YouTube Analytics

Now We are going to Explore the Top 3 Telugu Youtubers Video Contents Here

YouTube Overview

In [ ]:
```
import pandas as pd
import plotly
import plotly.offline as pyo
import plotly.graph_objs as go
import plotly.express as px
# Set notebook mode to work in offline
pyo.init_notebook_mode()
Ov = pd.read_excel(r"C:\Users\50510\Desktop\YouTube_Projects\YouTube_Overview.xlsx"
Ov.head()
```

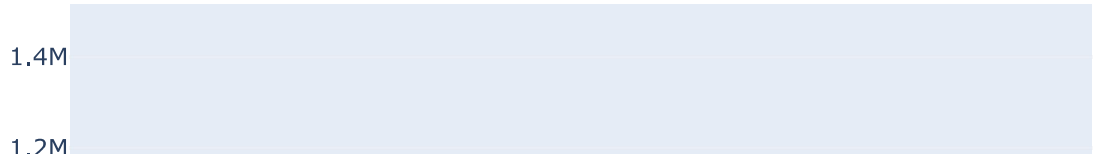
Out[ ]:

	Channel_name	Subscribers	Views	Total_Videos
0	Uma Telugu Traveller	816000	168577301	510
1	Ravi Telugu Traveller	700000	215070028	719
2	Naa Anveshana	1440000	369601360	1021

In [ ]:
```
Ov.info()
```

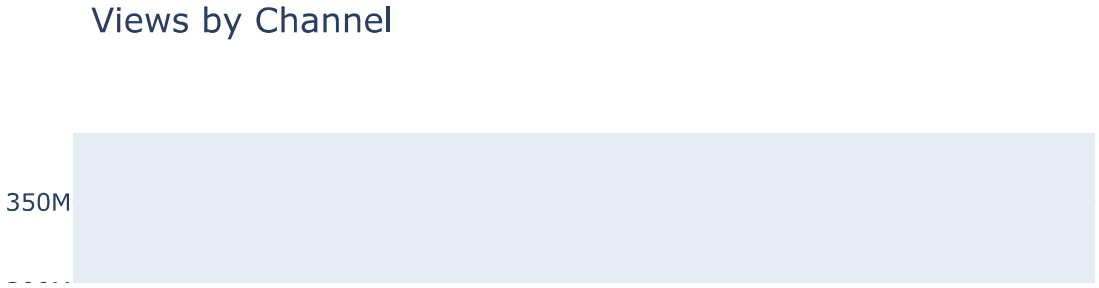
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3 entries, 0 to 2
Data columns (total 4 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   Channel_name  3 non-null      object
 1   Subscribers   3 non-null      int64 
 2   Views         3 non-null      int64 
 3   Total_Videos  3 non-null      int64 
dtypes: int64(3), object(1)
memory usage: 224.0+ bytes
```

In [ ]:
```
import plotly.express as px
# Create a bar plot with different colors for each channel
fig = px.bar(Ov, x='Channel_name', y='Subscribers', title='Subscribers by Channel',
             color='Channel_name')  # Specify 'color' parameter
fig.show()
```

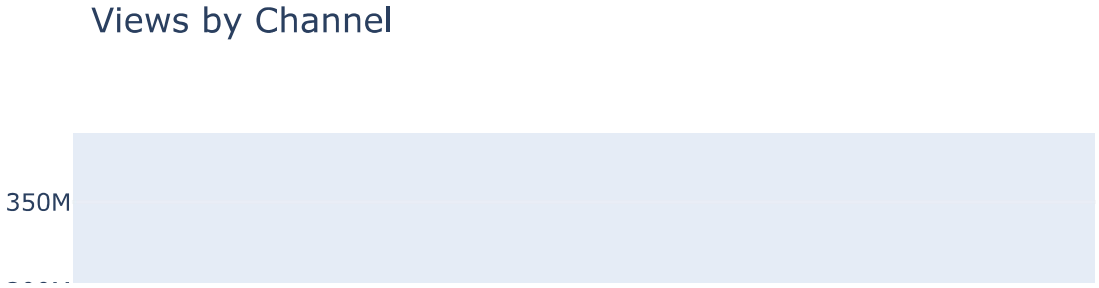


In [ ]:
```
import plotly.express as px

# Create a bar plot with different colors for each channel
fig = px.bar(Ov, x='Channel_name', y='Total_Videos', title='Posted Videos by Channe
             color='Channel_name')  # Specify 'color' parameter
fig.show()
```

In [ ]:
```
import plotly.express as px

# Create a bar plot with different colors for each channel
fig = px.bar(Ov, x='Channel_name', y='Views', title='Views by Channel',
             color='Channel_name')  # Specify 'color' parameter
fig.show()
```



In [ ]:
```
import plotly.express as px

# Create a bar plot with different colors for each channel
fig = px.bar(Ov, x='Subscribers', y='Views', title='Views by Channel',
             color='Channel_name')  # Specify 'color' parameter
fig.show()
```



Content Insights

In [ ]:
```
df = pd.read_excel(r"C:\Users\50510\Desktop\YouTube_Projects\Cleaned_YouTube_Data.x
df.drop(columns=['Unnamed: 0'], inplace=True)
df.head(1)
```

Out[ ]:

	Date_Published	Title	Tags	channel	Comments	viewCount	likeCount	fa
0	2023-08-22 02:30:05+00:00	Naa Anveshana meet up in OBJ Montreal and Vancouv...	['Naa Anveshana meet up in Montreal and Vancou...	Naa Anveshana	1169	333229	19698	

In [ ]:
```
df['Engagement'] = df['Comments'] + df['viewCount'] + df['likeCount'] + df['favorit
df.head()
```

Out[ ]:

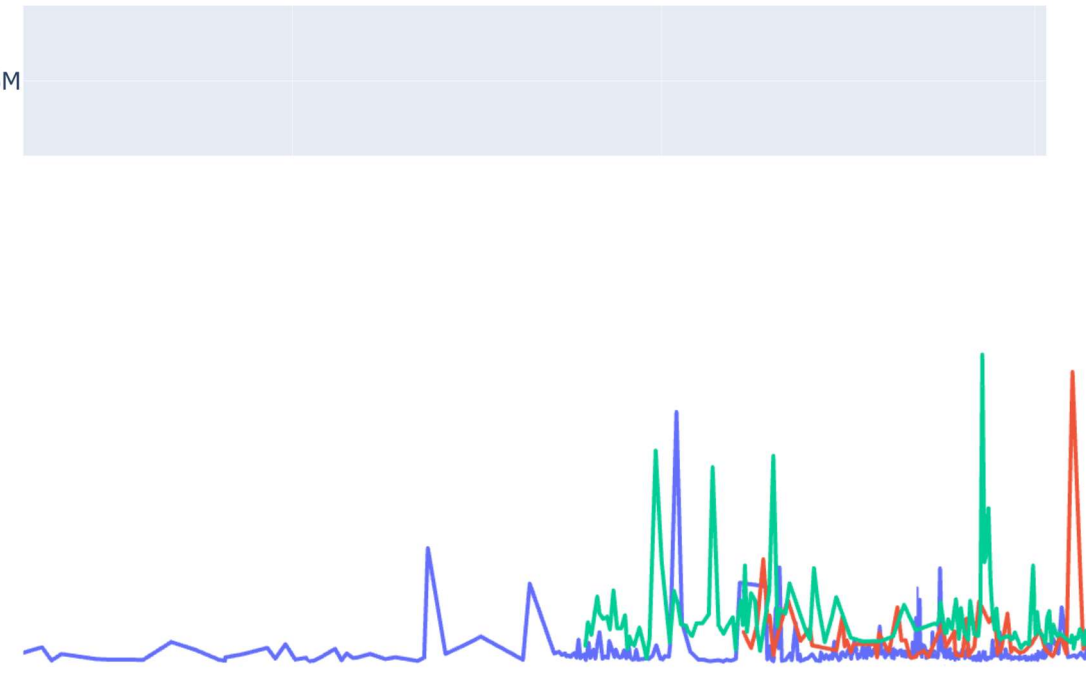
	Date_Published	Title	Tags	channel	Comments	viewCount	likeCou
0	2023-08-22 02:30:05+00:00	Naa Anveshana meet up in [OBJ] Montreal and Vancouv...	['Naa Anveshana meet up in Montreal and Vancou...	Naa Anveshana	1169	333229	196
1	2023-08-19 04:52:05+00:00	English Bay Beach Vancouver | Suspension bridg...	['Canada Place', 'English Bay Beach Vancouver'...	Naa Anveshana	2006	980291	510
2	2023-08-17 03:47:03+00:00	48 hrs On Canada's Most Luxury train journey B...	['Naaanveshana', 'banff', 'british columbia', ...	Naa Anveshana	3774	1686573	745
3	2023-08-15 02:30:31+00:00	Sulphur banff gondola | things to do in banff ...	['Naaanveshana', 'Sulphur banff gondola', 'ban...	Naa Anveshana	2462	2189913	638
4	2023-08-12 12:49:37+00:00	Columbia ice field Glacier Adventure Jasper Na...	['Naaanveshana', 'athabasca falls', 'athabasca...	Naa Anveshana	3430	1481449	646

In [ ]:
```
df['Date_Published'] = pd.to_datetime(df['Date_Published'])
```

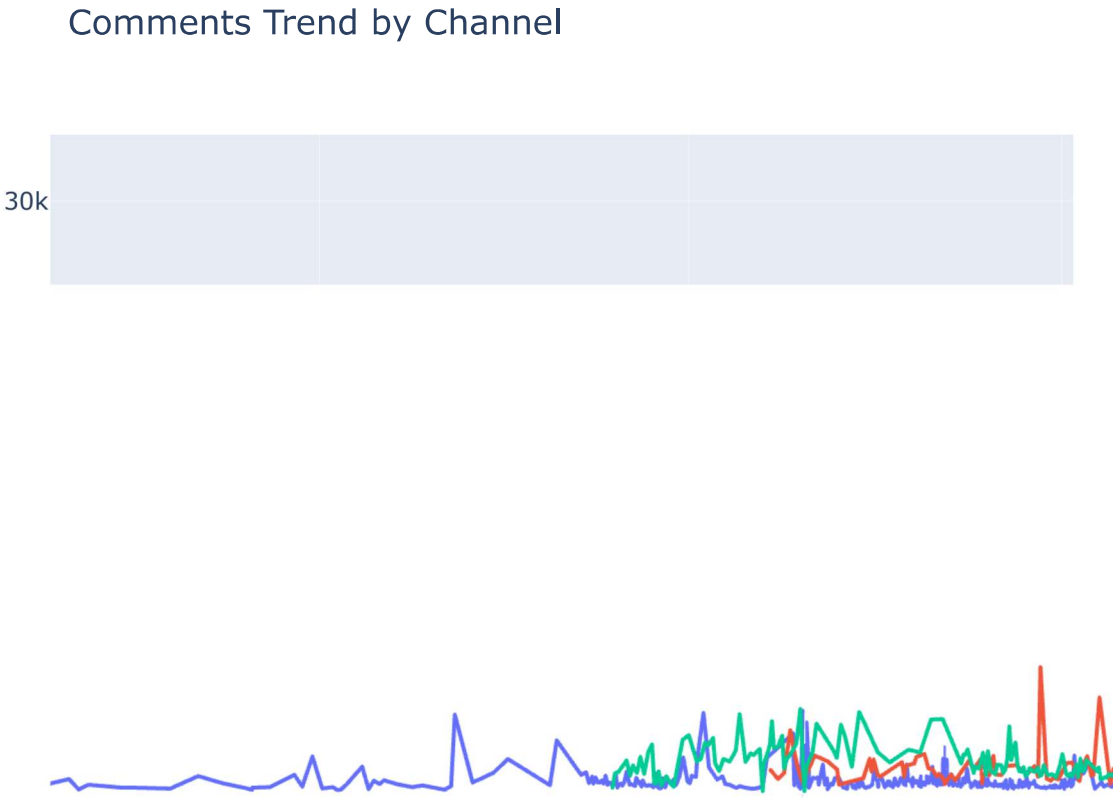
In [ ]:
```
df.columns
```

Out[ ]:
```
Index(['Date_Published', 'Title', 'Tags', 'channel', 'Comments', 'viewCount',
       'likeCount', 'favoriteCount', 'Cl_Text', 'English_Text', 'Text',
       'Genre', 'personalities_ents', 'Geographical_locations',
       'Nationalities', 'Locations', 'Engagement'],
      dtype='object')
```

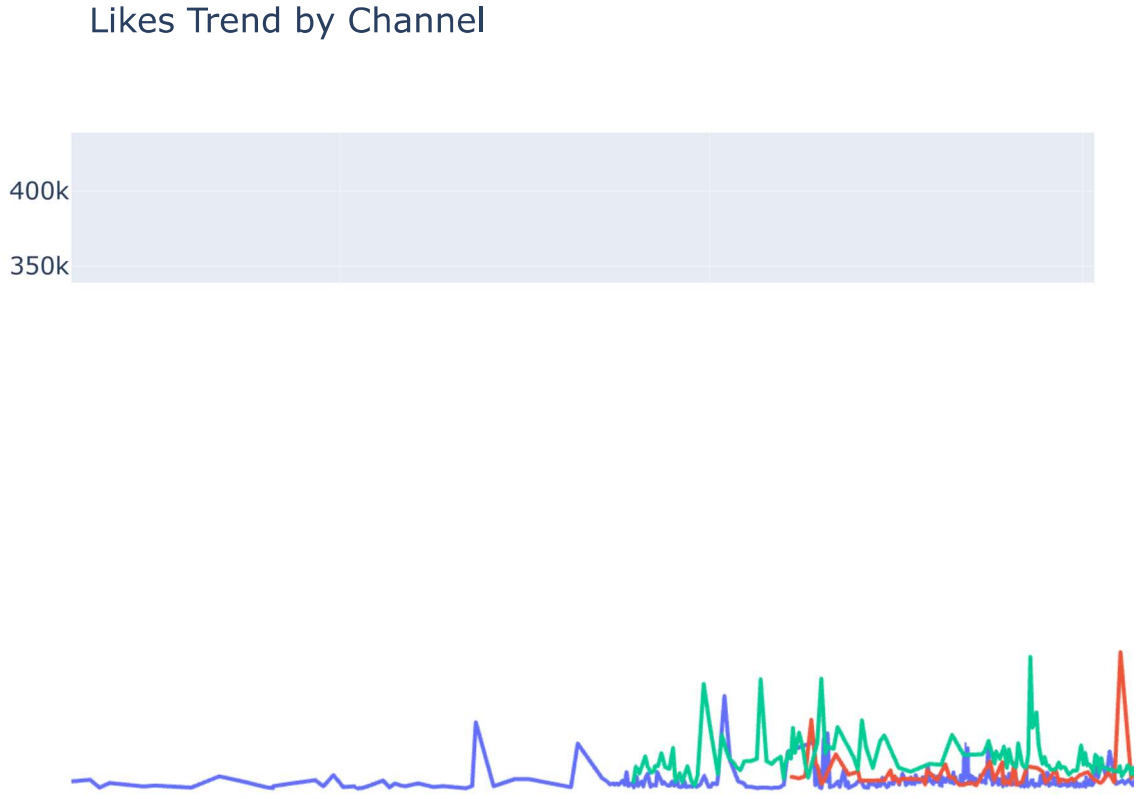
In [ ]:
```
### Engagement Trend By Date Published

# Create a line plot with different lines for each channel
fig = px.line(df, x='Date_Published', y='Engagement', color='channel', title='Engag

# Show the plot
fig.show()
```



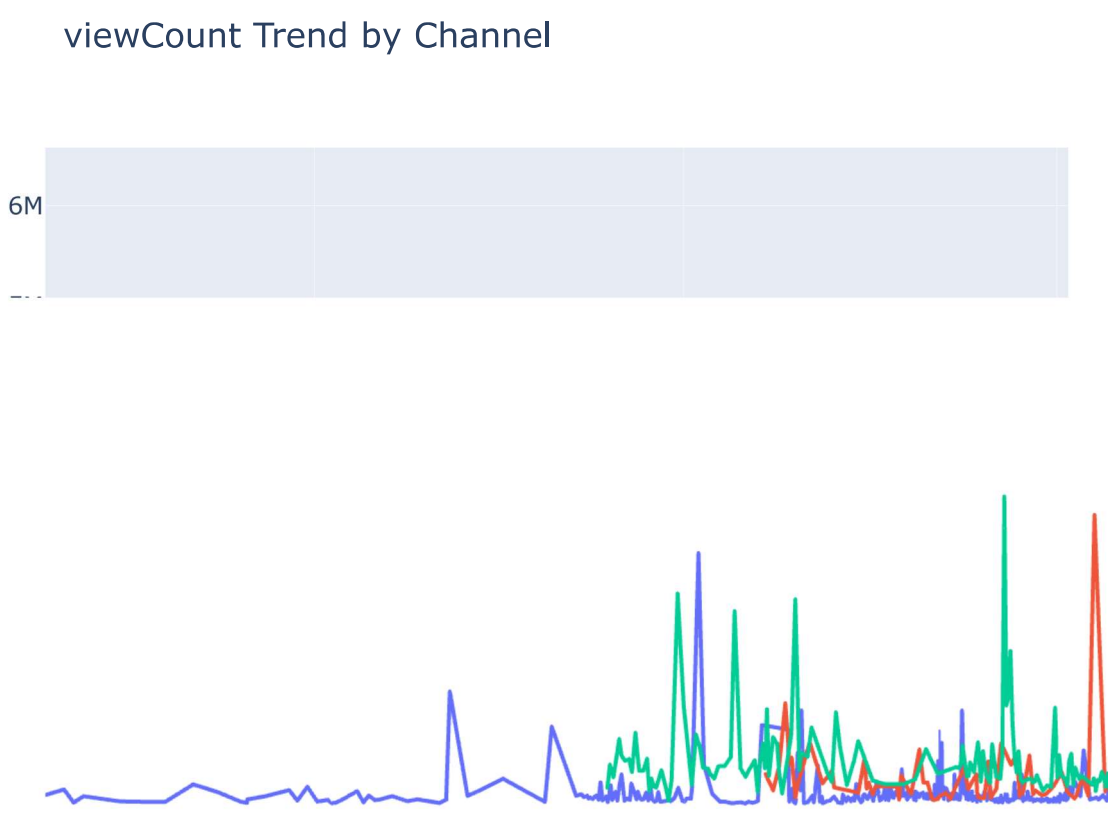
In [ ]:
```
### Comments Trend By Time

# Create a line plot with different lines for each channel
fig = px.line(df, x='Date_Published', y='Comments', color='channel', title='Comment

# Show the plot
fig.show()
```



In [ ]:
```
### Likes Trend By Time

# Create a line plot with different lines for each channel
fig = px.line(df, x='Date_Published', y='likeCount', color='channel', title='Likes

# Show the plot
fig.show()
```



In [ ]:
```
### viewCount Trend By Time

# Create a line plot with different lines for each channel
fig = px.line(df, x='Date_Published', y='viewCount', color='channel', title='viewCo

# Show the plot
fig.show()
```



In [ ]:
```
df['channel'].value_counts()
```

Out[ ]:
```
channel
Naa Anveshana            1022
Ravi Telugu Traveller     719
Uma Telugu Traveller      510
Name: count, dtype: int64
```

Top Videos

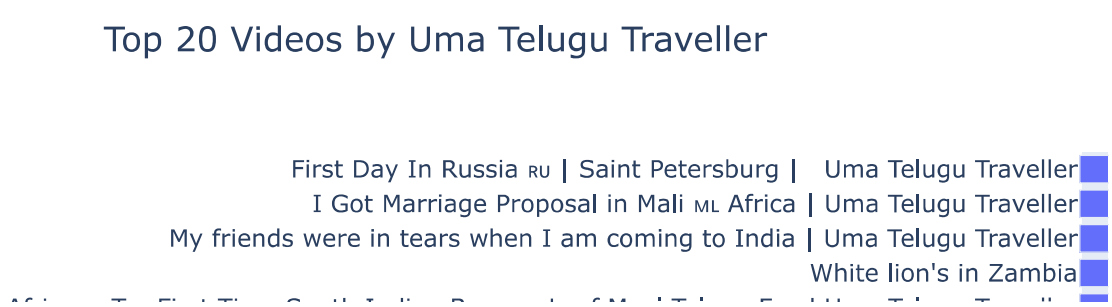
In [ ]:
```
anv = df.loc[df['channel']=='Naa Anveshana']
# Sort the DataFrame by Engagement and select top 10 videos
Anvesh_Top10_videos = anv.sort_values(by='Engagement', ascending=False).head(20)

# Create a bar plot for the top 10 videos by engagement
fig = px.bar(Anvesh_Top10_videos, x='Engagement', y='Title', color='channel',
             title='Top 20 Videos by Na Anveshana')

# Customize the layout if needed
fig.update_layout(xaxis_title='Engagement', yaxis_title='Video Title')
```

```
# Show the plot
fig.show()
```

In [ ]:
```
ravi = df.loc[df['channel']=='Ravi Telugu Traveller']
# Sort the DataFrame by Engagement and select top 10 videos
ravi_Top20_videos = ravi.sort_values(by='Engagement', ascending=False).head(20)

# Create a bar plot for the top 10 videos by engagement
fig = px.bar(ravi_Top20_videos, x='Engagement', y='Title', color='channel',
             title='Top 20 Videos by Ravi Telugu Traveller')

# Customize the layout if needed
fig.update_layout(xaxis_title='Engagement', yaxis_title='Video Title')

# Show the plot
fig.show()
```

Top 20 Videos by Ravi Telugu Traveller

In [ ]:
```
uma = df.loc[df['channel']=='Uma Telugu Traveller']
# Sort the DataFrame by Engagement and select top 10 videos
uma_Top20_videos = uma.sort_values(by='Engagement', ascending=False).head(20)

# Create a bar plot for the top 10 videos by engagement
fig = px.bar(uma_Top20_videos, x='Engagement', y='Title', color='channel',
             title='Top 20 Videos by Uma Telugu Traveller')

# Customize the layout if needed
fig.update_layout(xaxis_title='Engagement', yaxis_title='Video Title')

# Show the plot
fig.show()
```


Top 20 Videos by Uma Telugu Traveller
First Day In Russia RU | Saint Petersburg | Uma Telugu Traveller
I Got Marriage Proposal in Mali ML Africa | Uma Telugu Traveller
My friends were in tears when I am coming to India | Uma Telugu Traveller
White lion's in Zambia

In [ ]:
```
df.head()
```

Out[ ]:

	Date_Published	Title	Tags	channel	Comments	viewCount	likeCou
0	2023-08-22 02:30:05+00:00	Naa Anveshana meet up in [OBJ] Montreal and Vancouv...	['Naa Anveshana meet up in Montreal and Vancou...	Naa Anveshana	1169	333229	196
1	2023-08-19 04:52:05+00:00	English Bay Beach Vancouver | Suspension bridg...	['Canada Place', 'English Bay Beach Vancouver'...	Naa Anveshana	2006	980291	510
2	2023-08-17 03:47:03+00:00	48 hrs On Canada's Most Luxury train journey B...	['Naaanveshana', 'banff', 'british columbia', ...	Naa Anveshana	3774	1686573	745
3	2023-08-15 02:30:31+00:00	Sulphur banff gondola | things to do in banff ...	['Naaanveshana', 'Sulphur banff gondola', 'ban...	Naa Anveshana	2462	2189913	638
4	2023-08-12 12:49:37+00:00	Columbia ice field Glacier Adventure Jasper Na...	['Naaanveshana', 'athabasca falls', 'athabasca...	Naa Anveshana	3430	1481449	646

In [ ]:
```
df.columns
```

Out[ ]:
```
Index(['Date_Published', 'Title', 'Tags', 'channel', 'Comments', 'viewCount',
       'likeCount', 'favoriteCount', 'Cl_Text', 'English_Text', 'Text',
       'Genre', 'personalities_ents', 'Geographical_locations',
       'Nationalities', 'Locations', 'Engagement'],
      dtype='object')
```

Most Trended Nationalities

In [ ]:
```
# Clean the Nationalities data remove dupicates

df['Nationalities'] = df['Nationalities'].astype('str')
df['Nationalities'] = df['Nationalities'].str.strip()


## Removing duplicate words
```

```
def remove_duplicate_words(text):
    l = text.split(',')
    s = set([word.strip() for word in l])
    return ', '.join(s)
df['Nationalities'] = df['Nationalities'].apply(lambda x: remove_duplicate_words(x)

persons_df = df['Nationalities'].str.split(', ', expand=True).stack().reset_index(
                    level=1, drop=True).to_frame('Nationals')

# Join the new DataFrame with the original DataFrame
result= df.drop('Nationalities', axis=1).join(persons_df)

result.dropna(subset=['Nationals'], inplace=True)

result = result.loc[result['Nationals'] != 'nan']

result['Nationals'] = result['Nationals'].str.strip()


Nationals = result[['Date_Published', 'channel', 'Comments', 'viewCount',
       'likeCount', 'Engagement', 'Nationals']].reset_index(drop=True)
```

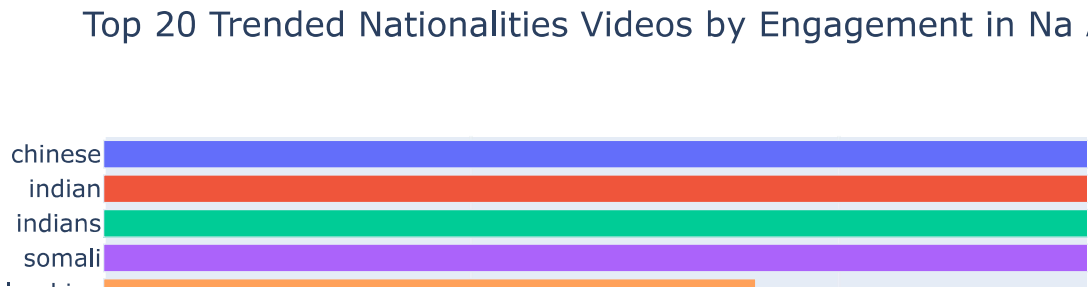
In [ ]:
```
Nationals['channel'].value_counts()
```

Out[ ]:
```
channel
Naa Anveshana            106
Ravi Telugu Traveller     89
Uma Telugu Traveller      58
Name: count, dtype: int64
```

Top Nationalities Videos in Na Anveshana

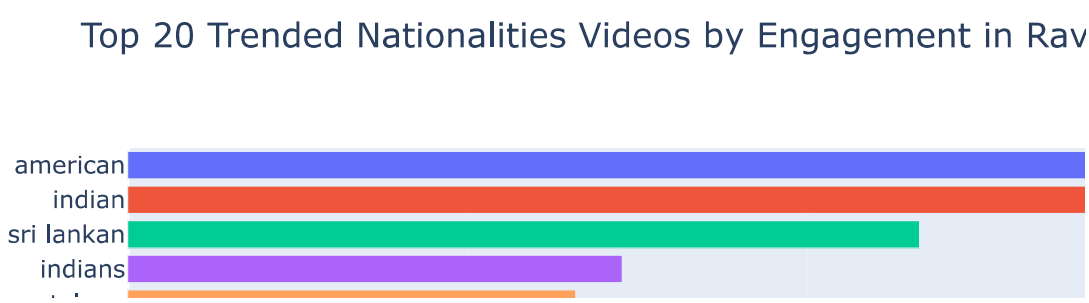
In [ ]:
```
Anveshana = Nationals.loc[Nationals['channel'] == 'Naa Anveshana']
```

In [ ]:
```
# Group by 'Nationals' and calculate sum of 'Engagement'
nationality_engagement = Anveshana.groupby('Nationals')['Engagement'].sum().reset_i

# Sort by 'Engagement' in descending order and select top 20 nationalities
top_20_nationalities = nationality_engagement.sort_values(by='Engagement', ascendin

# Create a bar plot for the top 20 trended nationalities
fig = px.bar(top_20_nationalities, y='Nationals', x='Engagement', color='Nationals'
             title='Top 20 Trended Nationalities Videos by Engagement in Na Anvesha

# Customize the layout if needed
fig.update_layout(yaxis_title='Nationality', xaxis_title='Engagement')
```



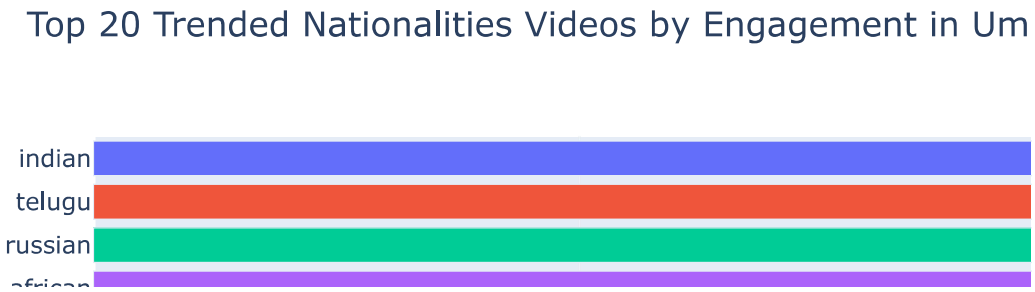
Top Nationalities Videos in Ravi Telugu Traveller

In [ ]:
```
Ravi = Nationals.loc[Nationals['channel'] == 'Ravi Telugu Traveller']

# Group by 'Nationals' and calculate sum of 'Engagement'
nationality_engagement = Ravi.groupby('Nationals')['Engagement'].sum().reset_index(

# Sort by 'Engagement' in descending order and select top 20 nationalities
top_20_nationalities = nationality_engagement.sort_values(by='Engagement', ascendin

# Create a bar plot for the top 20 trended nationalities
fig = px.bar(top_20_nationalities, y='Nationals', x='Engagement', color='Nationals'
             title='Top 20 Trended Nationalities Videos by Engagement in Ravi Telug

# Customize the layout if needed
fig.update_layout(yaxis_title='Nationality', xaxis_title='Engagement')
```



Top Nationalities Videos in Uma Telugu Traveller

In [ ]:
```
Uma = Nationals.loc[Nationals['channel'] == 'Uma Telugu Traveller']

# Group by 'Nationals' and calculate sum of 'Engagement'
nationality_engagement = Uma.groupby('Nationals')['Engagement'].sum().reset_index()

# Sort by 'Engagement' in descending order and select top 20 nationalities
top_20_nationalities = nationality_engagement.sort_values(by='Engagement', ascendin

# Create a bar plot for the top 20 trended nationalities
fig = px.bar(top_20_nationalities, y='Nationals', x='Engagement', color='Nationals'
             title='Top 20 Trended Nationalities Videos by Engagement in Uma Telugu

# Customize the layout if needed
fig.update_layout(yaxis_title='Nationality', xaxis_title='Engagement')
```



In [ ]:
```
df.columns
```

Out[ ]:
```
Index(['Date_Published', 'Title', 'Tags', 'channel', 'Comments', 'viewCount',
       'likeCount', 'favoriteCount', 'Cl_Text', 'English_Text', 'Text',
       'Genre', 'personalities_ents', 'Geographical_locations',
       'Nationalities', 'Locations', 'Engagement'],
      dtype='object')
```

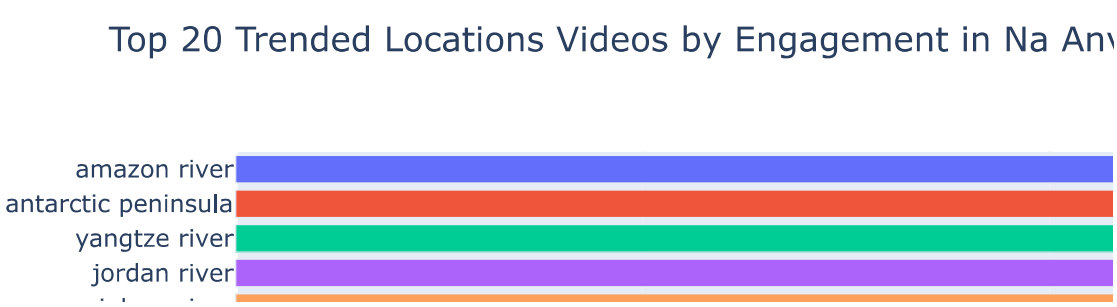
Top Locations in YouTube Videos

In [ ]:
```
# Clean the Nationalities data remove dupicates

df['Locations'] = df['Locations'].astype('str')
df['Locations'] = df['Locations'].str.strip()


## Removing duplicate words
def remove_duplicate_words(text):
    l = text.split(',')
    s = set([word.strip() for word in l])
    return ', '.join(s)
df['Locations'] = df['Locations'].apply(lambda x: remove_duplicate_words(x))
```

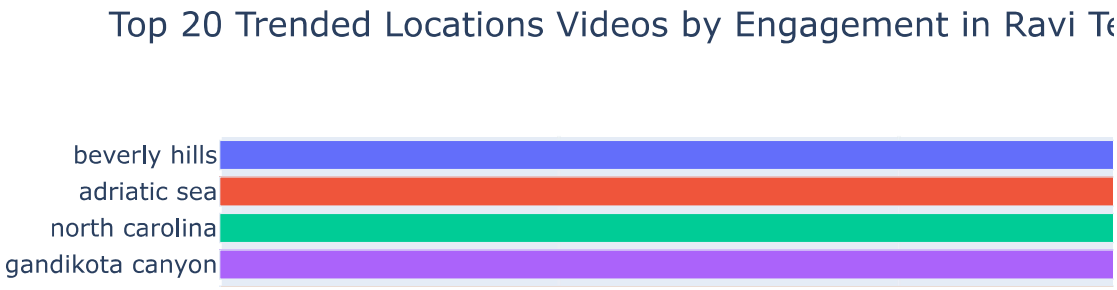
```
persons_df = df['Locations'].str.split(', ', expand=True).stack().reset_index(
                    level=1, drop=True).to_frame('Location')

# Join the new DataFrame with the original DataFrame
result= df.drop('Locations', axis=1).join(persons_df)

result.dropna(subset=['Location'], inplace=True)

result = result.loc[result['Location'] != 'nan']

result['Location'] = result['Location'].str.strip()


Location  = result[['Date_Published', 'channel', 'Comments', 'viewCount',
       'likeCount', 'Engagement', 'Location']].reset_index(drop=True)

Location['channel'].value_counts()
```

Out[ ]:
```
channel
Naa Anveshana            75
Uma Telugu Traveller     56
Ravi Telugu Traveller    21
Name: count, dtype: int64
```

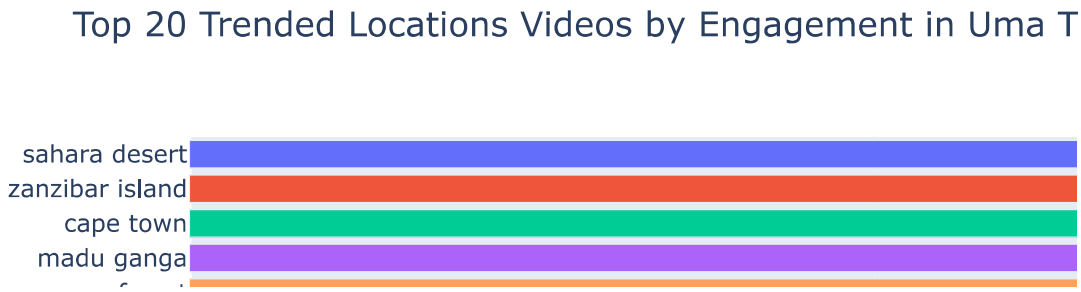
In [ ]:
```
Anveshana = Location.loc[Location['channel'] == 'Naa Anveshana']

# Group by 'Nationals' and calculate sum of 'Engagement'
nationality_engagement = Anveshana.groupby('Location')['Engagement'].sum().reset_in

# Sort by 'Engagement' in descending order and select top 20 nationalities
top_20_nationalities = nationality_engagement.sort_values(by='Engagement', ascendin

# Create a bar plot for the top 20 trended nationalities
fig = px.bar(top_20_nationalities, y='Location', x='Engagement', color='Location',
             title='Top 20 Trended Locations Videos by Engagement in Na Anveshana')

# Customize the layout if needed
fig.update_layout(yaxis_title='Areas', xaxis_title='Engagement')
```



In [ ]:
```
Ravi = Location.loc[Location['channel'] == 'Ravi Telugu Traveller']

# Group by 'Nationals' and calculate sum of 'Engagement'
nationality_engagement = Ravi.groupby('Location')['Engagement'].sum().reset_index()

# Sort by 'Engagement' in descending order and select top 20 nationalities
top_20_nationalities = nationality_engagement.sort_values(by='Engagement', ascendin

# Create a bar plot for the top 20 trended nationalities
fig = px.bar(top_20_nationalities, y='Location', x='Engagement', color='Location',
             title='Top 20 Trended Locations Videos by Engagement in Ravi Telugu Tr

# Customize the layout if needed
fig.update_layout(yaxis_title='Areas', xaxis_title='Engagement')
```



In [ ]:
```
Uma = Location.loc[Location['channel'] == 'Uma Telugu Traveller']

# Group by 'Nationals' and calculate sum of 'Engagement'
nationality_engagement = Uma.groupby('Location')['Engagement'].sum().reset_index()

# Sort by 'Engagement' in descending order and select top 20 nationalities
top_20_nationalities = nationality_engagement.sort_values(by='Engagement', ascendin

# Create a bar plot for the top 20 trended nationalities
fig = px.bar(top_20_nationalities, y='Location', x='Engagement', color='Location',
             title='Top 20 Trended Locations Videos by Engagement in Uma Telugu Tra

# Customize the layout if needed
fig.update_layout(yaxis_title='Areas', xaxis_title='Engagement')
```



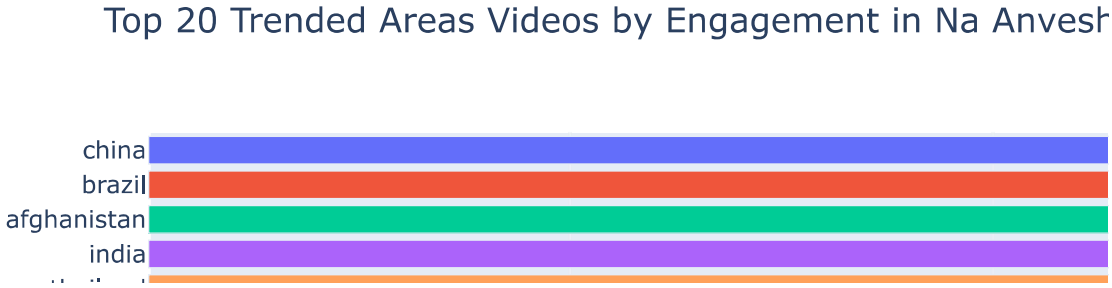
Top trended Geographical_locations in YouTube Videos

In [ ]:
```
# Clean the Nationalities data remove dupicates

df['Geographical_locations'] = df['Geographical_locations'].astype('str')
df['Geographical_locations'] = df['Geographical_locations'].str.strip()


## Removing duplicate words
def remove_duplicate_words(text):
    l = text.split(',')
    s = set([word.strip() for word in l])
    return ', '.join(s)
df['Geographical_locations'] = df['Geographical_locations'].apply(lambda x: remove_

persons_df = df['Geographical_locations'].str.split(', ', expand=True).stack().rese
                    level=1, drop=True).to_frame('Area')

# Join the new DataFrame with the original DataFrame
result= df.drop('Geographical_locations', axis=1).join(persons_df)

result.dropna(subset=['Area'], inplace=True)
```

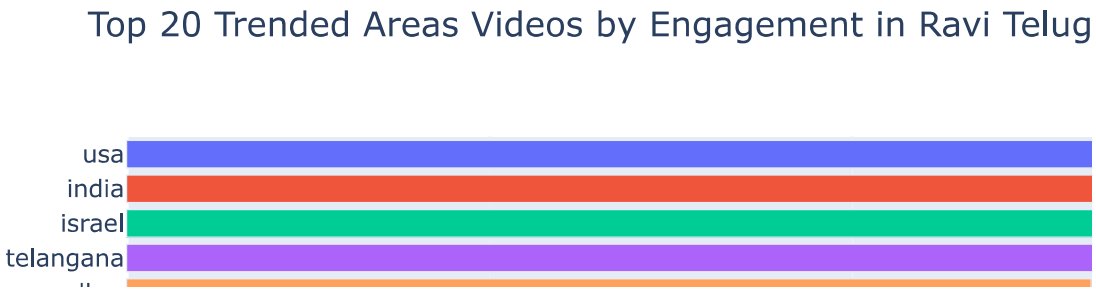
```
result = result.loc[result['Area'] != 'nan']

result['Area'] = result['Area'].str.strip()


Area = result[['Date_Published', 'channel', 'Comments', 'viewCount',
       'likeCount', 'Engagement', 'Area']].reset_index(drop=True)

Area['channel'].value_counts()
```

Out[ ]:
```
channel
Naa Anveshana            910
Ravi Telugu Traveller    590
Uma Telugu Traveller     579
Name: count, dtype: int64
```

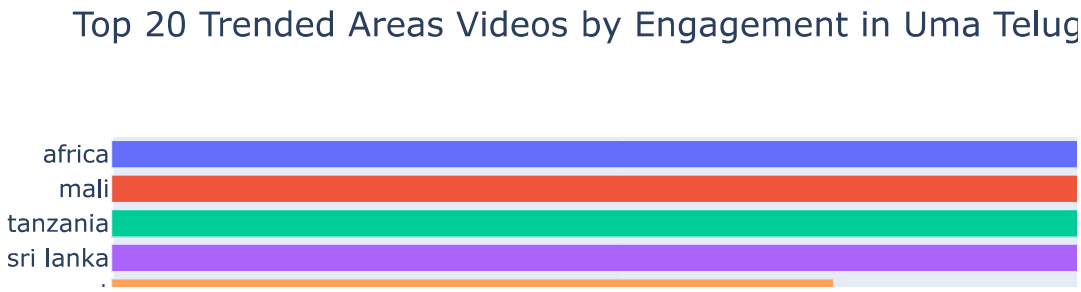
In [ ]:
```
Anveshana = Area.loc[Area['channel'] == 'Naa Anveshana']

# Group by 'Nationals' and calculate sum of 'Engagement'
nationality_engagement = Anveshana.groupby('Area')['Engagement'].sum().reset_index(

# Sort by 'Engagement' in descending order and select top 20 nationalities
top_20_nationalities = nationality_engagement.sort_values(by='Engagement', ascendin

# Create a bar plot for the top 20 trended nationalities
fig = px.bar(top_20_nationalities, y='Area', x='Engagement', color='Area',
             title='Top 20 Trended Areas Videos by Engagement in Na Anveshana')

# Customize the layout if needed
fig.update_layout(yaxis_title='Areas', xaxis_title='Engagement')
```



In [ ]:
```
Ravi = Area.loc[Area['channel'] == 'Ravi Telugu Traveller']

# Group by 'Nationals' and calculate sum of 'Engagement'
nationality_engagement = Ravi.groupby('Area')['Engagement'].sum().reset_index()

# Sort by 'Engagement' in descending order and select top 20 nationalities
top_20_nationalities = nationality_engagement.sort_values(by='Engagement', ascendin

# Create a bar plot for the top 20 trended nationalities
fig = px.bar(top_20_nationalities, y='Area', x='Engagement', color='Area',
             title='Top 20 Trended Areas Videos by Engagement in Ravi Telugu Travel

# Customize the layout if needed
fig.update_layout(yaxis_title='Areas', xaxis_title='Engagement')
```



In [ ]:
```
Uma = Area.loc[Area['channel'] == 'Uma Telugu Traveller']

# Group by 'Nationals' and calculate sum of 'Engagement'
nationality_engagement = Uma.groupby('Area')['Engagement'].sum().reset_index()

# Sort by 'Engagement' in descending order and select top 20 nationalities
top_20_nationalities = nationality_engagement.sort_values(by='Engagement', ascendin

# Create a bar plot for the top 20 trended nationalities
fig = px.bar(top_20_nationalities, y='Area', x='Engagement', color='Area',
             title='Top 20 Trended Areas Videos by Engagement in Uma Telugu Travell

# Customize the layout if needed
fig.update_layout(yaxis_title='Areas', xaxis_title='Engagement')
```


Top 20 Trended Areas Videos by Engagement in Uma Telug
africa
mali
tanzania
sri lanka