Hongfan Lu EDA

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
df = pd.read_csv('../data/yt_labeled.csv')
```

In [3]:
```
df.shape
```

Out[3]:
```
(138996, 33)
```

In [4]:
```
df.columns
```

Out[4]:
```
Index(['channel_id', 'channel_name', 'video_id', 'video_title',
       'video_creation_time', 'video_description', 'video_tags',
       'video_viewcount', 'video_likecount', 'video_commentcount',
       'comment_id', 'comment_author_id', 'comment_text', 'comment_time',
       'comment_likecount', 'comment_replycount', 'genre', 'cleaned_comment',
       'tokenized_comment', 'toxicity', 'severe_toxicity', 'identity_attack',
       'insult', 'profanity', 'threat', 'neg', 'neu', 'pos', 'compound',
       'polarity', 'subjectivity', 'negative_emotion', 'positive_emotion'],
      dtype='object')
```

In [5]:
```
simple_df = df[['channel_name','video_title', 'video_creation_time',
                 'video_viewcount','video_likecount', 'video_commentcount','comment_time',
         'comment_likecount', 'comment_replycount', 'genre', 'cleaned_comment',
         'tokenized_comment', 'toxicity', 'severe_toxicity', 'identity_attack',
         'insult', 'profanity', 'threat', 'neg', 'neu', 'pos', 'compound',
         'polarity', 'subjectivity', 'negative_emotion', 'positive_emotion']]
```

In [6]:
```
all_games = [
    "call of duty", "gta", "the last of us", "god of war", "batman",
    "red dead redemption", "assassin's creed", "star wars jedi",
    "resident evil", "cyberpunk", "fallout", "tomb raider", "elden ring",
    "minecraft", "pokemon go", "just dance", "it takes two", "uncharted",
    "brawl stars"]
```

In [7]:
```
simple_df['video_title_lower'] = simple_df['video_title'].str.lower()
simple_df['game'] = simple_df['video_title_lower'].apply(lambda title: ', '.join([keyword
simple_df.drop(columns=['video_title_lower'], inplace=True)
```

```
/var/folders/_1/f4gv917x3jv_z8t21wgjlcrw0000gn/T/ipykernel_25471/3244421465.py:1: Setting
WithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_g
uide/indexing.html#returning-a-view-versus-a-copy
  simple_df['video_title_lower'] = simple_df['video_title'].str.lower()
/var/folders/_1/f4gv917x3jv_z8t21wgjlcrw0000gn/T/ipykernel_25471/3244421465.py:2: Setting
WithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_g
uide/indexing.html#returning-a-view-versus-a-copy
  simple_df['game'] = simple_df['video_title_lower'].apply(lambda title: ', '.join([keywo
rd for keyword in all_games if keyword in title]))
/var/folders/_1/f4gv917x3jv_z8t21wgjlcrw0000gn/T/ipykernel_25471/3244421465.py:3: Setting
WithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_g
uide/indexing.html#returning-a-view-versus-a-copy
  simple_df.drop(columns=['video_title_lower'], inplace=True)
```

In [8]:
```
simple_df.head(2)
```

Out[8]:

	channel_name	video_title	video_creation_time	video_viewcount	video_likecount	video_commentcount
0	PewDiePie	I ~~tried to~~ beat Elden Ring Without Dyi...	2022-04-30 16:40:18+00:00	11540558.0	473052.0	15129.0
1	PewDiePie	I ~~tried to~~ beat Elden Ring Without Dyi...	2022-04-30 16:40:18+00:00	11540558.0	473052.0	15129.0

2 rows × 27 columns

In [9]:
```
sentiment_measures = ['toxicity','severe_toxicity', 'identity_attack','insult', 'profanit
```

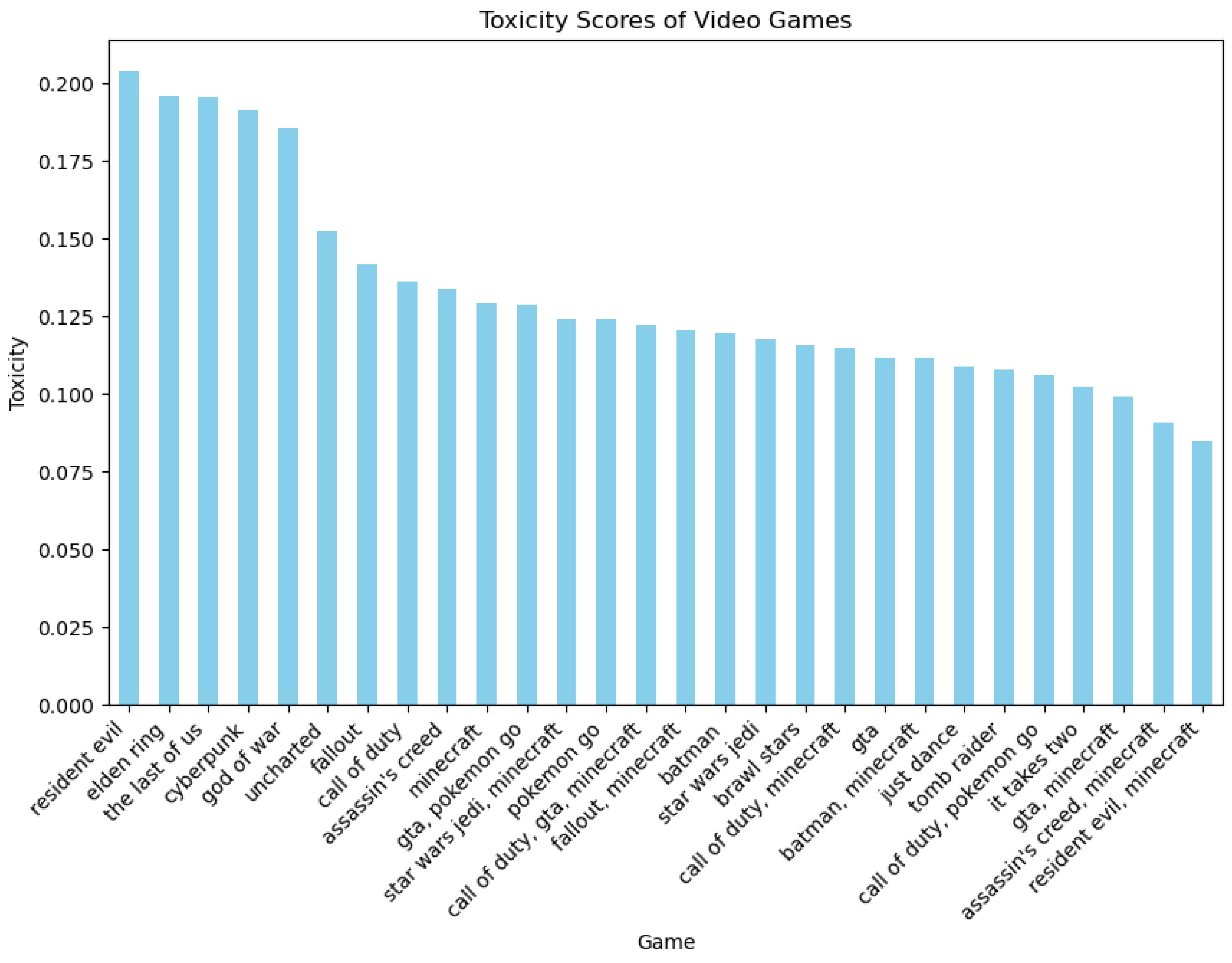
Toxicity by Game

In [10]:
```
game_toxicity = simple_df.groupby('game')[sentiment_measures].mean().sort_values(by = 'to
```

In [11]:
```
game_toxicity_plot = game_toxicity['toxicity']

plt.figure(figsize = (10, 6))
ax = game_toxicity_plot.plot(kind='bar', color='skyblue')
# Adding labels and title
plt.xlabel('Game')
ax.set_xticklabels(ax.get_xticklabels(), rotation=45, ha='right')
plt.ylabel('Toxicity')
plt.title('Toxicity Scores of Video Games')
```

Out[11]:
```
Text(0.5, 1.0, 'Toxicity Scores of Video Games')
```



```
In [12]: # simple_df.groupby(['genre','game']).mean()
```

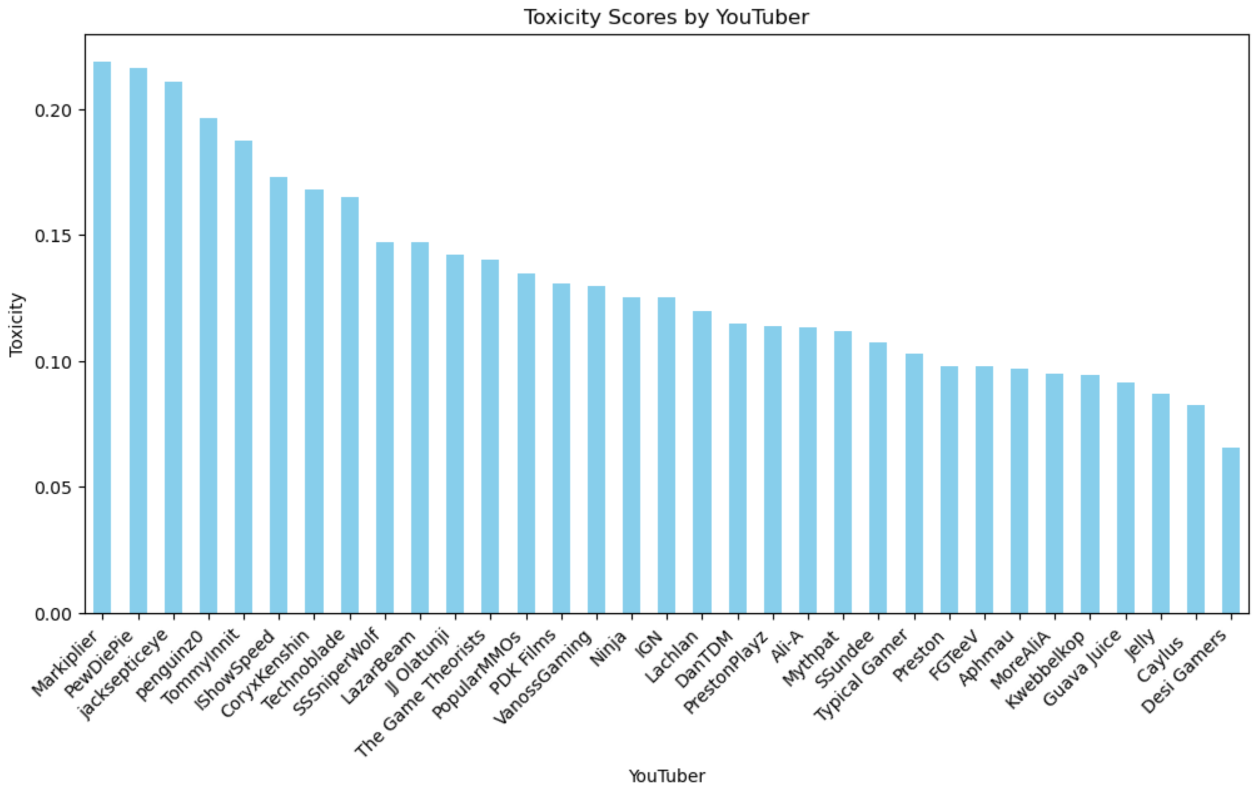
Toxicity by Channel

```
In [13]: channel_toxicity = simple_df.groupby('channel_name')[sentiment_measures].mean().sort_valu
```

```
In [14]: channel_toxicity_plot = channel_toxicity['toxicity']

         plt.figure(figsize = (12, 6))
         ax = channel_toxicity_plot.plot(kind='bar', color='skyblue')
         # Adding labels and title
         plt.xlabel('YouTuber')
         ax.set_xticklabels(ax.get_xticklabels(), rotation=45, ha='right')
         plt.ylabel('Toxicity')
         plt.title('Toxicity Scores by YouTuber')
```

```
Out[14]: Text(0.5, 1.0, 'Toxicity Scores by YouTuber')
```



Investigating Why different YouTuber has different Toxicity Means?

In [15]:
```
# Pivot to calculate unique count of video titles for each channel and genre
count_vid_by_channel_genre = simple_df.pivot_table(index='channel_name', columns='genre',

# Rename columns for clarity
count_vid_by_channel_genre.rename(columns={'action': 'action_count', 'non-action': 'non_a
```

In [16]:
```
count_vid_by_channel_genre.head(3)
```

Out[16]:

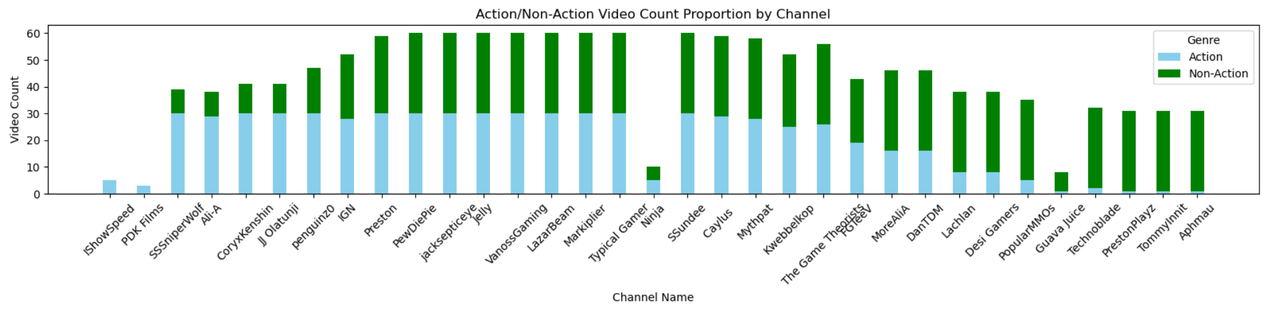
genre	channel_name	action_count	non_action_count
0	Ali-A	29	9
1	Aphmau	1	30
2	Caylus	29	30

In [24]:
```
plt.figure(figsize=(16, 4))

# Plot action counts
plt.bar(count_vid_by_channel_genre['channel_name'], count_vid_by_channel_genre['action_co

# Plot non-action counts
plt.bar(count_vid_by_channel_genre['channel_name'], count_vid_by_channel_genre['non_actio

# Adding labels and title
plt.xlabel('Channel Name')
plt.ylabel('Video Count')
plt.title('Action/Non-Action Video Count Proportion by Channel')
plt.legend(title='Genre')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



In [18]:
```
count_vid_by_channel_genre['action_video_%'] = round(100*(count_vid_by_channel_genre['act
```

In [19]:
```
count_vid_by_channel_genre = count_vid_by_channel_genre.merge(channel_toxicity['toxicity'
```

In [20]:
```
count_vid_by_channel_genre = count_vid_by_channel_genre.sort_values(by = 'action_video_%'
```

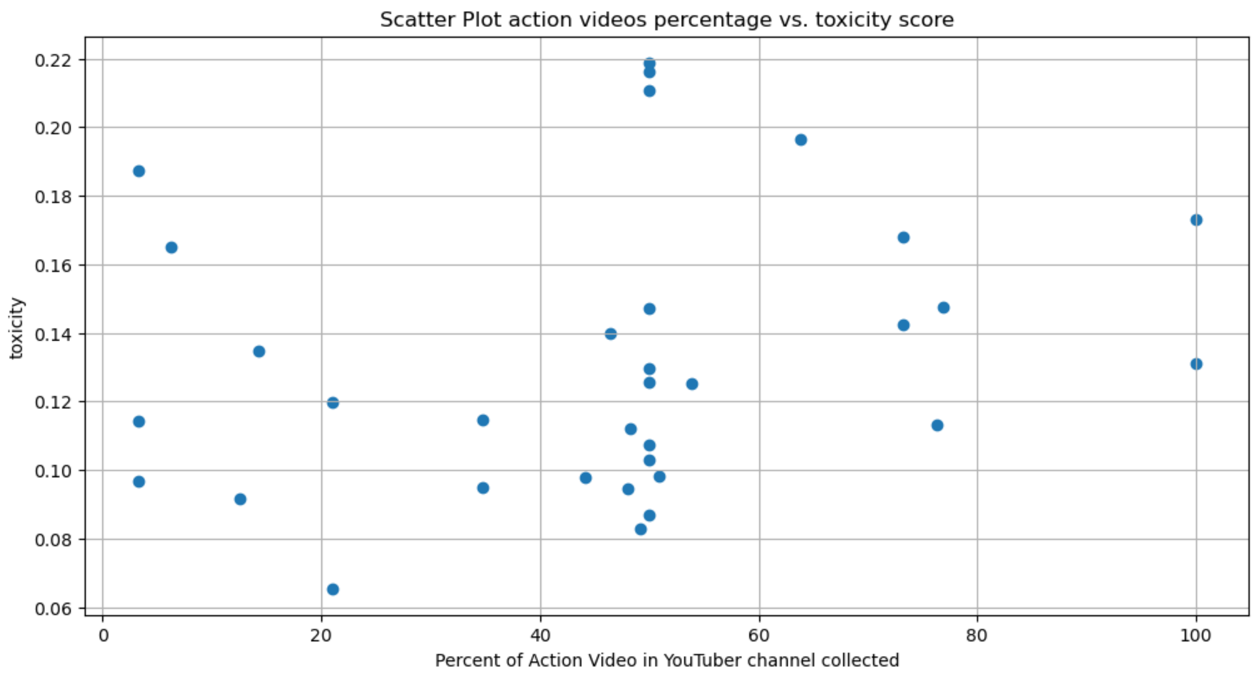
In [21]:
```
count_vid_by_channel_genre
```

Out[21]:

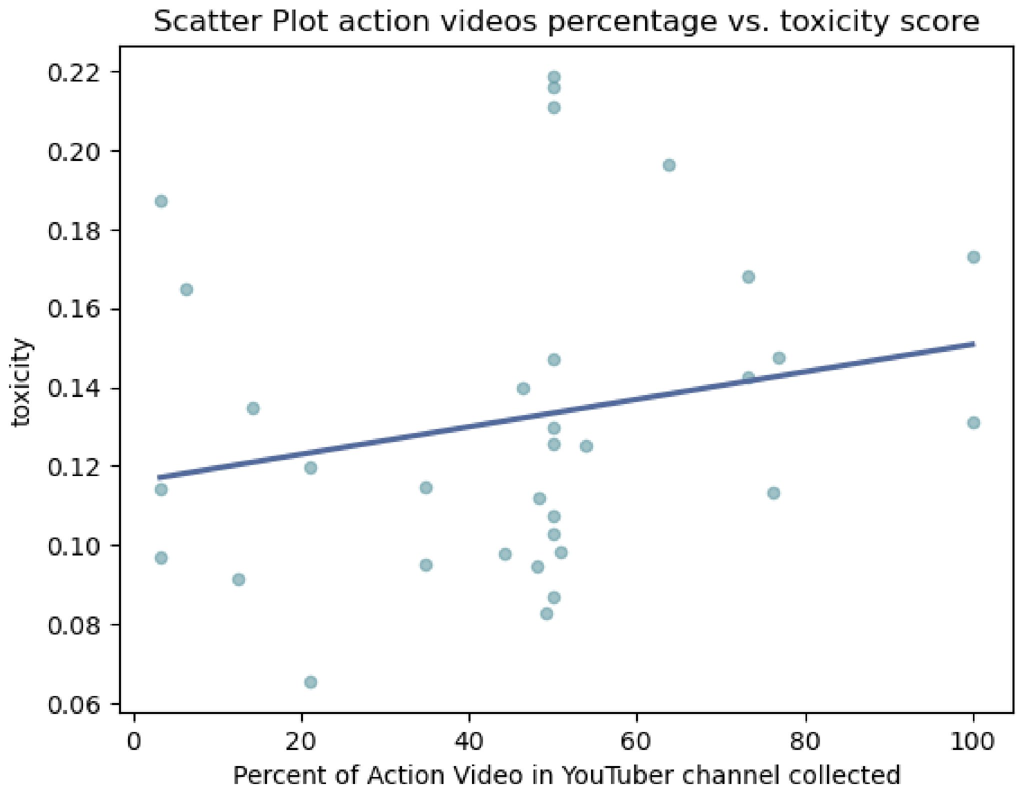
	channel_name	action_count	non_action_count	action_video_%	toxicity
0	IShowSpeed	5	0	100.00	0.173149
1	PDK Films	3	0	100.00	0.131037
2	SSSniperWolf	30	9	76.92	0.147461
3	Ali-A	29	9	76.32	0.113281
4	CoryxKenshin	30	11	73.17	0.168041
5	JJ Olatunji	30	11	73.17	0.142374
6	penguinz0	30	17	63.83	0.196428
7	IGN	28	24	53.85	0.125393
8	Preston	30	29	50.85	0.098132
9	PewDiePie	30	30	50.00	0.216254
10	jacksepticeye	30	30	50.00	0.210827
11	Jelly	30	30	50.00	0.086814
12	VanossGaming	30	30	50.00	0.129787
13	LazarBeam	30	30	50.00	0.147317
14	Markiplier	30	30	50.00	0.218835
15	Typical Gamer	30	30	50.00	0.102998
16	Ninja	5	5	50.00	0.125484
17	SSundee	30	30	50.00	0.107337
18	Caylus	29	30	49.15	0.082752
19	Mythpat	28	30	48.28	0.112104
20	Kwebbelkop	25	27	48.08	0.094685
21	The Game Theorists	26	30	46.43	0.140013
22	FGTeeV	19	24	44.19	0.097934
23	MoreAliA	16	30	34.78	0.094854
24	DanTDM	16	30	34.78	0.114798
25	Lachlan	8	30	21.05	0.119786
26	Desi Gamers	8	30	21.05	0.065423
27	PopularMMOs	5	30	14.29	0.134893
28	Guava Juice	1	7	12.50	0.091561
29	Technoblade	2	30	6.25	0.164925
30	PrestonPlayz	1	30	3.23	0.114104
31	TommyInnit	1	30	3.23	0.187283
32	Aphmau	1	30	3.23	0.096814

In [22]:

```python
# Scatter plot
plt.figure(figsize=(12, 6))
plt.scatter(count_vid_by_channel_genre['action_video_%'], count_vid_by_channel_genre['tox
plt.xlabel('Percent of Action Video in YouTuber channel collected')
plt.ylabel('toxicity')
plt.title('Scatter Plot action videos percentage vs. toxicity score')
plt.grid(True)
plt.show()
```



```
In [23]: sns.regplot(x = 'action_video_%', y = 'toxicity', data = count_vid_by_channel_genre, ci =
                     line_kws = {"color": "#526A9B"},
                     scatter_kws = {"s": 20, "color": "#619AA2", "alpha": 0.6})
         plt.xlabel('Percent of Action Video in YouTuber channel collected')
         plt.ylabel('toxicity')
         plt.title('Scatter Plot action videos percentage vs. toxicity score')
         # plt.grid(True)
         plt.show()
```



```
In [ ]:
```