In [1]:
```
import pandas as pd
import seaborn as sns
```

In [2]:
```
data = pd.read_csv(r"C:\Users\Praveen T\Downloads\Twitter_cleaned.csv")
```

In [3]:
```
data
```

Out[3]:

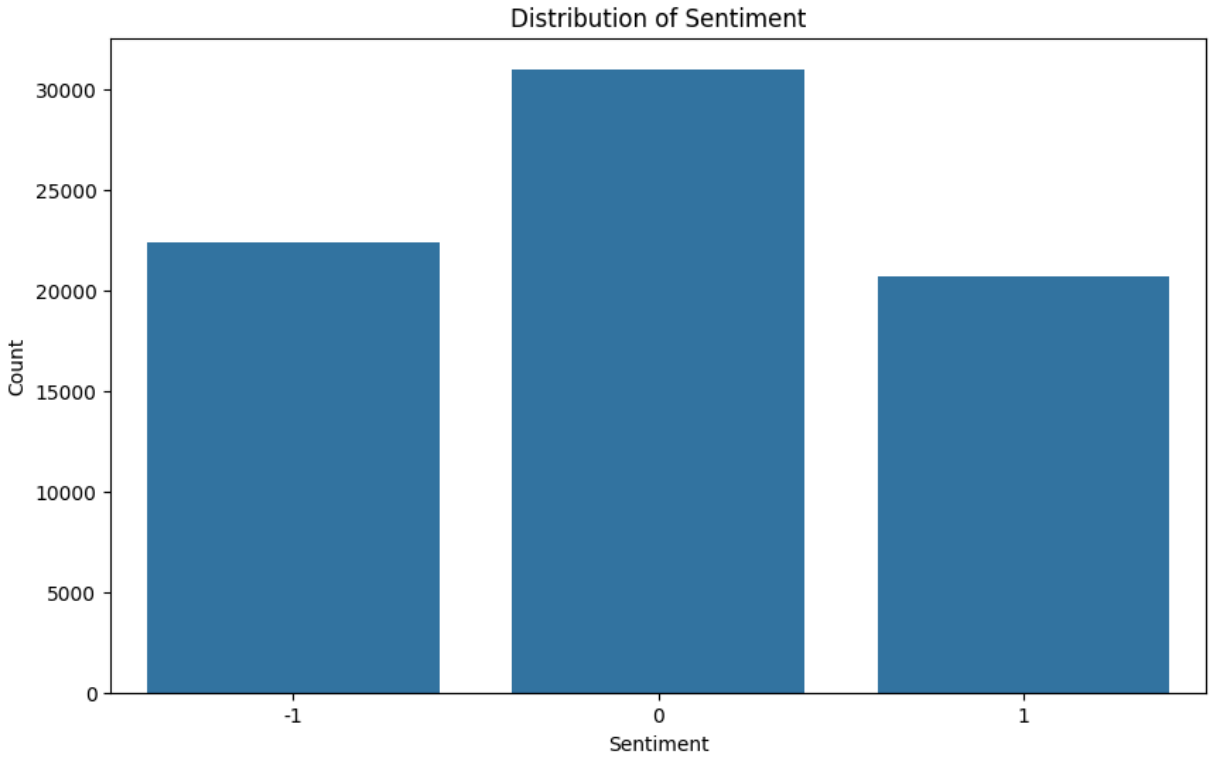
	Unnamed: 0	Sentiment	Tweet_word_count	Tweet_char_count	Tweet_clean	ApexLegends	Assa
0	0	1	11	43	im getting borderland murder	0	
1	1	1	12	40	coming border kill	0	
2	2	1	10	41	im getting borderland kill	0	
3	3	1	10	42	im coming borderland murder	0	
4	4	1	12	46	im getting borderland murder	0	
...	...	...	...	...	...	...	
73991	74677	1	26	103	realized window partition mac like year behind...	0	
73992	74678	1	23	95	realized mac window partition year behind nvid...	0	
73993	74679	1	25	101	realized window partition mac year behind nvid...	0	
73994	74680	1	32	128	realized window partition mac like year behind...	0	
73995	74681	1	25	95	like window partition mac like year behind dri...	0	

73996 rows × 36 columns

In [13]:
```
# Analyze the distribution of sentiment
```

In [14]:
```
import matplotlib.pyplot as plt
import seaborn as sns
plt.figure(figsize=(10, 6))
```

```
sns.countplot(data=data, x='Sentiment')
plt.title('Distribution of Sentiment')
plt.xlabel('Sentiment')
plt.ylabel('Count')
plt.show()
```



In [15]:
```
# Calculate the frequency of each topic
```

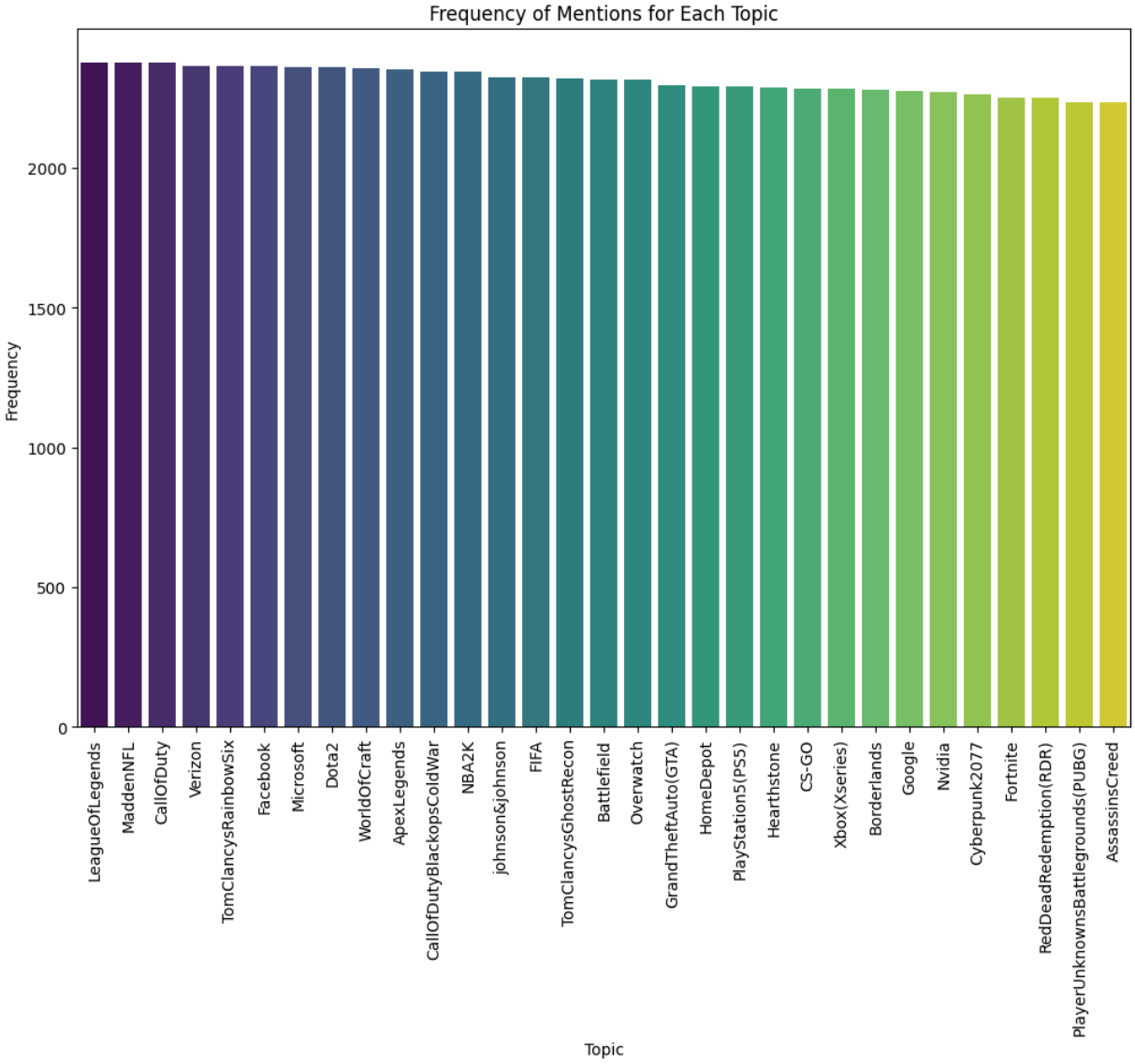
In [16]:
```
topic_columns = [col for col in data.columns if col not in ['Unnamed: 0', 'Sentiment'

topic_frequencies = data[topic_columns].sum().sort_values(ascending=False)

# Display the top topics
print(topic_frequencies)
```

```
LeagueOfLegends                      2377
MaddenNFL                            2377
CallOfDuty                           2376
Verizon                              2365
TomClancysRainbowSix                 2364
Facebook                             2362
Microsoft                            2361
Dota2                                2359
WorldOfCraft                         2357
ApexLegends                          2353
CallOfDutyBlackopsColdWar            2343
NBA2K                                2343
johnson&johnson                      2324
FIFA                                 2324
TomClancysGhostRecon                 2321
Battlefield                          2316
Overwatch                            2316
GrandTheftAuto(GTA)                  2293
HomeDepot                            2292
PlayStation5(PS5)                    2291
Hearthstone                          2286
CS-GO                                2284
Xbox(Xseries)                        2283
Borderlands                          2280
Google                               2274
Nvidia                               2271
Cyberpunk2077                        2262
Fortnite                             2249
RedDeadRedemption(RDR)               2249
PlayerUnknownsBattlegrounds(PUBG)    2234
AssassinsCreed                       2234
dtype: int64
```

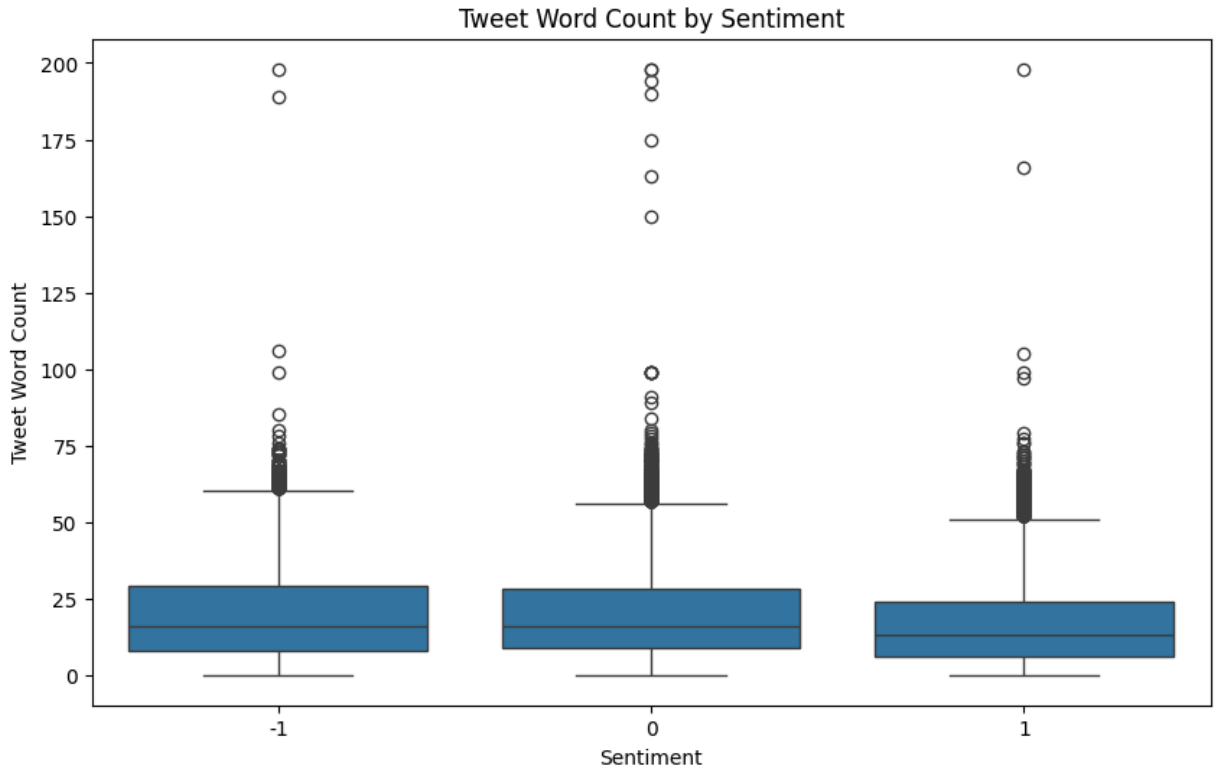
In [7]:

```
# Plot the frequency of mentions for each topic
plt.figure(figsize=(12, 8))
sns.barplot(x=topic_frequencies.index, y=topic_frequencies.values, palette='viridis')
plt.title('Frequency of Mentions for Each Topic')
plt.xlabel('Topic')
plt.ylabel('Frequency')
plt.xticks(rotation=90)
plt.show()
```

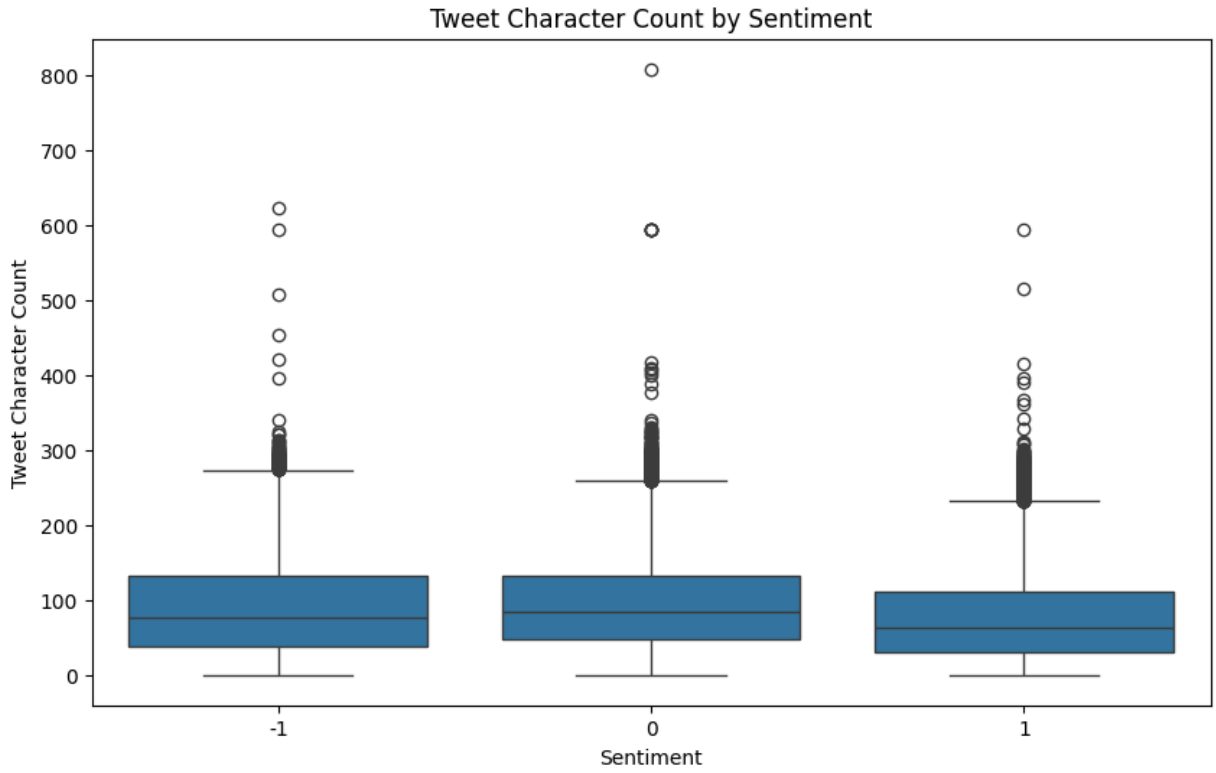
```
C:\Users\Praveen T\AppData\Local\Temp\ipykernel_15984\1691019718.py:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.
0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(x=topic_frequencies.index, y=topic_frequencies.values, palette='viridi
s')
```



In [10]:
```
# Sentiment vs. Tweet word count
plt.figure(figsize=(10, 6))
sns.boxplot(data=data, x='Sentiment', y='Tweet_word_count')
plt.title('Tweet Word Count by Sentiment')
plt.xlabel('Sentiment')
plt.ylabel('Tweet Word Count')
plt.show()
```

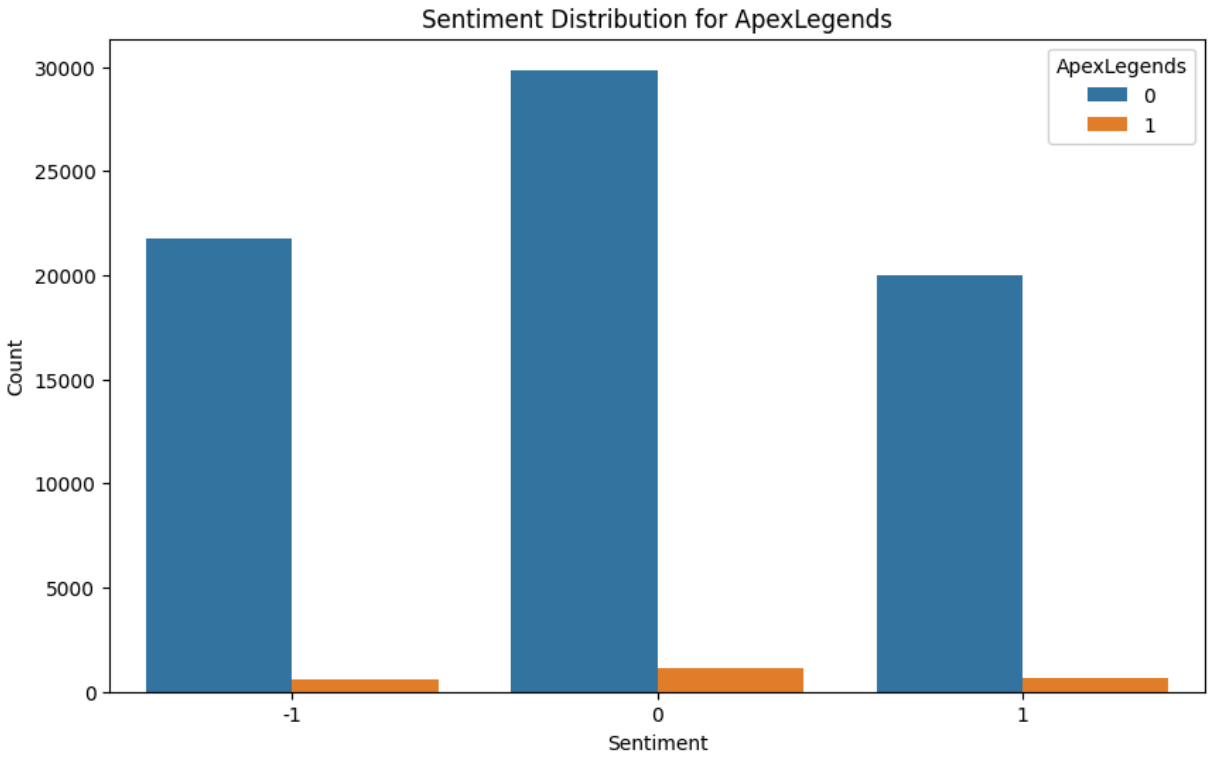


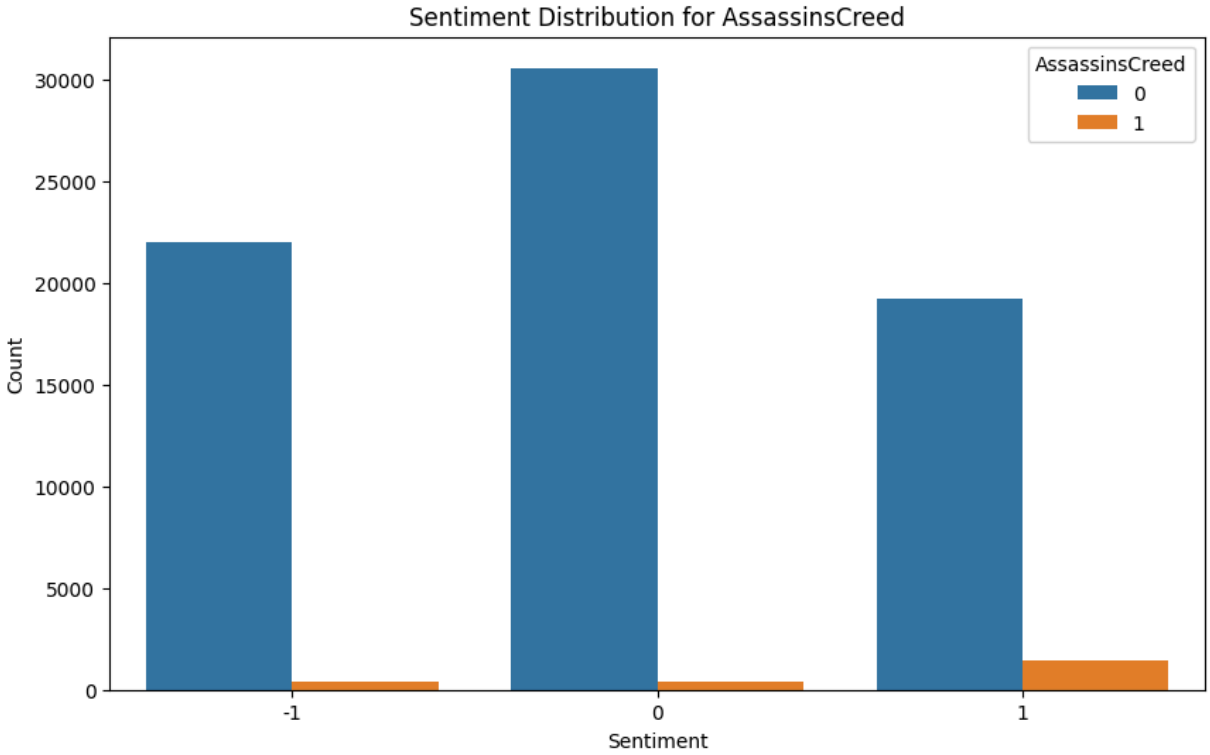
In [11]:
```
# Sentiment vs. Tweet character count
plt.figure(figsize=(10, 6))
sns.boxplot(data=data, x='Sentiment', y='Tweet_char_count')
plt.title('Tweet Character Count by Sentiment')
plt.xlabel('Sentiment')
plt.ylabel('Tweet Character Count')
plt.show()
```

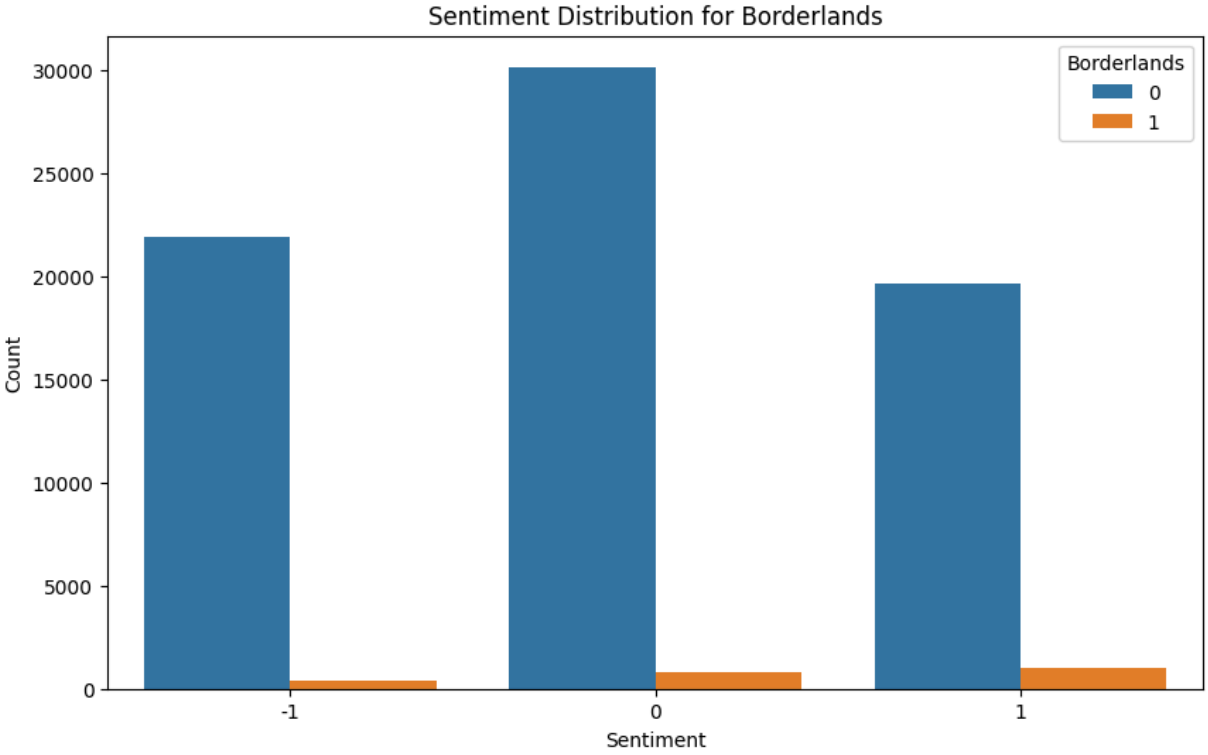


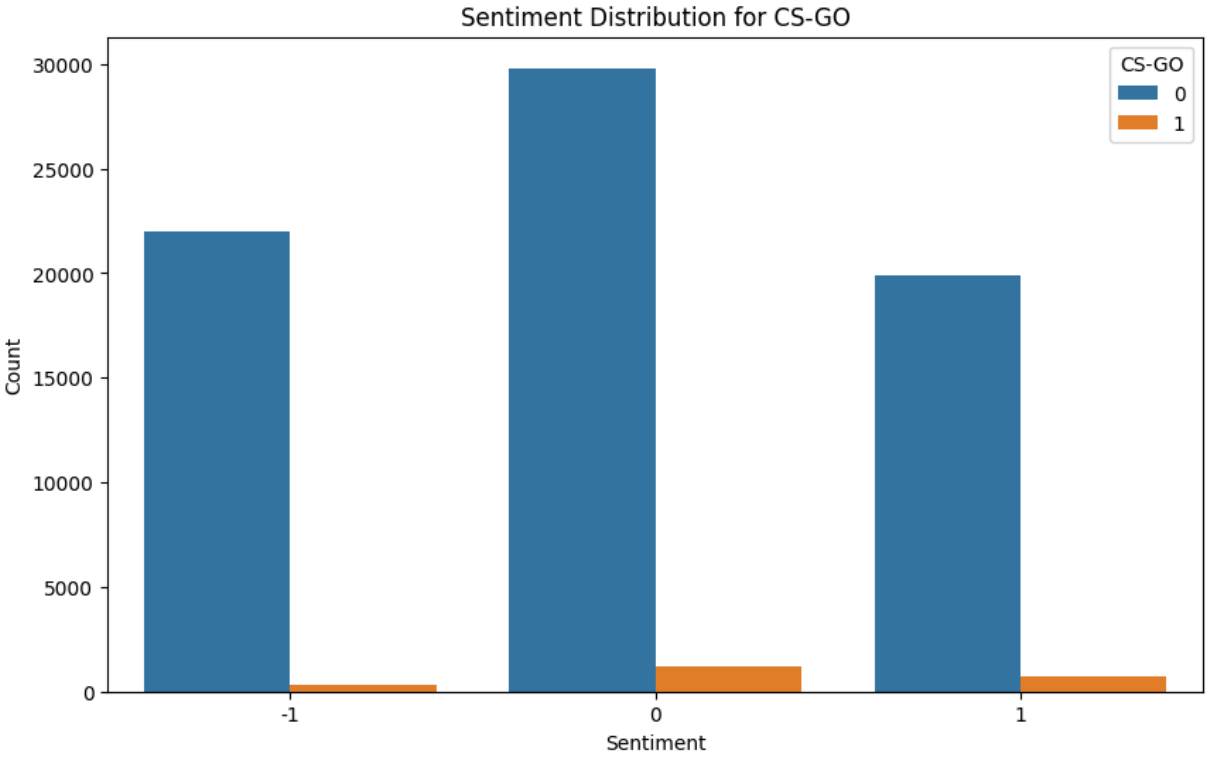
In [12]:
```
# Example: Sentiment distribution for different topics/brands
topics = ['ApexLegends', 'AssassinsCreed', 'Borderlands', 'CS-GO', 'Overwatch', 'Play

for topic in topics:
    plt.figure(figsize=(10, 6))
```

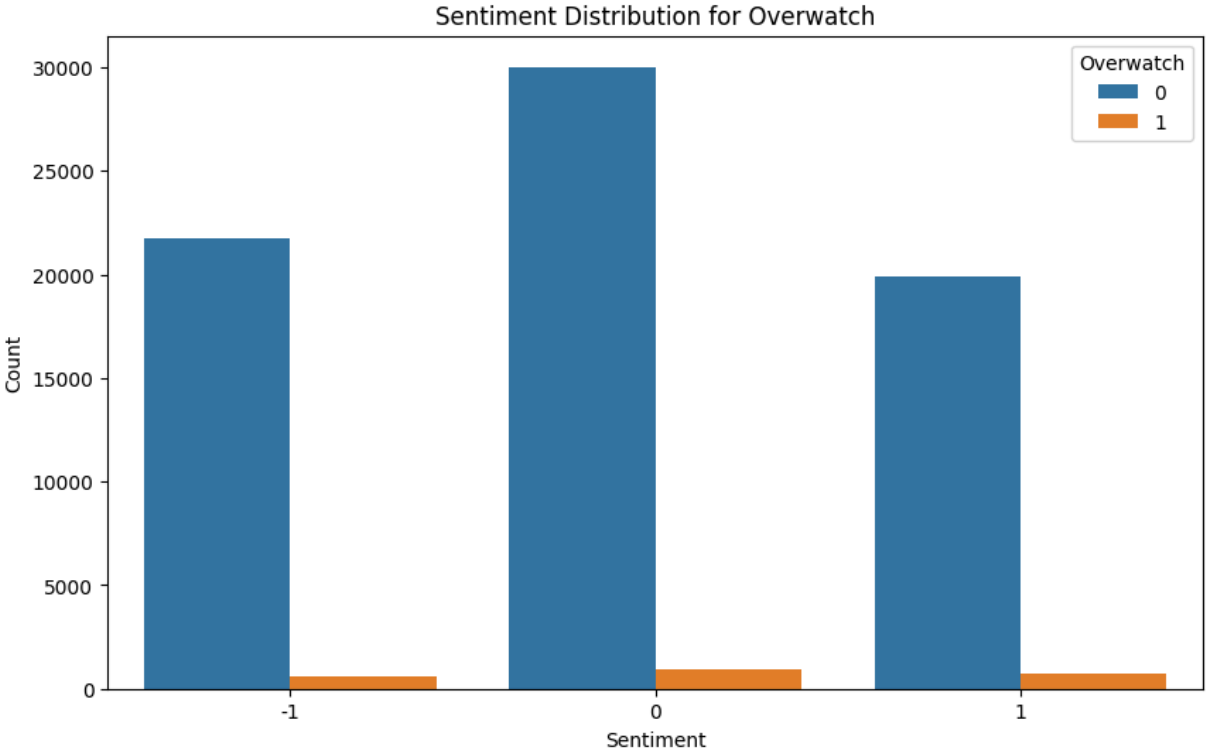
```
    sns.countplot(data=data, x='Sentiment', hue=topic)
    plt.title(f'Sentiment Distribution for {topic}')
    plt.xlabel('Sentiment')
    plt.ylabel('Count')
    plt.legend(title=topic)
    plt.show()
```

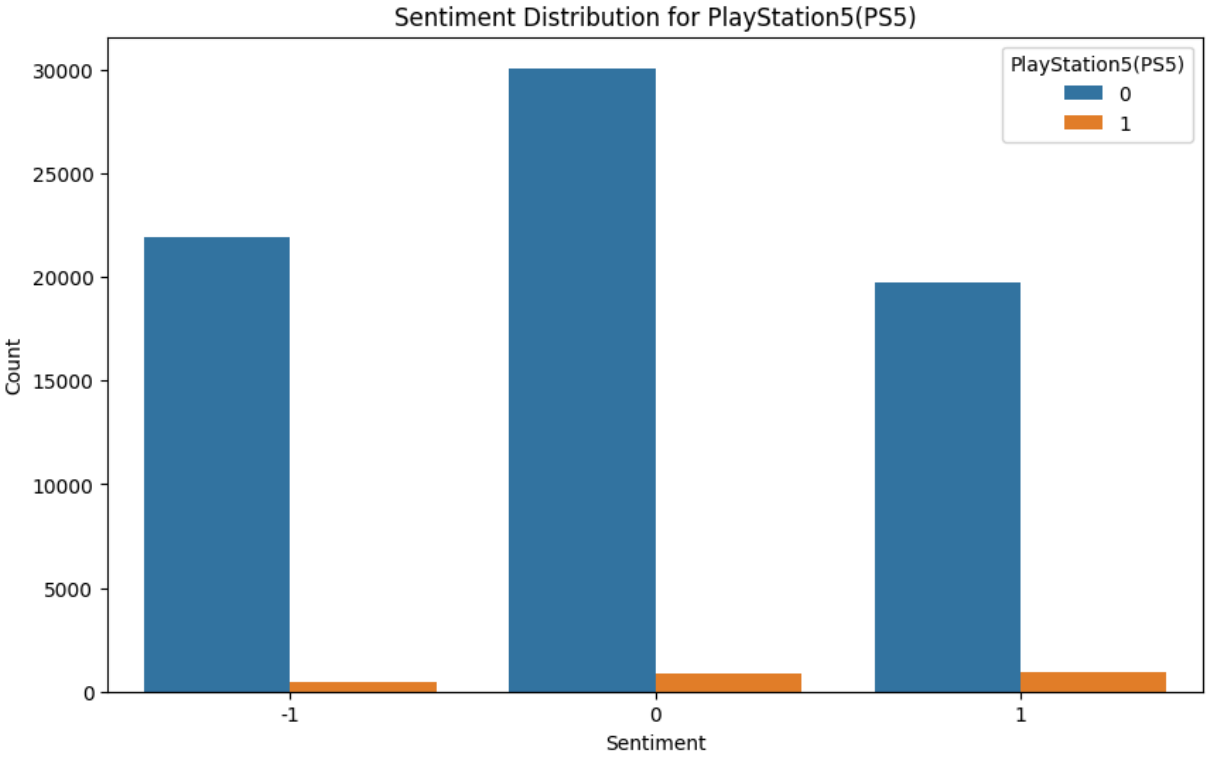


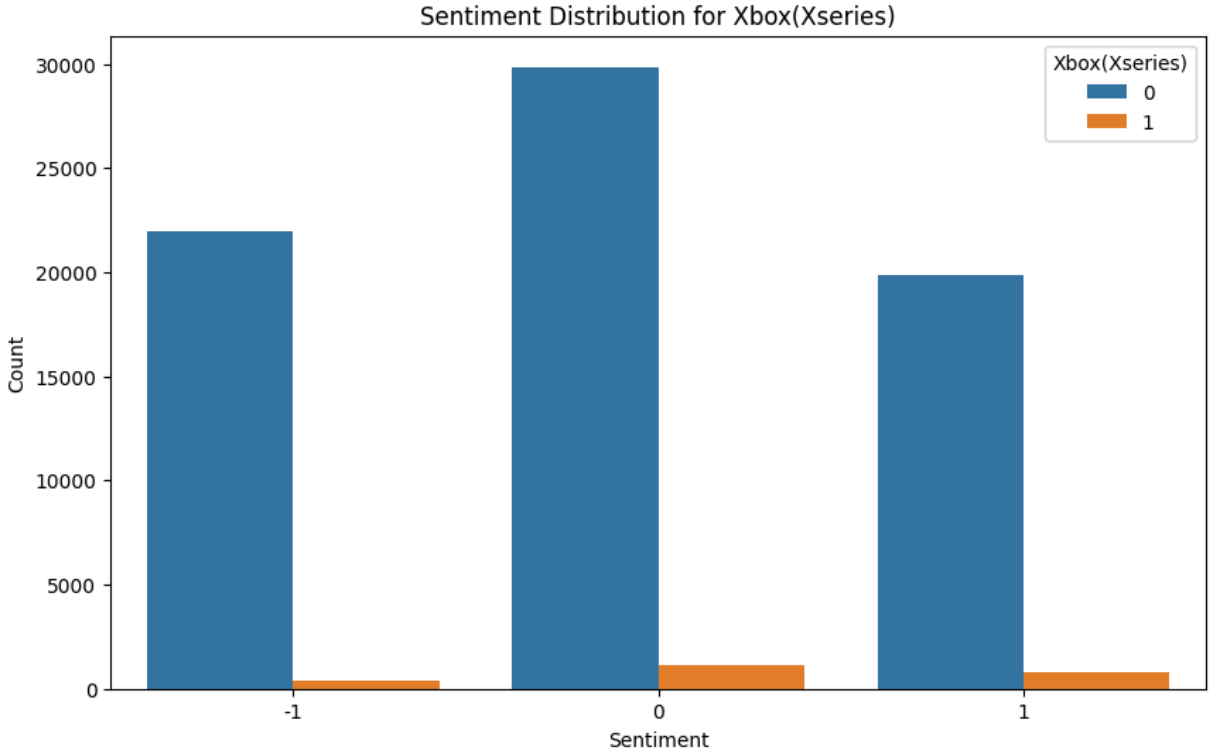



Sentiment Distribution for Borderlands
Borderlands
0
1
30000
25000
20000
15000
10000
5000
0
Count
-1
0
1
Sentiment


Sentiment Distribution for CS-GO
CS-GO
0
1
30000
25000
20000
15000
10000
5000
0
Count
-1
0
1
Sentiment


Sentiment Distribution for Overwatch
Overwatch
0
1
30000
25000
20000
15000
10000
5000
0
Count
-1
0
1
Sentiment


Sentiment Distribution for PlayStation5(PS5)
PlayStation5(PS5)
0
1
30000
25000
20000
15000
10000
5000
0
Count
-1
0
1
Sentiment


Sentiment Distribution for Xbox(Xseries)
Xbox(Xseries)
0
1
30000
25000
20000
15000
10000
5000
0
Count
-1
0
1
Sentiment

In [ ]: