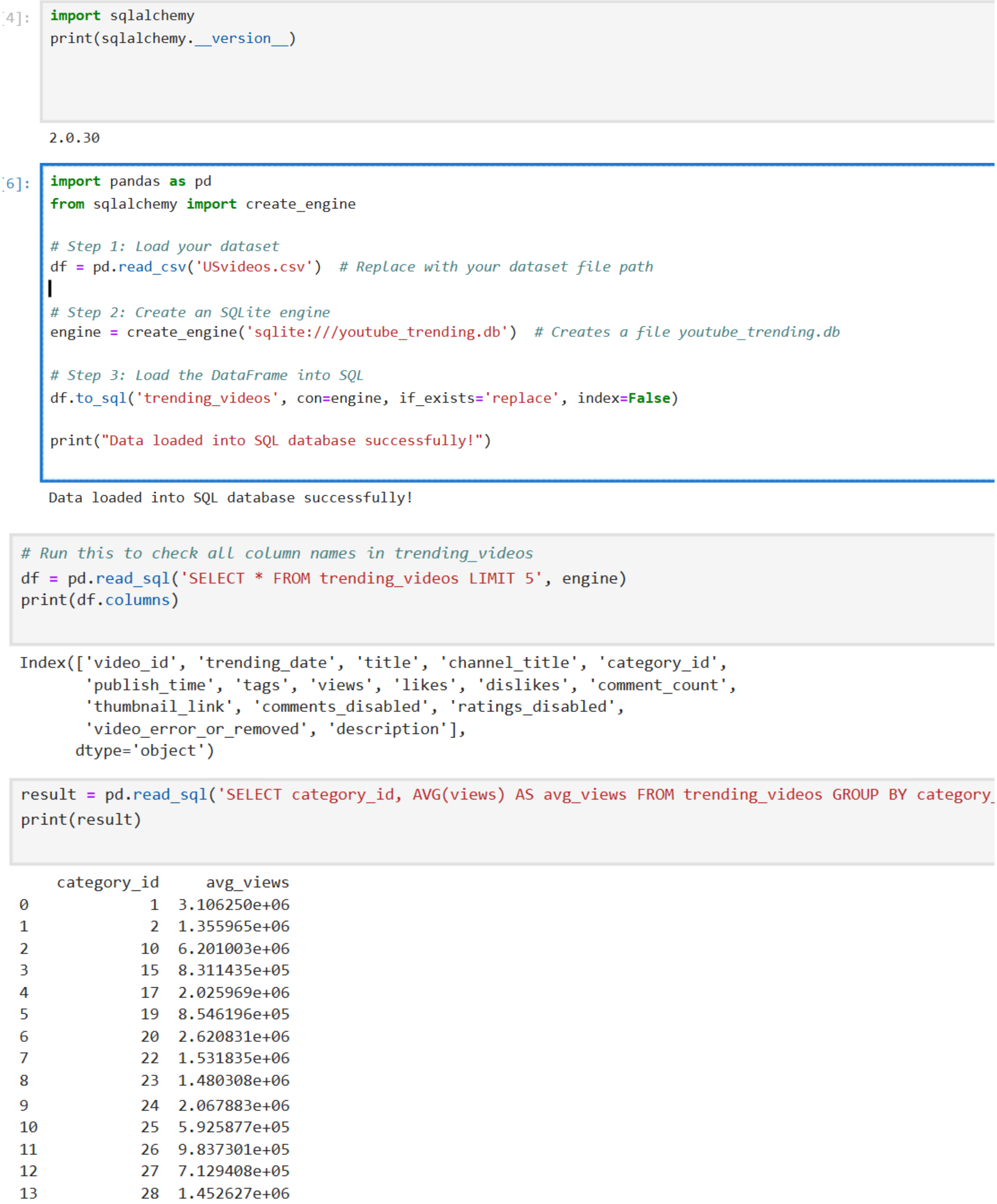


4]:

```
import sqlalchemy
print(sqlalchemy.__version__)
```

```
2.0.30
```

6]:

```
import pandas as pd
from sqlalchemy import create_engine

# Step 1: Load your dataset
df = pd.read_csv('USvideos.csv')  # Replace with your dataset file path

# Step 2: Create an SQLite engine
engine = create_engine('sqlite:///youtube_trending.db')  # Creates a file youtube_trending.db

# Step 3: Load the DataFrame into SQL
df.to_sql('trending_videos', con=engine, if_exists='replace', index=False)

print("Data loaded into SQL database successfully!")
```

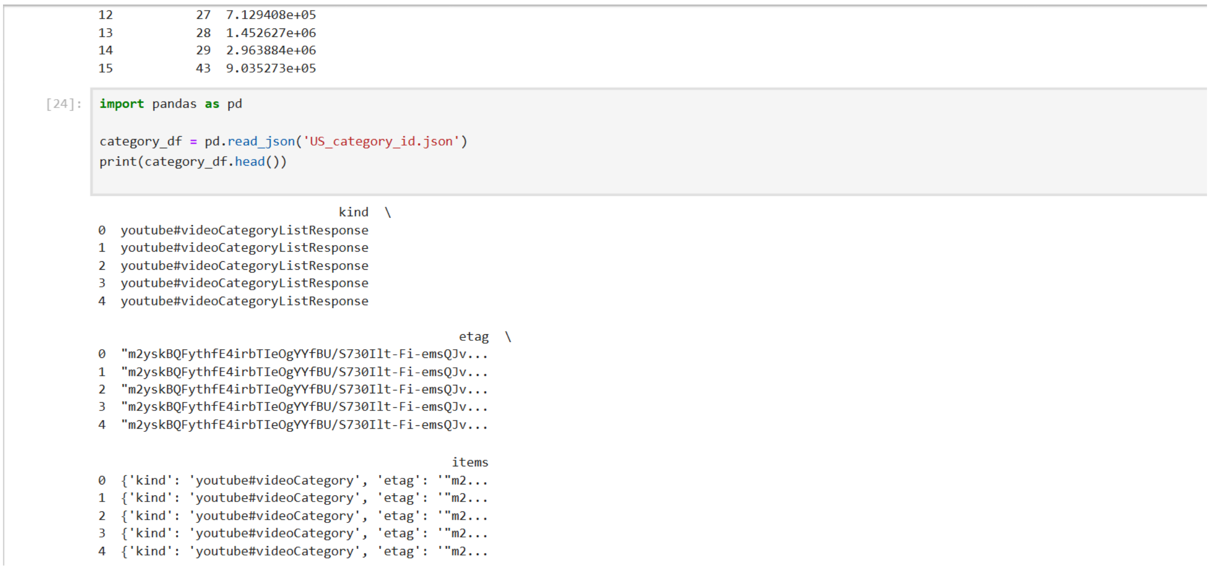
```
Data loaded into SQL database successfully!
```

```
# Run this to check all column names in trending_videos
df = pd.read_sql('SELECT * FROM trending_videos LIMIT 5', engine)
print(df.columns)
```

```
Index(['video_id', 'trending_date', 'title', 'channel_title', 'category_id',
       'publish_time', 'tags', 'views', 'likes', 'dislikes', 'comment_count',
       'thumbnail_link', 'comments_disabled', 'ratings_disabled',
       'video_error_or_removed', 'description'],
      dtype='object')
```

```
result = pd.read_sql('SELECT category_id, AVG(views) AS avg_views FROM trending_videos GROUP BY category_
print(result)
```

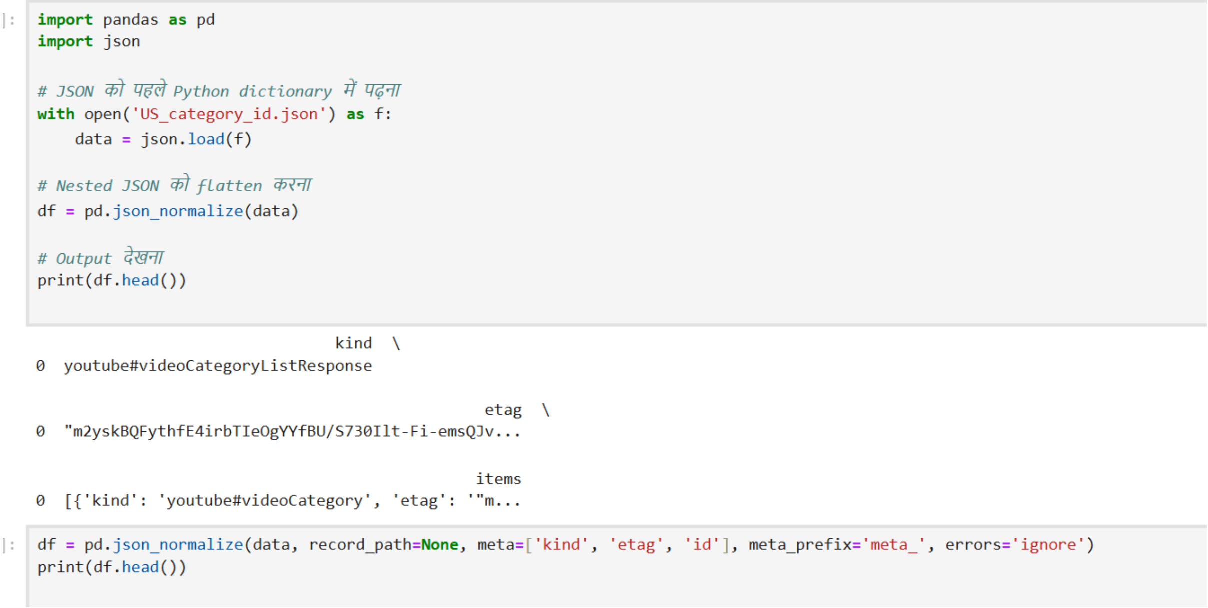
```
    category_id     avg_views
0             1  3.106250e+06
1             2  1.355965e+06
2            10  6.201003e+06
3            15  8.311435e+05
4            17  2.025969e+06
5            19  8.546196e+05
6            20  2.620831e+06
7            22  1.531835e+06
8            23  1.480308e+06
9            24  2.067883e+06
10           25  5.925877e+05
11           26  9.837301e+05
12           27  7.129408e+05
13           28  1.452627e+06
```



```
12          27  7.129408e+05
13          28  1.452627e+06
14          29  2.963884e+06
15          43  9.035273e+05
```

```
[24]: import pandas as pd

      category_df = pd.read_json('US_category_id.json')
      print(category_df.head())
```

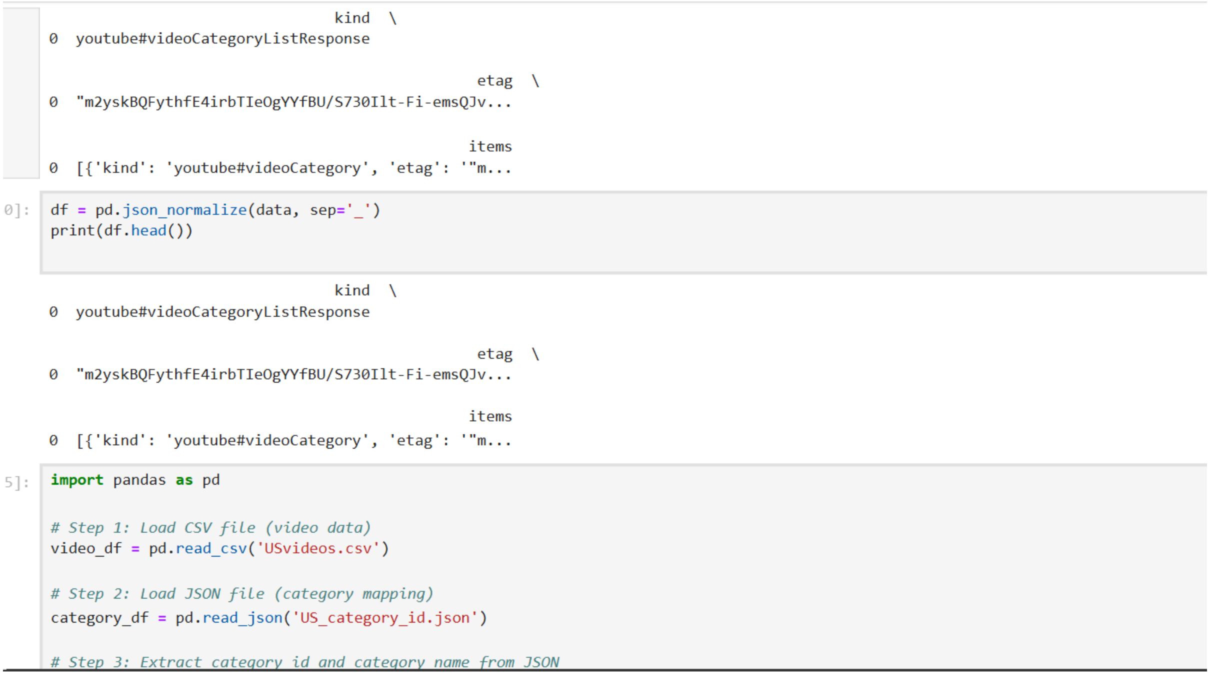
```
                                kind  \
0  youtube#videoCategoryListResponse
1  youtube#videoCategoryListResponse
2  youtube#videoCategoryListResponse
3  youtube#videoCategoryListResponse
4  youtube#videoCategoryListResponse

                                                etag  \
0  "m2yskBQFythfE4irbTIeOgYYfBU/S730Ilt-Fi-emsQJv...
1  "m2yskBQFythfE4irbTIeOgYYfBU/S730Ilt-Fi-emsQJv...
2  "m2yskBQFythfE4irbTIeOgYYfBU/S730Ilt-Fi-emsQJv...
3  "m2yskBQFythfE4irbTIeOgYYfBU/S730Ilt-Fi-emsQJv...
4  "m2yskBQFythfE4irbTIeOgYYfBU/S730Ilt-Fi-emsQJv...

                                               items
0  {'kind': 'youtube#videoCategory', 'etag': '"m2...
1  {'kind': 'youtube#videoCategory', 'etag': '"m2...
2  {'kind': 'youtube#videoCategory', 'etag': '"m2...
3  {'kind': 'youtube#videoCategory', 'etag': '"m2...
4  {'kind': 'youtube#videoCategory', 'etag': '"m2...
```



```
[ ]: import pandas as pd
     import json

     # JSON को पहले Python dictionary में पढ़ना
     with open('US_category_id.json') as f:
         data = json.load(f)

     # Nested JSON को flatten करना
     df = pd.json_normalize(data)

     # Output देखना
     print(df.head())
```

```
                                kind  \
0  youtube#videoCategoryListResponse

                                                etag  \
0  "m2yskBQFythfE4irbTIeOgYYfBU/S730Ilt-Fi-emsQJv...

                                               items
0  [{'kind': 'youtube#videoCategory', 'etag': '"m...
```

```
[ ]: df = pd.json_normalize(data, record_path=None, meta=['kind', 'etag', 'id'], meta_prefix='meta_', errors='ignore')
     print(df.head())
```



```
                                       kind  \
0  youtube#videoCategoryListResponse

                                                     etag  \
0  "m2yskBQFythfE4irbTIeOgYYfBU/S730Ilt-Fi-emsQJv...

                                                    items
0  [{'kind': 'youtube#videoCategory', 'etag': '"m...
```

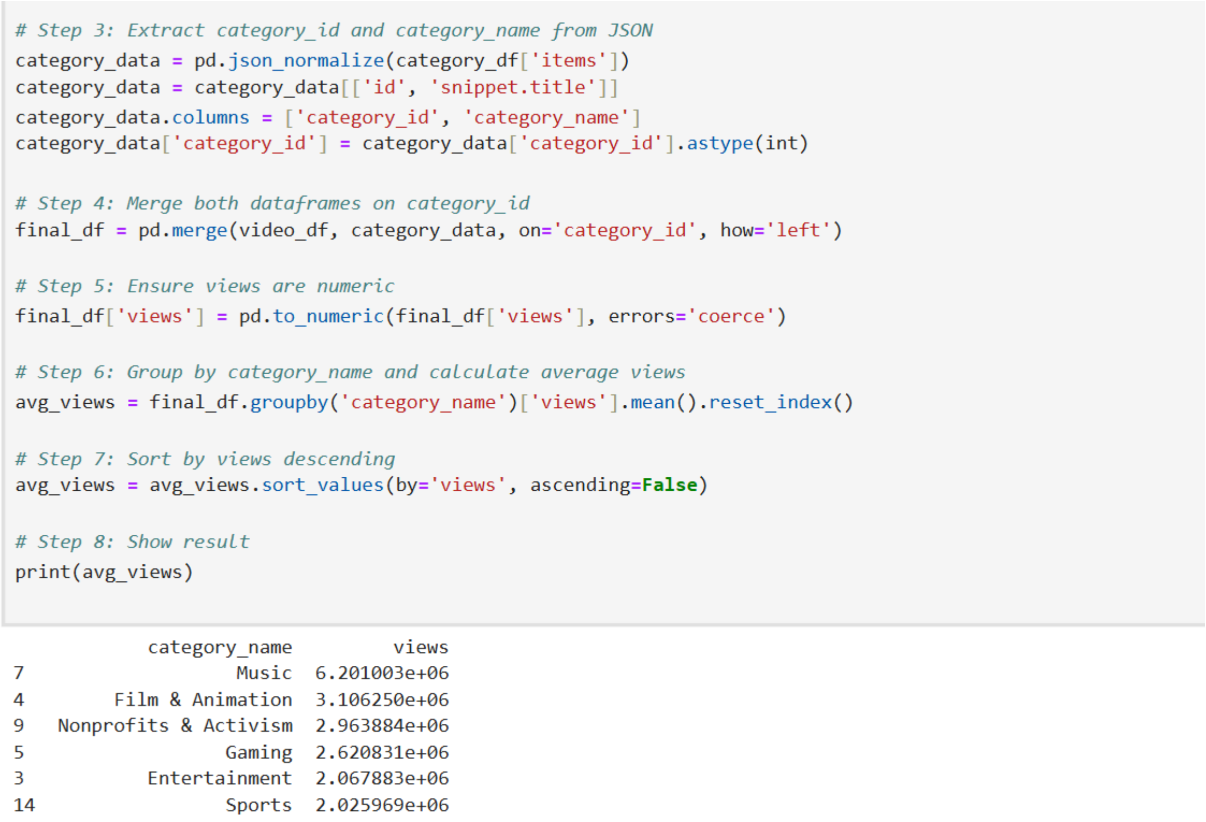
0]:

```
df = pd.json_normalize(data, sep='_')
print(df.head())
```

```
                                       kind  \
0  youtube#videoCategoryListResponse

                                                     etag  \
0  "m2yskBQFythfE4irbTIeOgYYfBU/S730Ilt-Fi-emsQJv...

                                                    items
0  [{'kind': 'youtube#videoCategory', 'etag': '"m...
```

5]:

```
import pandas as pd

# Step 1: Load CSV file (video data)
video_df = pd.read_csv('USvideos.csv')

# Step 2: Load JSON file (category mapping)
category_df = pd.read_json('US_category_id.json')

# Step 3: Extract category_id and category_name from JSON
```



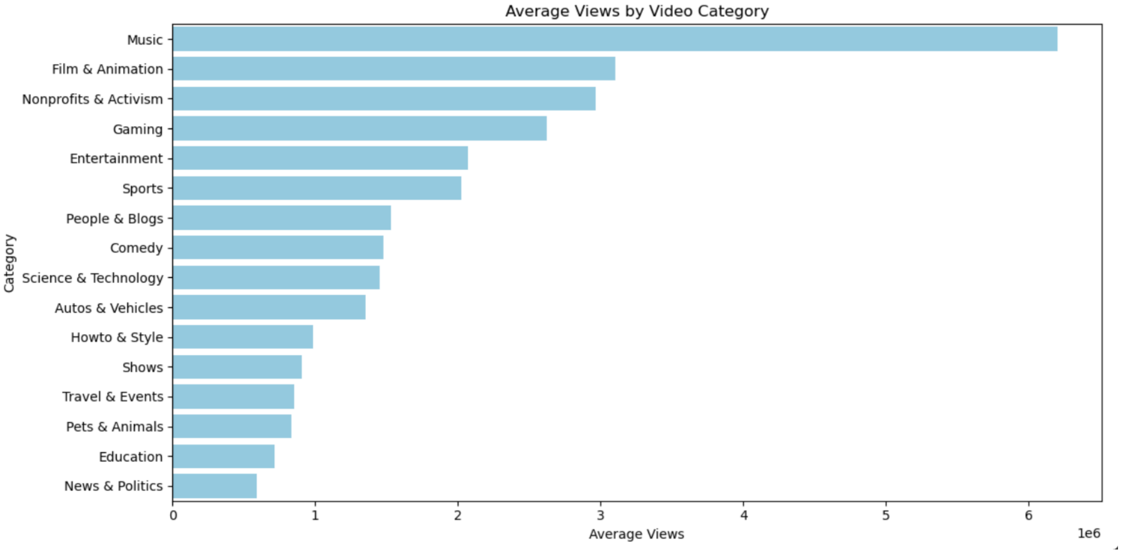
```
# Step 3: Extract category_id and category_name from JSON
category_data = pd.json_normalize(category_df['items'])
category_data = category_data[['id', 'snippet.title']]
category_data.columns = ['category_id', 'category_name']
category_data['category_id'] = category_data['category_id'].astype(int)

# Step 4: Merge both dataframes on category_id
final_df = pd.merge(video_df, category_data, on='category_id', how='left')

# Step 5: Ensure views are numeric
final_df['views'] = pd.to_numeric(final_df['views'], errors='coerce')

# Step 6: Group by category_name and calculate average views
avg_views = final_df.groupby('category_name')['views'].mean().reset_index()

# Step 7: Sort by views descending
avg_views = avg_views.sort_values(by='views', ascending=False)

# Step 8: Show result
print(avg_views)
```

```
           category_name         views
7                  Music  6.201003e+06
4       Film & Animation  3.106250e+06
9   Nonprofits & Activism  2.963884e+06
5                 Gaming  2.620831e+06
3          Entertainment  2.067883e+06
14                Sports  2.025969e+06
```

```
14                Sports  2.025969e+06
10         People & Blogs  1.531835e+06
1                 Comedy  1.480308e+06
12  Science & Technology  1.452627e+06
0        Autos & Vehicles  1.355965e+06
6           Howto & Style  9.837301e+05
13                 Shows  9.035273e+05
15         Travel & Events  8.546196e+05
11          Pets & Animals  8.311435e+05
2              Education  7.129408e+05
8         News & Politics  5.925877e+05
```

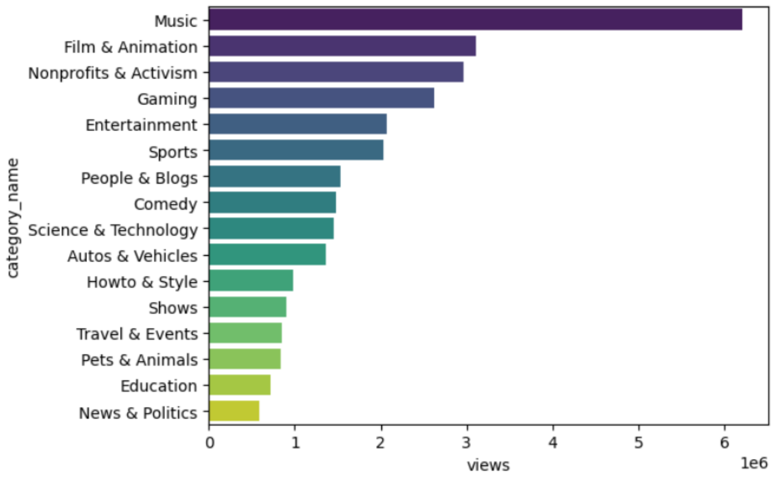
```python
import matplotlib.pyplot as plt
import seaborn as sns

# Plotting
plt.figure(figsize=(12,6))
sns.barplot(x='views', y='category_name', data=avg_views, color='skyblue')  # no palette warning
plt.title('Average Views by Video Category')
plt.xlabel('Average Views')
plt.ylabel('Category')
plt.tight_layout()
plt.show()
```

Average Views by Video Category

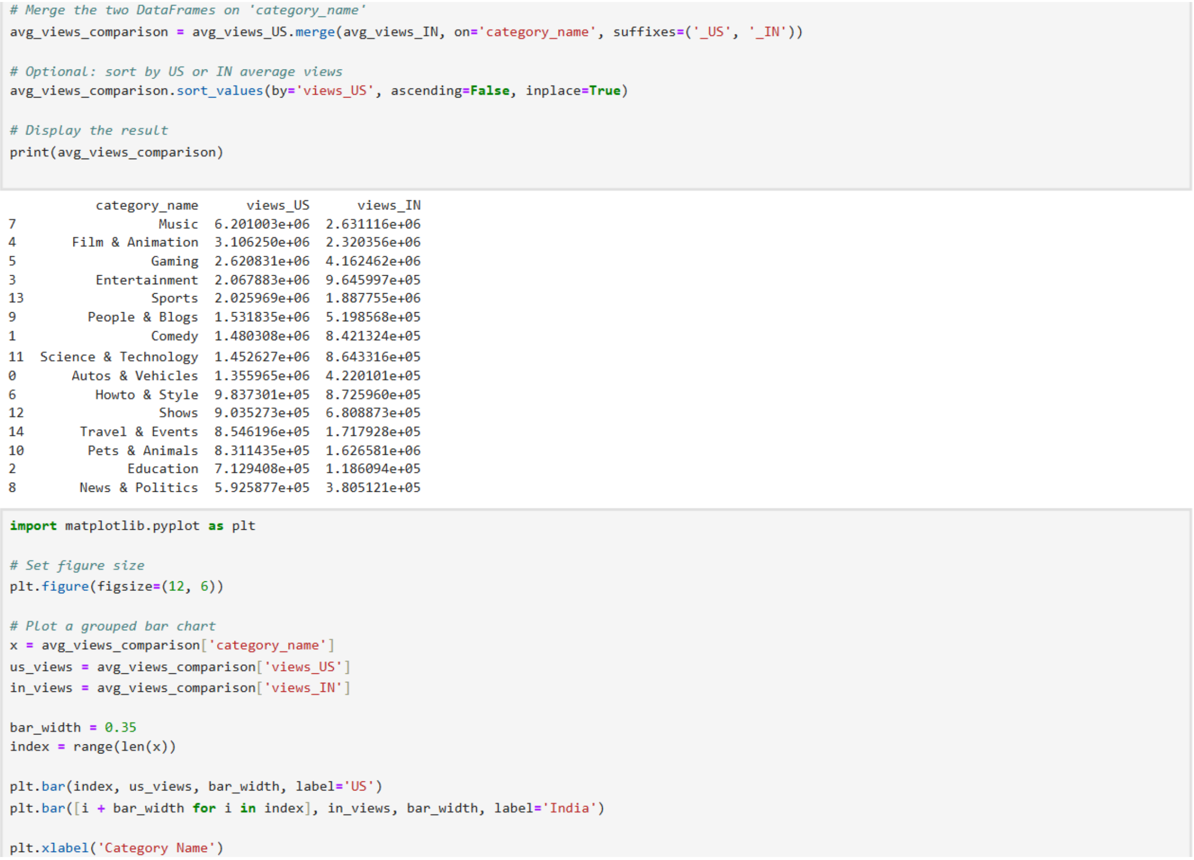


```python
sns.barplot(
    x='views',
    y='category_name',
    data=avg_views,
    hue='category_name',
    dodge=False,
    palette='viridis',
    legend=False
)
```

```
<Axes: xlabel='views', ylabel='category_name'>
```



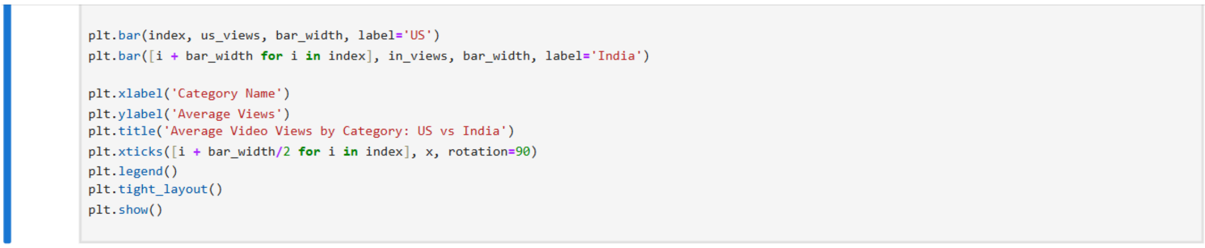
```python
import pandas as pd

# 1. Normalize the category data
category_us_df = pd.json_normalize(category_us['items'])
category_in_df = pd.json_normalize(category_in['items'])

# 2. Extract and rename the mapping columns
category_us_map = category_us_df[['id', 'snippet.title']].copy()
category_us_map.columns = ['category_id', 'category_name']
category_us_map['category_id'] = category_us_map['category_id'].astype(int)

category_in_map = category_in_df[['id', 'snippet.title']].copy()
category_in_map.columns = ['category_id', 'category_name']
category_in_map['category_id'] = category_in_map['category_id'].astype(int)

# 3. Make sure df_us and df_in have 'category_id' as int
df_us['category_id'] = df_us['category_id'].astype(int)
df_in['category_id'] = df_in['category_id'].astype(int)

# 4. Merge the category names into the US and IN DataFrames
df_us = df_us.merge(category_us_map, on='category_id', how='left')
df_in = df_in.merge(category_in_map, on='category_id', how='left')

# 5. Check if 'category_name' and 'views' columns exist
assert 'category_name' in df_us.columns, "category_name not found in df_us"
assert 'category_name' in df_in.columns, "category_name not found in df_in"
assert 'views' in df_us.columns, "views not found in df_us"
assert 'views' in df_in.columns, "views not found in df_in"

# 6. Group by category_name and calculate average views
avg_views_US = df_us.groupby('category_name')['views'].mean().reset_index()
avg_views_IN = df_in.groupby('category_name')['views'].mean().reset_index()
```



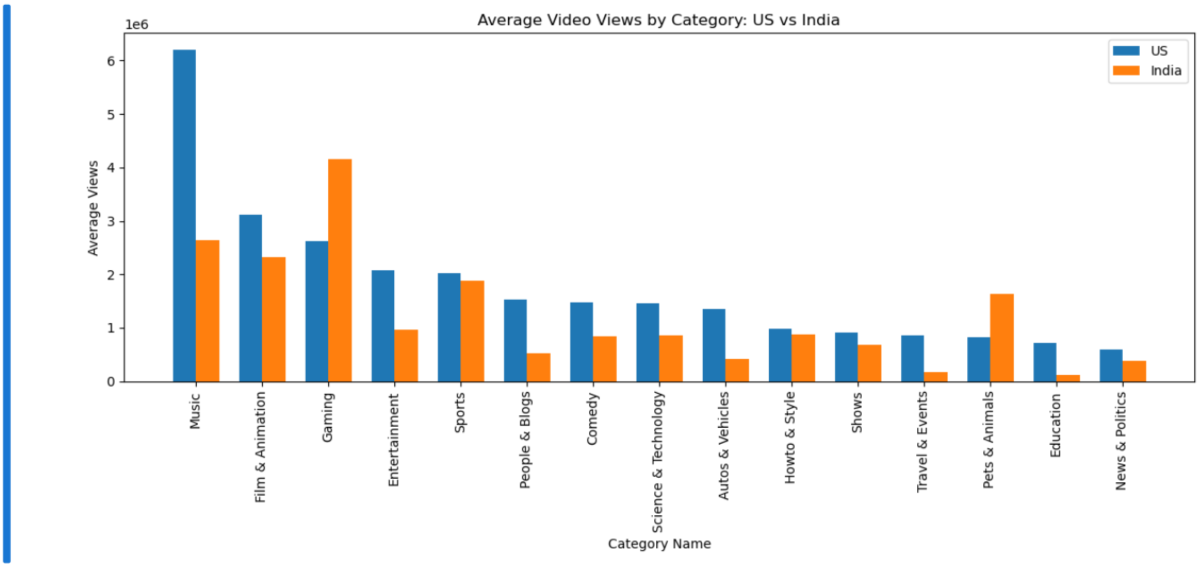
```
# Merge the two DataFrames on 'category_name'
avg_views_comparison = avg_views_US.merge(avg_views_IN, on='category_name', suffixes=('_US', '_IN'))

# Optional: sort by US or IN average views
avg_views_comparison.sort_values(by='views_US', ascending=False, inplace=True)

# Display the result
print(avg_views_comparison)
```

```
          category_name      views_US      views_IN
7                 Music  6.201003e+06  2.631116e+06
4      Film & Animation  3.106250e+06  2.320356e+06
5                Gaming  2.620831e+06  4.162462e+06
3         Entertainment  2.067883e+06  9.645997e+05
13               Sports  2.025969e+06  1.887755e+06
9        People & Blogs  1.531835e+06  5.198568e+05
1                Comedy  1.480308e+06  8.421324e+05
11  Science & Technology  1.452627e+06  8.643316e+05
0      Autos & Vehicles  1.355965e+06  4.220101e+05
6         Howto & Style  9.837301e+05  8.725960e+05
12                Shows  9.035273e+05  6.808873e+05
14      Travel & Events  8.546196e+05  1.717928e+05
10       Pets & Animals  8.311435e+05  1.626581e+06
2             Education  7.129408e+05  1.186094e+05
8       News & Politics  5.925877e+05  3.805121e+05
```

```
import matplotlib.pyplot as plt

# Set figure size
plt.figure(figsize=(12, 6))

# Plot a grouped bar chart
x = avg_views_comparison['category_name']
us_views = avg_views_comparison['views_US']
in_views = avg_views_comparison['views_IN']

bar_width = 0.35
index = range(len(x))

plt.bar(index, us_views, bar_width, label='US')
plt.bar([i + bar_width for i in index], in_views, bar_width, label='India')

plt.xlabel('Category Name')
```



```
plt.bar(index, us_views, bar_width, label='US')
plt.bar([i + bar_width for i in index], in_views, bar_width, label='India')

plt.xlabel('Category Name')
plt.ylabel('Average Views')
plt.title('Average Video Views by Category: US vs India')
plt.xticks([i + bar_width/2 for i in index], x, rotation=90)
plt.legend()
plt.tight_layout()
plt.show()
```



```python
# To CSV
avg_views_comparison.to_csv("average_views_comparison.csv", index=False)

# Or to Excel
avg_views_comparison.to_excel("average_views_comparison.xlsx", index=False)
```

```python
# Add a new column to see the difference
avg_views_comparison['view_difference'] = avg_views_comparison['views_US'] - avg_views_comparison['views_IN']

# Sort by the difference to find the biggest gap in views
avg_views_comparison_sorted = avg_views_comparison.sort_values(by='view_difference', ascending=False)

# Display the top 10 categories with the biggest difference
print(avg_views_comparison_sorted.head(10))
```

```
         category_name      views_US      views_IN  view_difference
7                Music  6.201003e+06  2.631116e+06     3.569887e+06
3        Entertainment  2.067883e+06  9.645997e+05     1.103283e+06
9       People & Blogs  1.531835e+06  5.198568e+05     1.011979e+06
0     Autos & Vehicles  1.355965e+06  4.220101e+05     9.339553e+05
4     Film & Animation  3.106250e+06  2.320356e+06     7.858946e+05
14     Travel & Events  8.546196e+05  1.717928e+05     6.828269e+05
1               Comedy  1.480308e+06  8.421324e+05     6.381760e+05
2            Education  7.129408e+05  1.186094e+05     5.943314e+05
11  Science & Technology  1.452627e+06  8.643316e+05     5.882952e+05
12               Shows  9.035273e+05  6.808873e+05     2.226401e+05
```

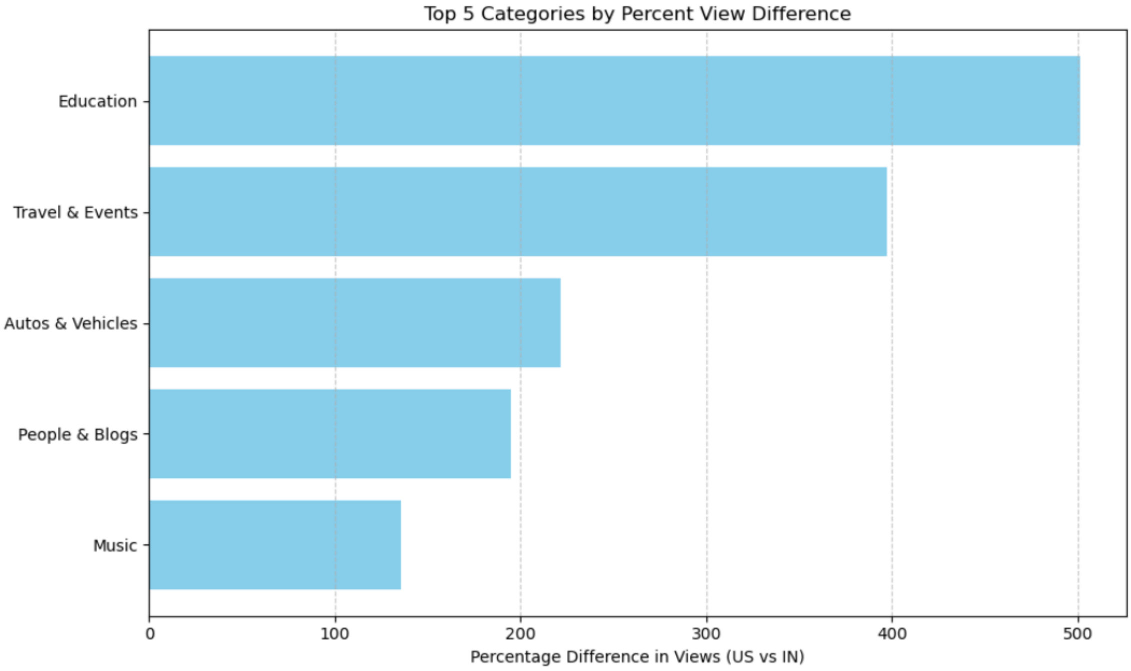
```python
# Top 5 categories in the US
top_us = avg_views_comparison.sort_values(by='views_US', ascending=False).head(5)
print("Top 5 Categories in US:")
print(top_us[['category_name', 'views_US']])

# Top 5 categories in India
top_in = avg_views_comparison.sort_values(by='views_IN', ascending=False).head(5)
print("\nTop 5 Categories in India:")
print(top_in[['category_name', 'views_IN']])
```

```
Top 5 Categories in US:
       category_name      views_US
7              Music  6.201003e+06
4   Film & Animation  3.106250e+06
5             Gaming  2.620831e+06
3      Entertainment  2.067883e+06
13            Sports  2.025969e+06

Top 5 Categories in India:
       category_name      views_IN
5             Gaming  4.162462e+06
7              Music  2.631116e+06
4   Film & Animation  2.320356e+06
13            Sports  1.887755e+06
10    Pets & Animals  1.626581e+06
```

```python
avg_views_comparison['percent_difference'] = (
    (avg_views_comparison['views_US'] - avg_views_comparison['views_IN']) / avg_views_comparison['views_IN']
) * 100

# Show top 5 most different categories by percentage
print(avg_views_comparison.sort_values(by='percent_difference', ascending=False).head())
```

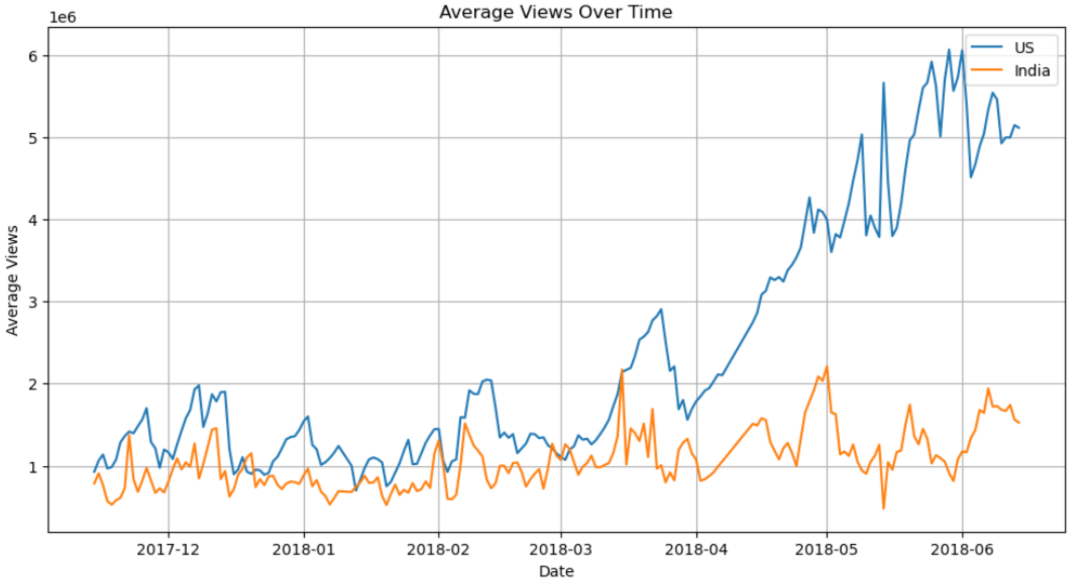
```
       category_name      views_US      views_IN  view_difference  \
2          Education  7.129408e+05  1.186094e+05     5.943314e+05
14   Travel & Events  8.546196e+05  1.717928e+05     6.828269e+05
0   Autos & Vehicles  1.355965e+06  4.220101e+05     9.339553e+05
9     People & Blogs  1.531835e+06  5.198568e+05     1.011979e+06
7              Music  6.201003e+06  2.631116e+06     3.569887e+06

    percent_difference
2           501.082659
14          397.471291
0           221.311150
9           194.664883
7           135.679604
```

```python
# Plot percent differences
top_diff = avg_views_comparison.sort_values(by='percent_difference', ascending=False).head(5)

plt.figure(figsize=(10, 6))
plt.barh(top_diff['category_name'], top_diff['percent_difference'], color='skyblue')
plt.xlabel("Percentage Difference in Views (US vs IN)")
plt.title("Top 5 Categories by Percent View Difference")
plt.gca().invert_yaxis()
plt.grid(axis='x', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
```



```python
avg_views_comparison.to_csv("average_views_comparison.csv", index=False)
```

```python
import pandas as pd

# Load the CSV files
us_df = pd.read_csv("USvideos.csv")
in_df = pd.read_csv("INvideos.csv")
# Convert trending_date to datetime
us_df['trending_date'] = pd.to_datetime(us_df['trending_date'], format='%y.%d.%m')
in_df['trending_date'] = pd.to_datetime(in_df['trending_date'], format='%y.%d.%m')
import matplotlib.pyplot as plt

us_trend = us_df.groupby('trending_date')['views'].mean()
in_trend = in_df.groupby('trending_date')['views'].mean()

plt.figure(figsize=(12,6))
plt.plot(us_trend.index, us_trend.values, label='US')
plt.plot(in_trend.index, in_trend.values, label='India')
plt.title("Average Views Over Time")
plt.xlabel("Date")
plt.ylabel("Average Views")
plt.legend()
plt.grid(True)
plt.show()
```



[56]:

```
# Top 10 most viewed videos in the US
top_us = us_df.sort_values('views', ascending=False).head(10)
print("Top 10 US Videos by Views:")
print(top_us[['title', 'views']])

# Top 10 most viewed videos in India
top_in = in_df.sort_values('views', ascending=False).head(10)
print("\nTop 10 India Videos by Views:")
print(top_in[['title', 'views']])
```

```
Top 10 US Videos by Views:
                                                   title      views
38547  Childish Gambino - This Is America (Official V...  225211923
38345  Childish Gambino - This Is America (Official V...  220490543
38146  Childish Gambino - This Is America (Official V...  217750076
37935  Childish Gambino - This Is America (Official V...  210338856
37730  Childish Gambino - This Is America (Official V...  205643016
37531  Childish Gambino - This Is America (Official V...  200820941
37333  Childish Gambino - This Is America (Official V...  196222618
37123  Childish Gambino - This Is America (Official V...  190950401
36913  Childish Gambino - This Is America (Official V...  184446490
36710  Childish Gambino - This Is America (Official V...  179045286

Top 10 India Videos by Views:
                                                  title      views
5408  YouTube Rewind: The Shape of 2017 | #YouTubeRe...  125432237
5119  YouTube Rewind: The Shape of 2017 | #YouTubeRe...  113876217
4936  YouTube Rewind: The Shape of 2017 | #YouTubeRe...  100911567
4477  Marvel Studios' Avengers: Infinity War Officia...   89930713
4236  Marvel Studios' Avengers: Infinity War Officia...   87449453
4013  Marvel Studios' Avengers: Infinity War Officia...   84281319
3823  Marvel Studios' Avengers: Infinity War Officia...   80360459
4743  YouTube Rewind: The Shape of 2017 | #YouTubeRe...   75969469
3639  Marvel Studios' Avengers: Infinity War Officia...   74789251
3456  Marvel Studios' Avengers: Infinity War Officia...   66637636
```

]:

```
# Count of videos per category in US
print("US Category Distribution:")
print(us_df['category_id'].value_counts())

# Count of videos per category in India
print("\nIndia Category Distribution:")
print(in_df['category_id'].value_counts())
```

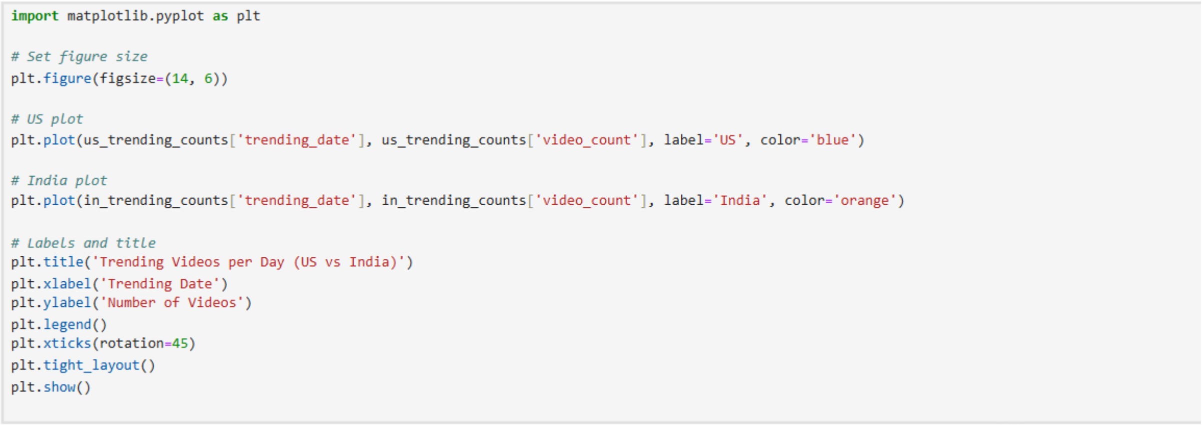
```
US Category Distribution:
category_id
```

```
US Category Distribution:
category_id
24    9964
10    6472
26    4146
23    3457
22    3210
25    2487
28    2401
1     2345
17    2174
27    1656
15     920
20     817
19     402
2      384
29      57
43      57
Name: count, dtype: int64

India Category Distribution:
category_id
24    16712
25     5241
10     3858
23     3429
22     2624
1      1658
27     1227
26      845
17      731
28      552
43      205
29      105
2        72
20       66
30       16
19        8
15        3
Name: count, dtype: int64
```

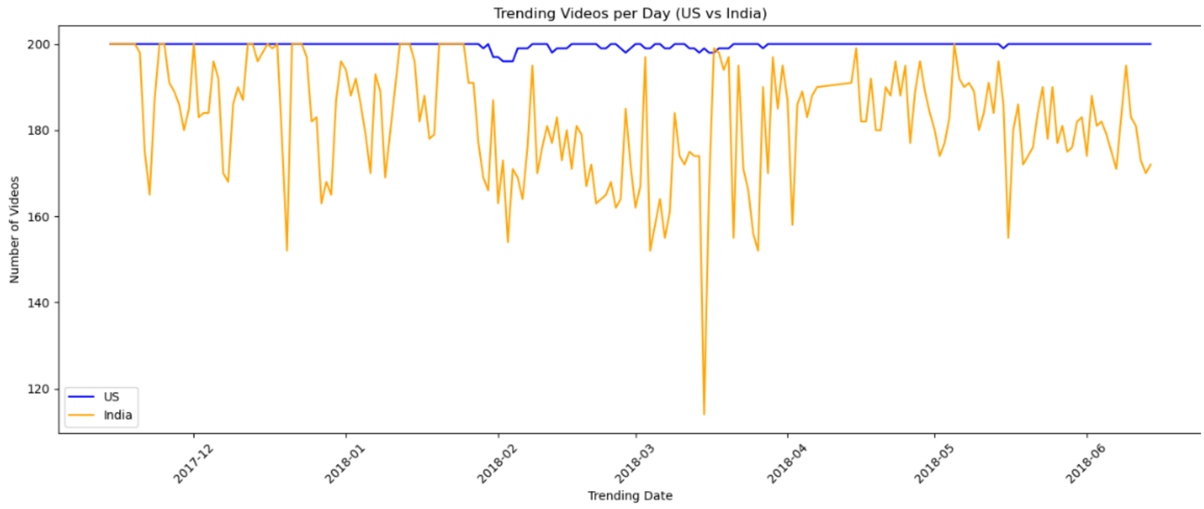
```
us_df['like_dislike_ratio'] = us_df['likes'] / (us_df['dislikes'] + 1)
in_df['like_dislike_ratio'] = in_df['likes'] / (in_df['dislikes'] + 1)

print("Top 5 US videos by like/dislike ratio:")
print(us_df.sort_values('like_dislike_ratio', ascending=False)[['title', 'like_dislike_ratio']].head())

print("\nTop 5 India videos by like/dislike ratio:")
print(in_df.sort_values('like_dislike_ratio', ascending=False)[['title', 'like_dislike_ratio']].head())
```

```
Top 5 US videos by like/dislike ratio:
                                         title  like_dislike_ratio
7933  Jonghyun Lonely (Feat. 태연) - Piano Cover              1303.0
8985  Jonghyun Lonely (Feat. 태연) - Piano Cover              1195.0
8762  Jonghyun Lonely (Feat. 태연) - Piano Cover              1175.0
8552  Jonghyun Lonely (Feat. 태연) - Piano Cover              1151.0
8347  Jonghyun Lonely (Feat. 태연) - Piano Cover              1123.6

Top 5 India videos by like/dislike ratio:
                                                   title  like_dislike_ratio
6025   Ronnie Singh - BARBAAD [Teaser] |Harman Buttar...          769.428571
22510  Daily Promise and Prayer by Bro. P. Satish Kum...          464.000000
8604   Gurj Sidhu | Yaad Kar | Full Video | Kaos | VI...          425.166667
7534           நாற்காலிச்  சண்டைக்குள் சிதைந்த மீனவன் .          382.047619
2056                              Why birds fly in V shape?          365.111111
```

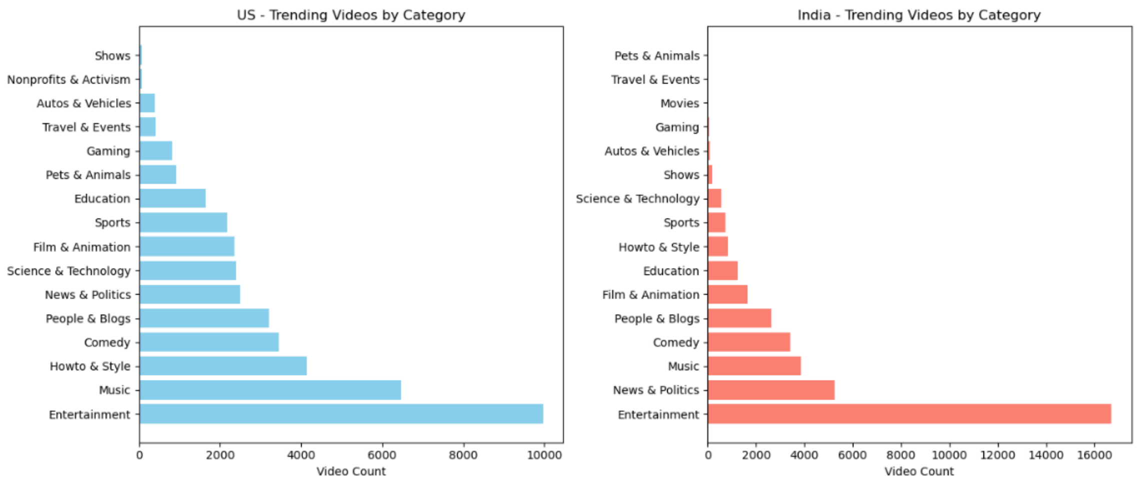
```
# Group by trending date and count number of videos
us_trending_counts = us_df.groupby('trending_date').size()
in_trending_counts = in_df.groupby('trending_date').size()

# Optional: convert to DataFrame for easy plotting
us_trending_counts = us_trending_counts.reset_index(name='video_count')
in_trending_counts = in_trending_counts.reset_index(name='video_count')
```



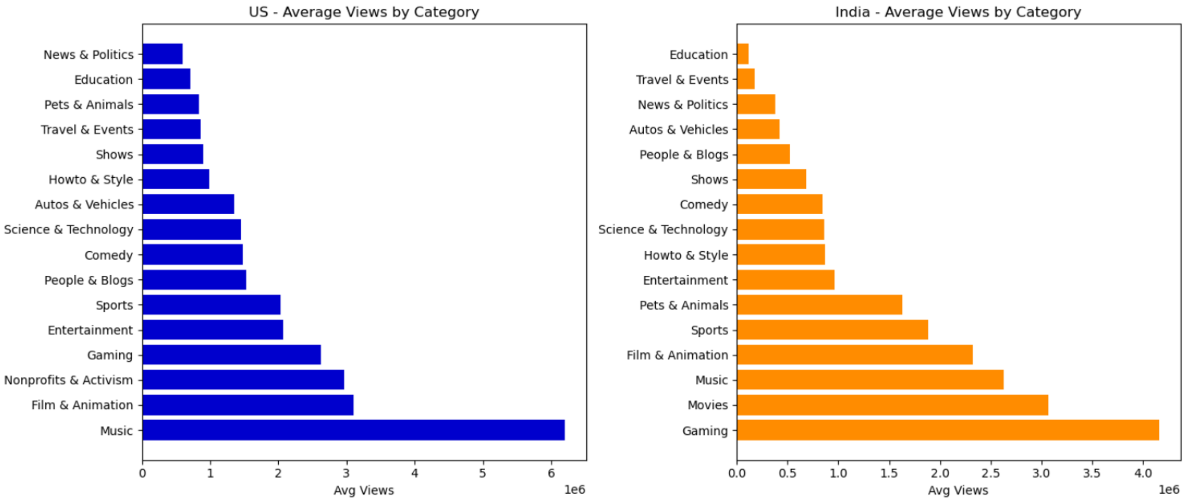
```
import matplotlib.pyplot as plt

# Set figure size
plt.figure(figsize=(14, 6))

# US plot
plt.plot(us_trending_counts['trending_date'], us_trending_counts['video_count'], label='US', color='blue')

# India plot
plt.plot(in_trending_counts['trending_date'], in_trending_counts['video_count'], label='India', color='orange')

# Labels and title
plt.title('Trending Videos per Day (US vs India)')
plt.xlabel('Trending Date')
plt.ylabel('Number of Videos')
plt.legend()
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



```
import json

# Load US categories
with open('US_category_id.json') as f:
    us_categories = json.load(f)

# Load India categories
with open('IN_category_id.json') as f:
    in_categories = json.load(f)

# Create a mapping: category_id -> category_name
def get_category_mapping(categories):
    mapping = {}
    for item in categories['items']:
        mapping[int(item['id'])] = item['snippet']['title']
    return mapping

us_cat_map = get_category_mapping(us_categories)
in_cat_map = get_category_mapping(in_categories)
us_df['category_name'] = us_df['category_id'].map(us_cat_map)
in_df['category_name'] = in_df['category_id'].map(in_cat_map)
# Count videos per category
us_category_counts = us_df['category_name'].value_counts().reset_index()
us_category_counts.columns = ['Category', 'Video Count']

in_category_counts = in_df['category_name'].value_counts().reset_index()
in_category_counts.columns = ['Category', 'Video Count']
# Plot side-by-side bar charts
plt.figure(figsize=(14, 6))

plt.subplot(1, 2, 1)
plt.barh(us_category_counts['Category'], us_category_counts['Video Count'], color='skyblue')
plt.title('US - Trending Videos by Category')
plt.xlabel('Video Count')

plt.subplot(1, 2, 2)
plt.barh(in_category_counts['Category'], in_category_counts['Video Count'], color='salmon')
plt.title('India - Trending Videos by Category')
plt.xlabel('Video Count')

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



```python
# US - Avg views per category
us_avg_views = us_df.groupby('category_name')['views'].mean().sort_values(ascending=False).reset_index()
us_avg_views.columns = ['Category', 'Average Views']

# India - Avg views per category
in_avg_views = in_df.groupby('category_name')['views'].mean().sort_values(ascending=False).reset_index()
in_avg_views.columns = ['Category', 'Average Views']
# US - Avg likes
us_avg_likes = us_df.groupby('category_name')['likes'].mean().sort_values(ascending=False).reset_index()

# India - Avg likes
in_avg_likes = in_df.groupby('category_name')['likes'].mean().sort_values(ascending=False).reset_index()
plt.figure(figsize=(14, 6))

plt.subplot(1, 2, 1)
plt.barh(us_avg_views['Category'], us_avg_views['Average Views'], color='mediumblue')
plt.title('US - Average Views by Category')
plt.xlabel('Avg Views')

plt.subplot(1, 2, 2)
plt.barh(in_avg_views['Category'], in_avg_views['Average Views'], color='darkorange')
plt.title('India - Average Views by Category')
plt.xlabel('Avg Views')

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


US - Average Views by Category
News & Politics
Education
Pets & Animals
Travel & Events
Shows
Howto & Style
Autos & Vehicles
Science & Technology
Comedy
People & Blogs
Sports
Entertainment
Gaming
Nonprofits & Activism
Film & Animation
Music
1e6
Avg Views
India - Average Views by Category
Education
Travel & Events
News & Politics
Autos & Vehicles
People & Blogs
Shows
Comedy
Science & Technology
Howto & Style
Entertainment
Pets & Animals
Sports
Film & Animation
Music
Movies
Gaming
1e6
Avg Views