
EDA for YouTube Data

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import os
```

```
channel_master = pd.read_csv("/content/Channel_Master.csv")
video_summary = pd.read_csv("/content/Language_Master.csv")
revenue_master = pd.read_csv("/content/Revenue_Master.csv")
language_master = pd.read_csv("/content/Revenue_Master.csv")
```

```
# -----------------------------
# Create output folders for plots/tables
# -----------------------------
os.makedirs("eda_outputs/plots", exist_ok=True)
os.makedirs("eda_outputs/tables", exist_ok=True)
```

```
# -----------------------------
# 1. Basic Info
# -----------------------------
datasets = {
    "channel_master": channel_master,
    "video_summary": video_summary,
    "revenue_master": revenue_master,
    "language_master": language_master
}

for name, df in datasets.items():
    print(f"--- {name} ---")
    print(df.shape)
    print(df.info())
    print(df.describe(include="all"))
    print("\n\n")
```

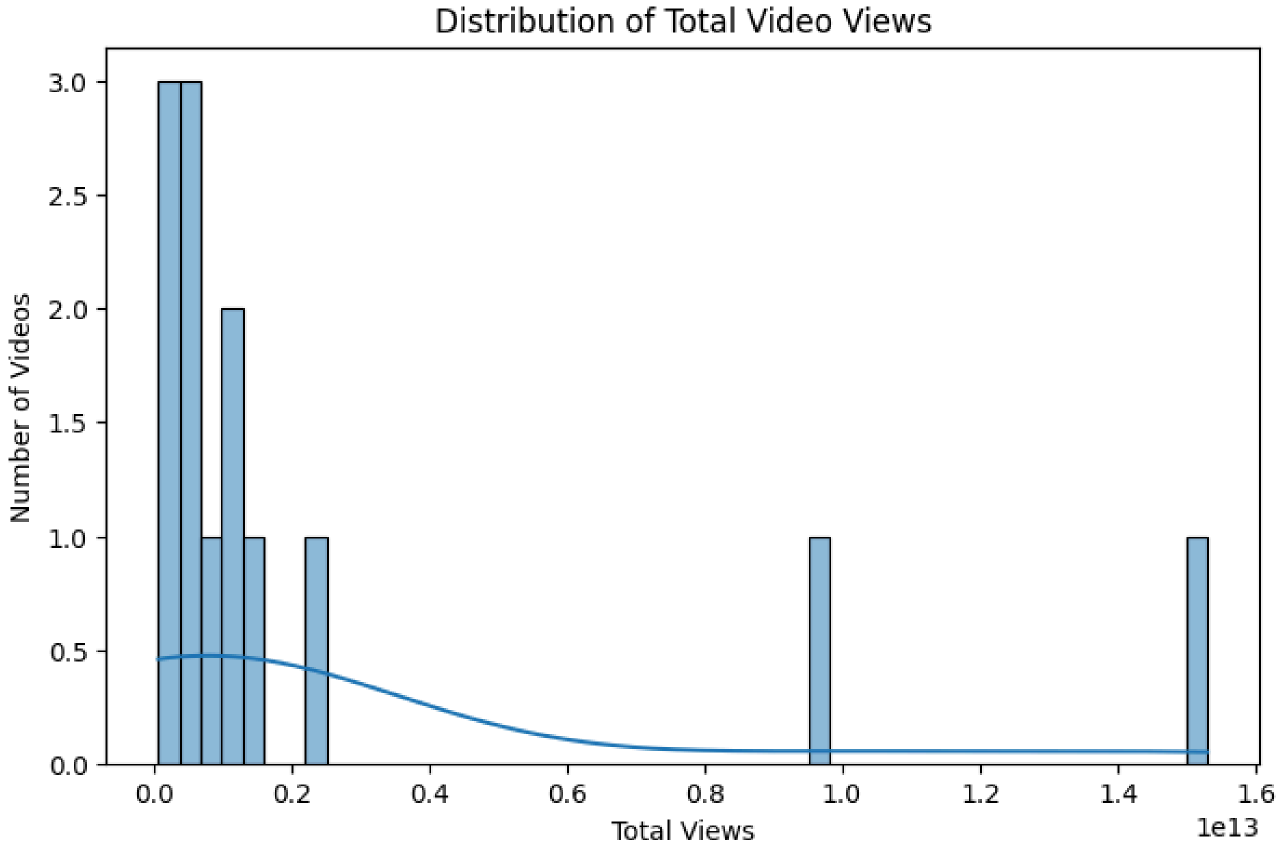
```
25%                 8.314000e+02  8.314000e+03      1.000000
50%                 2.269360e+04  2.269360e+05      1.000000
75%                 2.087746e+05  2.087746e+06      1.000000
max                 4.963990e+09  4.963990e+10    222.000000
```

```
--- language_master ---
(18261, 6)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 18261 entries, 0 to 18260
Data columns (total 6 columns):
 #   Column                   Non-Null Count  Dtype  
---  ------                   --------------  -----  
 0   Channelid                18261 non-null  object 
 1   subscribercount          18196 non-null  float64
 2   channelname              18186 non-null  object 
 3   total_estimated_revenue  18261 non-null  float64
 4   total_views              18261 non-null  int64  
 5   video_count              18261 non-null  int64  
dtypes: float64(2), int64(2), object(2)
memory usage: 856.1+ KB
None
                       Channelid  subscribercount               channelname  \
count                      18261     1.819600e+04                     18186   
unique                     18261              NaN                     18186   
top     UCz5VUqEp7ysXn4Z2FhMrkhQ              NaN  Aami Pohu Aaha by Jharna   
freq                           1              NaN                         1   
mean                         NaN     8.964333e+05                       NaN   
std                          NaN     6.180050e+06                       NaN   
min                          NaN     2.000000e+00                       NaN   
25%                          NaN     2.520000e+03                       NaN   
50%                          NaN     2.590000e+04                       NaN   
75%                          NaN     2.230000e+05                       NaN   
max                          NaN     4.240000e+08                       NaN   

        total_estimated_revenue   total_views   video_count  
count              1.826100e+04  1.826100e+04  18261.000000  
unique                      NaN           NaN           NaN  
top                         NaN           NaN           NaN  
freq                        NaN           NaN           NaN  
mean               2.320522e+06  2.320522e+07      1.903182  
std                4.475782e+07  4.475782e+08      4.838899  
min                0.000000e+00  0.000000e+00      1.000000  
25%                8.314000e+02  8.314000e+03      1.000000  
50%                2.269360e+04  2.269360e+05      1.000000  
75%                2.087746e+05  2.087746e+06      1.000000  
max                4.963990e+09  4.963990e+10    222.000000  
```

Video Summary Analysis

```
# Distribution of TotalViews
plt.figure(figsize=(8,5))
sns.histplot(video_summary["TotalViews"].fillna(0), bins=50, kde=True)
plt.title("Distribution of Total Video Views")
plt.xlabel("Total Views")
plt.ylabel("Number of Videos")
plt.savefig("eda_outputs/plots/totalviews_distribution.png")
plt.show()
```



Business insight: Most videos have moderate views; few viral videos contribute to a large audience reach.

Channel Revenue Analysis

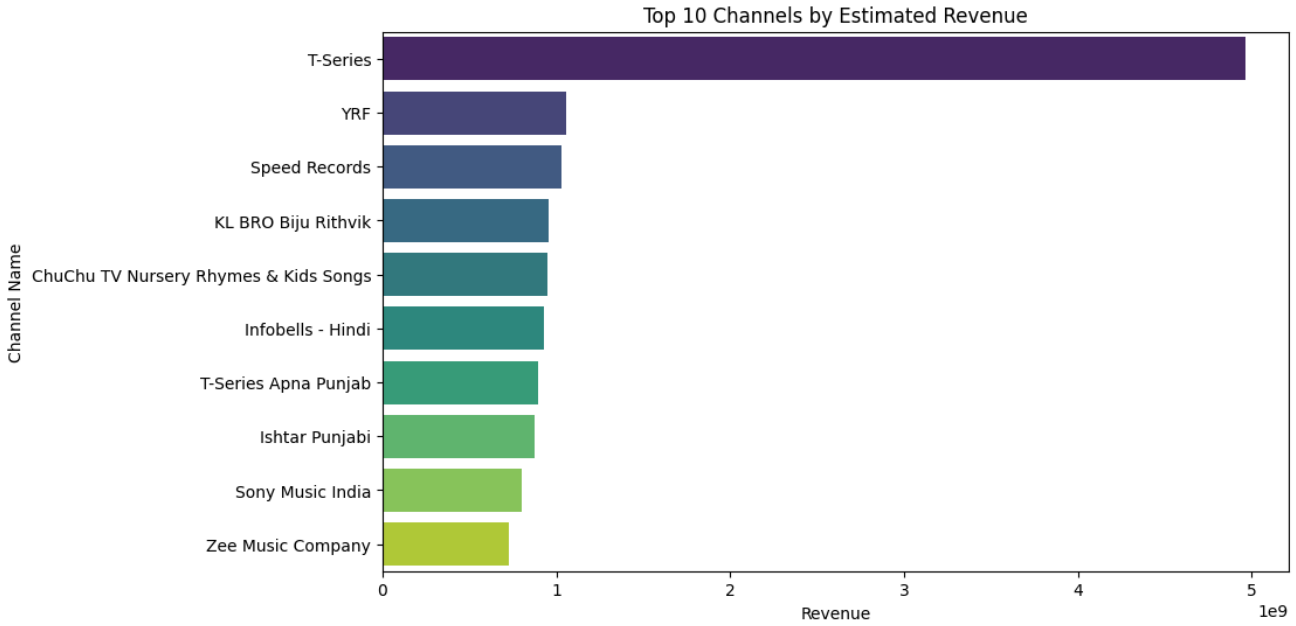
```
# Top 10 channels by estimated revenue
top_channels = revenue_master.sort_values("total_estimated_revenue", ascending=False).hea
top_channels.to_csv("eda_outputs/tables/top_10_channels_by_revenue.csv", index=False)

plt.figure(figsize=(10,6))
sns.barplot(x="total_estimated_revenue", y="channelname", data=top_channels, palette="vir
plt.title("Top 10 Channels by Estimated Revenue")
plt.xlabel("Revenue")
plt.ylabel("Channel Name")
plt.savefig("eda_outputs/plots/top_channels_revenue.png")
plt.show()
```

```
/tmp/ipython-input-278476792.py:6: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.

  sns.barplot(x="total_estimated_revenue", y="channelname", data=top_channels, palett
```



Business insight: Helps prioritize high-revenue channels for marketing or partnership focus.

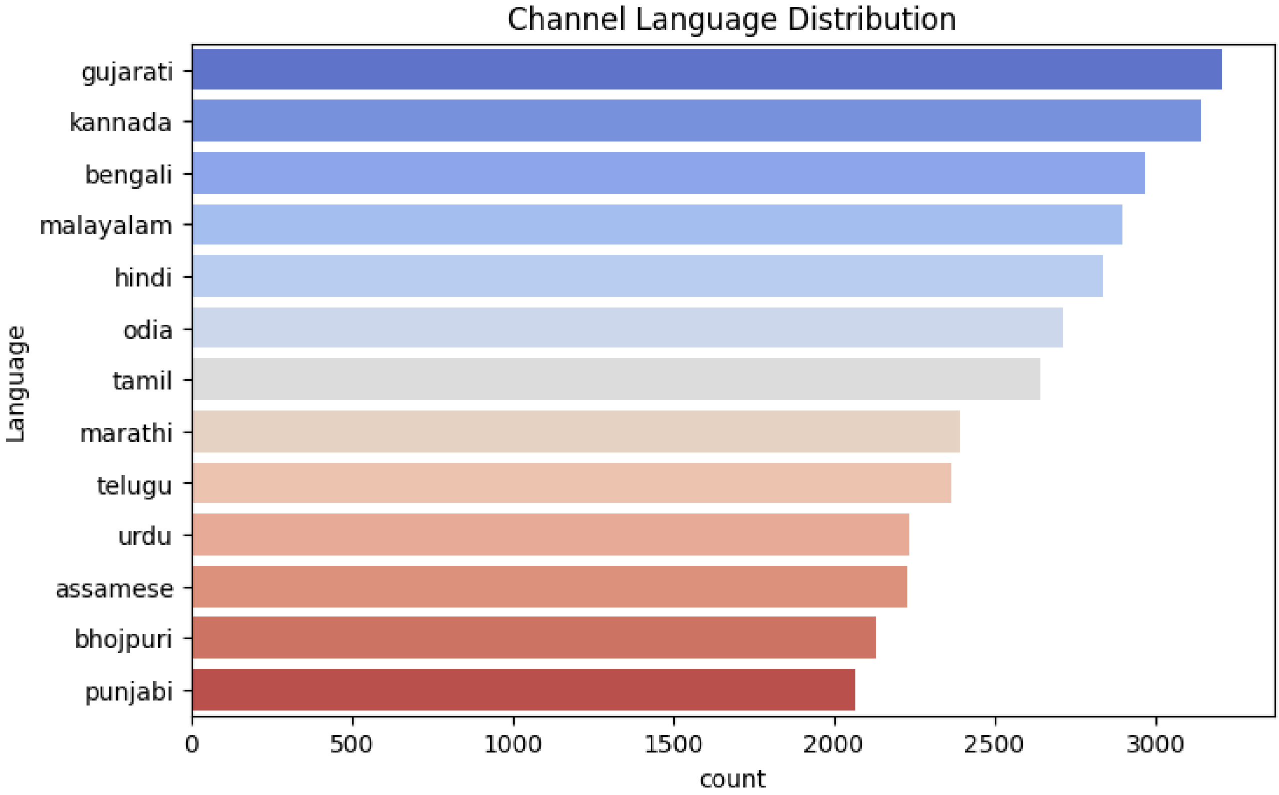
Language Distribution

```
plt.figure(figsize=(8,5))
sns.countplot(y="Language", data=channel_master, order=channel_master['Language'].value_c
plt.title("Channel Language Distribution")
plt.savefig("eda_outputs/plots/language_distribution.png")
plt.show()
```

```
/tmp/ipython-input-2030453952.py:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.

  sns.countplot(y="Language", data=channel_master, order=channel_master['Language'].v
```

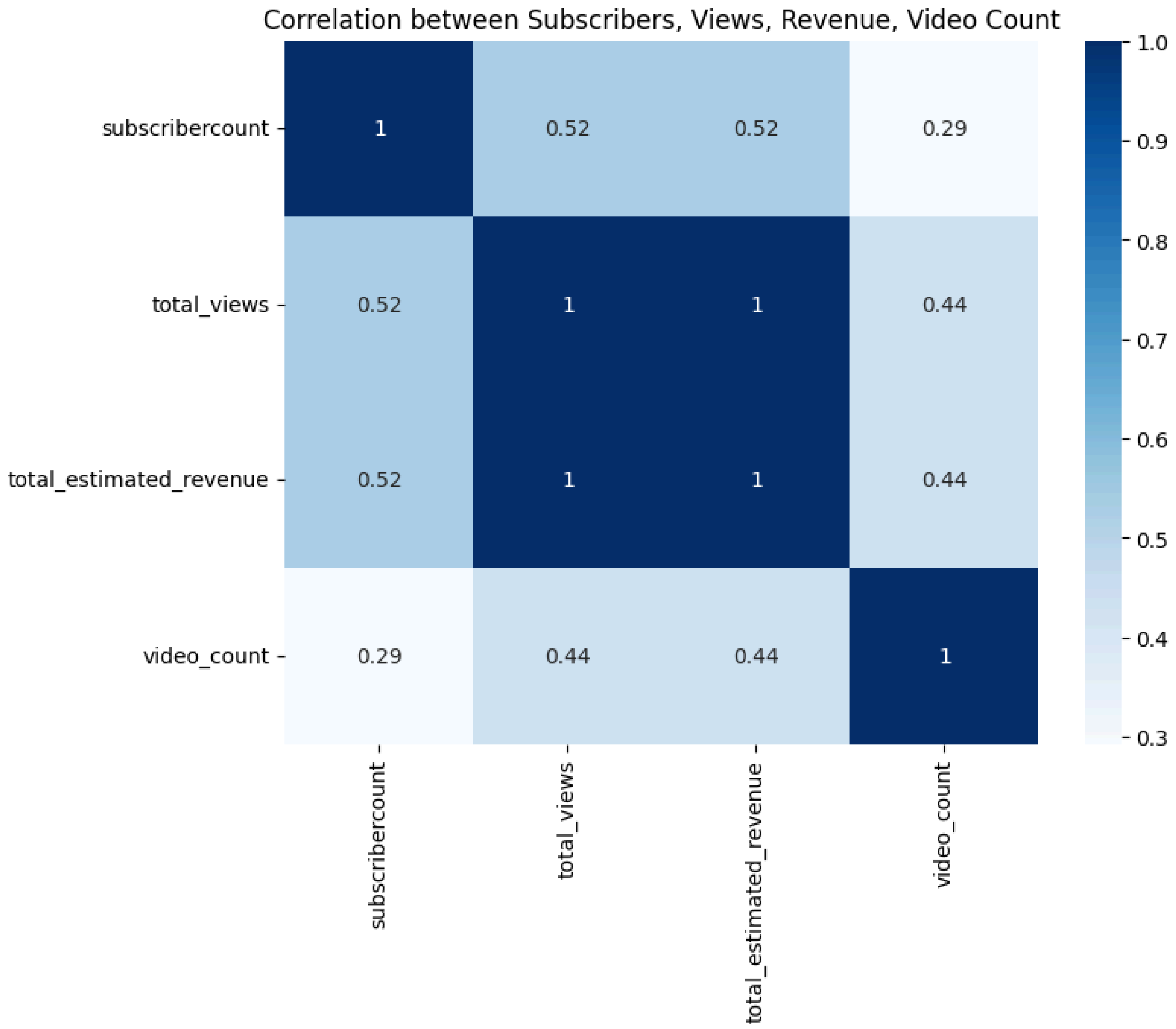


Business insight: Identify popular content languages to target audience growth.

Correlation Heatmap (numeric features)

```
numeric_cols = ["subscribercount", "total_views", "total_estimated_revenue", "video_count

plt.figure(figsize=(8,6))
sns.heatmap(revenue_master[numeric_cols].corr(), annot=True, cmap="Blues")
plt.title("Correlation between Subscribers, Views, Revenue, Video Count")
plt.savefig("eda_outputs/plots/correlation_heatmap.png")
plt.show()
```



Business insight: High correlation between subscribers and revenue indicates investing in audience growth is valuable.

Top 5 channels by subscribers (from revenue_master)

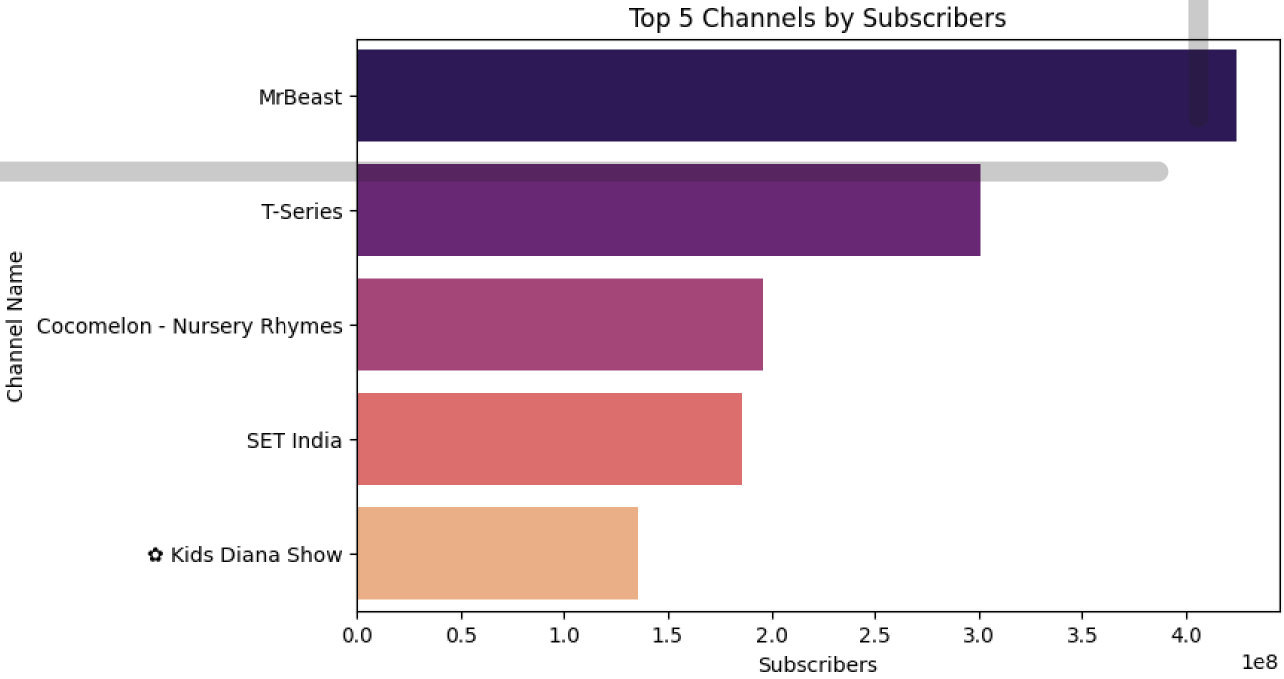
```
top_subs = revenue_master.sort_values("subscribercount", ascending=False).head(5)
top_subs.to_csv("eda_outputs/tables/top_5_channels_by_subscribers.csv", index=False

# Optional plot
plt.figure(figsize=(8,5))
sns.barplot(x="subscribercount", y="channelname", data=top_subs, palette="magma")
plt.title("Top 5 Channels by Subscribers")
plt.xlabel("Subscribers")
plt.ylabel("Channel Name")
plt.savefig("eda_outputs/plots/top_5_channels_by_subscribers.png")
```

```
/tmp/ipython-input-660349946.py:6: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.

  sns.barplot(x="subscribercount", y="channelname", data=top_subs, palette="magma")
```



Identifies top-performing channels by audience size.

Helps prioritize partnerships, promotions, or targeted marketing for maximum reach.