Anime Data Analysis: Trends, Genres, and Popularity

This project provides an in-depth analysis of anime trends, genres, and ratings. By examining a dataset of popular anime shows, the goal is to uncover patterns in viewer preferences, genre popularity, rating distributions etc.

The dataset was downloaded from Kaggle

In [89]:
```
import pandas as pd
import numpy as np
```

In [90]:
```
import matplotlib.pyplot as plt
import seaborn as sns
from langdetect import detect
from datetime import datetime
import pickle
import warnings
warnings.filterwarnings(action='ignore')
```

In [91]:
```
from sklearn.preprocessing import MinMaxScaler
from sklearn.preprocessing import LabelEncoder
```

In [92]:
```
from tensorflow.keras.layers import Add, Activation, Lambda, BatchNormali
```

In [93]:
```
from tensorflow.keras import layers
```

In [94]:
```
from tensorflow.keras.optimizers import Adam
```

In [95]:
```
from collections import defaultdict
from collections import Counter
```

In [96]:
```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import linear_kernel
```

In [97]:
```
anime_df = pd.read_csv('/Users/riswee/Desktop/anime-dataset-2023.csv')
```

In [98]:
```
anime_df.head()
```

Out[98]:

	anime_id	Name	English name	Other name	Score	Genres	Synopsis	Type	E
0	1	Cowboy Bebop	Cowboy Bebop	カウボーイビバップ	8.75	Action, Award Winning, Sci-Fi	Crime is timeless. By the year 2071, humanity ...	TV	
1	5	Cowboy Bebop: Tengoku no Tobira	Cowboy Bebop: The Movie	カウボーイビバップ 天国の扉	8.38	Action, Sci-Fi	Another day, another bounty—such is the life o...	Movie	
2	6	Trigun	Trigun	トライガン	8.22	Action, Adventure, Sci-Fi	Vash the Stampede is the man with a $$60,000,0...	TV	
3	7	Witch Hunter Robin	Witch Hunter Robin	Witch Hunter ROBIN (ウイッチハンターロビン)	7.25	Action, Drama, Mystery, Supernatural	Robin Sena is a powerful craft user drafted in...	TV	
4	8	Bouken Ou Beet	Beet the Vandel Buster	冒険王ビィト	6.94	Adventure, Fantasy, Supernatural	It is the dark century and the people are suff...	TV	

5 rows × 24 columns

In [99]:
```
anime_df.shape
```

Out[99]: (24905, 24)

In [100…
```
user_df = pd.read_csv('/Users/riswee/Desktop/users-details-2023.csv')
```

In [101…
```
user_df.head()
```

Out[101…

	Mal ID	Username	Gender	Birthday	Location	Joined	Wa
0	1	Xinil	Male	1985-03-04T00:00:00+00:00	California	2004-11-05T00:00:00+00:00	
1	3	Aokaado	Male	NaN	Oslo, Norway	2004-11-11T00:00:00+00:00	
2	4	Crystal	Female	NaN	Melbourne, Australia	2004-11-13T00:00:00+00:00	
3	9	Arcane	NaN	NaN	NaN	2004-12-05T00:00:00+00:00	
4	18	Mad	NaN	NaN	NaN	2005-01-03T00:00:00+00:00	

In [102…
```
user_df.shape
```

Out[102… (731290, 16)

In [103…
```
score_df = pd.read_csv('/Users/riswee/Desktop/users-score-2023.csv')
```

In [104…
```
score_df.head()
```

Out[104…

	user_id	Username	anime_id	Anime Title	rating
0	1	Xinil	21	One Piece	9
1	1	Xinil	48	.hack//Sign	7
2	1	Xinil	320	A Kite	5
3	1	Xinil	49	Aa! Megami-sama!	8
4	1	Xinil	304	Aa! Megami-sama! Movie	8

In [105…
```
score_df.shape
```

Out[105… (24325191, 5)

In [106…
```
anime_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 24905 entries, 0 to 24904
Data columns (total 24 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   anime_id      24905 non-null  int64 
 1   Name          24905 non-null  object
 2   English name  24905 non-null  object
 3   Other name    24905 non-null  object
 4   Score         24905 non-null  object
 5   Genres        24905 non-null  object
 6   Synopsis      24905 non-null  object
 7   Type          24905 non-null  object
 8   Episodes      24905 non-null  object
 9   Aired         24905 non-null  object
 10  Premiered     24905 non-null  object
 11  Status        24905 non-null  object
 12  Producers     24905 non-null  object
 13  Licensors     24905 non-null  object
 14  Studios       24905 non-null  object
 15  Source        24905 non-null  object
 16  Duration      24905 non-null  object
 17  Rating        24905 non-null  object
 18  Rank          24905 non-null  object
 19  Popularity    24905 non-null  int64 
 20  Favorites     24905 non-null  int64 
 21  Scored By     24905 non-null  object
 22  Members       24905 non-null  int64 
 23  Image URL     24905 non-null  object
dtypes: int64(4), object(20)
memory usage: 4.6+ MB
```

In [107…

```
anime_df.isna().any()
```

Out[107…
```
anime_id        False
Name            False
English name    False
Other name      False
Score           False
Genres          False
Synopsis        False
Type            False
Episodes        False
Aired           False
Premiered       False
Status          False
Producers       False
Licensors       False
Studios         False
Source          False
Duration        False
Rating          False
Rank            False
Popularity      False
Favorites       False
Scored By       False
Members         False
Image URL       False
dtype: bool
```

In [108…
```
anime_df.duplicated().sum()
```

Out[108… 0

In [109…
```
Type_ct = anime_df['Type'].value_counts()
print(Type_ct)
```

```
TV         7597
Movie      4381
OVA        4076
ONA        3533
Music      2686
Special    2558
UNKNOWN      74
Name: Type, dtype: int64
```

In [110…
```
anime_df = anime_df[(anime_df['Type'] == 'TV') | (anime_df['Type'] == 'Mo
anime_df = anime_df[(anime_df['Score'] != 'UNKNOWN') | (anime_df['Scored
anime_df
```

Out[110…

	anime_id	Name	English name	Other name	Score	Genres	Synopsi
0	1	Cowboy Bebop	Cowboy Bebop	カウボーイビバップ	8.75	Action, Award Winning, Sci-Fi	Crime i timeless. B the yea 207

humanity .

1	5	Cowboy Bebop: Tengoku no Tobira	Cowboy Bebop: The Movie	カウボーイビバップ 天国の扉	8.38	Action, Sci-Fi	Another day anothe bounty- such is th life o.
2	6	Trigun	Trigun	トライガン	8.22	Action, Adventure, Sci-Fi	Vash th Stampede i the man wit $$60,000,0.
3	7	Witch Hunter Robin	Witch Hunter Robin	Witch Hunter ROBIN (ウイッチハンターロビン)	7.25	Action, Drama, Mystery, Supernatural	Robin Sen is a powerfu craft use drafted in.
4	8	Bouken Ou Beet	Beet the Vandel Buster	冒険王ビィト	6.94	Adventure, Fantasy, Supernatural	It is the dar century an the peopl are suff.
...	...	...	...	...	...	...	.
23785	54040	Tousouchuu: Great Mission	Run For Money : The Great Mission	逃走中グレートミッション	5.72	Sci-Fi, Suspense	The anim series is se in the futur when hum.
23902	54209	Minna to Manner wo Manabou	UNKNOWN	みんなとマナーを学ぼう	5.85	Comedy	Manne movie fo Madogata exhibitior \n\nCr.
23928	54259	Rokudou no Onna-tachi	Rokudo's Bad Girls	六道の悪女〈おんな〉たち	6.19	Comedy, Romance	Tousuk Rokudou is first-yea attendin Aomo.
24129	54591	Kimetsu no Yaiba: Jougen Shuuketsu,	Demon Slayer: Kimetsu no Yaiba - To	「鬼滅の刃」上弦集結、そして刀	7.58	Action, Fantasy	A feature length cut c episodes 1

		Soshite Ka...	the Swords...	鍛冶の里へ			and 11 of .
24197	54738	Majutsushi Orphen Hagure Tabi: Seiiki-hen	Sorcerous Stabber Orphen - Doom of Dragon's San...	魔術士オーフェンはぐれ旅 聖域編	6.09	Action, Adventure, Drama, Fantasy	With n other choic left, Orphe is led by Wi.

6957 rows × 24 columns

In [111…
```
# Step 1: Identify animes with known score values and scored by in the an

valid_anime_ids = anime_df[(anime_df['Score'] != 'UNKNOWN') & (anime_df['
```

In [112…
```
# Step 2: Filter out the corresponding rows from the users table

score_df = score_df[score_df['anime_id'].isin(valid_anime_ids)]
```

In [113…
```
# Step 3: Filter out the corresponding rows from the anime table

anime_df = anime_df[anime_df['anime_id'].isin(valid_anime_ids)]
```

In [114…
```
anime_df.shape
```

Out[114… (6957, 24)

In [115…
```
anime_sorted_rank = anime_df.sort_values('Rank',ascending = True)
```

In [116…
```
anime_sorted_rank.loc[:,['Score','Scored By']] = anime_sorted_rank.loc[:,
anime_sorted_rank
```

Out[116…

	anime_id	Name	English name	Other name	Score	Genres
3961	5114	Fullmetal Alchemist: Brotherhood	Fullmetal Alchemist: Brotherhood	鋼の錬金術師 FULLMETAL ALCHEMIST	9.10	Action, Adventure, Drama, Fantasy
6456	11061	Hunter x Hunter (2011)	Hunter x Hunter	HUNTER×HUNTER（ハンター×ハンター）	9.04	Action, Adventure, Fantasy
		Shoujo☆Kageki	Revue	劇場版 少女☆歌劇		

16184	40664	Revue Starlight Movie	Starlight: The Movie	レヴュースタァライト	8.55	UNKNOWN
8776	22975	Kaibutsu-kun: Demon no Ken	UNKNOWN	怪物くん デーモンの剣	5.72	Comedy, Horror
5267	8213	Hoka Hoka Kazoku	The Affectuous Family	ほかほか家族	5.72	Comedy
...	...	...	...	...	...	...
5534	8861	Yosuga no Sora: In Solitude, Where We Are Leas...	Yosuga no Sora	ヨスガノソラ In solitude, where we are least alone.	6.07	Drama, Romance, Erotica
2159	2360	Choujin Densetsu Urotsukidouji 2	Urotsukidoji II: Legend of the Demon Womb (movie)	超神伝説うろつき童子 魔胎伝	5.84	Fantasy, Horror, Sci-Fi, Hentai
15684	39761	Saezuru Tori wa Habatakanai: The Clouds Gather	UNKNOWN	囀る鳥は羽ばたかない The clouds gather	7.41	Boys Love, Drama, Erotica
9413	26081	Yasuji no Pornorama: Yacchimae!!	UNKNOWN	ヤスジのポルノラマ やっちまえ!!	5.17	Hentai
15673	39731	Na Bbeun Sang Sa	UNKNOWN	나쁜상사	5.93	Drama, Romance, Slice of Life, Suspense, Hentai

6957 rows × 24 columns

In [117…

```
def PP (score, scoreby, maxscoreby):
    return score * (scoreby/maxscoreby)

max_score_by = anime_sorted_rank['Scored By'].max()
anime_clean_rank = anime_sorted_rank.copy()
```

```
anime_clean_rank['Rank'] = anime_clean_rank.apply(lambda row: PP(row['Sco
anime_clean_rank = anime_clean_rank.sort_values('Rank', ascending= False)
anime_clean_rank['Rank Number'] = range(1, len(anime_clean_rank) + 1)
anime_clean_rank = anime_clean_rank.loc[:]
anime_clean_rank.head(20)
```

Out[117…

	anime_id	Name	English name	Other name	Score	Genres	
0	16498	Shingeki no Kyojin	Attack on Titan	進撃の巨人	8.54	Action, Award Winning, Drama, Suspense	ag s
1	1535	Death Note	Death Note	デスノート	8.62	Supernatural, Suspense	p and
2	5114	Fullmetal Alchemist: Brotherhood	Fullmetal Alchemist: Brotherhood	鋼の錬金術師 FULLMETAL ALCHEMIST	9.10	Action, Adventure, Drama, Fantasy	Aft go
3	30276	One Punch Man	One Punch Man	ワンパンマン	8.50	Action, Comedy	ur Sa
4	38000	Kimetsu no Yaiba	Demon Slayer: Kimetsu no Yaiba	鬼滅の刃	8.50	Action, Award Winning, Fantasy	Eve
5	32281	Kimi no Na wa.	Your Name.	君の名は。	8.85	Award Winning, Drama, Supernatural	girl
6	31964	Boku no Hero Academia	My Hero Academia	僕のヒーローアカデミア	7.89	Action	app

7	20	Naruto	Naruto	ナルト	7.99	Action, Adventure, Fantasy	bi
8	25777	Shingeki no Kyojin Season 2	Attack on Titan Season 2	進撃の巨人 Season2	8.50	Action, Drama, Suspense	Fo hu b
9	11061	Hunter x Hunter (2011)	Hunter x Hunter	HUNTER×HUNTER（ハンター×ハンター）	9.04	Action, Adventure, Fantasy	the acc
10	11757	Sword Art Online	Sword Art Online	ソードアート・オンライン	7.20	Action, Adventure, Fantasy, Romance	Eve rel
11	22319	Tokyo Ghoul	Tokyo Ghoul	東京喰種-トーキョーグール-	7.79	Action, Fantasy, Horror	To
12	28851	Koe no Katachi	A Silent Voice	聲の形	8.94	Award Winning, Drama	stu
13	33486	Boku no Hero Academia 2nd Season	My Hero Academia Season 2	僕のヒーローアカデミア	8.11	Action	Ac eve att
14	38524	Shingeki no Kyojin Season 3	Attack on Titan Season 3	進撃の巨人 Season3 Part.2	9.05	Action, Drama	

			Part 2	Part 2			
15	35760	Shingeki no Kyojin Season 3	Attack on Titan Season 3	進撃の巨人 Season3	8.62	Action, Drama	thr t
16	1735	Naruto: Shippuuden	Naruto Shippuden	-ナルト- 疾風伝	8.26	Action, Adventure, Fantasy	tw
17	40748	Jujutsu Kaisen	Jujutsu Kaisen	呪術廻戦	8.64	Action, Award Winning, Fantasy	Id
18	9253	Steins;Gate	Steins;Gate	STEINS;GATE	9.07	Drama, Sci-Fi, Suspense	C
19	19815	No Game No Life	No Game, No Life	ノーゲーム・ノーライフ	8.08	Comedy, Fantasy, Ecchi	ser

20 rows × 25 columns

In [118…

```
import datetime
from datetime import date

def estandarizar_fechas(fecha):

    if " to " in fecha:
        fecha = fecha.split(" to ")[0]

    formats = ("%b %d, %Y", "%b %d %Y", "%m/%d/%Y", "%Y")
```

```
    for fmt in formats:
        try:
            return datetime.datetime.strptime(fecha, fmt).date()
        except ValueError:
            pass

    return datetime.date(1900, 1, 1)

anime_clean_rank["Aired"] = anime_clean_rank["Aired"].apply(estandarizar_
anime_clean_rank = anime_clean_rank[anime_clean_rank['Aired'] > date(1900
anime_clean_rank
```

Out[118…

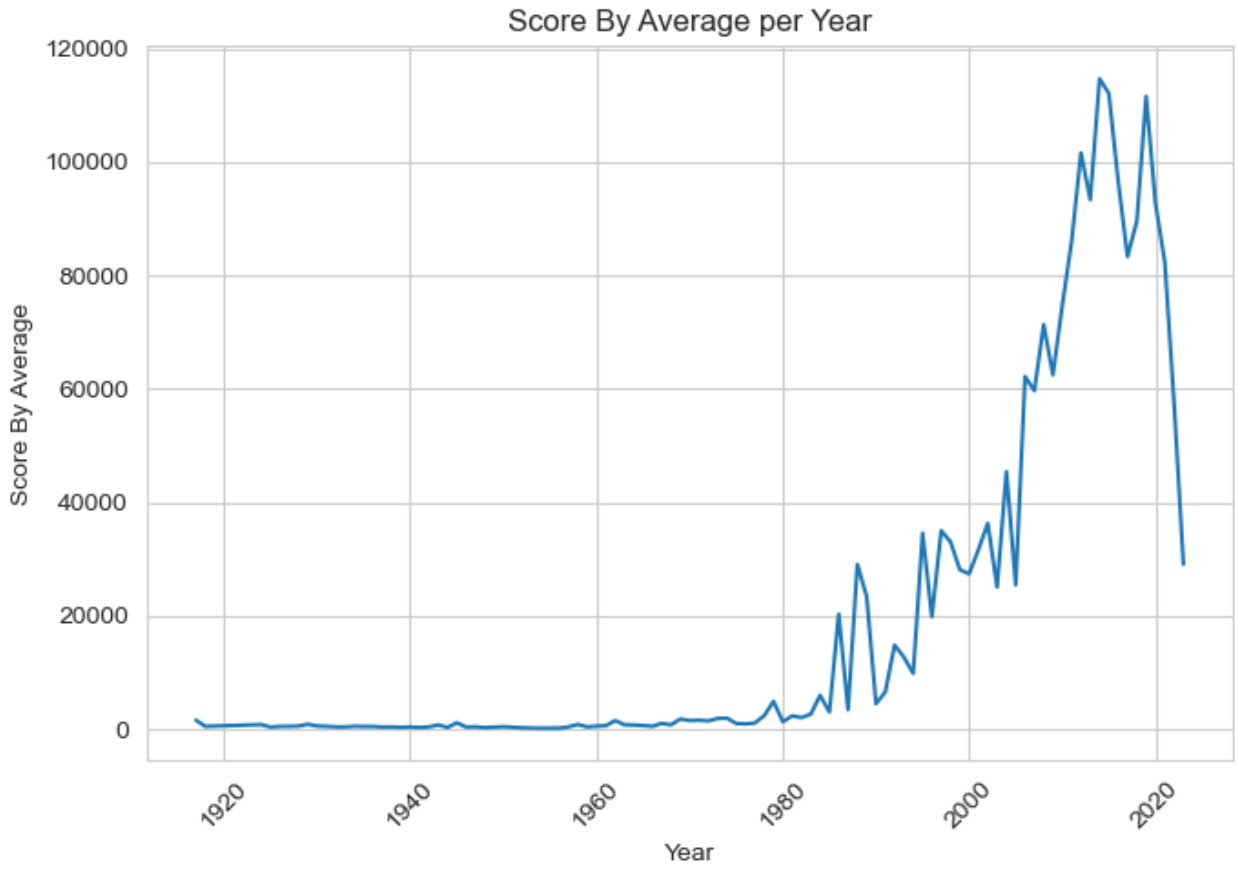
	anime_id	Name	English name	Other name	Score	Genres	S
0	16498	Shingeki no Kyojin	Attack on Titan	進撃の巨人	8.54	Action, Award Winning, Drama, Suspense	(mar sla
1	1535	Death Note	Death Note	デスノート	8.62	Supernatural, Suspense	pet s
2	5114	Fullmetal Alchemist: Brotherhood	Fullmetal Alchemist: Brotherhood	鋼の錬金術師 FULLMETAL ALCHEMIST	9.10	Action, Adventure, Drama, Fantasy	ex
3	30276	One Punch Man	One Punch Man	ワンパンマン	8.50	Action, Comedy	s unin Sai
4	38000	Kimetsu no Yaiba	Demon Slayer: Kimetsu no Yaiba	鬼滅の刃	8.50	Action, Award Winning, Fantasy	E the h the
…	…	…	…	…	…	…	
6952	30408	Tokyo SOS	UNKNOWN	tokyoSOS	4.78	Avant Garde	T a a
6953	42576	Houkago Pilot	UNKNOWN	放課後（パイロット版）	4.90	Slice of Life	de aft

							a
6954	34115	Hitori de Nayamanaide Issho ni Kangaeyou	UNKNOWN	一人で悩まないで一緒に考えよう	5.07	Drama	co air Ci Ka
6955	42272	Karasu	UNKNOWN	鴉	4.97	Avant Garde	M us a a
6956	51048	Imagination of Marathon Runners	UNKNOWN	IMAGINATION of MARATHON RUNNERS	4.95	Comedy, Sports	A sho

6899 rows × 25 columns

In [119…

```
anime_year_helper = anime_clean_rank.copy()
anime_year_helper['Aired'] = pd.to_datetime(anime_year_helper['Aired'])
anime_year_helper['Year'] = anime_year_helper['Aired'].dt.year
score_by_year = anime_year_helper.groupby('Year')['Scored By'].mean()
fig, ax = plt.subplots(figsize=(7, 5))
score_by_year.plot(kind='line', ax=ax)

ax.set_xlabel('Year')
ax.set_ylabel('Score By Average')
ax.set_title('Score By Average per Year')

plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



In [120…
```
anime_clean_rank = anime_clean_rank[anime_clean_rank['Aired'] > date(1980
anime_clean_rank
```

Out[120…

	anime_id	Name	English name	Other name	Score	Genres	S
0	16498	Shingeki no Kyojin	Attack on Titan	進撃の巨人	8.54	Action, Award Winning, Drama, Suspense	(mar sla
1	1535	Death Note	Death Note	デスノート	8.62	Supernatural, Suspense	pet s
2	5114	Fullmetal Alchemist: Brotherhood	Fullmetal Alchemist: Brotherhood	鋼の錬金術師 FULLMETAL ALCHEMIST	9.10	Action, Adventure, Drama, Fantasy	ex
3	30276	One Punch Man	One Punch Man	ワンパンマン	8.50	Action, Comedy	s unin Sai

4	38000	Kimetsu no Yaiba	Demon Slayer: Kimetsu no Yaiba	鬼滅の刃	8.50	Action, Award Winning, Fantasy	E the h the
...	...	...	...	...	...	...	
6952	30408	Tokyo SOS	UNKNOWN	tokyoSOS	4.78	Avant Garde	T a a
6953	42576	Houkago Pilot	UNKNOWN	放課後（パイロット版）	4.90	Slice of Life	de aft ac
6954	34115	Hitori de Nayamanaide Issho ni Kangaeyou	UNKNOWN	一人で悩まないで一緒に考えよう	5.07	Drama	co air Ci Ka
6955	42272	Karasu	UNKNOWN	鴉	4.97	Avant Garde	M us a a
6956	51048	Imagination of Marathon Runners	UNKNOWN	IMAGINATION of MARATHON RUNNERS	4.95	Comedy, Sports	A sho

6314 rows × 25 columns

In [121…
```
df_anime_expanded = anime_clean_rank.copy()

genres_expanded = df_anime_expanded["Genres"].str.get_dummies(sep=", ")
df_anime_expanded = df_anime_expanded.join(genres_expanded)
df_anime_expanded = df_anime_expanded.drop("Genres", axis=1)
df_anime_expanded = df_anime_expanded.rename(columns={'Type': 'Medio Emis
```

In [122…
```
def split_studios(x):
    return x.split(", ")
df_anime_expanded["Studios"] = df_anime_expanded["Studios"].apply(split_s
df_anime_expanded = df_anime_expanded.explode('Studios').reset_index(drop

df_anime_expanded.head()
```

Out[122…

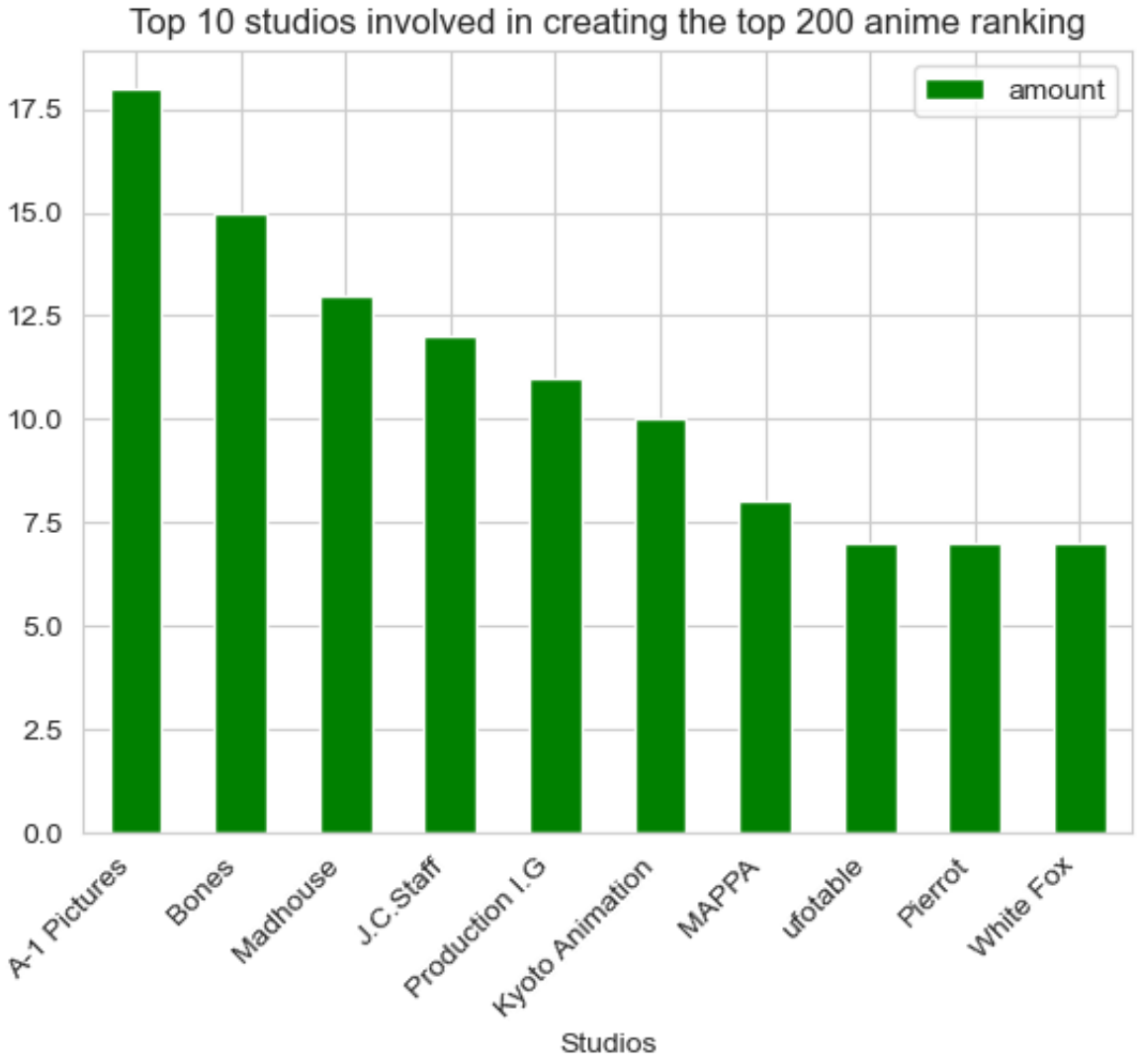
	anime_id	Name	English name	Other name	Score	Synopsis	Medio Emision	E
0	16498	Shingeki no Kyojin	Attack on Titan	進撃の巨人	8.54	Centuries ago, mankind was slaughtered to near...	TV	
1	1535	Death Note	Death Note	デスノート	8.62	Brutal murders, petty thefts, and senseless vi...	TV	
2	5114	Fullmetal Alchemist: Brotherhood	Fullmetal Alchemist: Brotherhood	鋼の錬金術師 FULLMETAL ALCHEMIST	9.10	After a horrific alchemy experiment goes wrong...	TV	
3	30276	One Punch Man	One Punch Man	ワンパンマン	8.50	The seemingly unimpressive Saitama has a rathe...	TV	
4	38000	Kimetsu no Yaiba	Demon Slayer: Kimetsu no Yaiba	鬼滅の刃	8.50	Ever since the death of his father, the burden...	TV	

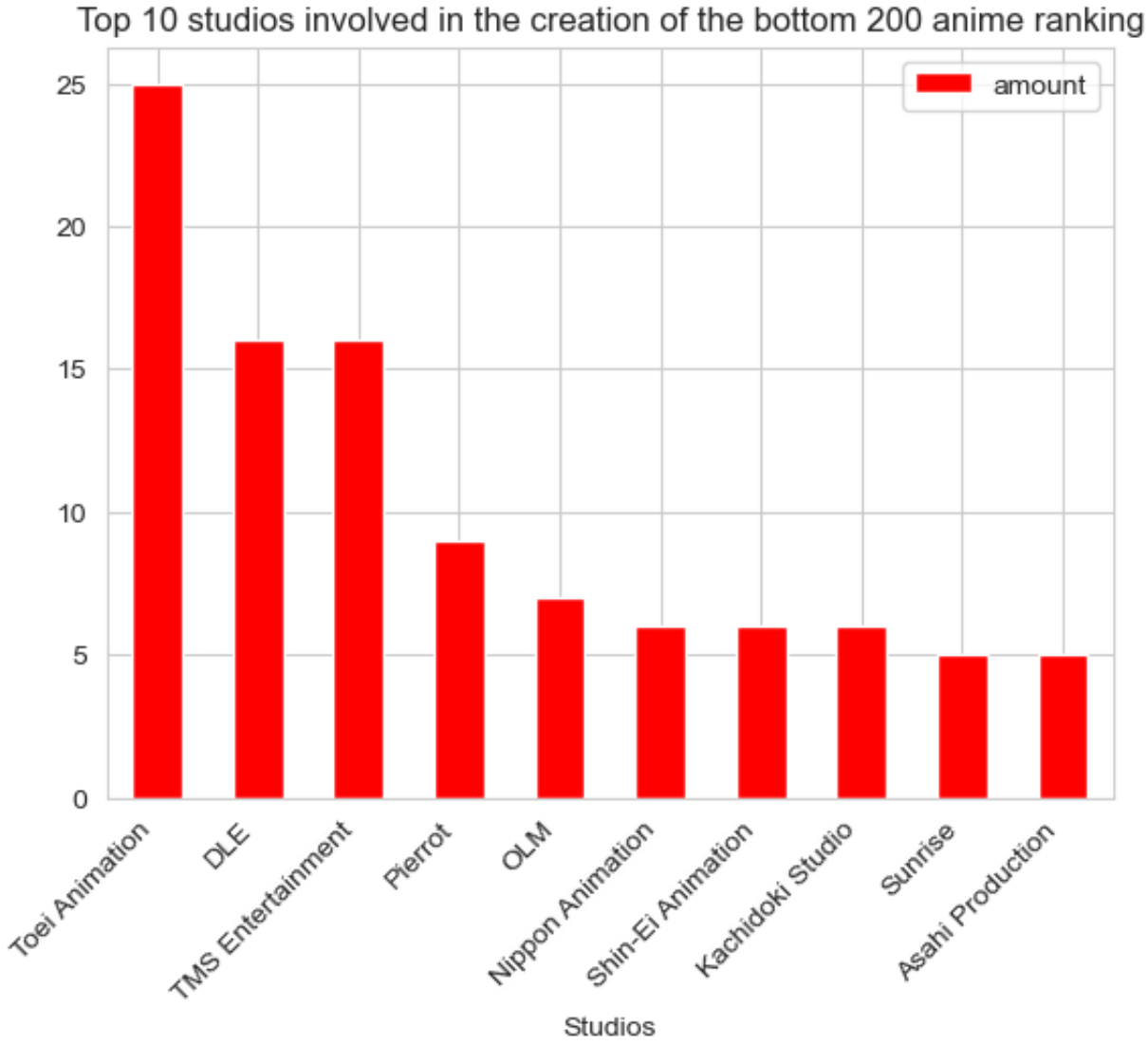
5 rows × 46 columns

In [123…

```
muestra_top = df_anime_expanded.head(200)
top_studios = (
    muestra_top.groupby('Studios').Name.count()
        .reset_index(name='amount')
        .sort_values('amount', ascending = False)
        .head(10)
)

muestra_bottom = (
    df_anime_expanded[df_anime_expanded['Studios'] != 'UNKNOWN']
        .sort_values('Rank Number', ascending = False)
        .head(200)
)

bottom_studios = (
    muestra_bottom.groupby('Studios').Name.count()
        .reset_index(name='amount')
        .sort_values('amount', ascending = False)
```

```
        .head(10)
)


top_studios.plot(kind = 'bar', x = 'Studios', title='Top 10 studios invol
plt.xticks(rotation=45, ha='right')
bottom_studios.plot(kind = 'bar', x= 'Studios', title='Top 10 studios inv
plt.xticks(rotation=45, ha='right')
plt.show()
```





In [124…

```
df_long = pd.melt(df_anime_expanded, id_vars=['Name', 'Studios', 'Rank'],
                  value_vars=['Adventure', 'Avant Garde', 'Award Winning'
       'Drama', 'Ecchi', 'Erotica', 'Fantasy', 'Girls Love', 'Gourmet',
       'Hentai', 'Horror', 'Mystery', 'Romance', 'Sci-Fi',
       'Slice of Life', 'Sports', 'Supernatural', 'Suspense', 'Action',
       'Adventure', 'Avant Garde', 'Award Winning', 'Boys Love', 'Comedy'
       'Drama', 'Ecchi', 'Fantasy', 'Girls Love', 'Gourmet', 'Horror',
       'Mystery', 'Romance', 'Sci-Fi', 'Slice of Life', 'Sports',
       'Supernatural', 'Suspense'],
                  var_name='Genre', value_name='Present')

# Filter only rows where a gender is present (assuming you use 1 or True
df_long = df_long[df_long['Present'] == 1]

# Group by study and genre, and count the occurrences
genre_count_by_studio = df_long.groupby(['Studios', 'Genre']).size().rese

top_genres_by_studio = genre_count_by_studio.groupby('Studios').apply(lam

top_genres_by_studio
```

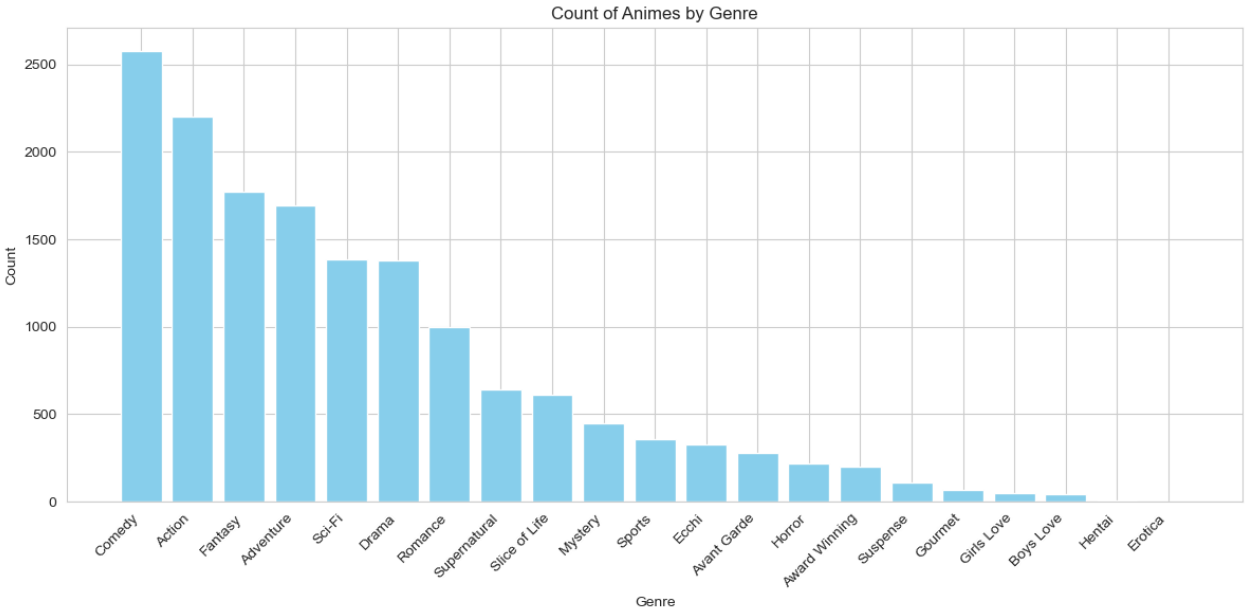
Out[124…

	Studios	Genre	Count
0	10Gauge	Slice of Life	1
1	10Gauge	Sports	1
2	8bit	Action	18
3	8bit	Comedy	13
4	8bit	Fantasy	13
...	...	...	...
1257	ufotable	Action	29
1258	ufotable	Fantasy	18
1259	ufotable	Supernatural	15
1260	yell	Action	1
1261	yell	Sci-Fi	1

1262 rows × 3 columns

In [125…

```
# Split the genres and count their occurrences
Genre_Counts = anime_df[anime_df['Genres'] != "UNKNOWN"]['Genres'].apply(

plt.figure(figsize=(12, 6))
plt.bar(Genre_Counts.index, Genre_Counts.values, color='skyblue')
plt.xlabel('Genre')
plt.ylabel('Count')
plt.title('Count of Animes by Genre')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```



In [126…
```
anime_df = anime_clean_rank
```

In [127…
```
user_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 731290 entries, 0 to 731289
Data columns (total 16 columns):
 #   Column            Non-Null Count   Dtype  
---  ------            --------------   -----  
 0   Mal ID            731290 non-null  int64  
 1   Username          731290 non-null  object 
 2   Gender            224383 non-null  object 
 3   Birthday          168068 non-null  object 
 4   Location          152806 non-null  object 
 5   Joined            731290 non-null  object 
 6   Days Watched      731282 non-null  float64
 7   Mean Score        731282 non-null  float64
 8   Watching          731282 non-null  float64
 9   Completed         731282 non-null  float64
 10  On Hold           731282 non-null  float64
 11  Dropped           731282 non-null  float64
 12  Plan to Watch     731282 non-null  float64
 13  Total Entries     731282 non-null  float64
 14  Rewatched         731282 non-null  float64
 15  Episodes Watched  731282 non-null  float64
dtypes: float64(10), int64(1), object(5)
memory usage: 89.3+ MB
```

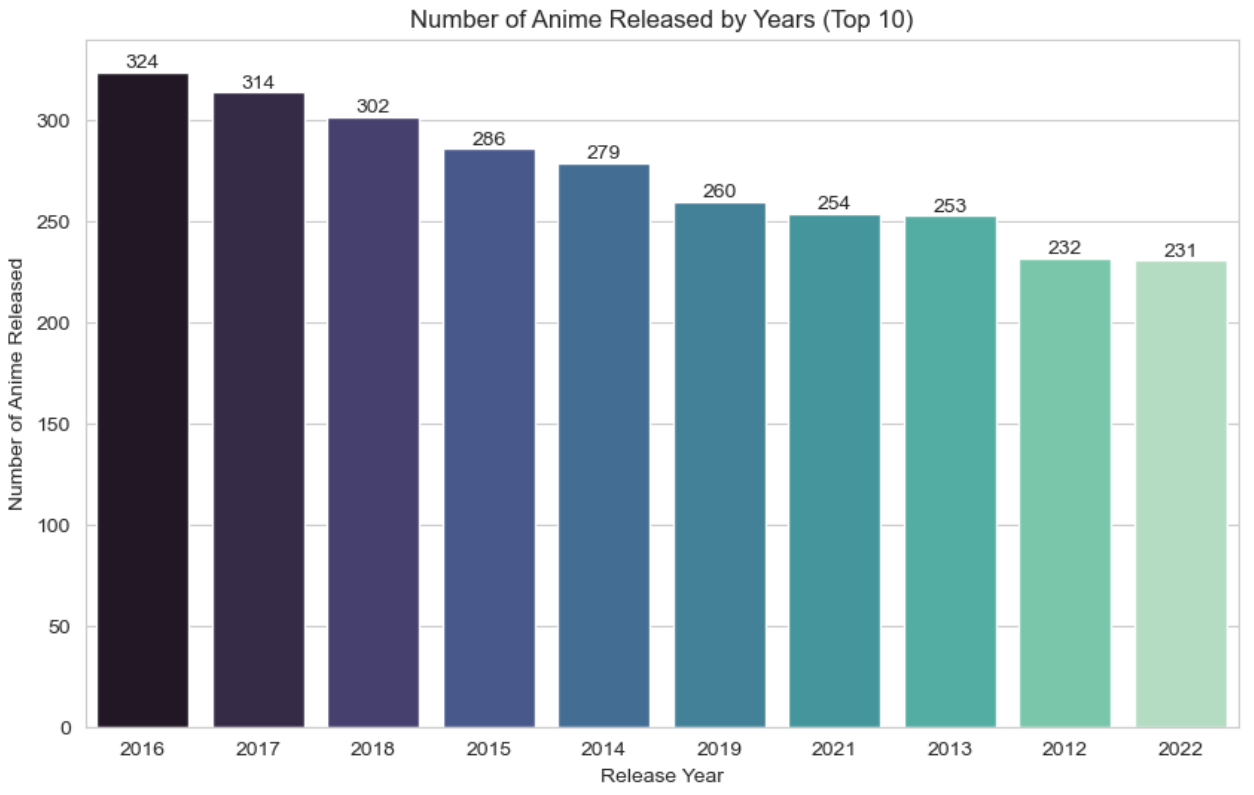
In [128…
```
# Convert the 'Aired' column to string type
anime_df['Aired'] = anime_df['Aired'].astype(str)

# Extract the release year from the 'Aired' column
anime_df['Release_Year'] = anime_df['Aired'].str.extract(r'(\d{4})')

# Group by release year and count the number of anime for each year
anime_cby = anime_df.groupby('Release_Year').size().reset_index(name='Ani

# Sort the result by anime count in descending order
k = anime_cby.sort_values(by='Anime_Count', ascending=False)

# Display the top 10 years with the highest number of anime
k.head(10)
```

Out[128…

	Release_Year	Anime_Count
36	2016	324
37	2017	314
38	2018	302
35	2015	286
34	2014	279
39	2019	260
41	2021	254
33	2013	253
32	2012	232
42	2022	231

In [129…

```
top_10 = k.head(10)
plt.figure(figsize=(10,6))
ax = sns.barplot(x='Release_Year', y='Anime_Count', hue='Release_Year', d
for i in ax.containers:
    ax.bar_label(i)
ax.set_xlabel('Release Year')
ax.set_ylabel('Number of Anime Released')
plt.title('Number of Anime Released by Years (Top 10)')
plt.show()
```

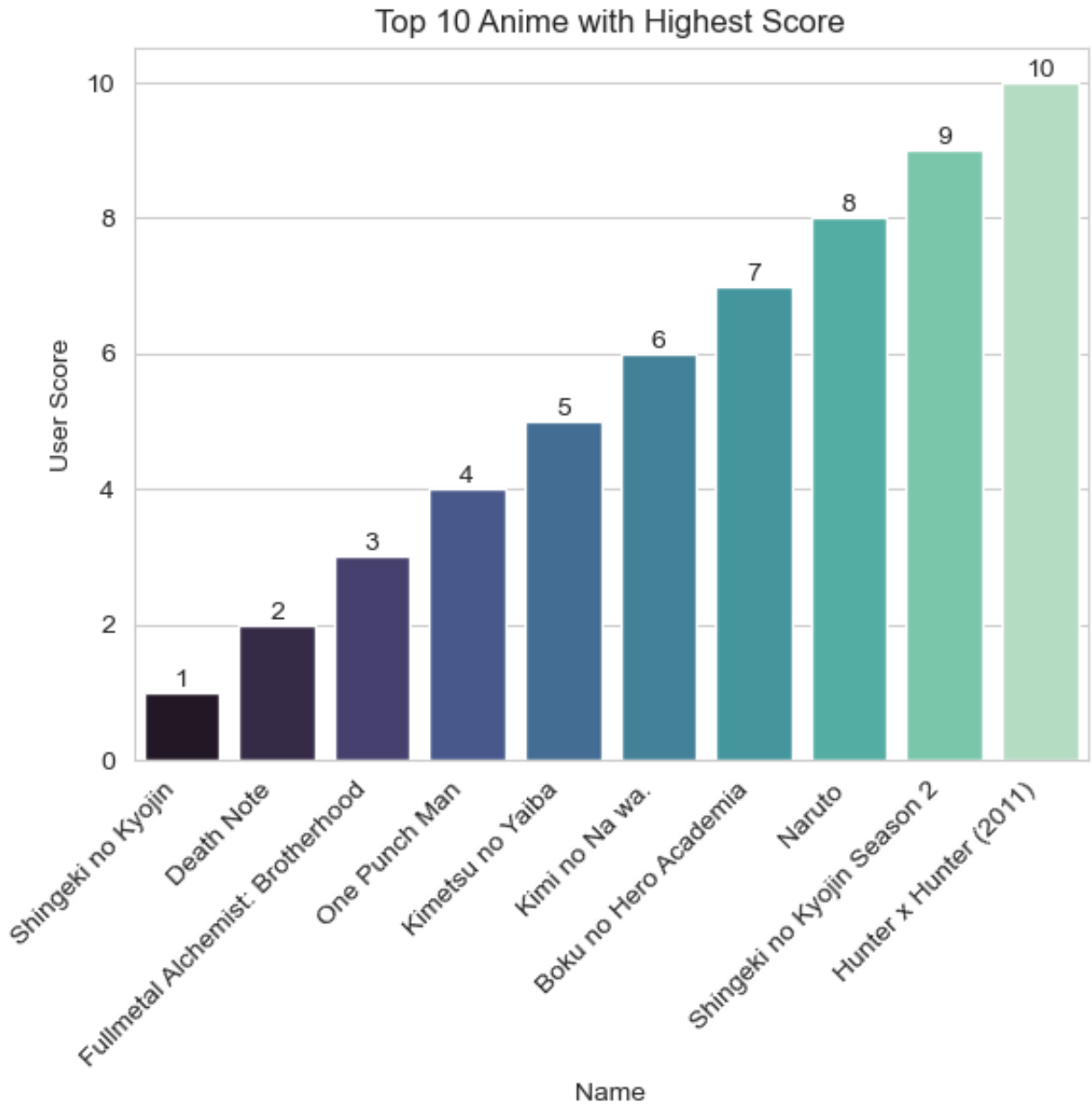


In [130…
```
name_score=anime_clean_rank[['Name','Rank Number']].sort_values(by='Rank
name_score.head(10)
```

Out[130…

	Name	Rank Number
0	Shingeki no Kyojin	1
1	Death Note	2
2	Fullmetal Alchemist: Brotherhood	3
3	One Punch Man	4
4	Kimetsu no Yaiba	5
5	Kimi no Na wa.	6
6	Boku no Hero Academia	7
7	Naruto	8
8	Shingeki no Kyojin Season 2	9
9	Hunter x Hunter (2011)	10

In [131…
```
ax=sns.barplot(y='Rank Number',x='Name',hue='Name',data=name_score,palett
ax.set_title('Top 10 Anime with Highest Score')
for i in ax.containers:
    ax.bar_label(i)
ax.set_xlabel('Name')
ax.set_ylabel('User Score')
plt.xticks(rotation=45, ha='right')
ax.title.set_text('Top 10 Anime with Highest Score')
plt.show()
```



In [132…

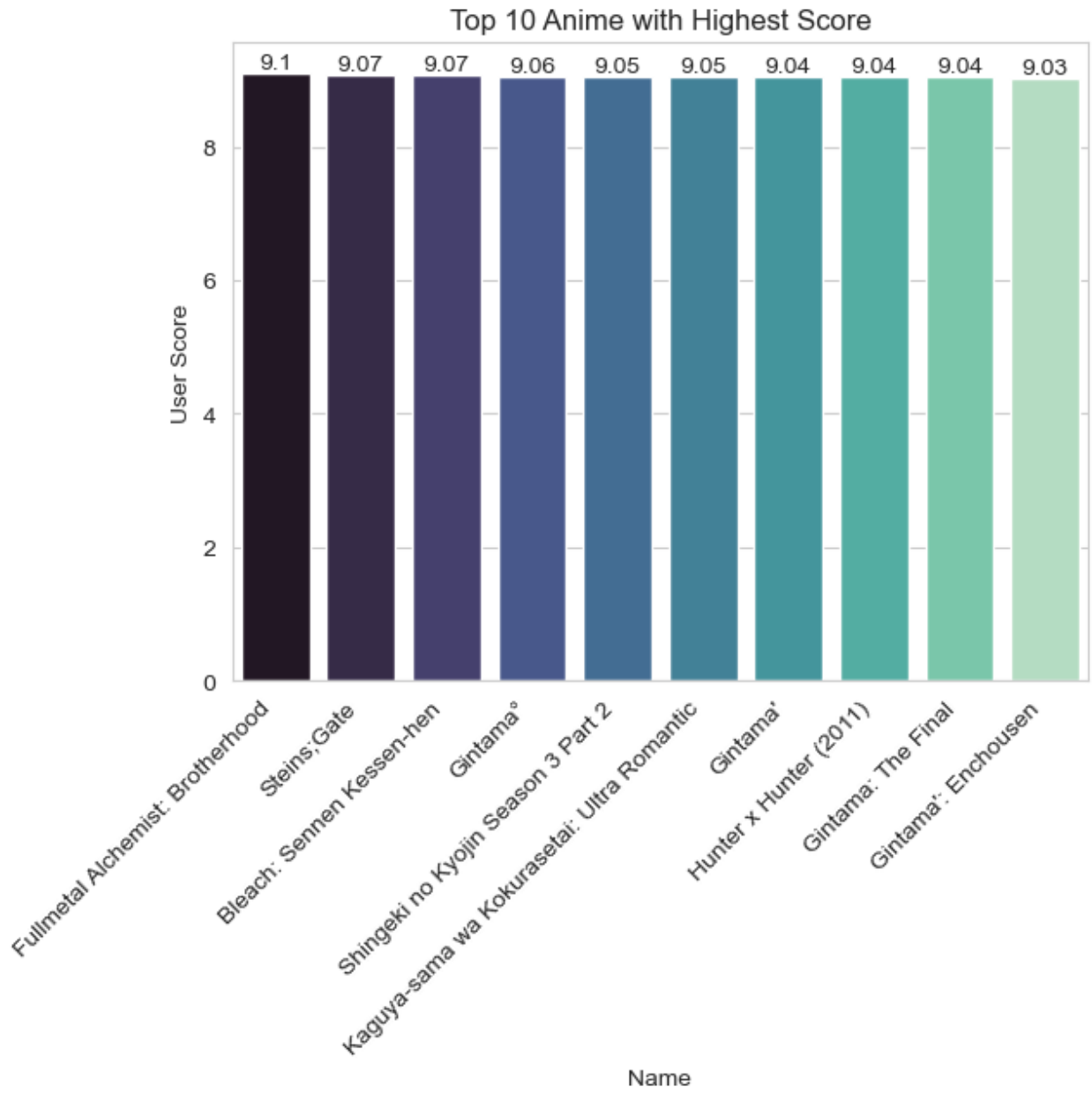
```
name_score=anime_clean_rank[['Name','Score']].sort_values(by='Score',asce
name_score.head(10)
```

Out[132…

	Name	Score
2	Fullmetal Alchemist: Brotherhood	9.10
18	Steins;Gate	9.07
414	Bleach: Sennen Kessen-hen	9.07
362	Gintama°	9.06
14	Shingeki no Kyojin Season 3 Part 2	9.05
163	Kaguya-sama wa Kokurasetai: Ultra Romantic	9.05
390	Gintama'	9.04
9	Hunter x Hunter (2011)	9.04
1161	Gintama: The Final	9.04
554	Gintama': Enchousen	9.03

In [133…

```
ax=sns.barplot(y='Score',x='Name',hue='Name',data=name_score,palette='mak
ax.set_title('Top 10 Anime with Highest Score')
for i in ax.containers:
    ax.bar_label(i)
ax.set_xlabel('Name')
ax.set_ylabel('User Score')
plt.xticks(rotation=45, ha='right')
ax.title.set_text('Top 10 Anime with Highest Score')
plt.show()
```



In [134…

```
sns.set_style('whitegrid')
name_score=anime_clean_rank[['Name','Score']].sort_values(by='Score',asce
name_score.head(10)
```

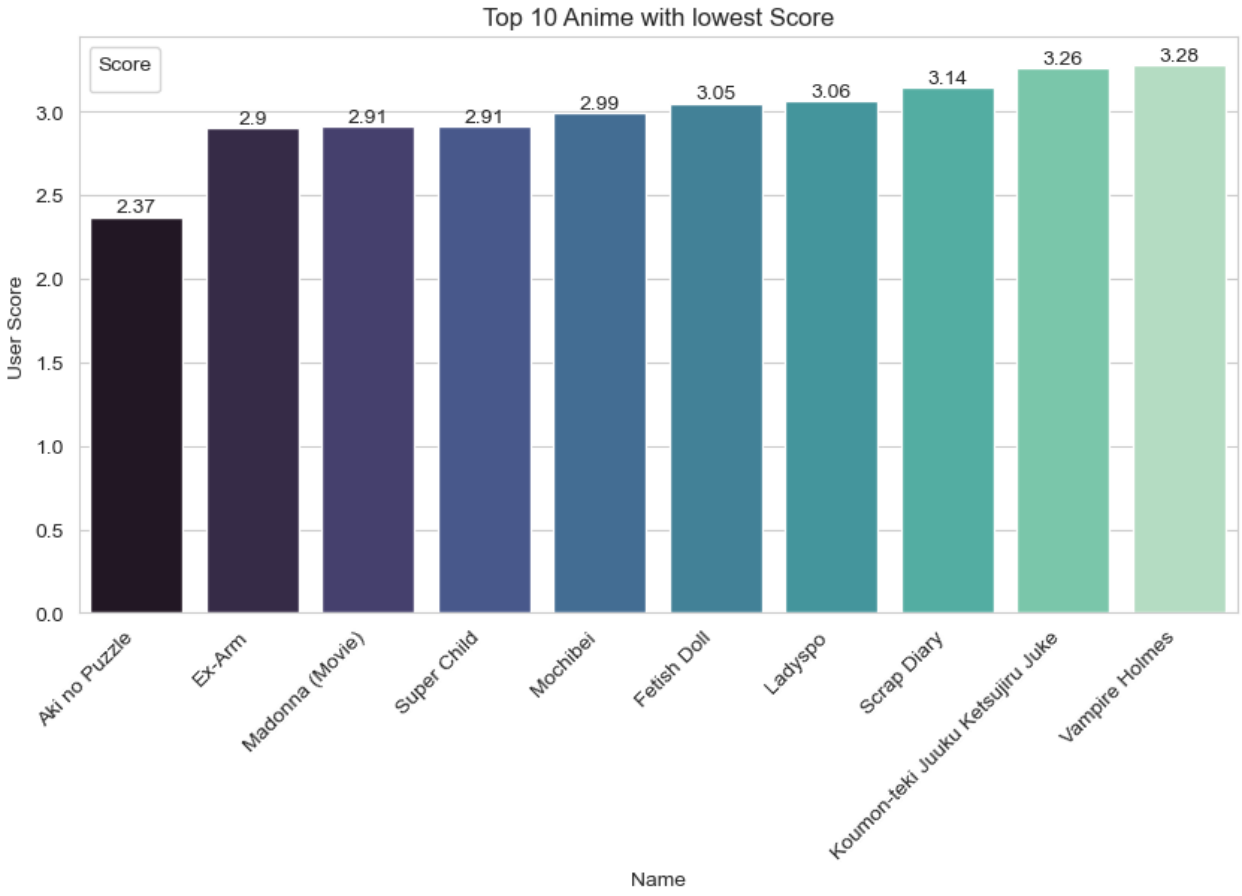
Out[134…

	Name	Score
5216	Aki no Puzzle	2.37
2452	Ex-Arm	2.90
5610	Madonna (Movie)	2.91
5781	Super Child	2.91
4520	Mochibei	2.99
5149	Fetish Doll	3.05
5273	Ladyspo	3.06
5345	Scrap Diary	3.14
5438	Koumon-teki Juuku Ketsujiru Juke	3.26
3586	Vampire Holmes	3.28

In [135…

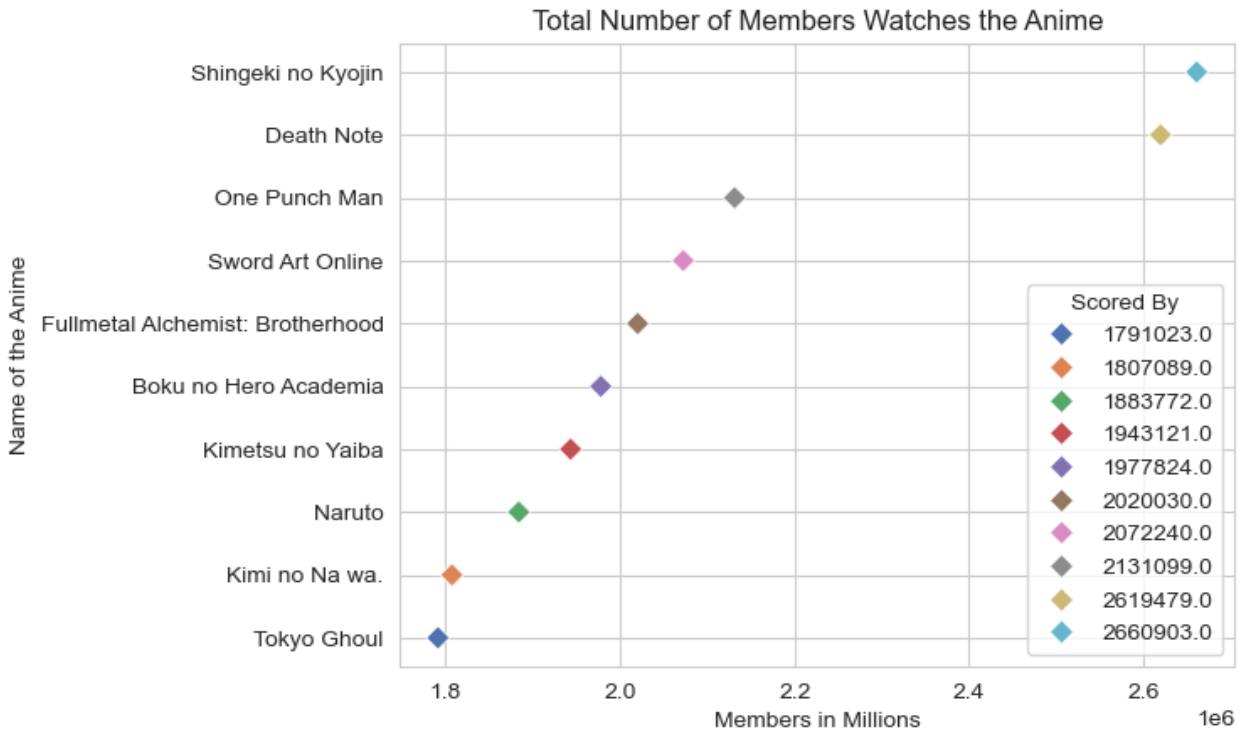
```
sns.set_style('whitegrid')
name_score=anime_clean_rank[['Name','Score']].sort_values(by='Score',asce
plt.figure(figsize=(10,5))
ax=sns.barplot(y='Score',x='Name',hue='Name',data=name_score,palette='mak
ax.set_title('Top 10 Anime with lowest  Score')
for i in ax.containers:
    ax.bar_label(i)
ax.set_xlabel('Name')
ax.set_ylabel('User Score')
plt.xticks(rotation=90)
ax.title.set_text('Top 10 Anime with lowest Score')
plt.legend(title='Score' ,loc='upper left')
plt.xticks(rotation=45, ha='right')
plt.show()
```

```
No artists with labels found to put in legend.  Note that artists whose la
bel start with an underscore are ignored when legend() is called with no a
rgument.
```

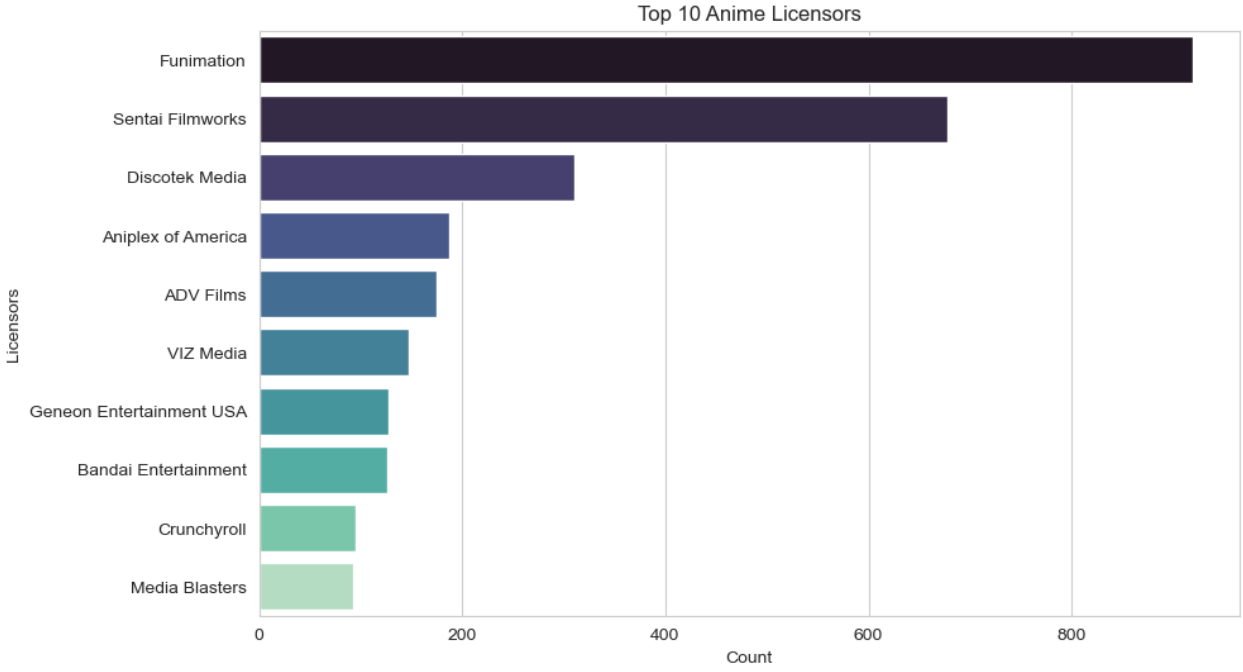


In [136…

```
famous=anime_clean_rank[['Scored By','Name']].sort_values(by='Scored By',
ax=sns.scatterplot(x='Scored By',y='Name',s=50,data=famous,marker='D',pal
plt.xlabel('Members in Millions')
plt.ylabel('Name of the Anime')
plt.title('Total Number of Members Watches the Anime')
plt.show()
```

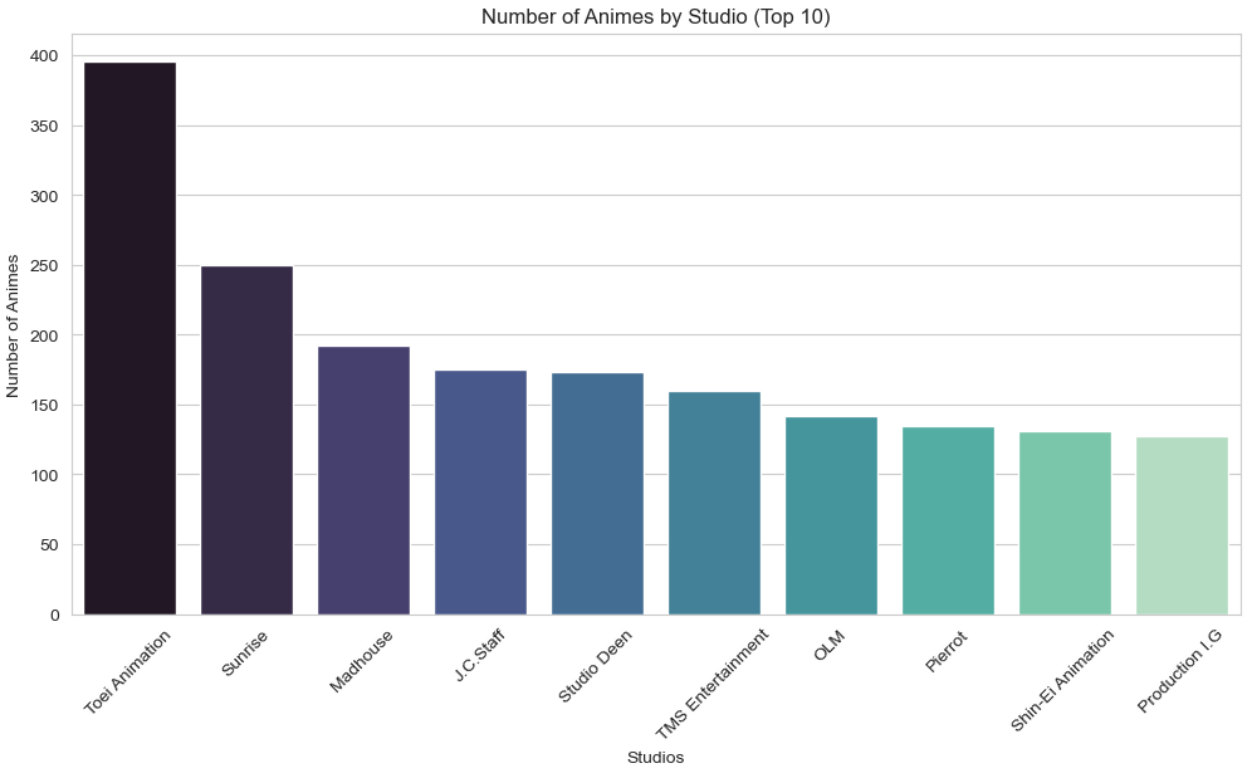


In [137…
```
# Create a list of all the individual licensors
licensors_list = [licensor.strip() for licensors in anime_df[anime_df['Li

# Count the occurrences of each licensor
licensor_counts = pd.Series(licensors_list).value_counts()

# Filter the licensor_counts series to exclude 'Unknown'
filtered_licensor_counts = licensor_counts[licensor_counts.index != 'Unkn

# Select the top 10 licensors
top_10_licensors = filtered_licensor_counts.head(10)

# Create the bar plot using Seaborn
plt.figure(figsize=(10, 6))
sns.barplot(x=top_10_licensors.values, y=top_10_licensors.index, palette=
plt.title('Top 10 Anime Licensors')
plt.xlabel('Count')
plt.ylabel('Licensors')
plt.show()
```



In [138…
```
# Count the occurrences of each studio
studio_counts = anime_df['Studios'].value_counts()

# Filter the studio_counts series to exclude 'Unknown'
studio_counts = studio_counts[studio_counts.index != 'UNKNOWN']

# Select the top 10 studios with the highest number of animes
top_studios = studio_counts.head(10)

# Create the bar plot using Seaborn
plt.figure(figsize=(12, 6))
sns.barplot(x=top_studios.index, y=top_studios.values, hue=top_studios.in
plt.title('Number of Animes by Studio (Top 10)')
```

```
plt.xlabel('Studios')
plt.ylabel('Number of Animes')
plt.xticks(rotation=45)
plt.show()
```



In [ ]: