

# Visualize the data
plt.figure(figsize=(12, 8))

# Bar plot for average view counts per category
plt.subplot(2, 2, 1)
sns.barplot(x='categoryId', y='view_count', data=category_stats)
plt.title('Average View Counts per Category')

# Bar plot for average likes per category
plt.subplot(2, 2, 2)
sns.barplot(x='categoryId', y='likes', data=category_stats)
plt.title('Average Likes per Category')

# Bar plot for average dislikes per category
plt.subplot(2, 2, 3)
sns.barplot(x='categoryId', y='dislikes', data=category_stats)
plt.title('Average Dislikes per Category')

# Bar plot for average comment counts per category
plt.subplot(2, 2, 4)
sns.barplot(x='categoryId', y='comment_count', data=category_stats)
plt.title('Average Comment Counts per Category')

plt.tight_layout()
plt.show()
```



Hypothesis Testing

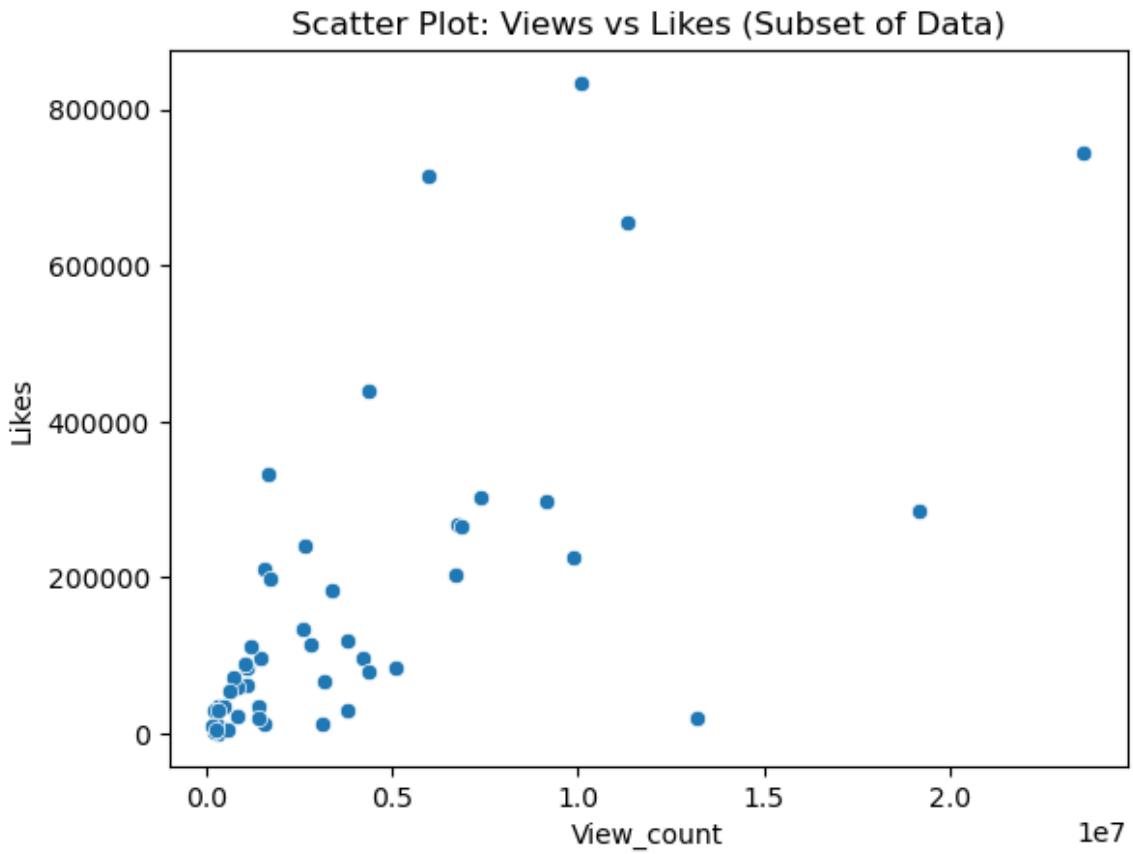
In [58]:
```
# importing libraries
import pandas as pd
import numpy as np
from scipy import stats
import seaborn as sns
import matplotlib.pyplot as plt

#load Dataset
india_df = pd.read_csv('IN_youtube_trending_data.csv', encoding='latin1')

#Explore Data
# Display the first few rows of the dataset
india_df.head()

# Explore the summary statistics
india_df.describe()


#Visualize
# Visualize the data
sns.boxplot(x='comments_disabled', y='likes', data=india_df)
plt.show()
# Separate data into two groups: videos with comments enabled and videos wit
enabled_likes = india_df[india_df['comments_disabled'] == False]['likes']
disabled_likes = india_df[india_df['comments_disabled'] == True]['likes']

# Perform an independent t-test
t_stat, p_value = stats.ttest_ind(enabled_likes, disabled_likes)

# Display the results
print(f'T-statistic: {t_stat}')
print(f'P-value: {p_value}')
```




```
T-statistic: 1.4765600254487634
P-value: 0.13979503919731065
```

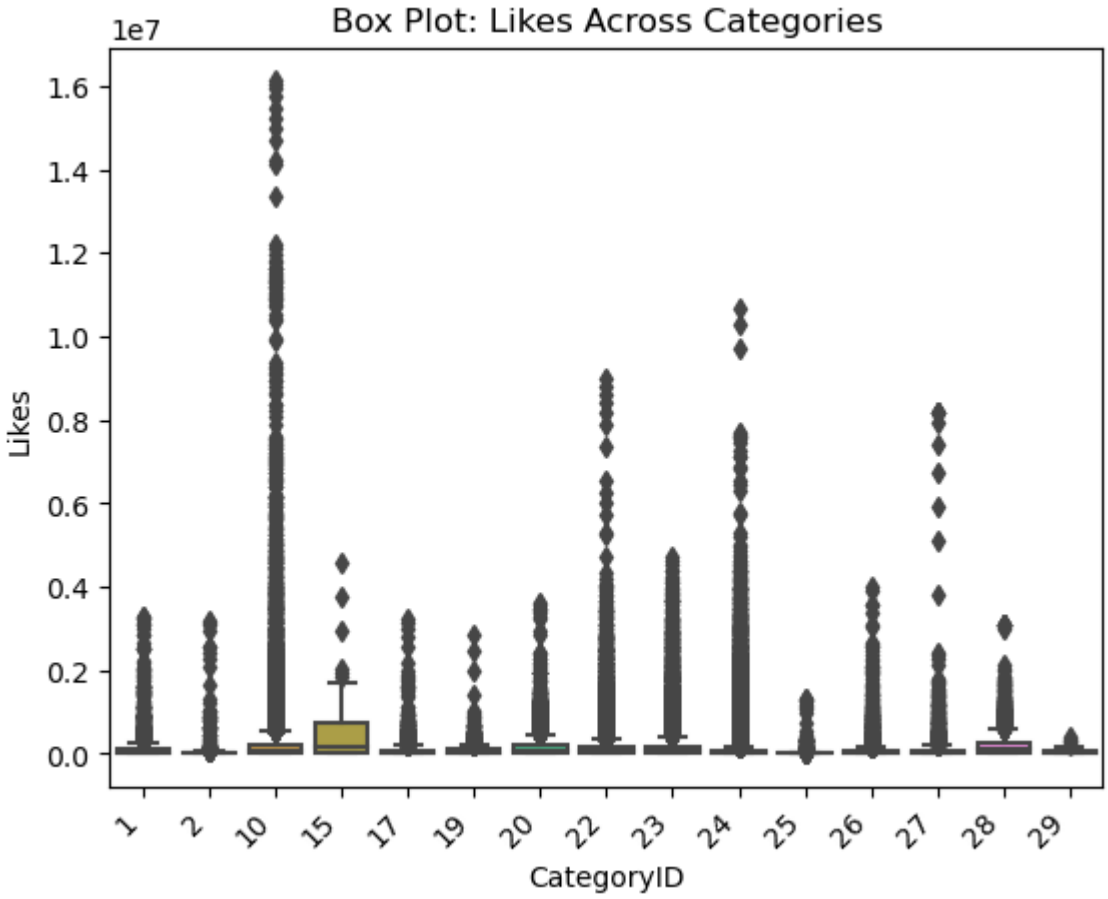
Visualization by scatterplot

In [63]:
```
# importing modules
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression

# Load data
india_df = pd.read_csv('IN_youtube_trending_data.csv', encoding='latin1')

#sns.scatterplot(x='views', y='likes', data=df)
# Select a specific range of rows, for example, from row 0 to 49
subset_df = india_df.iloc[:50]

#visualize
sns.scatterplot(x='view_count', y='likes', data=subset_df)
plt.title('Scatter Plot: Views vs Likes (Subset of Data)')
plt.xlabel('View_count')
plt.ylabel('Likes')
plt.show()
```



Box Plot

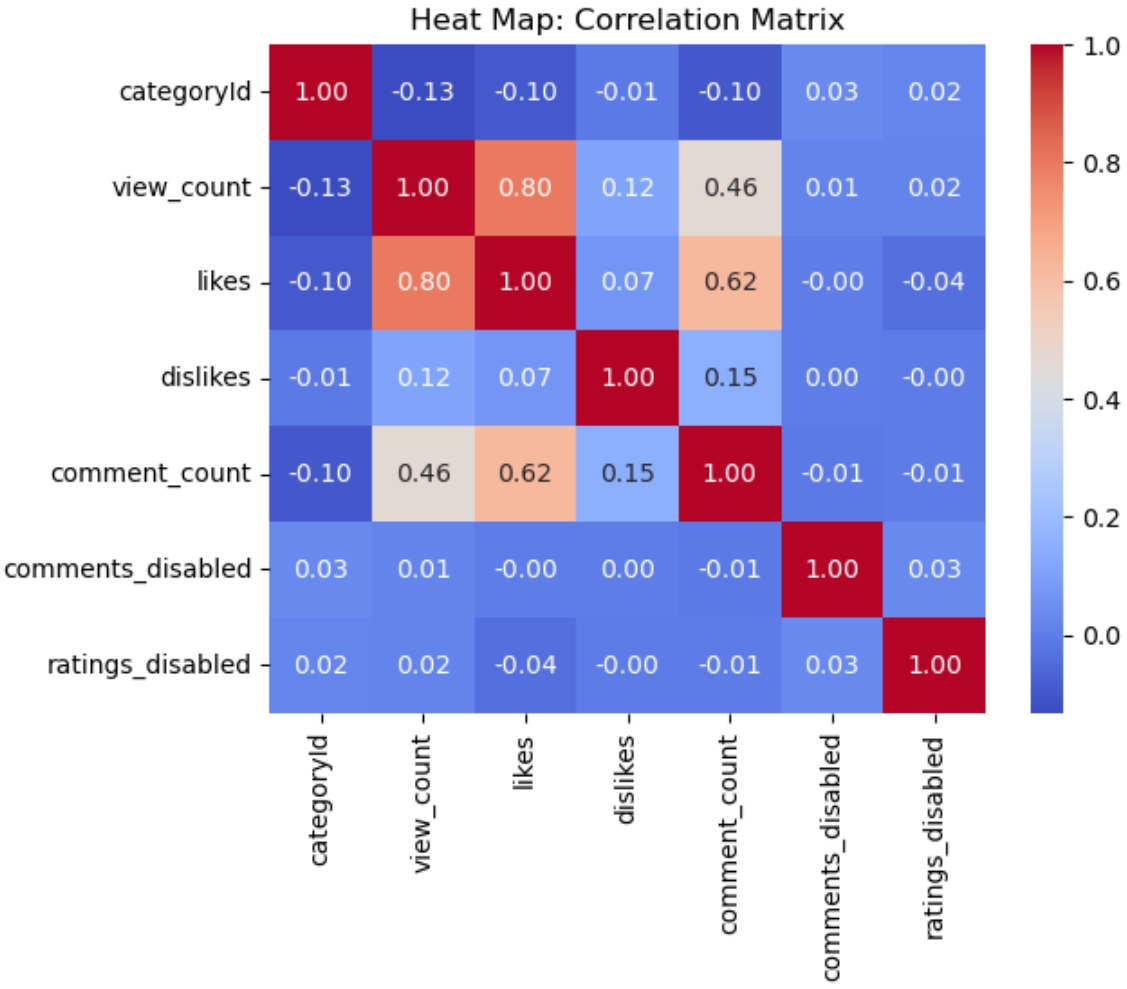
In [66]:
```
# importing modules
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression

# Load data
india_df = pd.read_csv('IN_youtube_trending_data.csv', encoding='latin1')

#sns.scatterplot(x='views', y='likes', data=df)
# Select a specific range of rows, for example, from row 0 to 49
subset_df = india_df.iloc[:50]

#visualize
sns.boxplot(x='categoryId', y='likes', data=india_df)
plt.title('Box Plot: Likes Across Categories')
plt.xlabel('CategoryID')
plt.ylabel('Likes')
plt.xticks(rotation=45, ha='right')  # Rotate x-axis labels for better visib
plt.show()
```



Heat Map

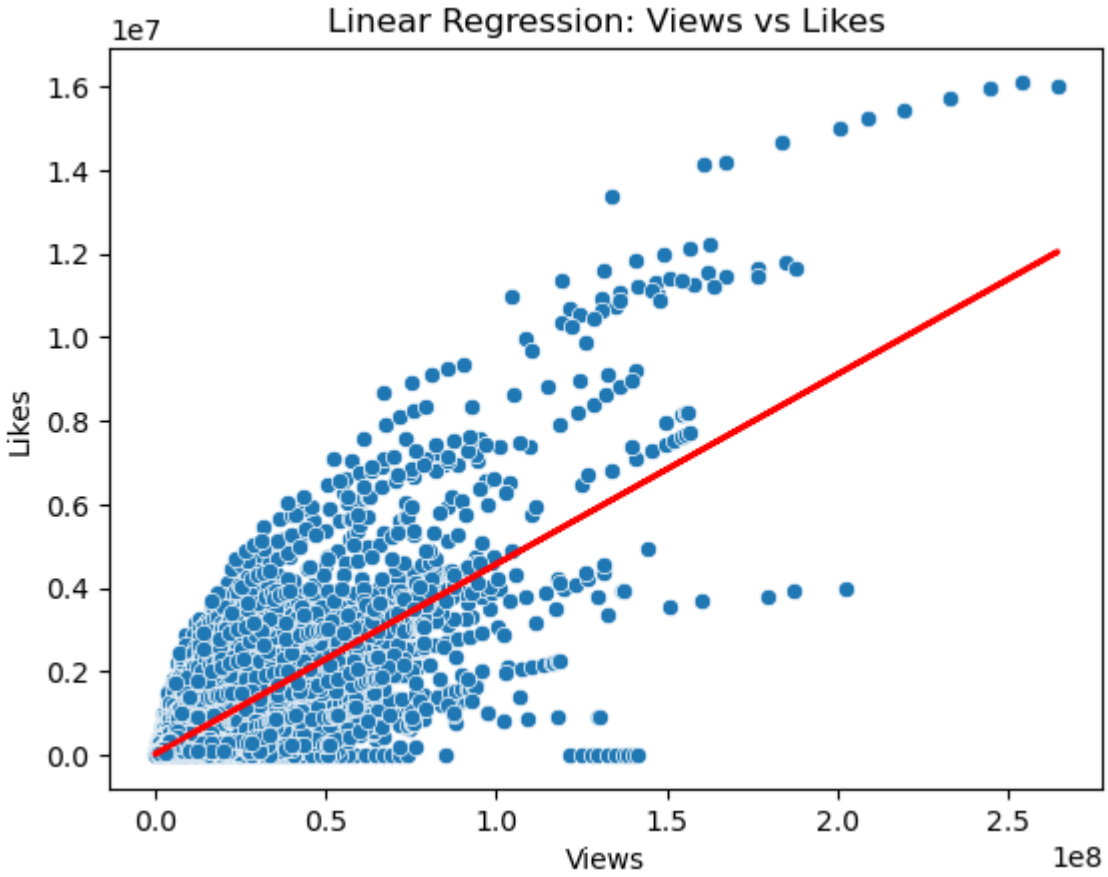
In [52]:
```
# importing modules
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression

# Load data
india_df = pd.read_csv('IN_youtube_trending_data.csv', encoding='latin1')

#sns.scatterplot(x='views', y='likes', data=df)
# Select a specific range of rows, for example, from row 0 to 49
subset_df = india_df.iloc[:50]

correlation_matrix = india_df.corr()
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f')
plt.title('Heat Map: Correlation Matrix')
plt.show()
```

```
C:\Users\tiwar\AppData\Local\Temp\ipykernel_13972\1158286515.py:16: FutureW
arning: The default value of numeric_only in DataFrame.corr is deprecated.
In a future version, it will default to False. Select only valid columns or
specify the value of numeric_only to silence this warning.
  correlation_matrix = india_df.corr()
```



Linear Regression

In [69]:
```
# importing modules
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression

# Load data
india_df = pd.read_csv('IN_youtube_trending_data.csv', encoding='latin1')

#sns.scatterplot(x='views', y='likes', data=df)
# Select a specific range of rows, for example, from row 0 to 49
subset_df = india_df.iloc[:50]

#visualization
# Assuming 'views' is the independent variable and 'likes' is the dependent
X = india_df[['view_count']]
y = india_df['likes']

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, ran

# Create and fit the linear regression model
model = LinearRegression()
model.fit(X_train, y_train)

# Plot the linear regression line on the scatter plot
sns.scatterplot(x='view_count', y='likes', data=india_df)
plt.plot(X, model.predict(X), color='red', linewidth=2)
plt.title('Linear Regression: Views vs Likes')
plt.xlabel('Views')
plt.ylabel('Likes')
plt.show()
```



Type *Markdown* and LaTeX: α^2

In [ ]:

In [ ]:

In [ ]: