Introduction to EDA

The main objective of this program is to cover the steps involved in Data pre-processing, Feature Engineering, and different stages of Exploratory Data Analysis, which is an essential step in any research analysis. Data pre-processing, Feature Engineering, and EDA are fundamental early steps after data collection. Still, they are not limited to where the data is simply visualized, plotted, and manipulated, without any assumptions, to assess the quality of the data and building models. This program will guide you through data pre-processing, feature engineering, and EDA using Python.

Steps for performing EDA with Python

1. **Data Collection**: Gather the relevant data from various sources such as databases, APIs, or files. This step involves compiling all the necessary information for analysis.

2. **Data Cleaning**: Identify and handle missing, incomplete, or erroneous data. Cleaning involves tasks like imputation, removing duplicates, and dealing with outliers to ensure data quality.

3. **Data Exploration**: Perform initial exploration to understand the structure, patterns, and relationships within the data. This step involves summary statistics, visualizations, and identifying potential trends or anomalies.

4. **Feature Engineering**: Create new features or transform existing ones to improve model performance. Feature engineering aims to extract relevant information from the data and enhance its predictive power.

5. **Data Visualization**: Utilize graphs, charts, and plots to visually represent the data. Visualization aids in understanding complex relationships and patterns that may not be apparent from raw data alone.

6. **Statistical Analysis**: Apply statistical techniques to gain deeper insights into the data. Statistical analysis helps in hypothesis testing, assessing correlations, and making inferences about the population from the sample.

7. **Documentation and Reporting**: Document the findings, methodologies, and insights obtained from the analysis. Reporting involves communicating results effectively to stakeholders through reports, presentations, or dashboards.

Data Collection:

The data utilized in this project was gathered through web scraping techniques employing a web scraping tool available at Chrome Web Store from the website goodcreator.co.

Link of Dataset: Micro-Influencer_Dataset

#Data Cleaning

Import all libraries which are required for our analysis, such as Data Loading, Statistical analysis, Visualizations, Data Transformations, Merge and Joins, etc.

Pandas and Numpy have been used for Data Manipulation and numerical Calculations

Matplotlib and Seaborn have been used for Data visualizations.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
#to ignore warnings
import warnings
warnings.filterwarnings('ignore')

data = pd.read_csv("Raw_Dataset.csv")

data.shape
```

```
(3299, 20)
```

```
data.head(10)
```

```
{"summary":"{\n  \"name\": \"data\",\n  \"rows\": 3299,\n  \"fields\":
[\n    {\n      \"column\": \"web-scraper-order\",\n
\"properties\": {\n        \"dtype\": \"string\",\n
\"num_unique_values\": 3299,\n        \"samples\": [\n
\"1708356967-2130\",\n          \"1708351151-1279\",\n
\"1708363186-3154\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"web-scraper-start-url\",\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 74,\n
\"samples\": [\n          \"https://goodcreator.co/instagram-
influencers/movie-shows/under-10k-followers?sourcePage=Creator
%20listing%20page&sourceCTA=Filters\",\n
\"https://goodcreator.co/instagram-influencers/sports/under-20k-
followers?sourcePage=Creator%20listing%20page&sourceCTA=Filters\",\n
\"https://goodcreator.co/instagram-influencers/food-drinks/under-10k-
followers?sourcePage=Creator%20listing%20page&sourceCTA=Filters\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Detail\",\n      \"properties\":
{\n        \"dtype\": \"category\",\n        \"num_unique_values\":
1,\n        \"samples\": [\n          \"Details\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Detail-href\",\n
\"properties\": {\n        \"dtype\": \"string\",\n
\"num_unique_values\": 3299,\n        \"samples\": [\n
\"https://goodcreator.co/influencer/littlebigpixar.goa/IA_31200?
```

```
sourceCTA=Details%20option&soucePage=Creator%20listing%20page\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Name\",\n      \"properties\":
{\n        \"dtype\": \"string\",\n        \"num_unique_values\":
3292,\n        \"samples\": [\n          \"Dietitian Sumaiyya
Shoaib\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Image-src\",\n      \"properties\": {\n        \"dtype\":
\"string\",\n        \"num_unique_values\": 2987,\n
\"samples\": [\n
\"https://d24w28i6lzk071.cloudfront.net/assets/profiles/instagram/
1507016449.jpg\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Category\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 23,\n
\"samples\": [\n          \"Science & Technology\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Ig_Follow\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
799761.944987238,\n        \"min\": 15.0,\n        \"max\":
32230163.0,\n        \"num_unique_values\": 2312,\n
\"samples\": [\n          49017.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"eng_rate\",\n      \"properties\":
{\n        \"dtype\": \"category\",\n        \"num_unique_values\":
798,\n        \"samples\": [\n          \"18.32%\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"est_Reel_play\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
261664.90367386656,\n        \"min\": 0.0,\n        \"max\":
4776812.0,\n        \"num_unique_values\": 2253,\n        \"samples\":
[\n          4634.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Avg_like\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 17632.847423515133,\n        \"min\":
0.0,\n        \"max\": 413120.0,\n        \"num_unique_values\":
1055,\n        \"samples\": [\n          580.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Avg_Comments\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
1203.7465185769481,\n        \"min\": 0.0,\n        \"max\": 60546.0,\
n        \"num_unique_values\": 170,\n        \"samples\": [\n
149.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"yt_subs\",\n      \"properties\": {\n        \"dtype\": \"number\",\
n        \"std\": 2710527.3595734197,\n        \"min\": 409.0,\n
\"max\": 23400000.0,\n        \"num_unique_values\": 722,\n
\"samples\": [\n          3030000.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
```

```
n    },\n    {\n      \"column\": \"views(30 days)\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
52437860.40603506,\n        \"min\": -1550886728.0,\n        \"max\":
401452781.0,\n        \"num_unique_values\": 1229,\n
\"samples\": [\n          1818.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"total_views\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
837403120.6754007,\n        \"min\": 0.0,\n        \"max\":
12800740251.0,\n        \"num_unique_values\": 1293,\n
\"samples\": [\n          4335939637.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"uplaods\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\":
2376.048567133539,\n        \"min\": 0.0,\n        \"max\": 74336.0,\n
\"num_unique_values\": 782,\n        \"samples\": [\n          7240.0\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"yt_links\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": null,\n        \"min\": null,\n
\"max\": null,\n        \"num_unique_values\": 0,\n
\"samples\": [],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"yt_links-href\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 1305,\n
\"samples\": [],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"insta_link\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": null,\n        \"min\": null,\n
\"max\": null,\n        \"num_unique_values\": 0,\n
\"samples\": [],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"insta_link-href\",\n      \"properties\": {\n        \"dtype\":
\"string\",\n        \"num_unique_values\": 3296,\n
\"samples\": [],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"data"}
```


```
data.tail(10)
```


```
{"repr_error":"0","type":"dataframe"}
```


```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3299 entries, 0 to 3298
Data columns (total 20 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   web-scraper-order     3299 non-null   object
```

```
 1   web-scraper-start-url  3299 non-null   object
 2   Detail                 3299 non-null   object
 3   Detail-href            3299 non-null   object
 4   Name                   3299 non-null   object
 5   Image-src              2990 non-null   object
 6   Category               3299 non-null   object
 7   Ig_Follow              2667 non-null   float64
 8   eng_rate               2667 non-null   object
 9   est_Reel_play          2500 non-null   float64
 10  Avg_like               2660 non-null   float64
 11  Avg_Comments           2573 non-null   float64
 12  yt_subs                1308 non-null   float64
 13  views(30 days)         1308 non-null   float64
 14  total_views            1308 non-null   float64
 15  uplaods                1308 non-null   float64
 16  yt_links               0 non-null      float64
 17  yt_links-href          1308 non-null   object
 18  insta_link             0 non-null      float64
 19  insta_link-href        3299 non-null   object
dtypes: float64(10), object(10)
memory usage: 515.6+ KB

data.nunique()

web-scraper-order        3299
web-scraper-start-url      74
Detail                      1
Detail-href              3299
Name                     3292
Image-src                2987
Category                   23
Ig_Follow                2312
eng_rate                  798
est_Reel_play            2253
Avg_like                 1055
Avg_Comments              170
yt_subs                   722
views(30 days)           1229
total_views              1293
uplaods                   782
yt_links                    0
yt_links-href            1305
insta_link                  0
insta_link-href          3296
dtype: int64

(data.nunique()/(len(data)))*100

web-scraper-order        100.000000
web-scraper-start-url      2.243104
```

```
Detail                      0.030312
Detail-href               100.000000
Name                       99.787814
Image-src                  90.542589
Category                    0.697181
Ig_Follow                  70.081843
eng_rate                   24.189148
est_Reel_play              68.293422
Avg_like                   31.979388
Avg_Comments                5.153077
yt_subs                    21.885420
views(30 days)             37.253713
total_views                39.193695
uplaods                    23.704153
yt_links                    0.000000
yt_links-href              39.557442
insta_link                  0.000000
insta_link-href            99.909063
dtype: float64
```

```
data.isnull().sum()
```

```
web-scraper-order           0
web-scraper-start-url       0
Detail                      0
Detail-href                 0
Name                        0
Image-src                 309
Category                    0
Ig_Follow                 632
eng_rate                  632
est_Reel_play             799
Avg_like                  639
Avg_Comments              726
yt_subs                  1991
views(30 days)           1991
total_views              1991
uplaods                  1991
yt_links                 3299
yt_links-href            1991
insta_link               3299
insta_link-href             0
dtype: int64
```

```
(data.isnull().sum()/(len(data)))*100
```

```
web-scraper-order           0.000000
web-scraper-start-url       0.000000
Detail                      0.000000
Detail-href                 0.000000
```

```
Name                          0.000000
Image-src                     9.366475
Category                      0.000000
Ig_Follow                    19.157320
eng_rate                     19.157320
est_Reel_play                24.219460
Avg_like                     19.369506
Avg_Comments                 22.006669
yt_subs                      60.351622
views(30 days)               60.351622
total_views                  60.351622
uplaods                      60.351622
yt_links                    100.000000
yt_links-href                60.351622
insta_link                  100.000000
insta_link-href               0.000000
dtype: float64
```

```
data.columns
```

```
Index(['web-scraper-order', 'web-scraper-start-url', 'Detail',
'Detail-href',
       'Name', 'Image-src', 'Category', 'Ig_Follow', 'eng_rate',
       'est_Reel_play', 'Avg_like', 'Avg_Comments', 'yt_subs',
       'views(30 days)', 'total_views', 'uplaods', 'yt_links',
'yt_links-href',
       'insta_link', 'insta_link-href'],
      dtype='object')
```

```
cols_remove=['web-scraper-start-url', 'Detail', 'Detail-
href','yt_links','insta_link']
data.drop(cols_remove, axis=1, inplace=True)
```

```
data.shape
```

```
(3299, 15)
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3299 entries, 0 to 3298
Data columns (total 15 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   web-scraper-order  3299 non-null   object
 1   Name               3299 non-null   object
 2   Image-src          2990 non-null   object
 3   Category           3299 non-null   object
 4   Ig_Follow          2667 non-null   float64
 5   eng_rate           2667 non-null   object
 6   est_Reel_play      2500 non-null   float64
```

```
 7   Avg_like           2660 non-null   float64
 8   Avg_Comments       2573 non-null   float64
 9   yt_subs            1308 non-null   float64
 10  views(30 days)     1308 non-null   float64
 11  total_views        1308 non-null   float64
 12  uplaods            1308 non-null   float64
 13  yt_links-href      1308 non-null   object
 14  insta_link-href    3299 non-null   object
dtypes: float64(8), object(7)
memory usage: 386.7+ KB
```

```
data.describe(include='all').T
```


```
{"summary":"{\n  \"name\": \"data\",\n  \"rows\": 15,\n  \"fields\":
[\n    {\n      \"column\": \"count\",\n      \"properties\": {\n
\"dtype\": \"date\",\n        \"min\": 1308.0,\n        \"max\":
\"3299\",\n        \"num_unique_values\": 7,\n        \"samples\": [\n
\"3299\",\n          \"2990\",\n          2573.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"unique\",\n      \"properties\":
{\n        \"dtype\": \"date\",\n        \"min\": 23,\n
\"max\": 3299,\n        \"num_unique_values\": 7,\n
\"samples\": [\n          3299,\n          3292,\n          1305\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"top\",\n      \"properties\": {\
n        \"dtype\": \"category\",\n        \"num_unique_values\": 7,\n
\"samples\": [\n          \"1708342125-1\",\n          \"Amit
Sharma\",\n
\"https://www.youtube.com/channel/UC6x8yEi3ZNFfwlnPzFI1GIA\"\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"freq\",\n      \"properties\": {\n        \"dtype\": \"date\",\n
\"min\": \"1\",\n        \"max\": \"288\",\n
\"num_unique_values\": 4,\n        \"samples\": [\n          \"2\",\n
\"43\",\n          \"1\"\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"mean\",\n      \"properties\": {\n        \"dtype\":
\"date\",\n        \"min\": 50.953750485814226,\n        \"max\":
288668995.96941894,\n        \"num_unique_values\": 8,\n
\"samples\": [\n          42480.1296,\n          5924248.713302752,\n
85014.4083239595\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"std\",\n      \"properties\": {\n        \"dtype\": \"date\",\n
\"min\": 1203.7465185769481,\n        \"max\": 837403120.6754007,\n
\"num_unique_values\": 8,\n        \"samples\": [\n
261664.90367386656,\n          52437860.40603506,\n
799761.944987238\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"min\",\n      \"properties\": {\n        \"dtype\": \"date\",\n
\"min\": -1550886728.0,\n        \"max\": 409.0,\n
```

\"num_unique_values\": 4,\n   \"samples\": [\n   0.0,\n
-1550886728.0,\n   15.0\n   ],\n   \"semantic_type\":
\"\",\n   \"description\": \"\"\n   }\n   },\n   {\n
\"column\": \"25%\",\n   \"properties\": {\n   \"dtype\":
\"date\",\n   \"min\": 1.0,\n   \"max\": 1585443.75,\n
\"num_unique_values\": 8,\n   \"samples\": [\n   1165.75,\
n   2396.5,\n   9786.5\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"50%\",\n   \"properties\": {\n
\"dtype\": \"date\",\n   \"min\": 5.0,\n   \"max\":
7657846.0,\n   \"num_unique_values\": 8,\n   \"samples\": [\
n   3558.5,\n   48137.5,\n   18498.0\n   ],\
n   \"semantic_type\": \"\",\n   \"description\": \"\"\n
}\n   },\n   {\n   \"column\": \"75%\",\n   \"properties\": {\
n   \"dtype\": \"date\",\n   \"min\": 15.0,\n
\"max\": 147188892.0,\n   \"num_unique_values\": 8,\n
\"samples\": [\n   11468.5,\n   1015849.75,\n
42324.0\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"max\",\n   \"properties\": {\n   \"dtype\": \"date\",\n
\"min\": 60546.0,\n   \"max\": 12800740251.0,\n
\"num_unique_values\": 8,\n   \"samples\": [\n
4776812.0,\n   401452781.0,\n   32230163.0\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   }\n]\n}","type":"dataframe"}

```
data.head(10)
```

{"summary":"{\n \"name\": \"data\",\n \"rows\": 3299,\n \"fields\":
[\n   {\n   \"column\": \"web-scraper-order\",\n
\"properties\": {\n   \"dtype\": \"string\",\n
\"num_unique_values\": 3299,\n   \"samples\": [\n
\"1708356967-2130\",\n   \"1708351151-1279\",\n
\"1708363186-3154\"\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"Name\",\n   \"properties\": {\n   \"dtype\": \"string\",\n
\"num_unique_values\": 3292,\n   \"samples\": [\n
\"Dietitian Sumaiyya Shoaib\",\n   \"DJ Vispi\",\n
\"Sachin Kamble\"\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"Image-src\",\n   \"properties\": {\n   \"dtype\":
\"string\",\n   \"num_unique_values\": 2987,\n
\"samples\": [\n
\"https://d24w28i6lzk071.cloudfront.net/assets/profiles/instagram/
1507016449.jpg\",\n
\"https://d24w28i6lzk071.cloudfront.net/assets/profiles/instagram/
2261574578.jpg\",\n
\"https://d24w28i6lzk071.cloudfront.net/assets/profiles/instagram/
3441966902.jpg\"\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":

```
\"Category\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 23,\n
\"samples\": [\n          \"Science & Technology\",\n
\"Films\",\n          \"Defence\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Ig_Follow\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
799761.944987238,\n        \"min\": 15.0,\n        \"max\":
32230163.0,\n        \"num_unique_values\": 2312,\n
\"samples\": [\n          49017.0,\n          8980.0,\n
17720.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"eng_rate\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 798,\n
\"samples\": [\n          \"18.32%\",\n          \"4.57%\",\n
\"3.39%\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"est_Reel_play\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 261664.90367386656,\n        \"min\":
0.0,\n        \"max\": 4776812.0,\n        \"num_unique_values\":
2253,\n        \"samples\": [\n          4634.0,\n          145438.0,\
n          27134.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Avg_like\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 17632.847423515133,\n        \"min\":
0.0,\n        \"max\": 413120.0,\n        \"num_unique_values\":
1055,\n        \"samples\": [\n          580.0,\n          931.0,\n
817.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Avg_Comments\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 1203.7465185769481,\n        \"min\":
0.0,\n        \"max\": 60546.0,\n        \"num_unique_values\": 170,\n
\"samples\": [\n          149.0,\n          18.0,\n          935.0\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"yt_subs\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
2710527.3595734197,\n        \"min\": 409.0,\n        \"max\":
23400000.0,\n        \"num_unique_values\": 722,\n        \"samples\":
[\n          3030000.0,\n          48800.0,\n          272000.0\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"views(30 days)\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
52437860.40603506,\n        \"min\": -1550886728.0,\n        \"max\":
401452781.0,\n        \"num_unique_values\": 1229,\n
\"samples\": [\n          1818.0,\n          61641045.0,\n
368.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"total_views\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 837403120.6754007,\n        \"min\":
```

```
0.0,\n        \"max\": 12800740251.0,\n        \"num_unique_values\":
1293,\n        \"samples\": [\n          4335939637.0,\n
2058613.0,\n          420088.0\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"uplaods\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 2376.048567133539,\n        \"min\":
0.0,\n        \"max\": 74336.0,\n        \"num_unique_values\": 782,\n
\"samples\": [\n          7240.0,\n          864.0,\n          99.0\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"yt_links-href\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 1305,\n        \"samples\": [\n
\"https://www.youtube.com/channel/UCNUibbzRuV8gyRmeYguuKvQ\",\n
\"https://www.youtube.com/channel/UCAw6biCii_E94umEzVVzTTQ\",\n
\"https://www.youtube.com/channel/UCVGckd9Wz9y2EN-xWa5emRA\"\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"insta_link-href\",\n      \"properties\": {\n        \"dtype\":
\"string\",\n        \"num_unique_values\": 3296,\n
\"samples\": [\n
\"https://www.instagram.com/therealsumaiya_shoaib\",\n
\"https://www.instagram.com/djvispi\",\n
\"https://www.instagram.com/namanchhabra_\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    }\n  ]\n}","type":"dataframe","variable_name":"data"}
```


```
print(data.Category.unique())
print(data.Category.nunique())
```

```
['Defence' 'IT & ITES' 'Supernatural' 'Blogs and Travel' 'Movie &
Shows'
 'Real Estate' 'Finance' 'Health & Fitness' 'Vloging' 'Films' 'Gaming'
 'Travel & Leisure' 'Sports' 'Education' 'Autos & Vehicles'
 'Science & Technology' 'Fashion & Style' 'Beauty' 'Food & Drinks'
'Music'
 'Comedy' 'Reviews' 'Entertainment']
23
```

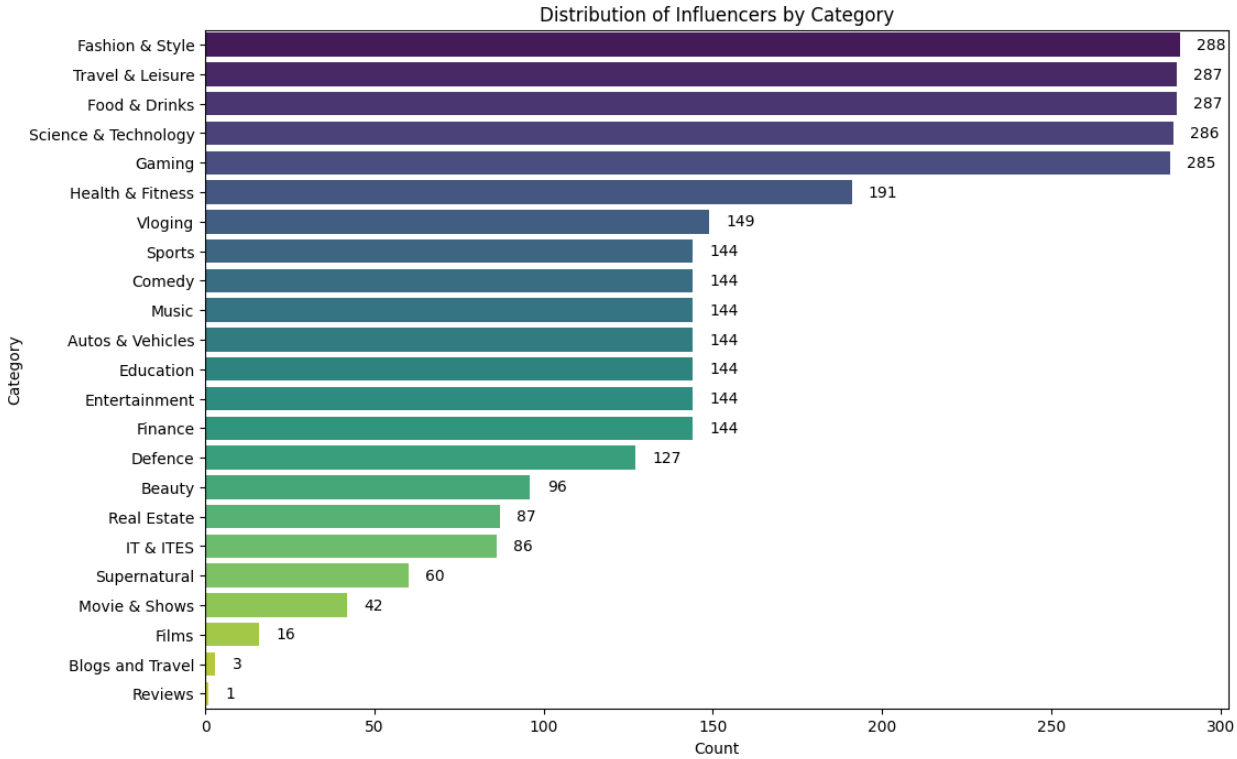
Key insights from the provided code snippets:

- **The dataset contains 3299 entries and 20 columns initially.**
- The **Category** column has 23 unique categories, representing different niches or industries the influencers belong to.
- Initial data cleaning involved dropping irrelevant columns and handling missing values.
- The **Image-src** column has some missing values (around 9.37%).

- Columns related to YouTube links and Instagram links seem to have no useful data.
- **After data cleaning, the dataset contains 3299 entries and 15 columns.**

Data Exploration:

```
category_counts = data['Category'].value_counts()

# Sort categories by frequency
sorted_categories = category_counts.index

plt.figure(figsize=(12, 8))
sns.countplot(y='Category', data=data, order=sorted_categories,
palette='viridis')

# Adding data labels
for i, category in enumerate(sorted_categories):
    count = category_counts[category]
    plt.text(count + 5, i, f'{count:}', va='center')

plt.title('Distribution of Influencers by Category')
plt.xlabel('Count')
plt.ylabel('Category')
plt.show()
```



```
# Define a mapping dictionary to merge similar categories
category_mapping = {
    'IT & ITES': 'Science & Technology',
    'Supernatural': 'Science & Technology',
    'Blogs and Travel': 'Vloging',
    'Reviews': 'Vloging',
    'Films':'Movie & Shows',
    'Real Estate':'Finance'
}

# Replace old categories with new ones using the mapping dictionary
data['Category'] = data['Category'].replace(category_mapping)

# Check the unique values in the 'Category' column after merging
print(data['Category'].unique())
print(data['Category'].nunique())
```

```
['Defence' 'Science & Technology' 'Vloging' 'Movie & Shows' 'Finance'
 'Health & Fitness' 'Gaming' 'Travel & Leisure' 'Sports' 'Education'
 'Autos & Vehicles' 'Fashion & Style' 'Beauty' 'Food & Drinks' 'Music'
 'Comedy' 'Entertainment']
17
```

After merging the specified categories, we have simplified the dataset by grouping similar themes together. Here are some insights based on the merged categories:

1. **Science & Technology:** The merging of *'IT & ITES'* and *'Supernatural'* into *'Science & Technology'* suggests a focus on technological and scientific content. This category likely encompasses topics related to information technology, IT-enabled services, and supernatural or futuristic subjects.
2. **Vlogging:** The merging of *'Blogs and Travel'* and *'Reviews'* into *'Vlogging'* indicates a shift towards content creation and user-generated reviews. This category may include blogs, travel-related content, and reviews of various products or services.
3. **Movie & Shows:** The merging of *'Films'* into *'Movie & Shows'* suggests a consolidation of content related to movies, television shows, and other visual entertainment media. This category likely covers a broad range of cinematic and episodic content.
4. **Finance:** The merging of *'Real Estate'* into *'Finance'* hints at a broader focus on financial topics. This category may include discussions about real estate investments, property markets, and other financial aspects related to real estate.

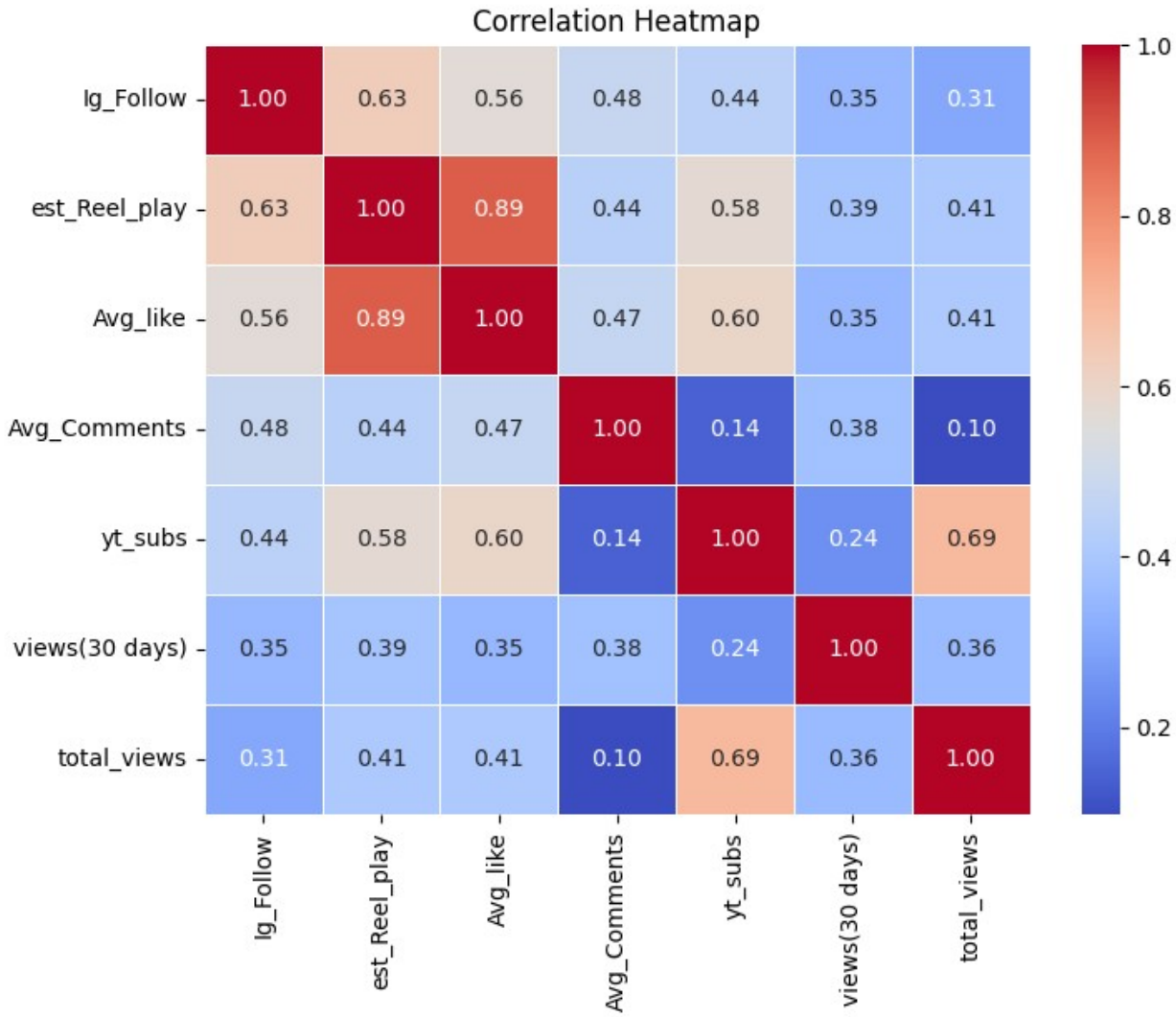
Overall, these mergers have streamlined the dataset and grouped similar categories together, making it easier to analyze and interpret. The insights derived from these merged categories can help in understanding the overarching themes present in the dataset and identifying trends or patterns related to these themes.

```
cat_cols=data.select_dtypes(include=['object']).columns
num_cols = data.select_dtypes(include=np.number).columns.tolist()
print("Categorical Variables:")
print(cat_cols)
```

```
print("Numerical Variables:")
print(num_cols)
```

```
Categorical Variables:
Index(['web-scraper-order', 'Name', 'Image-src', 'Category',
'eng_rate',
       'yt_links-href', 'insta_link-href'],
      dtype='object')
Numerical Variables:
['Ig_Follow', 'est_Reel_play', 'Avg_like', 'Avg_Comments', 'yt_subs',
'views(30 days)', 'total_views', 'uplaods']
```

```
# Compute the correlation matrix
num_colss = ['Ig_Follow', 'est_Reel_play', 'Avg_like', 'Avg_Comments',
'yt_subs', 'views(30 days)', 'total_views']
corr_matrix = data[num_colss].corr()

plt.figure(figsize=(8,6))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', fmt=".2f",
linewidths=0.5)
plt.title('Correlation Heatmap')
plt.show()
```



Insights from Correlation Heatmap

Strong Positive Correlation:

1. *Est. Reel Play* and *Avg. Like*: 0.89
   - This correlation indicates a very strong positive relationship between the estimated reel play count and the average number of likes received. A coefficient of *0.89* suggests that as the estimated reel play count increases, the average number of likes also tends to increase significantly.

Moderate Positive Correlation:

1. *YT Subs* and *Total Views*: 0.69
   - This correlation demonstrates a moderate positive relationship between YouTube subscribers and total views. A coefficient of *0.69* indicates that as total number of views increases, the number of YouTube subscribers tends to increase moderately.
2. *IG Follow* and *Est. Reel Play*: 0.63

- This correlation signifies a moderate positive relationship between Instagram followers and estimated reel plays. A coefficient of *0.63* suggests that as the estimated reel play count increase, the number of Instagram followers also tends to grow moderately.

Positive Correlation:

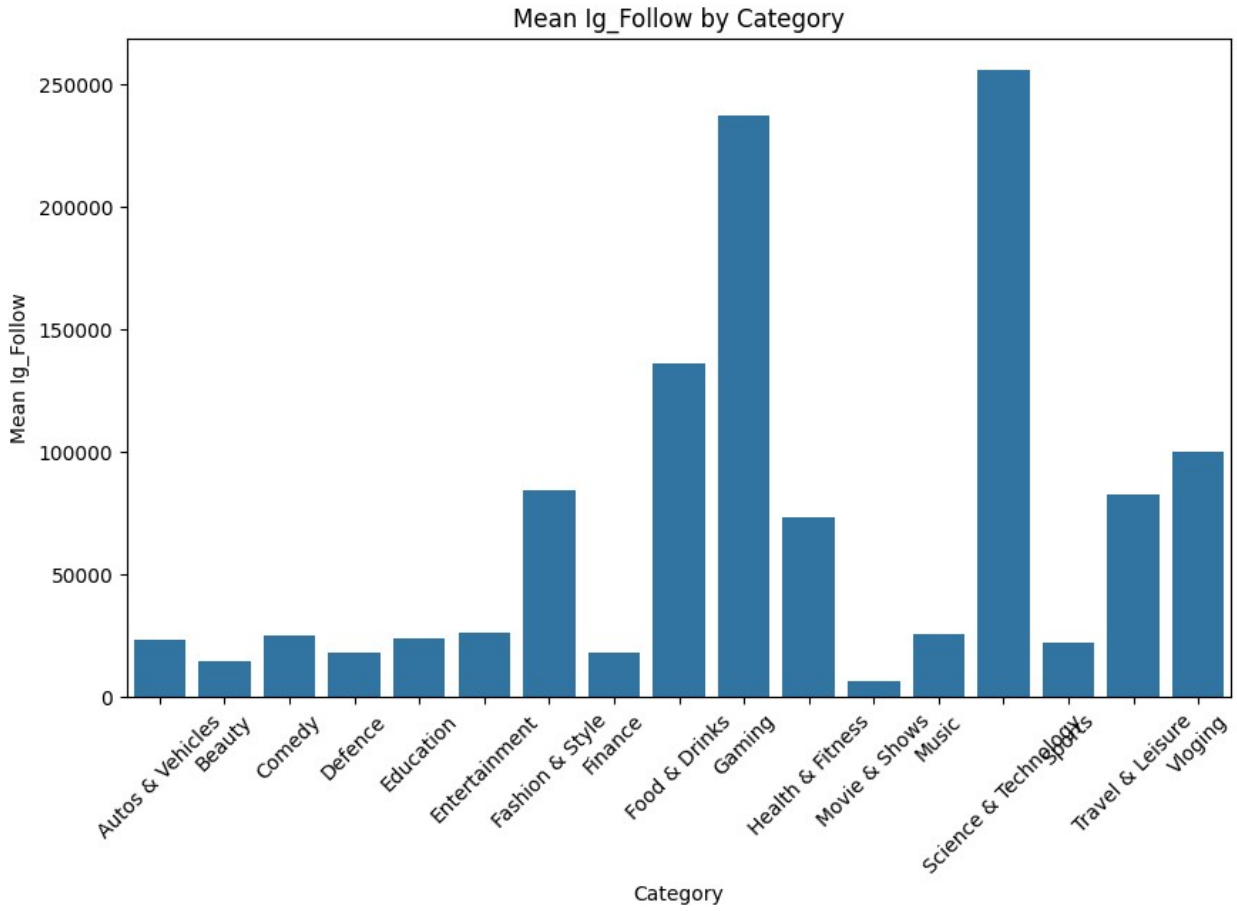
1. *IG Follow* and *Avg. Comments*: 0.48
   - This signifies a positive relationship between Instagram followers and the average number of comments. With a coefficient of *0.48*, it suggests a tendency for an increase in Instagram followers to be associated with a slight increase in average comments.
2. *Avg. Comments* and *Avg. Like*: 0.47
   - This also indicates a positive relationship between the average number of comments and the average number of likes. A coefficient of *0.47* suggests a tendency for an increase in average comments to be associated with a slight increase in average likes.

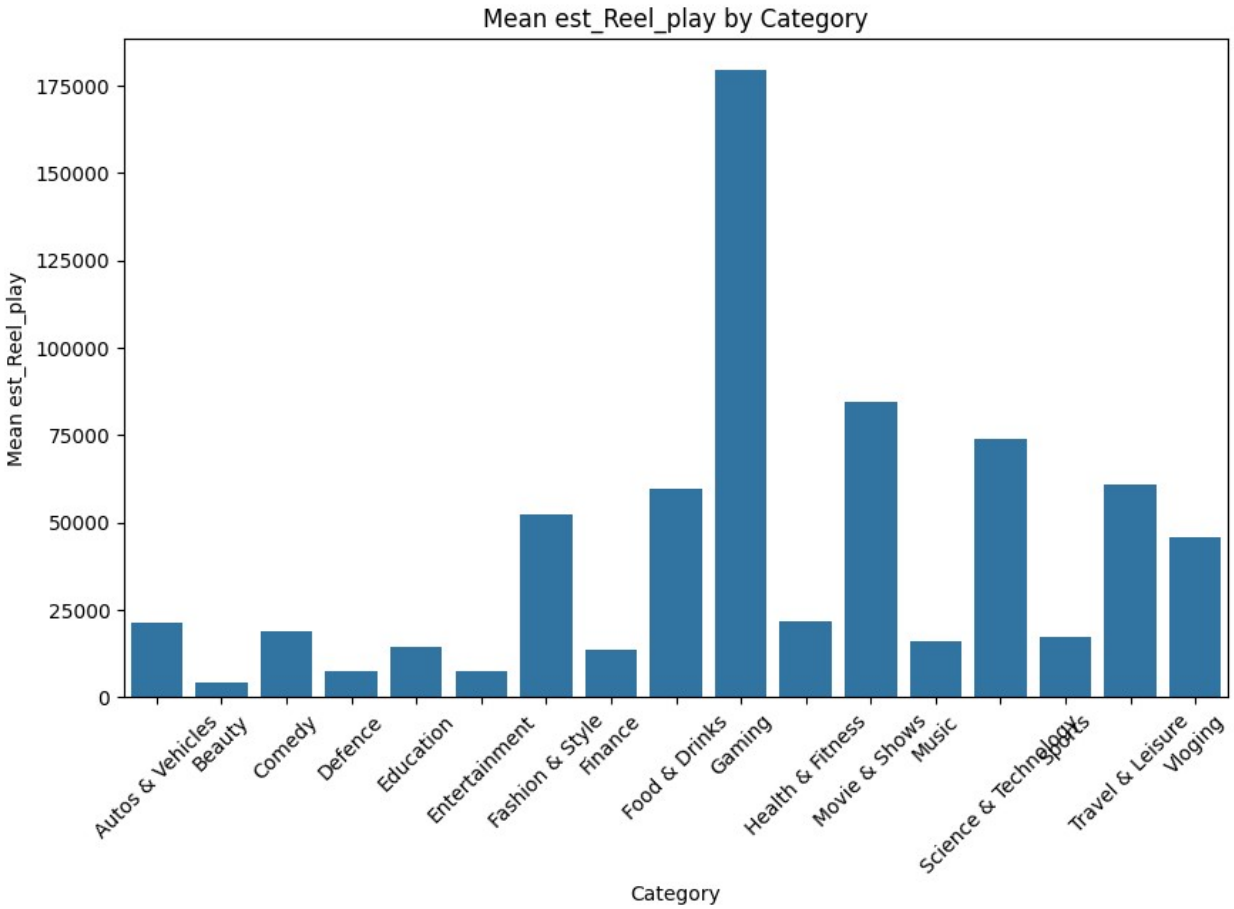
Weak Positive Correlation:

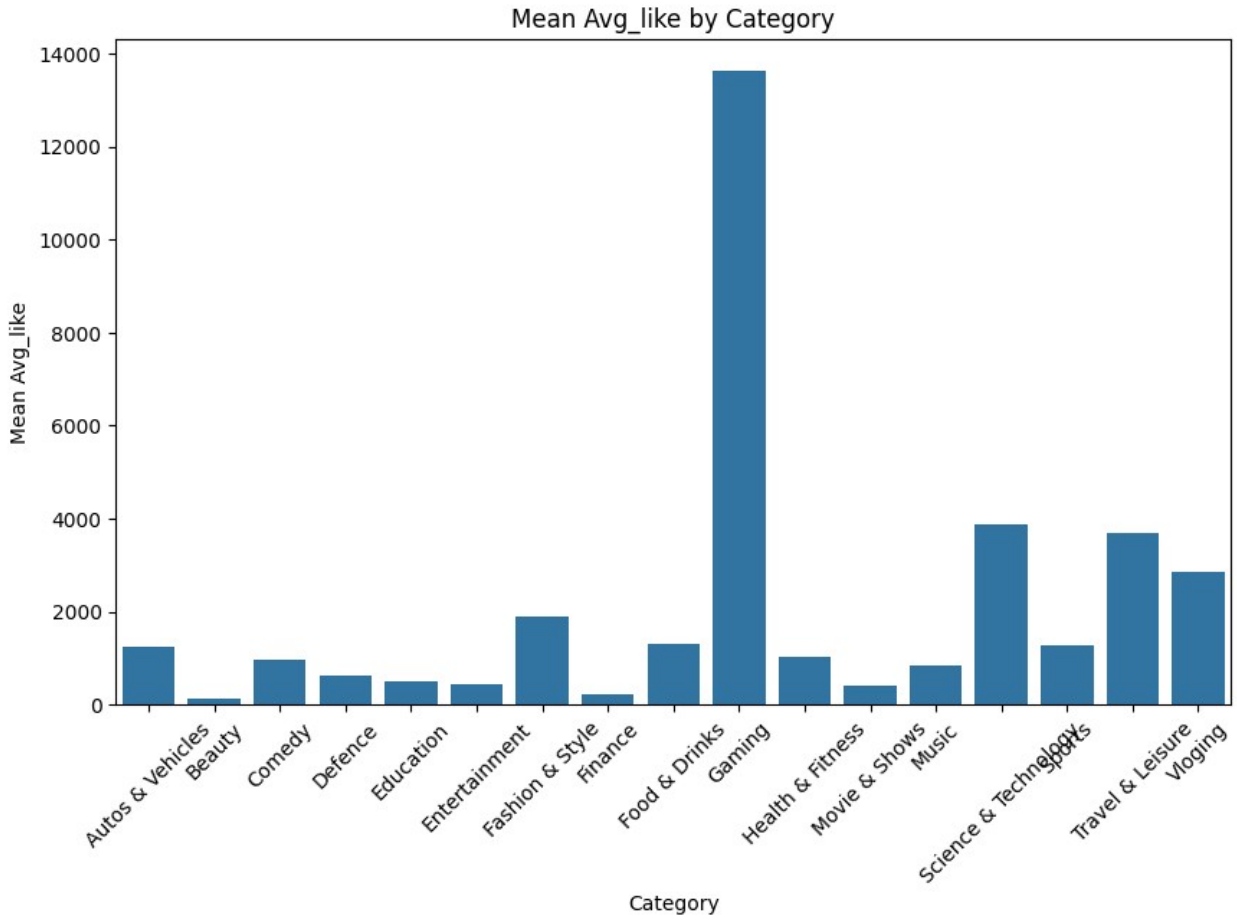
1. *Views (30 days)* and *Total Views*: 0.36
   - This correlation suggests a weak positive relationship between views in the last 30 days and total views. A coefficient of *0.36* indicates a slight tendency for total views to increase as views in the last 30 days increase.
2. *YT Subs* and *Views (30 days)*: 0.24
   - This correlation indicates a weak positive relationship between YouTube subscribers and views in the last 30 days. A coefficient of *0.24* suggests a minor tendency for views in the last 30 days to increase as the number of YouTube subscribers increases.
3. *YT Subs* and *Avg. Comments*: 0.14
   - This correlation reveals a very weak positive relationship between YouTube subscribers and the average number of comments. A coefficient of *0.14* suggests a minimal tendency for the average number of comments to increase as the number of YouTube subscribers increases.

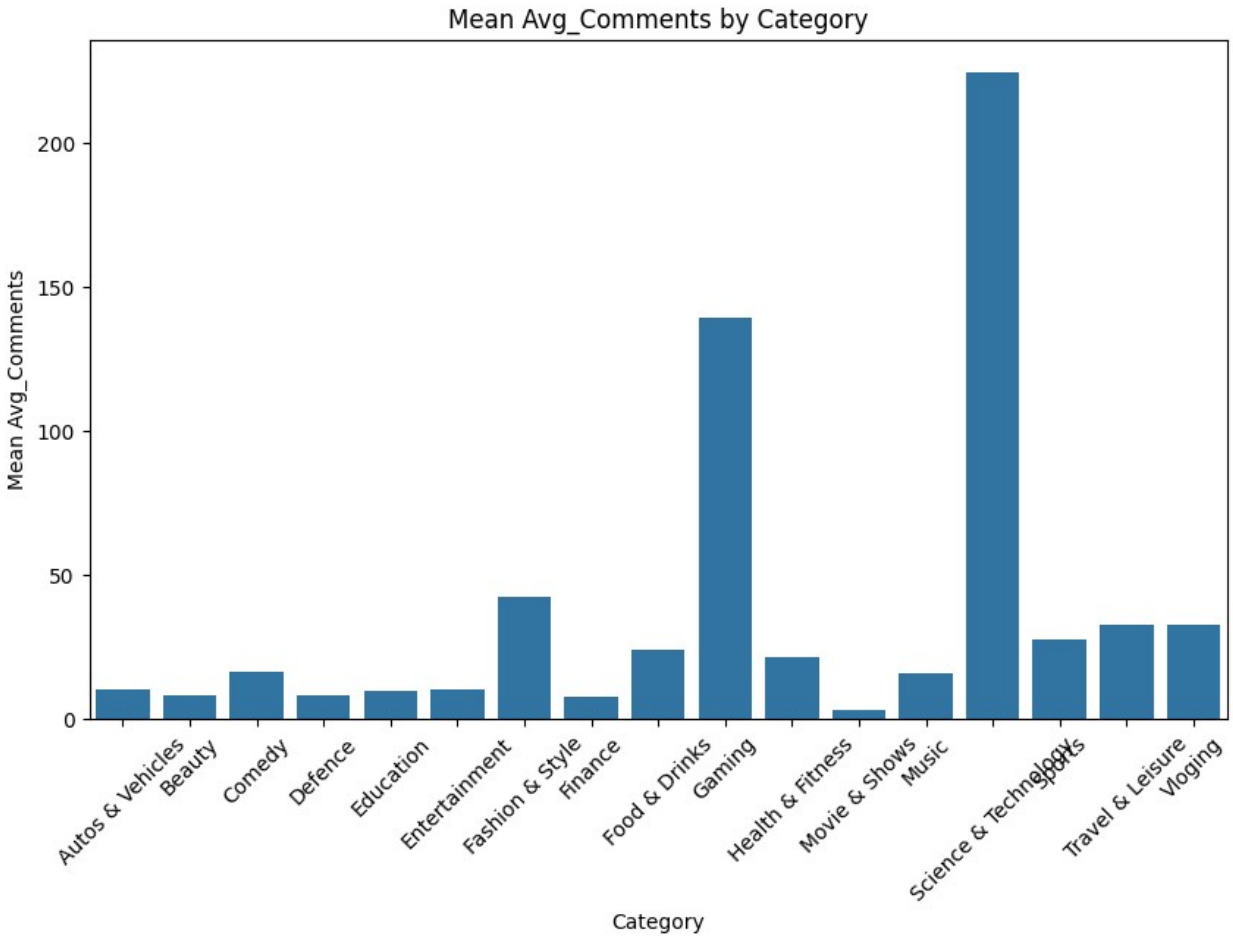
```
numerical_columns = ['Ig_Follow', 'est_Reel_play', 'Avg_like',
'Avg_Comments', 'yt_subs', 'views(30 days)', 'total_views', 'uplaods']
category_column = 'Category'

# Calculate the mean value for each numerical column grouped by
category
mean_values_by_category = data.groupby(category_column)
[numerical_columns].mean()

# Plot individual graphs for each numerical column
for numerical_column in numerical_columns:
    plt.figure(figsize=(10, 6))
    sns.barplot(x=mean_values_by_category.index,
y=mean_values_by_category[numerical_column])
```

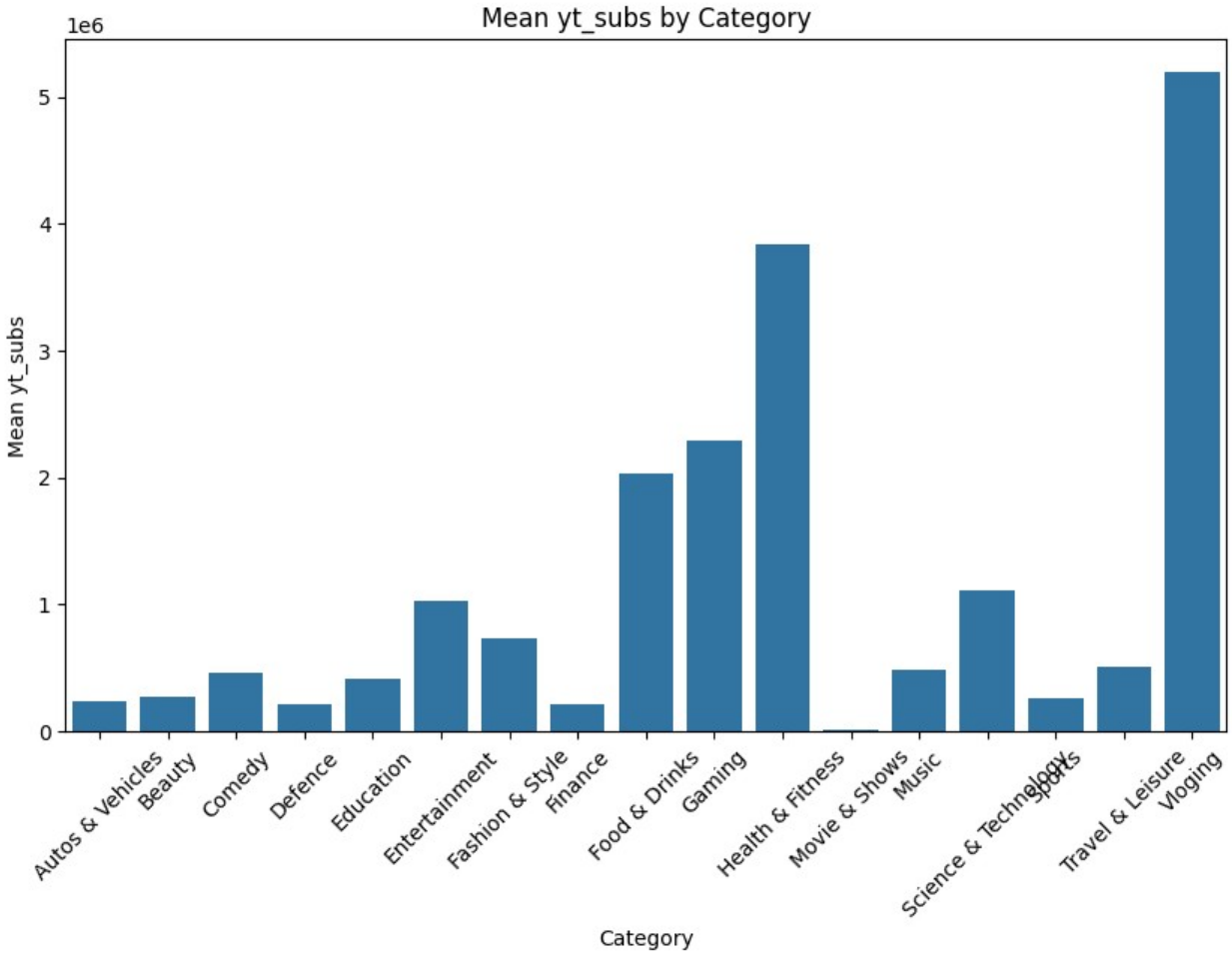
```
    plt.title(f'Mean {numerical_column} by {category_column}')
    plt.xlabel(category_column)
    plt.ylabel(f'Mean {numerical_column}')
    plt.xticks(rotation=45)
    plt.show()
```

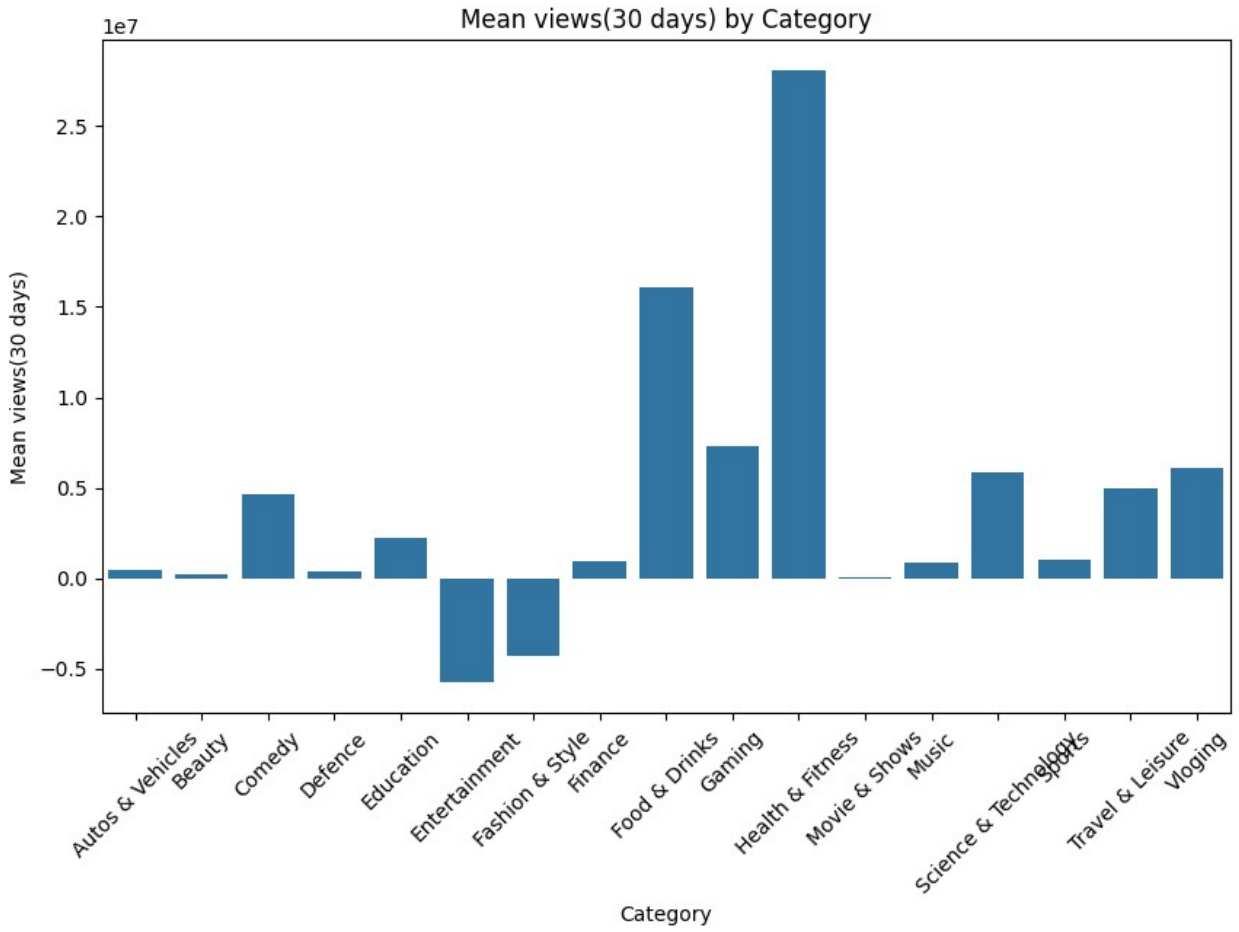


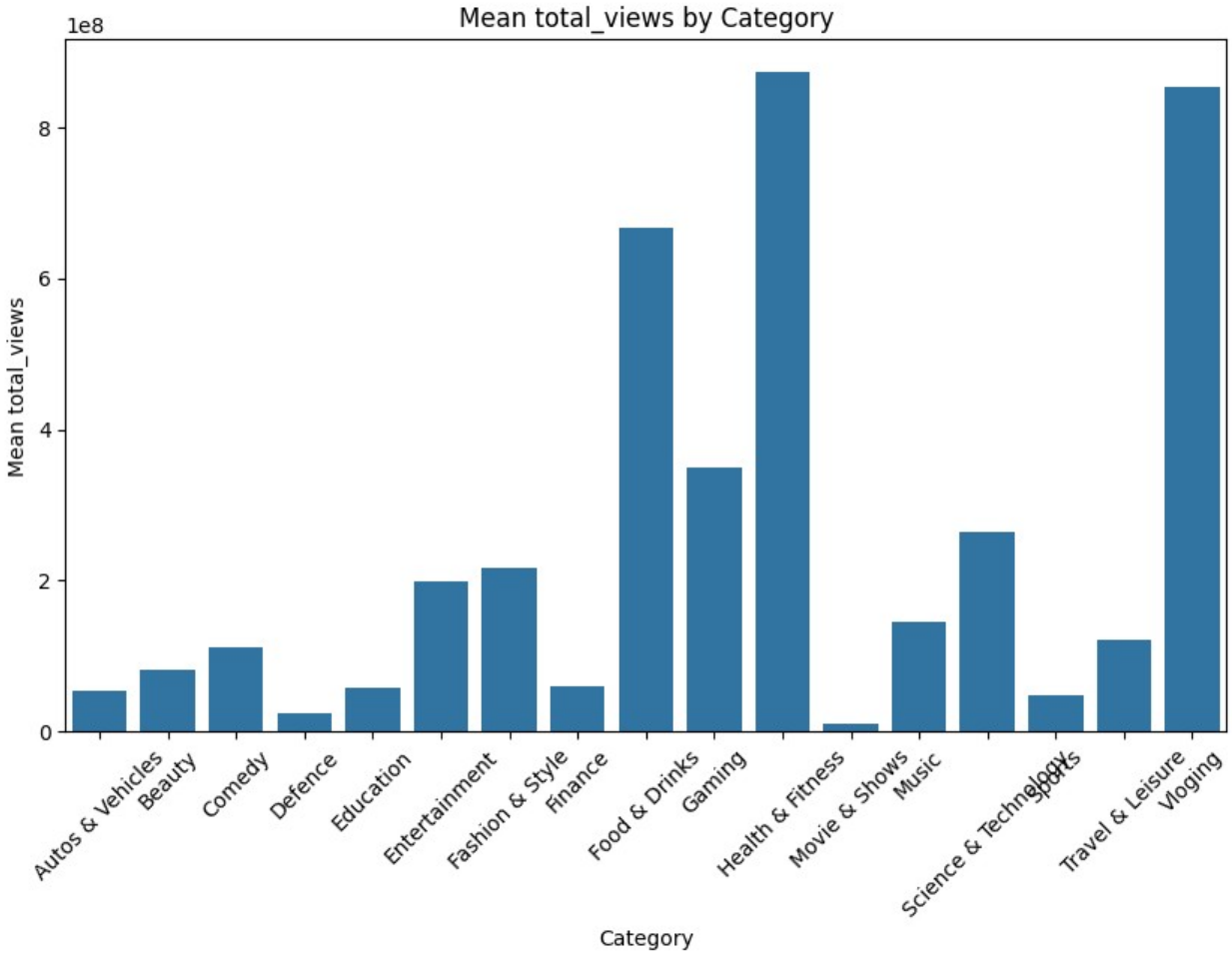

Mean est_Reel_play by Category
Mean est_Reel_play
175000
150000
125000
100000
75000
50000
25000
0
Autos & Vehicles
Beauty
Comedy
Defence
Education
Entertainment
Fashion & Style
Finance
Food & Drinks
Gaming
Health & Fitness
Movie & Shows
Music
Science & Technology
Sports
Travel & Leisure
Vloging
Category

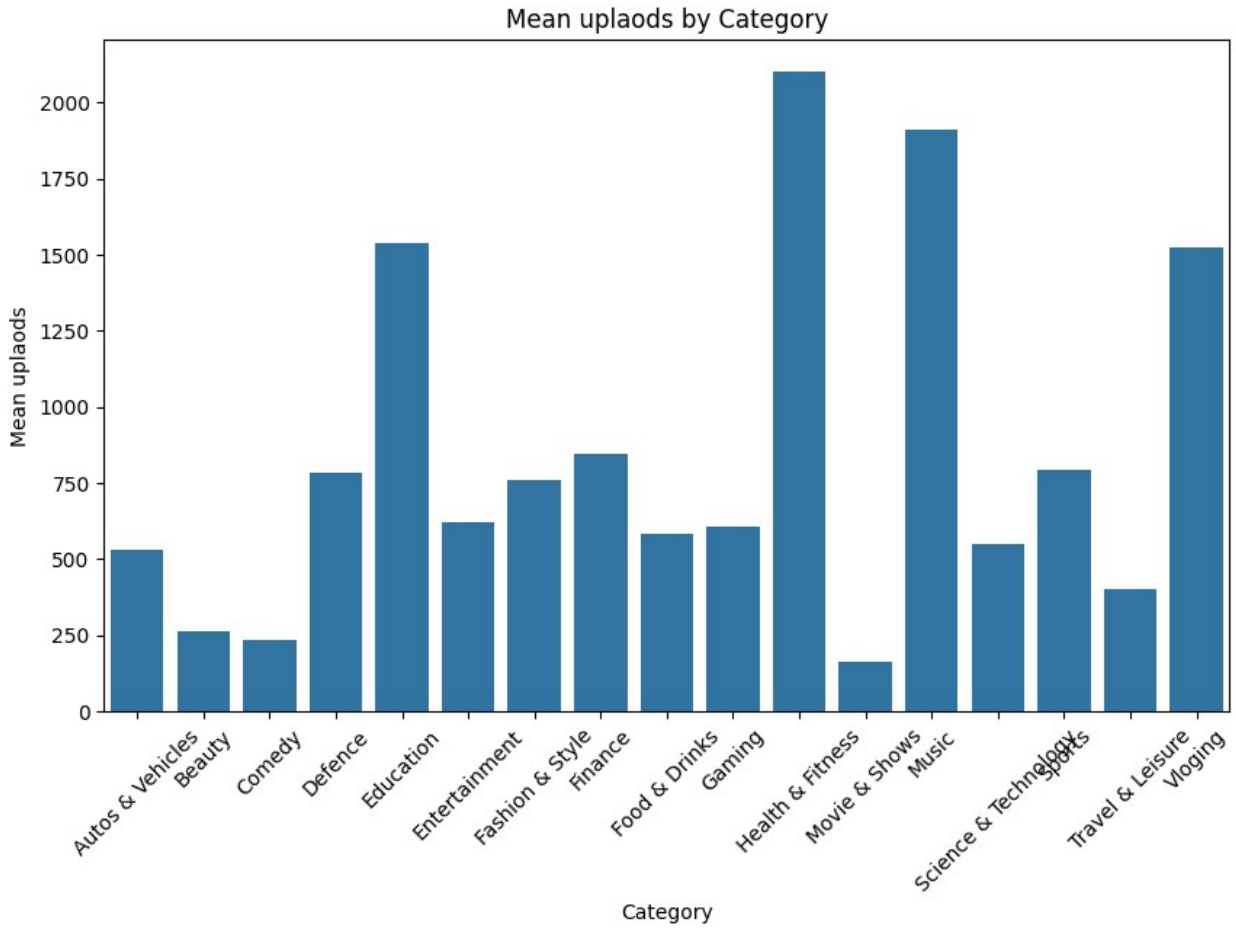

Mean Avg_like by Category
Mean Avg_like
14000
12000
10000
8000
6000
4000
2000
0
Autos & Vehicles
Beauty
Comedy
Defence
Education
Entertainment
Fashion & Style
Finance
Food & Drinks
Gaming
Health & Fitness
Movie & Shows
Music
Science & Technology
Sports
Travel & Leisure
Vloging
Category


Mean Avg_Comments by Category
Mean Avg_Comments
0
50
100
150
200
Autos & Vehicles
Beauty
Comedy
Defence
Education
Entertainment
Fashion & Style
Finance
Food & Drinks
Gaming
Health & Fitness
Movie & Shows
Music
Science & Technology
Sports
Travel & Leisure
Vloging
Category


Mean yt_subs by Category
1e6
Mean yt_subs
0
1
2
3
4
5
Autos & Vehicles
Beauty
Comedy
Defence
Education
Entertainment
Fashion & Style
Finance
Food & Drinks
Gaming
Health & Fitness
Movie & Shows
Music
Science & Technology
Sports
Travel & Leisure
Vloging
Category


Mean views(30 days) by Category
1e7
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
Mean views(30 days)
Autos & Vehicles
Beauty
Comedy
Defence
Education
Entertainment
Fashion & Style
Finance
Food & Drinks
Gaming
Health & Fitness
Movie & Shows
Music
Science & Technology
Sports
Travel & Leisure
Vloging
Category


Mean total_views by Category
1e8
Mean total_views
0
2
4
6
8
Autos & Vehicles
Beauty
Comedy
Defence
Education
Entertainment
Fashion & Style
Finance
Food & Drinks
Gaming
Health & Fitness
Movie & Shows
Music
Science & Technology
Sports
Travel & Leisure
Vloging
Category




```
output_file_path = "preprocessed_data.csv"
data.to_csv(output_file_path, index=False)

print("Preprocessed data has been saved successfully.")
```

```
Preprocessed data has been saved successfully.
```