Exploratory Data Statistics - Global Youtube Analysis

D.Sathyakala, Data analyst Intern @ Senchola Technology Solutions

1. Import library files & Reading Dataset

Importing libarary files

In [182…
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

Loading the dataset.....

In [182…
```
df = pd.read_csv("Global YouTube Statistics.csv", encoding="Latin")
df
```

Out[182…

	rank	Youtuber	subscribers	video views	category	Title	uploads	Country	Abbreviation	channel_type
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

2. Initial Data exploration

Display "Top 5 rows" of the dataset

In [182…
```
df.head(5)
```

Out[182…

	rank	Youtuber	subscribers	video views	category	Title	uploads	Country	Abbreviation	channel_type	...	su
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	...	

5 rows × 28 columns

Display "Bottom 5 rows" of the dataset

In [183…
```
df.tail(5)
```

Out[183…

	rank	Youtuber	subscribers	video views	category	Title	uploads	Country	Abbreviation	channel_type	...	s
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	...	

5 rows × 28 columns

Display the size of the dataset

In [183…
```
df.shape
```

Out[183… (995, 28)

Review of descriptive statistics of numerical datatypes

In [183…
```
df.describe()
```

Out[183…

	rank	subscribers	video views	uploads	video_views_rank	country_rank	channel_type_rank	video_views_for
count	995.00000	9.950000e+02	9.950000e+02	995.000000	9.940000e+02	879.000000	962.000000	
mean	498.00000	2.298241e+07	1.103954e+10	9187.125628	5.542489e+05	386.053470	745.719335	
std	287.37606	1.752611e+07	1.411084e+10	34151.352254	1.362782e+06	1232.244746	1944.386561	
min	1.00000	1.230000e+07	0.000000e+00	0.000000	1.000000e+00	1.000000	1.000000	
25%	249.50000	1.450000e+07	4.288145e+09	194.500000	3.230000e+02	11.000000	27.000000	
50%	498.00000	1.770000e+07	7.760820e+09	729.000000	9.155000e+02	51.000000	65.500000	
75%	746.50000	2.460000e+07	1.355470e+10	2667.500000	3.584500e+03	123.000000	139.750000	
max	995.00000	2.450000e+08	2.280000e+11	301308.000000	4.057944e+06	7741.000000	7741.000000	

8 rows × 21 columns

Display all the details of dataset

In [183…
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 995 entries, 0 to 994
Data columns (total 28 columns):
 #   Column                                  Non-Null Count  Dtype  
---  ------                                  --------------  -----  
 0   rank                                    995 non-null    int64  
 1   Youtuber                                995 non-null    object 
 2   subscribers                             995 non-null    int64  
 3   video views                             995 non-null    float64
 4   category                                949 non-null    object 
 5   Title                                   995 non-null    object 
 6   uploads                                 995 non-null    int64  
 7   Country                                 873 non-null    object 
 8   Abbreviation                            873 non-null    object 
 9   channel_type                            965 non-null    object 
 10  video_views_rank                        994 non-null    float64
 11  country_rank                            879 non-null    float64
 12  channel_type_rank                       962 non-null    float64
 13  video_views_for_the_last_30_days        939 non-null    float64
 14  lowest_monthly_earnings                 995 non-null    float64
 15  highest_monthly_earnings                995 non-null    float64
 16  lowest_yearly_earnings                  995 non-null    float64
 17  highest_yearly_earnings                 995 non-null    float64
 18  subscribers_for_last_30_days            658 non-null    float64
 19  created_year                            990 non-null    float64
 20  created_month                           990 non-null    object 
 21  created_date                            990 non-null    float64
 22  Gross tertiary education enrollment (%) 872 non-null    float64
 23  Population                              872 non-null    float64
 24  Unemployment rate                       872 non-null    float64
 25  Urban_population                        872 non-null    float64
 26  Latitude                                872 non-null    float64
 27  Longitude                               872 non-null    float64
dtypes: float64(18), int64(3), object(7)
memory usage: 217.8+ KB
```

3. Data Cleaning & Transformation

Display the column names

In [183…
```
df.columns
```

Out[183…
```
Index(['rank', 'Youtuber', 'subscribers', 'video views', 'category', 'Title',
       'uploads', 'Country', 'Abbreviation', 'channel_type',
       'video_views_rank', 'country_rank', 'channel_type_rank',
       'video_views_for_the_last_30_days', 'lowest_monthly_earnings',
       'highest_monthly_earnings', 'lowest_yearly_earnings',
       'highest_yearly_earnings', 'subscribers_for_last_30_days',
       'created_year', 'created_month', 'created_date',
       'Gross tertiary education enrollment (%)', 'Population',
       'Unemployment rate', 'Urban_population', 'Latitude', 'Longitude'],
      dtype='object')
```

Replace '_' with 'Space' & Capitalize the first letter of column name

In [183…
```
df.columns = df.columns.str.replace('_', ' ').str.title()
df
```

Out[183…

	Rank	Youtuber	Subscribers	Video Views	Category	Title	Uploads	Country	Abbreviation	Channel Type
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

Check the duplicate data

In [183…
```
df.duplicated().sum()
```

Out[183…
```
0
```

Identify the missing values

In [183…
```
df.isna().sum()
```

Out[183…
```
Rank                                    0
Youtuber                                0
Subscribers                             0
Video Views                             0
Category                               46
Title                                   0
Uploads                                 0
Country                               122
Abbreviation                          122
Channel Type                           30
Video Views Rank                        1
Country Rank                          116
Channel Type Rank                      33
Video Views For The Last 30 Days       56
Lowest Monthly Earnings                 0
Highest Monthly Earnings                0
Lowest Yearly Earnings                  0
Highest Yearly Earnings                 0
Subscribers For Last 30 Days          337
Created Year                            5
Created Month                           5
Created Date                            5
Gross Tertiary Education Enrollment (%) 123
Population                            123
Unemployment Rate                     123
Urban Population                      123
Latitude                              123
Longitude                             123
dtype: int64
```

Replace empty cell as '0' values

In [183…
```
df = df.dropna()
df
```

Out[183…

	Rank	Youtuber	Subscribers	Video Views	Category	Title	Uploads	Country	Abbreviation	Channel Type
0	1	T-Series	245000000	2.280000e+11	Music	T-Series	20082	India	IN	Music
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
8	9	Like Nastya	106000000	9.047906e+10	People & Blogs	Like Nastya Vlog	493	Russia	RU	People
...	...	...	...	...	...	...	...	...	...	...
989	990	Migos ATL	12400000	6.993406e+09	Music	Migos ATL	99	United States	US	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
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

554 rows × 28 columns

Remove & Replace Unwanted Characters

In [183…
```
# Define remove & replace logic as replacing any non-alphanumeric characters
pattern = r'[^a-zA-Z0-9\s.,!?&\'-]'

# Replace characters 'Youtuber' & 'Title' with an empty string
df['Youtuber'] = df['Youtuber'].str.replace(pattern, '')
df['Title'] = df['Title'].str.replace(pattern, '')
```

In [184…
```
# Remove any trailing/ leading whitespace from values
df['Youtuber'] = df['Youtuber'].str.strip()
df['Title'] = df['Title'].str.strip()
df['Title']
```

Out[184…
```
0                       T-Series
2                        MrBeast
3      Cocomelon - Nursery Rhymes
4                       SET India
8                  Like Nastya Vlog
                    ...
989                     Migos ATL
990                   Natan por Aï¿
991        Free Fire India Official
993                    RobTopGames
994                   Make Joke Of
Name: Title, Length: 554, dtype: object
```

In [184…
```
# Create filtered views of df to validate remove & replace
filter_youtuber_rows = df['Youtuber'].str.contains(pattern, regex = True)
filter_title_rows = df['Title'].str.contains(pattern, regex = True)
```

In [184…
```
# Validate results of string remove & replace
filter_youtuber_rows_results = filter_youtuber_rows[filter_youtuber_rows == True]
filter_youtuber_rows_results
```

Out[184…
```
55     True
64     True
76     True
93     True
105    True
       ...
920    True
945    True
970    True
979    True
990    True
Name: Youtuber, Length: 61, dtype: bool
```

Remove unwanted rows

In [184…
```
# Filter for YouTube channels with 0 video views
filtered_df = df[df['Video Views'] == 0]
filtered_df
```

Out[184…

Rank	Youtuber	Subscribers	Video Views	Category	Title	Uploads	Country	Abbreviation	Channel Type	...	Subscribers For Last 30 Days	Created Year	Create Mont

0 rows × 28 columns

Some YouTube channels have 0 video views, these appear to be not valid channels, So these rows will be removed.

In [184…
```
# Remove rows from with 0 video views
for x in df.index:
    if df.loc[x, 'Video Views'] == 0:
        df.drop(x, inplace = True)

# Validate the results
filtered_df = df[df['Video Views'] == 0]
filtered_df
```

Out[184…

Rank	Youtuber	Subscribers	Video Views	Category	Title	Uploads	Country	Abbreviation	Channel Type	...	Subscribers For Last 30 Days	Created Year	Create Mont

0 rows × 28 columns

In [184…
```
df
```

Out[184…

	Rank	Youtuber	Subscribers	Video Views	Category	Title	Uploads	Country	Abbreviation	Channel Type
0	1	T-Series	245000000	2.280000e+11	Music	T-Series	20082	India	IN	Music
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
8	9	Like Nastya	106000000	9.047906e+10	People & Blogs	Like Nastya Vlog	493	Russia	RU	People
...	...	...	...	...	...	...	...	...	...	...
989	990	Migos ATL	12400000	6.993406e+09	Music	Migos ATL	99	United States	US	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
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

554 rows × 28 columns

In [184…
```
# Resetting the index due to rows which have been dropped from the df
df = df.reset_index(drop = True)
df
```

Out[184…

	Rank	Youtuber	Subscribers	Video Views	Category	Title	Uploads	Country	Abbreviation	Channel Type
0	1	T-Series	245000000	2.280000e+11	Music	T-Series	20082	India	IN	Music
1	3	MrBeast	166000000	2.836884e+10	Entertainment	MrBeast	741	United States	US	Entertainment
2	4	Cocomelon - Nursery Rhymes	162000000	1.640000e+11	Education	Cocomelon - Nursery Rhymes	966	United States	US	Education
3	5	SET India	159000000	1.480000e+11	Shows	SET India	116536	India	IN	Entertainment
4	9	Like Nastya	106000000	9.047906e+10	People & Blogs	Like Nastya Vlog	493	Russia	RU	People
...	...	...	...	...	...	...	...	...	...	...
549	990	Migos ATL	12400000	6.993406e+09	Music	Migos ATL	99	United States	US	Entertainment
550	991	Natan por Aï¿	12300000	9.029610e+09	Sports	Natan por Aï¿	1200	Brazil	BR	Entertainment
551	992	Free Fire India Official	12300000	1.674410e+09	People & Blogs	Free Fire India Official	1500	India	IN	Games
552	994	RobTopGames	12300000	3.741235e+08	Gaming	RobTopGames	39	Sweden	SE	Games
553	995	Make Joke Of	12300000	2.129774e+09	Comedy	Make Joke Of	62	India	IN	Comedy

554 rows × 28 columns

In [184…
```
df.dropna()
```

Out[184…

	Rank	Youtuber	Subscribers	Video Views	Category	Title	Uploads	Country	Abbreviation	Channel Type
0	1	T-Series	245000000	2.280000e+11	Music	T-Series	20082	India	IN	Music
1	3	MrBeast	166000000	2.836884e+10	Entertainment	MrBeast	741	United States	US	Entertainment
2	4	Cocomelon - Nursery Rhymes	162000000	1.640000e+11	Education	Cocomelon - Nursery Rhymes	966	United States	US	Education
3	5	SET India	159000000	1.480000e+11	Shows	SET India	116536	India	IN	Entertainment
4	9	Like Nastya	106000000	9.047906e+10	People & Blogs	Like Nastya Vlog	493	Russia	RU	People
...	...	...	...	...	...	...	...	...	...	...
549	990	Migos ATL	12400000	6.993406e+09	Music	Migos ATL	99	United States	US	Entertainment
550	991	Natan por Aï¿	12300000	9.029610e+09	Sports	Natan por Aï¿	1200	Brazil	BR	Entertainment
551	992	Free Fire India Official	12300000	1.674410e+09	People & Blogs	Free Fire India Official	1500	India	IN	Games
552	994	RobTopGames	12300000	3.741235e+08	Gaming	RobTopGames	39	Sweden	SE	Games
553	995	Make Joke Of	12300000	2.129774e+09	Comedy	Make Joke Of	62	India	IN	Comedy

554 rows × 28 columns

Cleaned data output

In [184…
```
data.to_csv('Global youtube statistics cleaned dataset.csv',index = False, encoding = 'latin')
```

4. Data Visualisation & Analysis

In [184…
```
data = pd.read_csv("Global youtube statistics cleaned dataset.csv", encoding="Latin")
data
```

Out[184…

	Rank	Youtuber	Subscribers	Video Views	Category	Title	Uploads	Country	Abbreviation	Channel Type
0	1	T-Series	245000000	2.280000e+11	Music	T-Series	20082	India	IN	Music
1	3	MrBeast	166000000	2.836884e+10	Entertainment	MrBeast	741	United States	US	Entertainment
2	4	Cocomelon - Nursery Rhymes	162000000	1.640000e+11	Education	Cocomelon - Nursery Rhymes	966	United States	US	Education
3	5	SET India	159000000	1.480000e+11	Shows	SET India	116536	India	IN	Entertainment
4	9	Like Nastya	106000000	9.047906e+10	People & Blogs	Like Nastya Vlog	493	Russia	RU	People
...	...	...	...	...	...	...	...	...	...	...
549	990	Migos ATL	12400000	6.993406e+09	Music	Migos ATL	99	United States	US	Entertainment
550	991	Natan por Aï¿	12300000	9.029610e+09	Sports	Natan por Aï¿	1200	Brazil	BR	Entertainment
551	992	Free Fire India Official	12300000	1.674410e+09	People & Blogs	Free Fire India Official	1500	India	IN	Games
552	994	RobTopGames	12300000	3.741235e+08	Gaming	RobTopGames	39	Sweden	SE	Games
553	995	Make Joke Of	12300000	2.129774e+09	Comedy	Make Joke Of	62	India	IN	Comedy

554 rows × 28 columns

In [185…
```
data.columns
```

Out[185…
```
Index(['Rank', 'Youtuber', 'Subscribers', 'Video Views', 'Category', 'Title',
       'Uploads', 'Country', 'Abbreviation', 'Channel Type',
       'Video Views Rank', 'Country Rank', 'Channel Type Rank',
       'Video Views For The Last 30 Days', 'Lowest Monthly Earnings',
       'Highest Monthly Earnings', 'Lowest Yearly Earnings',
       'Highest Yearly Earnings', 'Subscribers For Last 30 Days',
       'Created Year', 'Created Month', 'Created Date',
       'Gross Tertiary Education Enrollment (%)', 'Population',
       'Unemployment Rate', 'Urban Population', 'Latitude', 'Longitude'],
      dtype='object')
```

Separate columns by object type

In [185…
```
categorical_columns = data.select_dtypes(include = ['object']).columns
categorical_columns
```

Out[185…
```
Index(['Youtuber', 'Category', 'Title', 'Country', 'Abbreviation',
       'Channel Type', 'Created Month', 'Created Date'],
      dtype='object')
```

In [185…
```
# Replace object datatype column missing values with 'Unknown'
data[categorical_columns] = data[categorical_columns].fillna('Unknown')
data[categorical_columns]
```

Out[185…

	Youtuber	Category	Title	Country	Abbreviation	Channel Type	Created Month	Created Date
0	T-Series	Music	T-Series	India	IN	Music	Mar	1970-01-01 00:00:00.000000013
1	MrBeast	Entertainment	MrBeast	United States	US	Entertainment	Feb	1970-01-01 00:00:00.000000020
2	Cocomelon - Nursery Rhymes	Education	Cocomelon - Nursery Rhymes	United States	US	Education	Sep	1970-01-01 00:00:00.000000001
3	SET India	Shows	SET India	India	IN	Entertainment	Sep	1970-01-01 00:00:00.000000020
4	Like Nastya	People & Blogs	Like Nastya Vlog	Russia	RU	People	Jan	1970-01-01 00:00:00.000000014
...	...	...	...	...	...	...	...	...
549	Migos ATL	Music	Migos ATL	United States	US	Entertainment	Jan	1970-01-01 00:00:00.000000017
550	Natan por Aï¿	Sports	Natan por Aï¿	Brazil	BR	Entertainment	Feb	1970-01-01 00:00:00.000000012
551	Free Fire India Official	People & Blogs	Free Fire India Official	India	IN	Games	Sep	1970-01-01 00:00:00.000000014
552	RobTopGames	Gaming	RobTopGames	Sweden	SE	Games	May	1970-01-01 00:00:00.000000009
553	Make Joke Of	Comedy	Make Joke Of	India	IN	Comedy	Aug	1970-01-01 00:00:00.000000001

554 rows × 8 columns

In [185…
```
numerical_columns = data.select_dtypes(include = ['float', 'int64']).columns
numerical_columns
```

Out[185…
```
Index(['Rank', 'Subscribers', 'Video Views', 'Uploads', 'Video Views Rank',
       'Country Rank', 'Channel Type Rank', 'Video Views For The Last 30 Days',
       'Lowest Monthly Earnings', 'Highest Monthly Earnings',
       'Lowest Yearly Earnings', 'Highest Yearly Earnings',
       'Subscribers For Last 30 Days', 'Created Year',
       'Gross Tertiary Education Enrollment (%)', 'Population',
       'Unemployment Rate', 'Urban Population', 'Latitude', 'Longitude'],
      dtype='object')
```

In [185…
```
# Replace numerical datatype column missing values with 'Unknown'
data[numerical_columns] = data[numerical_columns].fillna(0)
data[numerical_columns]
```

Out[185…

	Rank	Subscribers	Video Views	Uploads	Video Views Rank	Country Rank	Channel Type Rank	Video Views For The Last 30 Days	Lowest Monthly Earnings	Highest Monthly Earnings	Lowest Yearly Earnings	Hi Ear
0	1	245000000	2.280000e+11	20082	1.0	1.0	1.0	2.258000e+09	564600.0	9000000.0	6800000.0	10840
1	3	166000000	2.836884e+10	741	48.0	1.0	1.0	1.348000e+09	337000.0	5400000.0	4000000.0	6470
2	4	162000000	1.640000e+11	966	2.0	2.0	1.0	1.975000e+09	493800.0	7900000.0	5900000.0	9480
3	5	159000000	1.480000e+11	116536	3.0	2.0	2.0	1.824000e+09	455900.0	7300000.0	5500000.0	8750
4	9	106000000	9.047906e+10	493	630.0	5.0	25.0	4.894700e+07	12200.0	195800.0	146800.0	230
...	...	...	...	...	...	...	...	...	...	...	...	
549	990	12400000	6.993406e+09	99	833.0	175.0	171.0	4.941200e+07	12400.0	197600.0	148200.0	240
550	991	12300000	9.029610e+09	1200	525.0	55.0	172.0	5.525130e+08	138100.0	2200000.0	1700000.0	2650
551	992	12300000	1.674410e+09	1500	6141.0	125.0	69.0	6.473500e+07	16200.0	258900.0	194200.0	310
552	994	12300000	3.741235e+08	39	35112.0	4.0	69.0	3.871000e+06	968.0	15500.0	11600.0	185
553	995	12300000	2.129774e+09	62	4568.0	125.0	44.0	2.400000e+07	6000.0	96000.0	72000.0	120

554 rows × 20 columns

Find the correlation of data

In [185…
```
data[numerical_columns].corr().round(2)
```

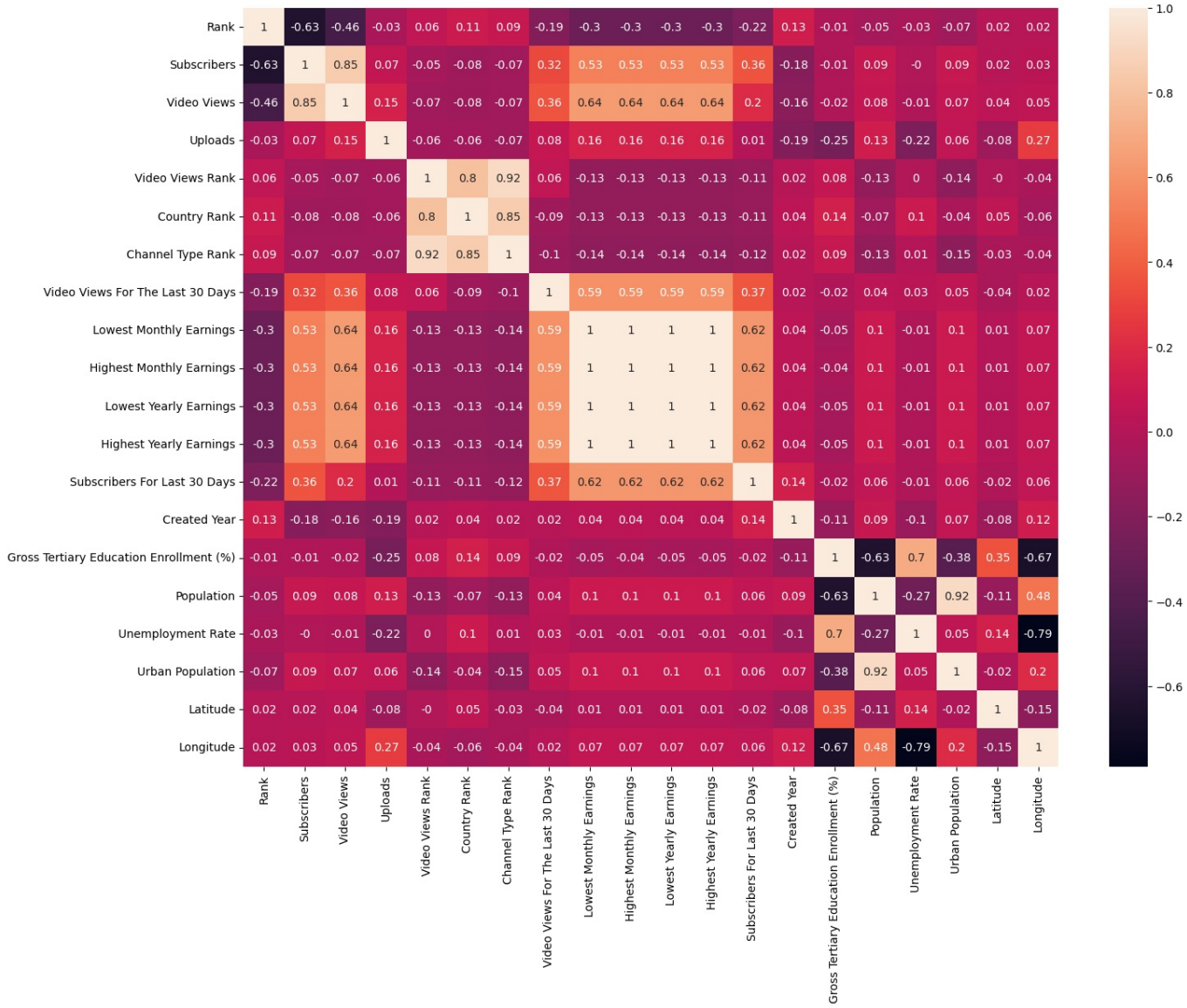
Out[185…

	Rank	Subscribers	Video Views	Uploads	Video Views Rank	Country Rank	Channel Type Rank	Video Views For The Last 30 Days	Lowest Monthly Earnings	Highest Monthly Earnings	Lowest Yearly Earnings	Highest Yearly Earnings	Su F
Rank	1.00	-0.63	-0.46	-0.03	0.06	0.11	0.09	-0.19	-0.30	-0.30	-0.30	-0.30	
Subscribers	-0.63	1.00	0.85	0.07	-0.05	-0.08	-0.07	0.32	0.53	0.53	0.53	0.53	
Video Views	-0.46	0.85	1.00	0.15	-0.07	-0.08	-0.07	0.36	0.64	0.64	0.64	0.64	
Uploads	-0.03	0.07	0.15	1.00	-0.06	-0.06	-0.07	0.08	0.16	0.16	0.16	0.16	
Video Views Rank	0.06	-0.05	-0.07	-0.06	1.00	0.80	0.92	0.06	-0.13	-0.13	-0.13	-0.13	
Country Rank	0.11	-0.08	-0.08	-0.06	0.80	1.00	0.85	-0.09	-0.13	-0.13	-0.13	-0.13	
Channel Type Rank	0.09	-0.07	-0.07	-0.07	0.92	0.85	1.00	-0.10	-0.14	-0.14	-0.14	-0.14	
Video Views For The Last 30 Days	-0.19	0.32	0.36	0.08	0.06	-0.09	-0.10	1.00	0.59	0.59	0.59	0.59	
Lowest Monthly Earnings	-0.30	0.53	0.64	0.16	-0.13	-0.13	-0.14	0.59	1.00	1.00	1.00	1.00	
Highest Monthly Earnings	-0.30	0.53	0.64	0.16	-0.13	-0.13	-0.14	0.59	1.00	1.00	1.00	1.00	
Lowest Yearly Earnings	-0.30	0.53	0.64	0.16	-0.13	-0.13	-0.14	0.59	1.00	1.00	1.00	1.00	
Highest Yearly Earnings	-0.30	0.53	0.64	0.16	-0.13	-0.13	-0.14	0.59	1.00	1.00	1.00	1.00	
Subscribers For Last 30 Days	-0.22	0.36	0.20	0.01	-0.11	-0.11	-0.12	0.37	0.62	0.62	0.62	0.62	
Created Year	0.13	-0.18	-0.16	-0.19	0.02	0.04	0.02	0.02	0.04	0.04	0.04	0.04	
Gross Tertiary Education Enrollment (%)	-0.01	-0.01	-0.02	-0.25	0.08	0.14	0.09	-0.02	-0.05	-0.04	-0.05	-0.05	
Population	-0.05	0.09	0.08	0.13	-0.13	-0.07	-0.13	0.04	0.10	0.10	0.10	0.10	
Unemployment Rate	-0.03	-0.00	-0.01	-0.22	0.00	0.10	0.01	0.03	-0.01	-0.01	-0.01	-0.01	
Urban Population	-0.07	0.09	0.07	0.06	-0.14	-0.04	-0.15	0.05	0.10	0.10	0.10	0.10	
Latitude	0.02	0.02	0.04	-0.08	-0.00	0.05	-0.03	-0.04	0.01	0.01	0.01	0.01	
Longitude	0.02	0.03	0.05	0.27	-0.04	-0.06	-0.04	0.02	0.07	0.07	0.07	0.07	

4.1 Correlation matrix Heatmap

In [185…

```
# Adjust the default figure size for matplotlib plots
plt.rcParams['figure.figsize'] = (16, 12)

# Generate correlation matrix heatmap
sns.heatmap(data[numerical_columns].corr().round(2), annot = True)

# Display the correlation matrix heatmap
plt.show()
```

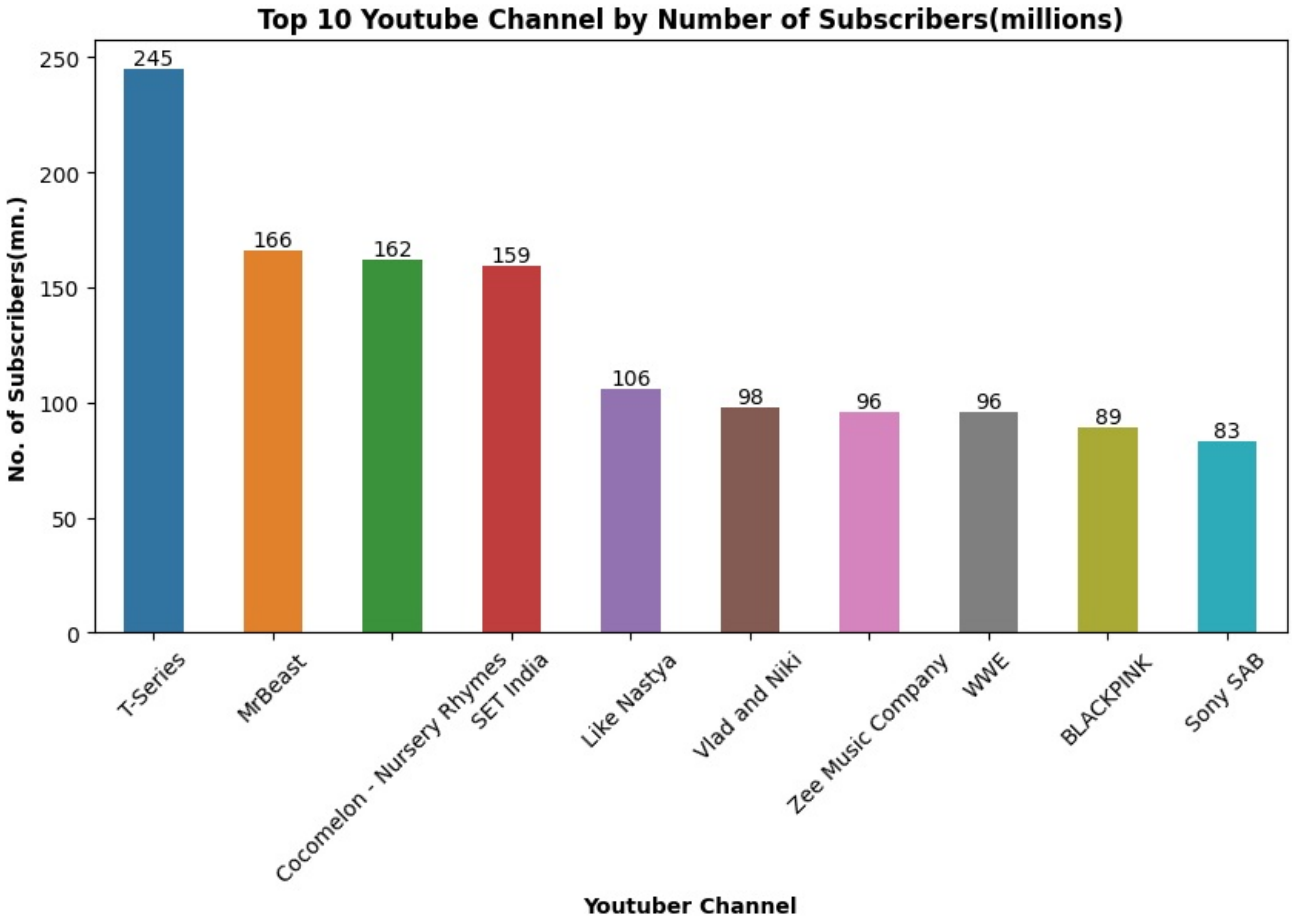


4.2 Top 10 Channels by subscribers(millions)

In [185…

```
# Select columns to show in output
selected_columns = ['Youtuber', 'Subscribers']

# Filter df for top 10 most subscribed YouTube channels
top_10_channels = df.loc[0:9, selected_columns]

# Sort values so that highest output is descending, this is for the horizontal bar chart
top_10_channels_desc = top_10_channels.sort_values(by = 'Subscribers', ascending = False)

# Validate filtered df output
print(top_10_channels_desc)

# Reflect 'Subscriber' in millions
top_10_channels['Subscribers (mn.)'] = (top_10_channels['Subscribers'] / 1000000).astype(int)
top_10_channels_desc['Subscribers (mn.)'] = (top_10_channels_desc['Subscribers'] / 1000000).astype(int)
# Select columns to show in output
top_10_channels_desc = top_10_channels_desc[['Youtuber', 'Subscribers (mn.)']]

plt.figure(figsize=(10,5))
a = sns.barplot(x=top_10_channels_desc['Youtuber'], y=top_10_channels_desc['Subscribers (mn.)'], width = 0.5)
plt.xlabel('Youtuber Channel', weight = 'bold')
plt.ylabel('No. of Subscribers(mn.)', weight = 'bold')
plt.title('Top 10 Youtube Channel by Number of Subscribers(millions)', weight = 'bold')
plt.xticks(rotation=45)
a.bar_label(a.containers[0])
plt.show()
```

	Youtuber	Subscribers
0	T-Series	245000000
1	MrBeast	166000000
2	Cocomelon - Nursery Rhymes	162000000
3	SET India	159000000
4	Like Nastya	106000000
5	Vlad and Niki	98900000
6	Zee Music Company	96700000
7	WWE	96000000
8	BLACKPINK	89800000
9	Sony SAB	83000000



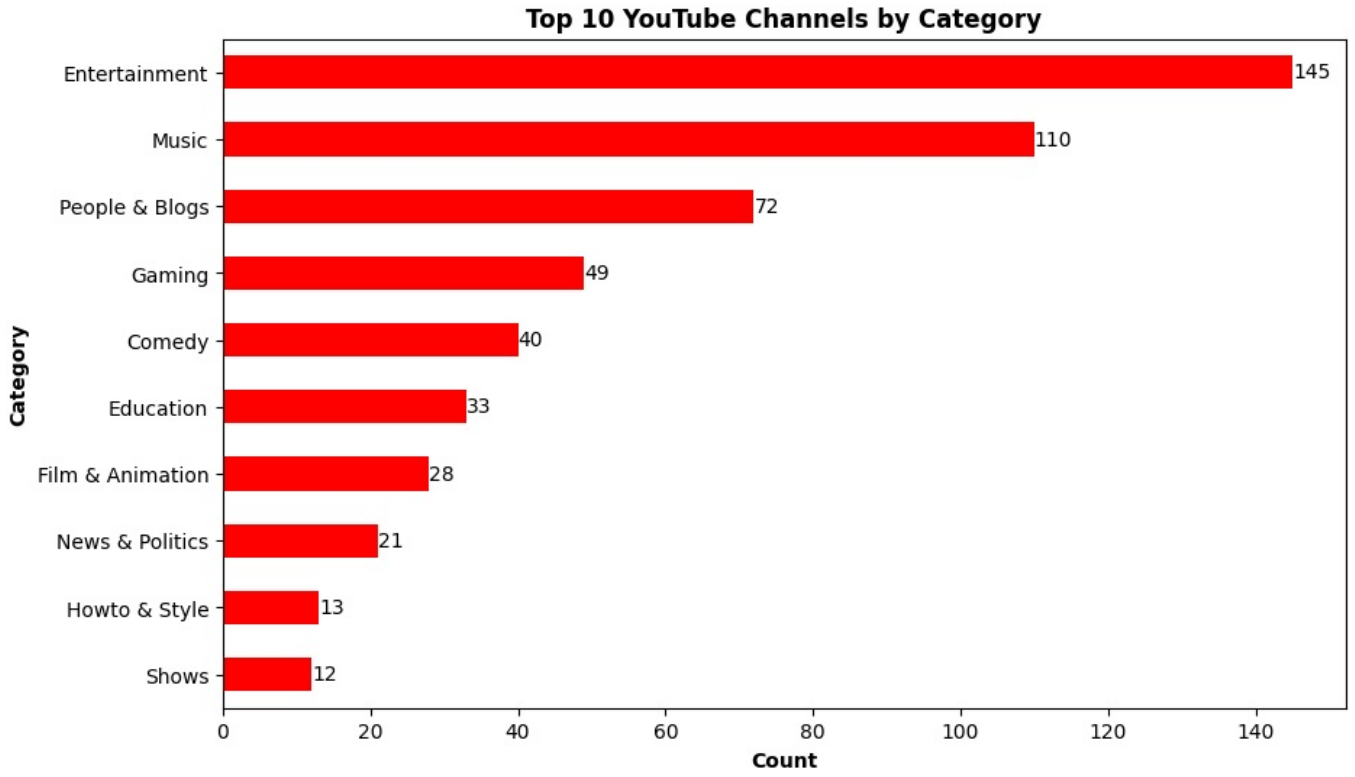
4.3. Top 10 Channels category by Youtube channels

In [185…
```
category_counts = data['Category'].value_counts(ascending = True)
top_10_categories = category_counts.tail(10)
top_10_categories
```

Out[185…
```
Category
Shows                12
Howto & Style        13
News & Politics      21
Film & Animation     28
Education            33
Comedy               40
Gaming               49
People & Blogs       72
Music               110
Entertainment       145
Name: count, dtype: int64
```

In [185…
```
category_counts = data['Category'].value_counts(ascending = False)

# Select the top 10 categories
top_10_categories = category_counts.head(10)

# Create a horizontal bar chart
plt.figure(figsize=(10, 6))
ax = top_10_categories.plot(kind='barh', color='red')
plt.xlabel('Count', weight = 'bold')
plt.ylabel('Category', weight = 'bold')
plt.title('Top 10 YouTube Channels by Category', weight = 'bold')
plt.gca().invert_yaxis()  # Invert the y-axis to display the highest average subscribers at the top

# Add labels near the bars
for i, v in enumerate(top_10_categories):
```

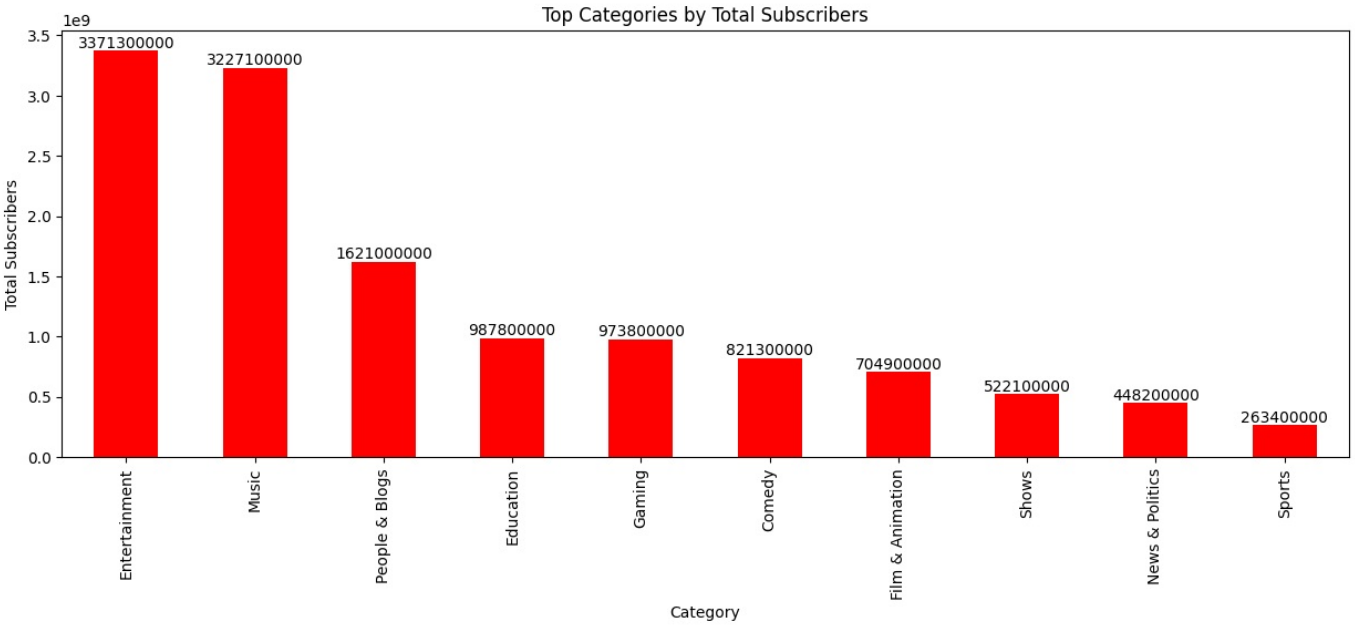
```
    ax.text(v, i, str(round(v, 2)), color='black', va='center')

plt.show()
```



4.4. Top Categories by Subscribers

In [186…

```
plt.figure(figsize=(15,5))

top_categories_by_subscribers = data.groupby('Category')['Subscribers'].sum().sort_values(ascending=False).head

top_categories_by_subscribers.plot(kind='bar', color='red', width = 0.5)
plt.xlabel('Category')
plt.ylabel('Total Subscribers')
plt.title('Top Categories by Total Subscribers')

# Add labels near the bars
for i, v in enumerate(top_categories_by_subscribers):
    plt.text(i, v, str(v), ha='center', va='bottom')

plt.show()
```



4.5. Top Channels in Each Category

In [186…

```
top_channels_by_category = data.groupby('Category').apply(lambda x: x.nlargest(1, 'Subscribers'))
top_channels_by_category
```

Out[186…

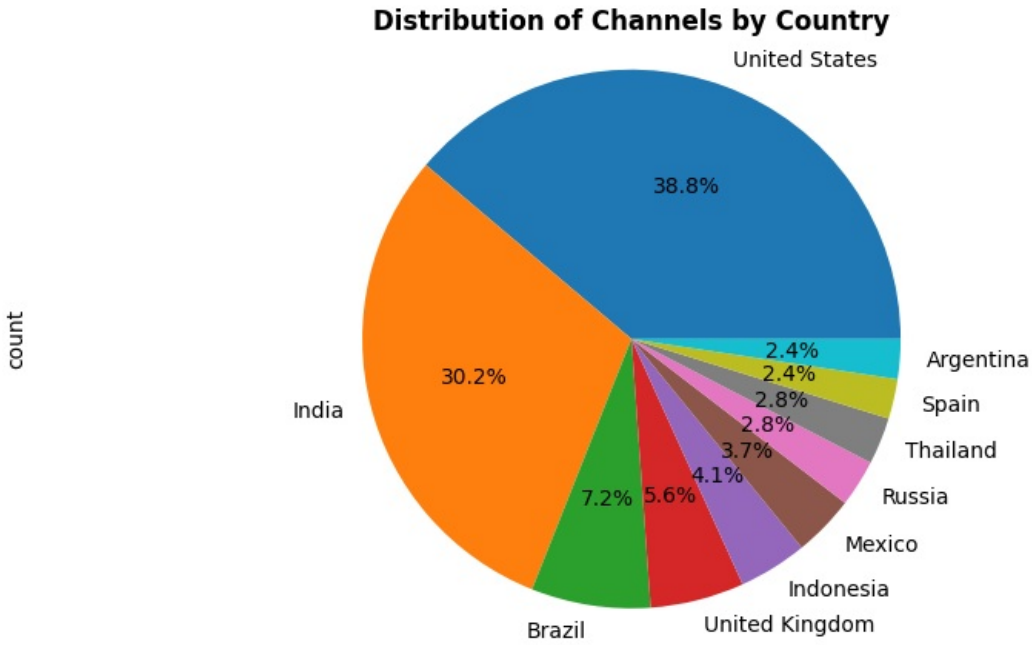
Category	Rank		Youtuber	Subscribers	Video Views	Category	Title	Uploads	Country	Abbreviation
Autos & Vehicles	**211**	334	DUDU e CAROL	21600000	9.597895e+09	Autos & Vehicles	DUDU e CAROL	2942	Brazil	BR
Comedy	**56**	79	CarryMinati	39200000	3.294013e+09	Comedy	CarryMinati	186	India	IN
Education	**2**	4	Cocomelon - Nursery Rhymes	162000000	1.640000e+11	Education	Cocomelon - Nursery Rhymes	966	United States	US
Entertainment	**1**	3	MrBeast	166000000	2.836884e+10	Entertainment	MrBeast	741	United States	US
Film & Animation	**19**	30	Movieclips	59500000	5.931647e+10	Film & Animation	Movieclips	39113	United States	US
Gaming	**33**	45	JuegaGerman	48100000	1.463171e+10	Gaming	JuegaGerman	2052	Chile	CL
Howto & Style	**172**	271	Troom Troom	23800000	1.041448e+10	Howto & Style	Troom Troom	2425	United States	US
Movies	**123**	183	Aditya Movies	28400000	1.006277e+10	Movies	Aditya Movies	5436	India	IN
Music	**0**	1	T-Series	245000000	2.280000e+11	Music	T-Series	20082	India	IN
News & Politics	**23**	34	Aaj Tak	57600000	2.530775e+10	News & Politics	Aaj Tak	283775	India	IN
Nonprofits & Activism	**62**	85	TEDx Talks	38600000	7.339333e+09	Nonprofits & Activism	TEDx Talks	200933	United States	US
People & Blogs	**4**	9	Like Nastya	106000000	9.047906e+10	People & Blogs	Like Nastya Vlog	493	Russia	RU
Pets & Animals	**174**	276	That Little Puff	23700000	2.028969e+10	Pets & Animals	That Little Puff	769	United States	US
Science & Technology	**103**	144	MR. INDIAN HACKER	31700000	5.711208e+09	Science & Technology	MR. INDIAN HACKER	929	India	IN
Shows	**3**	5	SET India	159000000	1.480000e+11	Shows	SET India	116536	India	IN
Sports	**7**	12	WWE	96000000	7.742847e+10	Sports	WWE	70127	United States	US
Trailers	**52**	71	Ishtar Music	41400000	1.760893e+10	Trailers	Ishtar Music	4510	India	IN

17 rows × 28 columns

4.6. Geographical Analysis

In [186…

```
plt.figure(figsize=(10,5))

top_countries_by_channels = data['Country'].value_counts().head(10)
top_countries_by_channels.plot(kind='pie', autopct='%1.1f%%')
plt.axis('equal')
plt.title('Distribution of Channels by Country', weight = 'bold')
plt.show()
```

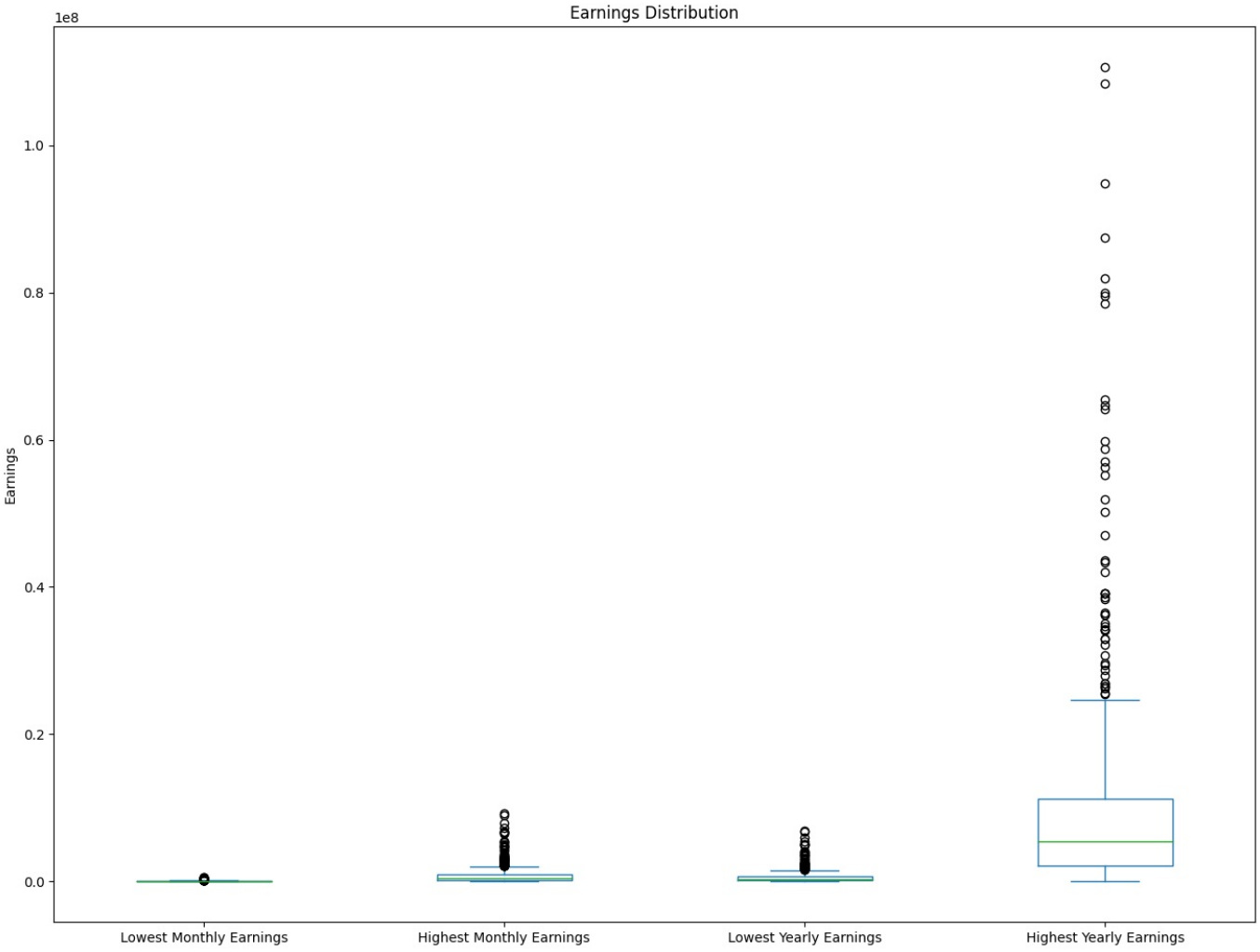


4.7. Earnings Analysis

In [186…

```
plt.figure(figsize=(18,5))

earnings_data = data[['Lowest Monthly Earnings', 'Highest Monthly Earnings', 'Lowest Yearly Earnings', 'Highest
earnings_data.plot(kind='box')
plt.ylabel('Earnings')
plt.title('Earnings Distribution')
plt.show()
```

```
<Figure size 1800x500 with 0 Axes>
```

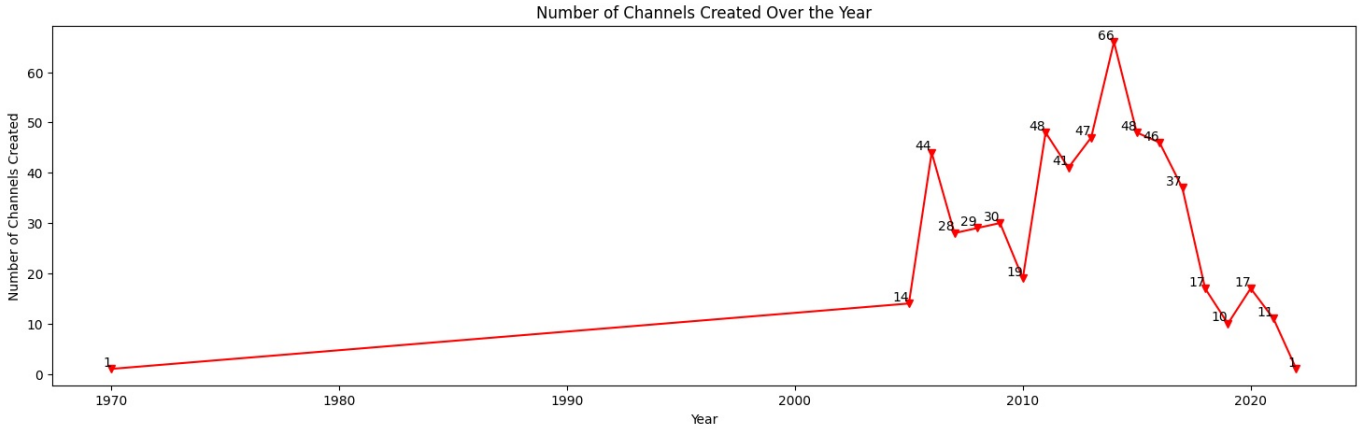


4.8 Channels created over the year

In [186…

```
plt.figure(figsize=(18,5))
```

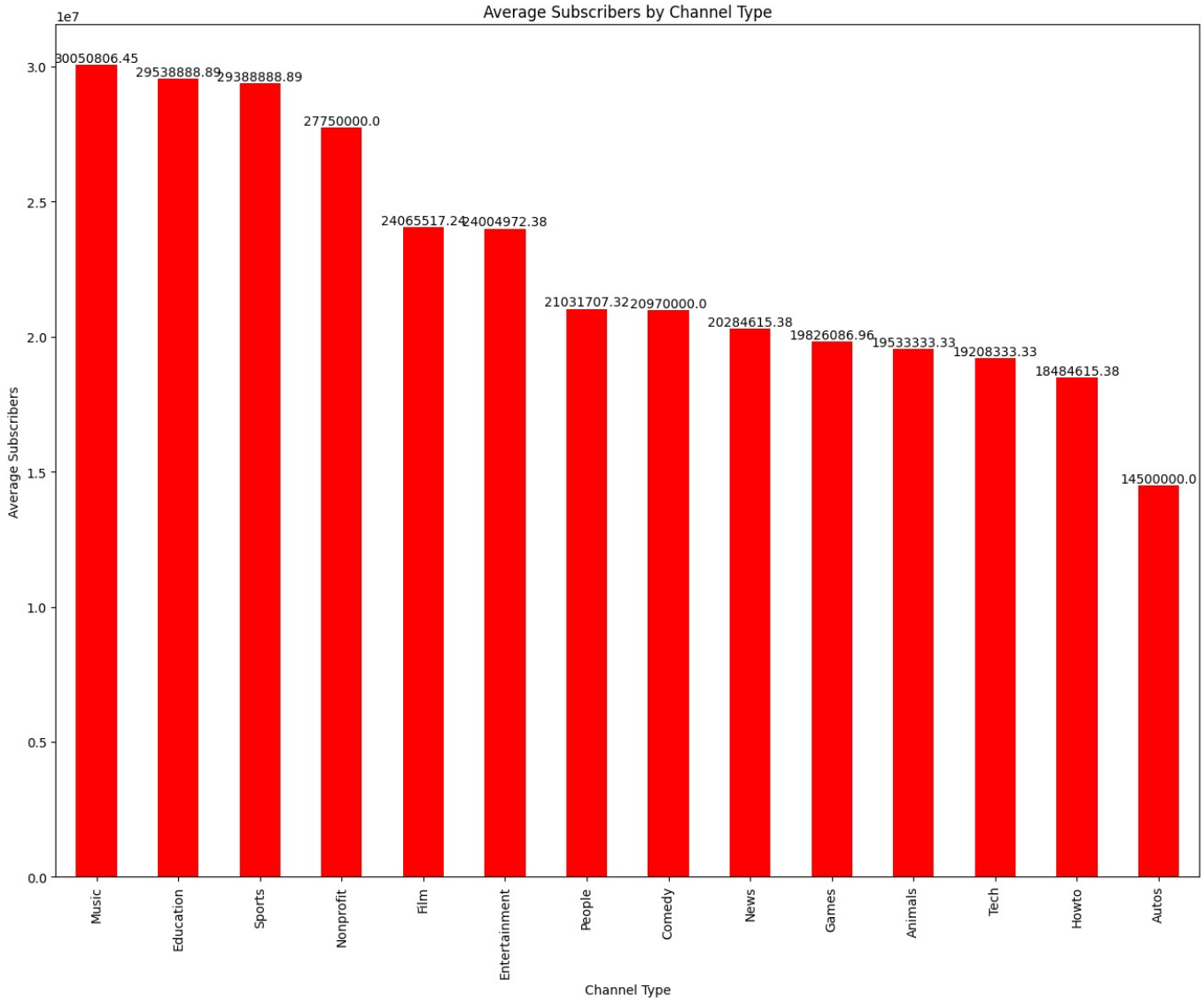
```
channels_by_year = data['Created Year'].value_counts().sort_index()
channels_by_year.plot(kind='line', marker='v', color='red')
plt.xlabel('Year')
plt.ylabel('Number of Channels Created')
plt.title('Number of Channels Created Over the Year')

# Add labels near the markers
for year, count in zip(channels_by_year.index, channels_by_year.values):
    plt.text(year, count, str(count), ha='right', va='bottom')

plt.show()
```



4.9. Average Subscribers by channel type*

In [186…

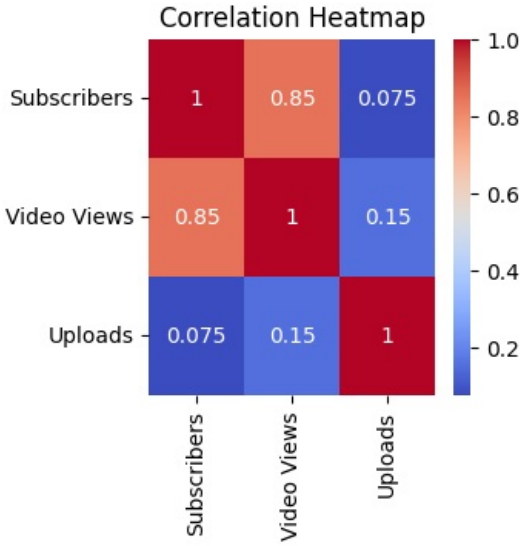
```
channel_type_comparison = data.groupby('Channel Type')['Subscribers'].mean().sort_values(ascending=False)
channel_type_comparison.plot(kind='bar', color='red')
plt.xlabel('Channel Type')
plt.ylabel('Average Subscribers')
plt.title('Average Subscribers by Channel Type')

# Add labels near the bars
for i, v in enumerate(channel_type_comparison):
    plt.text(i, v, str(round(v, 2)), ha='center', va='bottom')

plt.show()
```



4.10. Heat map

In [186…

```
plt.figure(figsize=(3,3))
correlation_matrix = data[['Subscribers', 'Video Views', 'Uploads']].corr()
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()
```



4.11. Average_education_enrollment

In [186…

```
average_education_enrollment = data['Gross Tertiary Education Enrollment (%)'].mean()
print("Average_education_enrollment: ",average_education_enrollment)
```

```
Average_education_enrollment:  60.533574007220224
```

4.12. Trending channels

In [186…
```
trending_channels = data.nlargest(10, 'Video Views For The Last 30 Days')
trending_channels
```

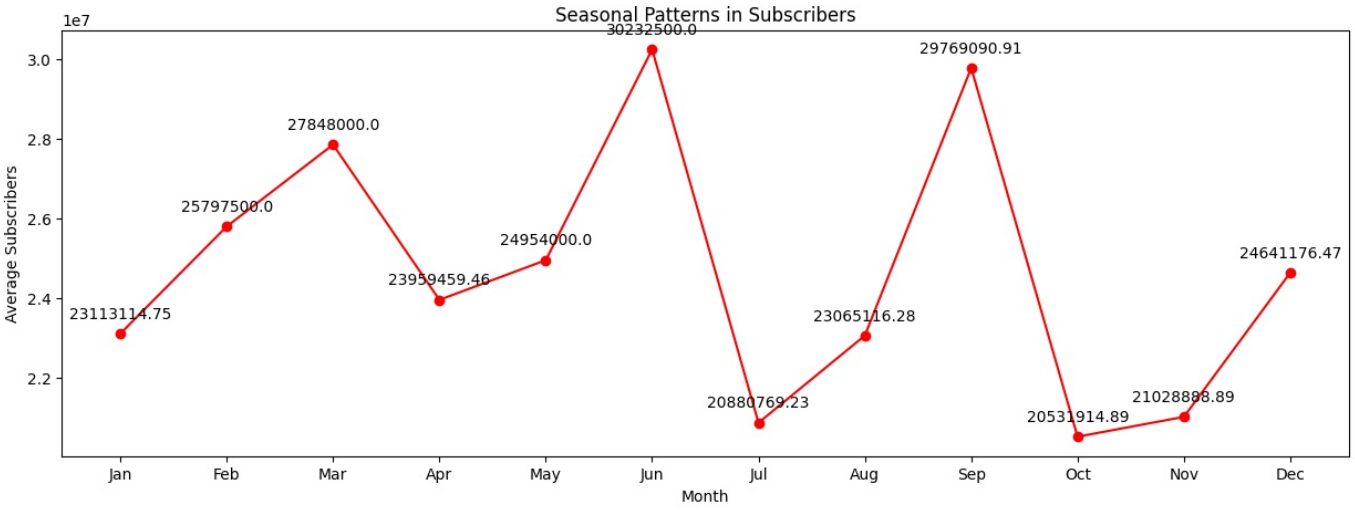
Out[186…

	Rank	Youtuber	Subscribers	Video Views	Category	Title	Uploads	Country	Abbreviation	Channel Type	...	S… F…
180	287	Happy Lives	23200000	2.634000e+03	Science & Technology	Happy Lives	1	United States	US	Entertainment	...	
282	456	Dan-Sa / Daniel Saboya	18500000	2.908121e+09	Music	Dan-Sa / Daniel Saboya	1329	Brazil	BR	Music	...	
261	418	DaFuq!? Boom!	19600000	7.906182e+09	Film & Animation	DaFuq!? Boom!	214	United States	US	Entertainment	...	
514	903	Calon Sarjana	13000000	1.066458e+07	Entertainment	Calon Sarjana	29	Indonesia	ID	Entertainment	...	
0	1	T-Series	245000000	2.280000e+11	Music	T-Series	20082	India	IN	Music	...	
2	4	Cocomelon - Nursery Rhymes	162000000	1.640000e+11	Education	Cocomelon - Nursery Rhymes	966	United States	US	Education	...	
3	5	SET India	159000000	1.480000e+11	Shows	SET India	116536	India	IN	Entertainment	...	
13	22	Zee TV	70500000	7.313905e+10	Entertainment	Zee TV	129204	India	IN	Entertainment	...	
101	140	StarPlus	32000000	2.680067e+10	Entertainment	StarPlus	44892	India	IN	Entertainment	...	
9	16	Sony SAB	83000000	1.010000e+11	Shows	Sony SAB	71270	India	IN	Entertainment	...	

10 rows × 28 columns

4.13. Monthwise Avg. subscribers

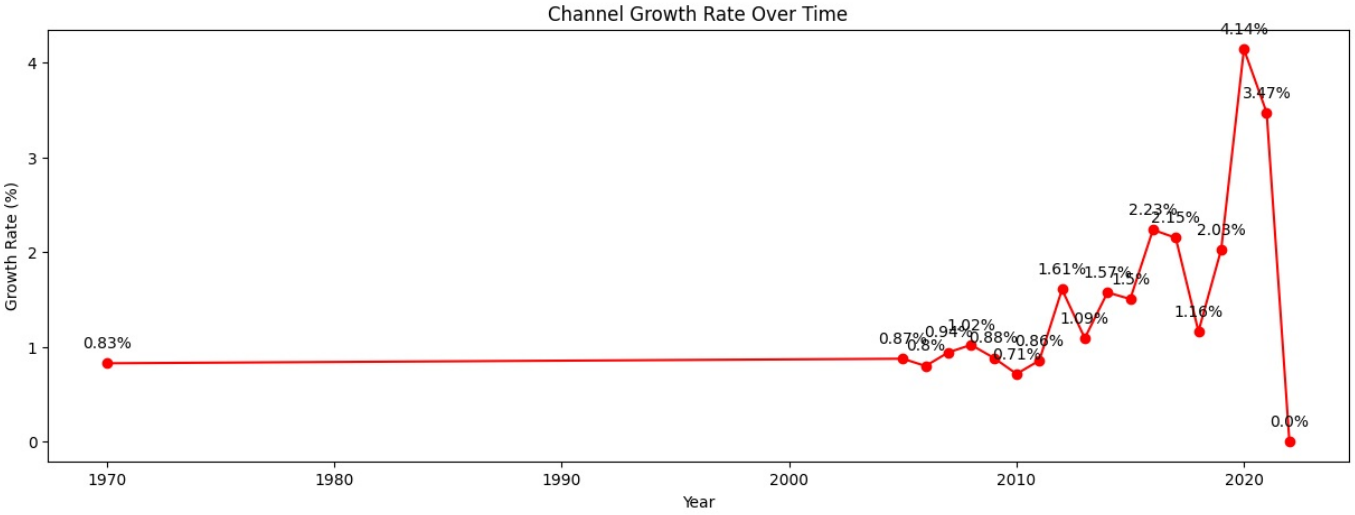
In [186…
```
plt.figure(figsize=(15, 5))

# Define a dictionary to map month names to numerical values
month_to_num = {
    'Jan': 1, 'Feb': 2, 'Mar': 3, 'Apr': 4, 'May': 5, 'Jun': 6,
    'Jul': 7, 'Aug': 8, 'Sep': 9, 'Oct': 10, 'Nov': 11, 'Dec': 12
}

# Apply the conversion to 'Created Month' and sort by numerical month
df['Month_Num'] = df['Created Month'].map(month_to_num)
seasonal_pattern = df.groupby('Month_Num')['Subscribers'].mean().sort_index()

# Plot the data
seasonal_pattern.plot(kind='line', marker='o', color='red')
plt.xticks(seasonal_pattern.index, seasonal_pattern.index.map({v: k for k, v in month_to_num.items()}))  # Set r
plt.xlabel('Month')
plt.ylabel('Average Subscribers')
plt.title('Seasonal Patterns in Subscribers')

# Annotate data points with labels
for month, avg_subs in zip(seasonal_pattern.index, seasonal_pattern.values):
    plt.annotate(f'{round(avg_subs, 2)}', (month, avg_subs), textcoords="offset points", xytext=(0, 10), ha='ce

plt.show()
```



4.14. Channel Growth Rate

In [187…

```
plt.figure(figsize=(15, 5))

growth_rate = data.groupby('Created Year')['Subscribers For Last 30 Days'].sum() / data.groupby('Created Year')
growth_rate.plot(kind='line', marker='o', color='red')
plt.xlabel('Year')
plt.ylabel('Growth Rate (%)')
plt.title('Channel Growth Rate Over Time')

# Add labels near the markers
for year, rate in zip(growth_rate.index, growth_rate):
    plt.annotate(f'{round(rate, 2)}%', (year, rate), textcoords="offset points", xytext=(0, 10), ha='center')

plt.show()
```

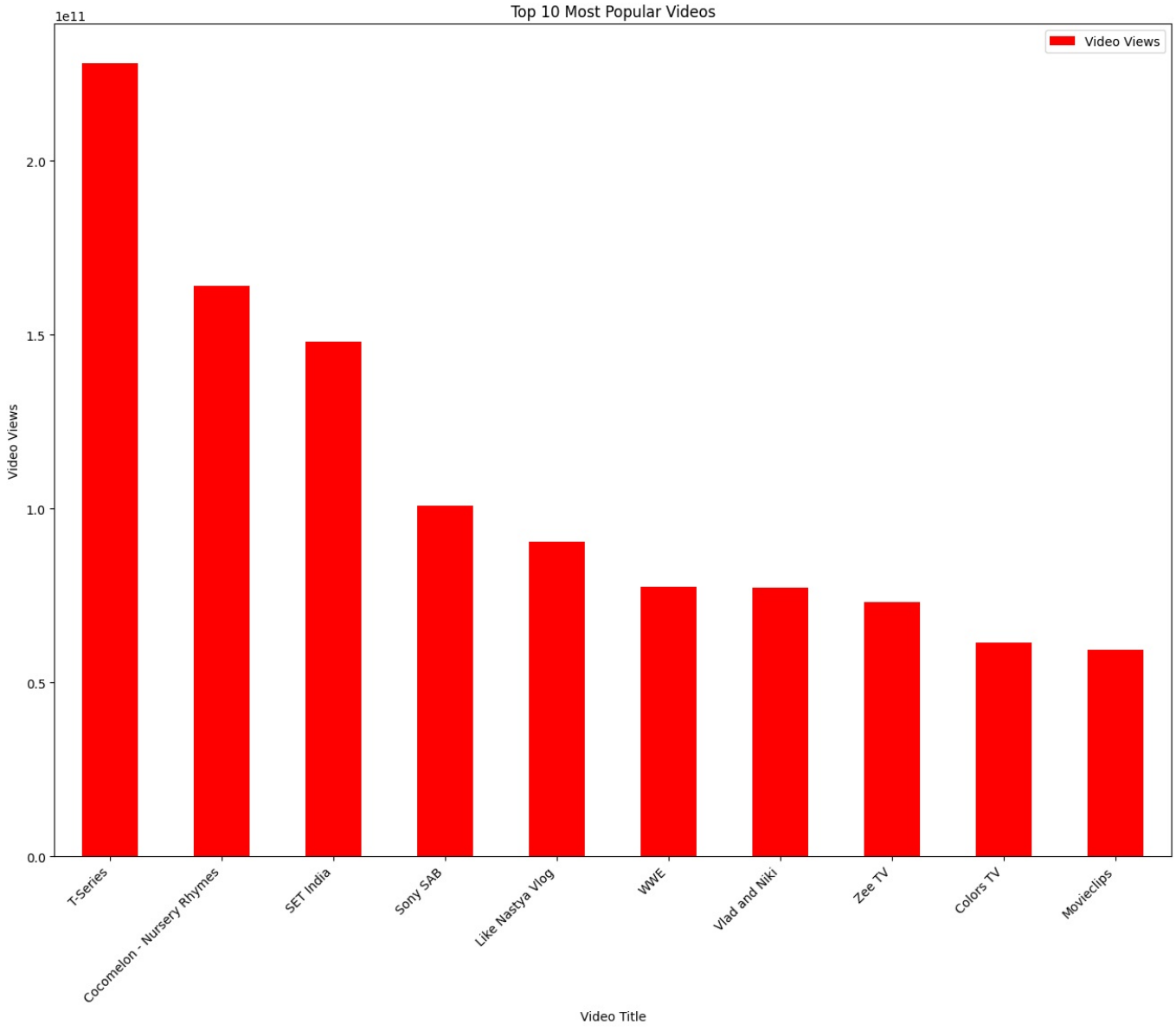


4.15. Video Popularity by views

In [187…

```
plt.figure(figsize=(15, 5))

top_videos = data.nlargest(10, 'Video Views')
top_videos.plot(kind='bar', x='Title', y='Video Views', color='red')
plt.xlabel('Video Title')
plt.ylabel('Video Views')
plt.title('Top 10 Most Popular Videos')
plt.xticks(rotation=45, ha='right')

plt.show()
```

```
<Figure size 1500x500 with 0 Axes>
```

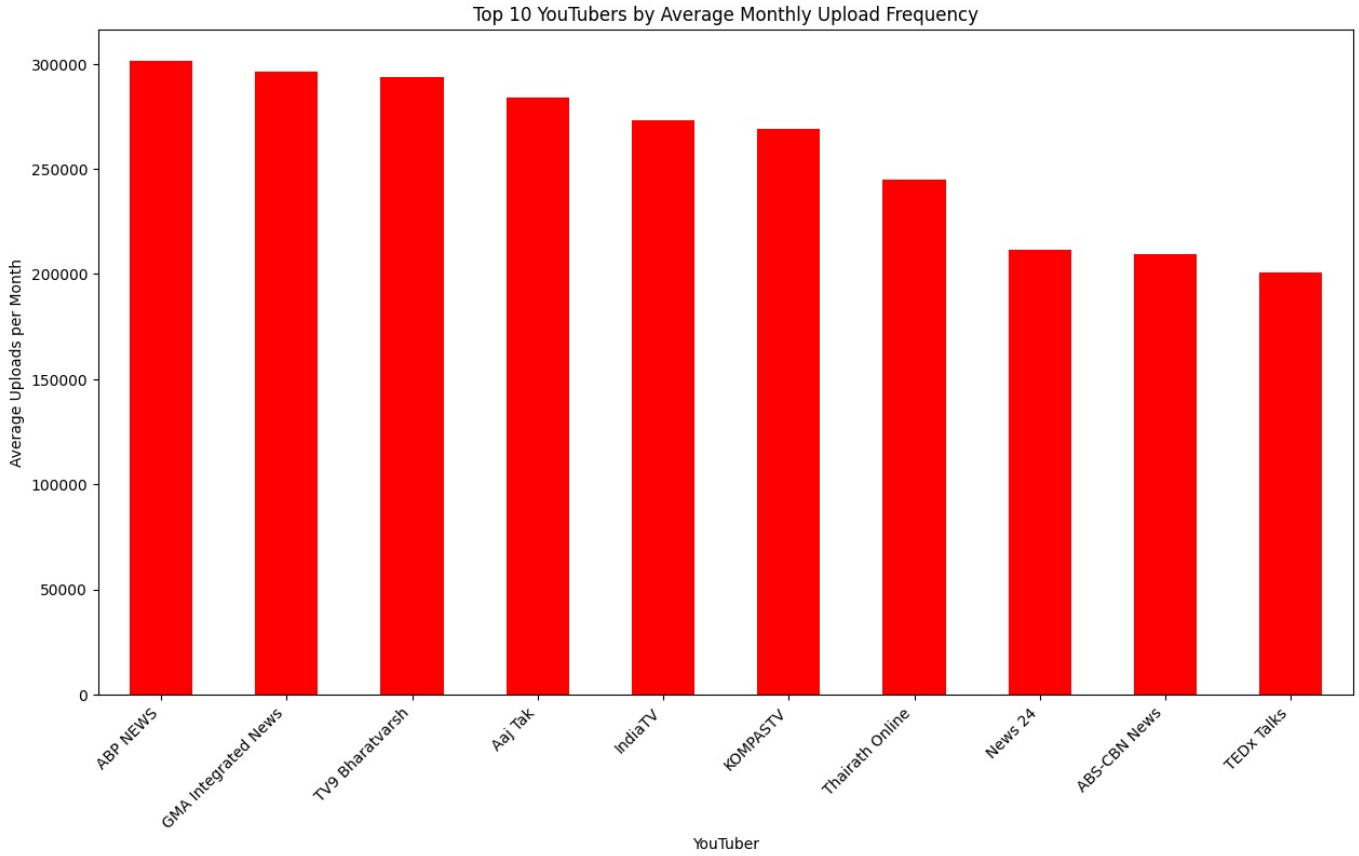


Monthly upload frequency

In [187…

```
# Convert the 'Created Date' column to a pandas datetime object
data['Created Date'] = pd.to_datetime(data['Created Date'])

# Group the data by 'Youtuber' and count the number of uploads per month
uploads_per_month = data.groupby(['Youtuber', data['Created Date'].dt.to_period('M')])['Uploads'].sum().reset_i

# Calculate the mean upload frequency per month
mean_upload_frequency = uploads_per_month.groupby('Youtuber')['Uploads'].mean()

# Sort the data by mean upload frequency in descending order
mean_upload_frequency = mean_upload_frequency.sort_values(ascending=False)

# Select the top 10 channels by upload frequency
top_10_uploaders = mean_upload_frequency.head(10)

# Create a bar chart to visualize the upload frequency
plt.figure(figsize=(15, 8))
top_10_uploaders.plot(kind='bar', color='red')
plt.xlabel('YouTuber')
plt.ylabel('Average Uploads per Month')
plt.title('Top 10 YouTubers by Average Monthly Upload Frequency')
plt.xticks(rotation=45, ha='right')
```

```
plt.show()
```


Top 10 YouTubers by Average Monthly Upload Frequency
Average Uploads per Month
0
50000
100000
150000
200000
250000
300000
ABP NEWS
GMA Integrated News
TV9 Bharatvarsh
Aaj Tak
IndiaTV
KOMPASTV
Thairath Online
News 24
ABS-CBN News
TEDx Talks
YouTuber