

Anime Data Analysis

Exploratory Data Analysis and Visualization of Anime Trends using
Python (Pandas, Seaborn & Matplotlib)

By:

Hager Salah



Project Objective



The main goal of this project is to perform a comprehensive analysis of anime data using Python, covering data cleaning, preprocessing, and extracting valuable insights through visualizations.

The focus is on exploring the most popular anime, scores, genres, release years, sources, and studios — helping us understand audience preferences and industry trends.



Libraries Used & Dataset Loading

- **Libraries :**

This project uses essential Python libraries for data analysis and visualization, including: **pandas, numpy, matplotlib.pyplot, and seaborn.**

- **Dataset :**

The Anime dataset was loaded from a CSV file containing information about anime titles, including features such as **Score, Type, Source, Genre, Studios, Popularity, Members, and more.**



1) Importing Libraries & Dataset

```
[1]: #import Libraries
import numpy as np
import pandas as pd

import matplotlib.pyplot as plt
import seaborn as sns
from PIL import Image
```

```
[2]: #import dataset
Anime_df = pd.read_csv('anime.csv')    # read uploaded file >> csv
```

[3]: Anime_df

[3]:	Score	Popularity	Rank	Members	Description	Synonyms	Japanese	English	Type	Episodes	...	Premiered	Broadcast	Producers	Licensors
0	9.38	284	1	710	During their decade-long quest to defeat the D...	Frieren at the Funeral	葬送のフリーレン	Frieren: Beyond Journey's End	TV	28	...	Fall 2023	Fridays at 23:00 (JST)	Aniplex, Dentsu, Shogakukan-Shueisha Productio...	None found, add some
1	9.09	3	2	3	After a horrific alchemy experiment goes wrong...	Hagane no Renkinjutsushi: Fullmetal Alchemist,...	鋼の錬金術師 FULLMETAL ALCHEMIST	Fullmetal Alchemist: Brotherhood	TV	64	...	Spring 2009	Sundays at 17:00 (JST)	Aniplex, Square Enix, Mainichi Broadcasting Sy...	Funimation, Aniplex of America
2	9.07	13	3	2	Eccentric scientist Rintarou Okabe has a never...	NaN	STEINS;GATE	Steins;Gate	TV	24	...	Spring 2011	Wednesdays at 02:05 (JST)	Frontier Works, Media Factory, Kadokawa	Funimation

2) Exploratory Data Analysis (EDA)



- Viewed the top and bottom rows using `.head()` & `.tail()`

```
[4]: #Check the head of the DataFrame
Anime_df.head()
```

[4]:	Score	Popularity	Rank	Members	Description	Synonyms	Japanese	English	Type	Episodes	...	Premiered	Broadcast	Producers	Licensors
0	9.38	284	1	710	During their decade-long quest to defeat the D...	Frieren at the Funeral	葬送のフリーレン	Frieren: Beyond Journey's End	TV	28	...	Fall 2023	Fridays at 23:00 (JST)	Aniplex, Dentsu, Shogakukan-Shueisha Productio...	None found, add some
1	9.09	3	2	3	After a horrific alchemy experiment goes wrong...	Hagane no Renkinjutsushi: Fullmetal Alchemist...	鋼の錬金術師 FULLMETAL ALCHEMIST	Fullmetal Alchemist: Brotherhood	TV	64	...	Spring 2009	Sundays at 17:00 (JST)	Aniplex, Square Enix, Mainichi Broadcasting Sy...	Funimation, Aniplex of America
2	9.07	13	3	2	Eccentric scientist Rintarou Okabe has a never...	NaN	STEINS;GATE	Steins;Gate	TV	24	...	Spring 2011	Wednesdays at 02:05 (JST)	Frontier Works, Media Factory, Kadokawa Shoten...	Funimation
3	9.06	342	4	630	Gintoki, Shinpachi, and Kagura return as the f...	Gintama' (2015)	銀魂	Gintama Season 4	TV	51	...	Spring 2015	Wednesdays at 18:00 (JST)	TV Tokyo, Aniplex, Dentsu	Funimation, Crunchyroll
4	9.05	21	5	2	Seeking to restore humanity's diminishing hope...	NaN	進撃の巨人 Season3 Part.2	Attack on Titan Season 3 Part 2	TV	10	...	Spring 2019	Mondays at 00:10 (JST)	Production I.G, Dentsu, Mainichi Broadcasting ...	Funimation

5 rows x 22 columns

```
[5]: #Check the tail of the DataFrame
Anime_df.tail()
```

[5]:	Score	Popularity	Rank	Members	Description	Synonyms	Japanese	English	Type	Episodes	...	Premiered	Broadcast	Producers	Licensors
995	7.8	4003	996	31	Half retelling of the original Frontier series...	Macross Frontier the Movie, Gekijouban Macross...	劇場版 マクロス F 虚空歌姫 ~イツワリノウタヒメ~	Macross Frontier: The False Songstress	Movie	1	...	NaN	NaN	Bandai Visual, Big West, KlockWorx, Bandai Nam...	None found, add some
996	7.8	1658	997	136	Not so long ago, mysterious structures called ...	Magi: Adventure of Sinbad OVA	マジック・オブ・パッドの冒険	NaN	OVA	5	...	NaN	NaN	Aniplex, Shogakukan	None found, add some
997	7.8	3114	998	50	Fifteen years ago in a barren stretch of the P...	NaN	名探偵コナン 水平線上の陰謀	Detective Conan Movie 09: Strategy Above the D...	Movie	1	...	NaN	NaN	None found, add some	None found, add some
998	7.8	3194	999	47	Serial murders involving all kinds have happen...	Meitantei Conan: Senritsu no Gakufu [Full Score]	名探偵コナン 戦慄の楽譜 (フルスコア)	Detective Conan Movie 12: Full Score of Fear	Movie	1	...	NaN	NaN	Shogakukan-Shueisha Productions, Tokyo Movie S...	None found, add some
999	7.8	3598	1000	38	One foggy morning, a black and white Toyota AE...	Shin Gekijouban Initial D: Legend 1 - Kakusei	新劇場版 頭文字 [イニシャル] D Legend1 -覚醒-	Initial D Legend 1 Awakening	Movie	1	...	NaN	NaN	Shochiku, Kodansha, Ultra Super Pictures, Avex...	Sentai Filmworks

5 rows x 22 columns

2) Exploratory Data Analysis (EDA)



- Checked the dataset shape (rows and columns)

```
[6]: #rows-cols
      Anime_df.shape    #>> 1000 rows & 22 columns

[6]: (1000, 22)
```

- Used .info() to inspect column types and missing values

```
[8]: #Details
      Anime_df.info()
```

- Displayed column names using .columns

```
[10]: #Column names
       Anime_df.columns
```

```
[10]: Index(['Score', 'Popularity', 'Rank', 'Members', 'Description', 'Synonyms',
            'Japanese', 'English', 'Type', 'Episodes', 'Status', 'Aired',
            'Premiered', 'Broadcast', 'Producers', 'Licensors', 'Studios', 'Source',
            'Genres', 'Demographic', 'Duration', 'Rating'],
            dtype='object')
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 22 columns):
#   Column                Non-Null Count  Dtype  
---  -
0   Score                 1000 non-null  float64
1   Popularity            1000 non-null  int64  
2   Rank                 1000 non-null  int64  
3   Members              1000 non-null  int64  
4   Description           1000 non-null  object  
5   Synonyms              709 non-null   object  
6   Japanese              999 non-null   object  
7   English               859 non-null   object  
8   Type                 1000 non-null   object  
9   Episodes              1000 non-null   object  
10  Status                1000 non-null   object  
11  Aired                 1000 non-null   object  
12  Premiered             569 non-null   object  
13  Broadcast             569 non-null   object  
14  Producers             1000 non-null   object  
15  Licensors             1000 non-null   object  
16  Studios               1000 non-null   object  
17  Source                1000 non-null   object  
18  Genres                771 non-null   object  
19  Demographic           521 non-null   object  
20  Duration              1000 non-null   object  
21  Rating                1000 non-null   object  
dtypes: float64(1), int64(3), object(18)
memory usage: 172.0+ KB
```


2) Exploratory Data Analysis (EDA)



- Used `.describe()` for statistical summary

```
[9]: #dataset Statistics
     Anime_df.describe()
```

```
[9]:
```

	Score	Popularity	Rank	Members
count	1000.000000	1000.000000	1000.000000	1000.000000
mean	8.156580	1805.637000	500.500000	223.648000
std	0.284344	1888.308553	288.819436	246.288299
min	7.800000	1.000000	1.000000	1.000000
25%	7.920000	413.750000	250.750000	31.750000
50%	8.100000	1139.500000	500.500000	132.000000
75%	8.312500	2633.750000	750.250000	330.000000
max	9.380000	12043.000000	1000.000000	998.000000

```
[10]: Anime_df.describe(include = 'object') # categorical values
```

```
[10]:
```

	Description	Synonyms	Japanese	English	Type	Episodes	Status	Aired	Premiered	Broadcast	Producers	Licensors
count	1000	709	999	859	1000	1000	1000	1000	569	569	1000	1000
unique	999	701	953	856	6	84	2	961	129	184	641	90
top	Following their success in the qualifying round...	The Magician's Bride, Mahoyome	僕のヒーローアカデミア	Time of Eve	TV	1	Finished Airing	Oct 6, 2013 to Mar 30, 2014	Fall 2022	Unknown	None found, add some	None found, add some
freq	2	2	5	2	569	292	978	3	15	58	111	369

3) Data Preprocessing



Check Null values using
.isnull().sum()

```
[11]: #Check Null values and handle it
      Anime_df.isnull().sum()
```

```
[11]: Score          0
      Popularity     0
      Rank           0
      Members        0
      Description    0
      Synonyms       291
      Japanese        1
      English        141
      Type           0
      Episodes       0
      Status         0
      Aired          0
      Premiered      431
      Broadcast      431
      Producers      0
      Licensors      0
      Studios        0
      Source         0
      Genres         229
      Demographic    479
      Duration       0
      Rating         0
      dtype: int64
```

- Handled null values using fillna() and dropna()

```
[12]: # Hundle Null values
      ## fillna
      Anime_df['Synonyms'] = Anime_df['Synonyms'].fillna('Unknown')
      Anime_df['English'] = Anime_df['English'].fillna('Unknown')
      Anime_df['Japanese'] = Anime_df['Japanese'].fillna('Unknown')
      Anime_df['Premiered'] = Anime_df['Premiered'].fillna('Unknown')
      Anime_df['Broadcast'] = Anime_df['Broadcast'].fillna('Unknown')
      Anime_df['Genres'] = Anime_df['Genres'].fillna('Unknown')
      Anime_df['Demographic'] = Anime_df['Demographic'].fillna('Unknown')
```

```
[13]: ## dropna
      Anime_df = Anime_df.dropna()
```

```
[14]: Score          0
      Popularity     0
      Rank           0
      Members        0
      Description    0
      Synonyms       0
      Japanese       0
      English        0
      Type           0
      Episodes       0
      Status         0
      Aired          0
      Premiered      0
      Broadcast      0
      Producers      0
      Licensors      0
      Studios        0
      Source         0
      Genres         0
      Demographic    0
      Duration       0
      Rating         0
      dtype: int64
```



3) Data Preprocessing



Check Duplicates
using `.duplicated().sum()`

```
[15]: #Check Duplicate and handle it
      Anime_df.duplicated().sum()
```

```
[15]: 0
```

- Removed duplicates using `drop_duplicates()`

```
[16]: # Hundle / Drop Duplicates
      Anime_df = Anime_df.drop_duplicates()
```



Dropped unnecessary columns
(Description, Synonyms, etc.) using `.drop()`

```
[17]: #Drop unnecessary columns
      Anime_df = Anime_df.drop('Description',axis =1 )
      Anime_df = Anime_df.drop('Synonyms',axis =1 )
```

```
[18]: #Column names
      Anime_df.columns
```

```
[18]: Index(['Score', 'Popularity', 'Rank', 'Members', 'Japanese', 'English', 'Type',
          'Episodes', 'Status', 'Aired', 'Premiered', 'Broadcast', 'Producers',
          'Licensors', 'Studios', 'Source', 'Genres', 'Demographic', 'Duration',
          'Rating'],
          dtype='object')
```

```
[19]: #rows-cols
      Anime_df.shape
```

```
[19]: (1000, 20)
```



4) Data Analysis & Visualization



1) Top 10 Anime by Popularity

2) Score distribution

3) Yearwise Members

4) Most Common Anime Types

5) Top 5 Genres

6) Top Studios

7) Top 4 Anime Sources

1) Top 10 Anime by Popularity

```
[21]: # 1) Top 10 Anime by Popularity

top_10_popular = Anime_df.sort_values(by='Popularity', ascending=True).head(10)

top_10_popular[['English', 'Score', 'Popularity', 'Members', 'Type']] #select multiple columns
```

[21]:

	English	Score	Popularity	Members	Type
107	Attack on Titan	8.55	1	3	TV
79	Death Note	8.62	2	3	TV
1	Fullmetal Alchemist: Brotherhood	9.09	3	3	TV
132	One Punch Man	8.50	4	3	TV
144	Demon Slayer: Kimetsu no Yaiba	8.47	6	3	TV
875	My Hero Academia	7.86	7	3	TV
636	Naruto	8.00	8	2	TV
6	Hunter x Hunter	9.04	9	2	TV
28	Your Name.	8.84	11	2	Movie
125	Attack on Titan Season 2	8.51	12	2	TV

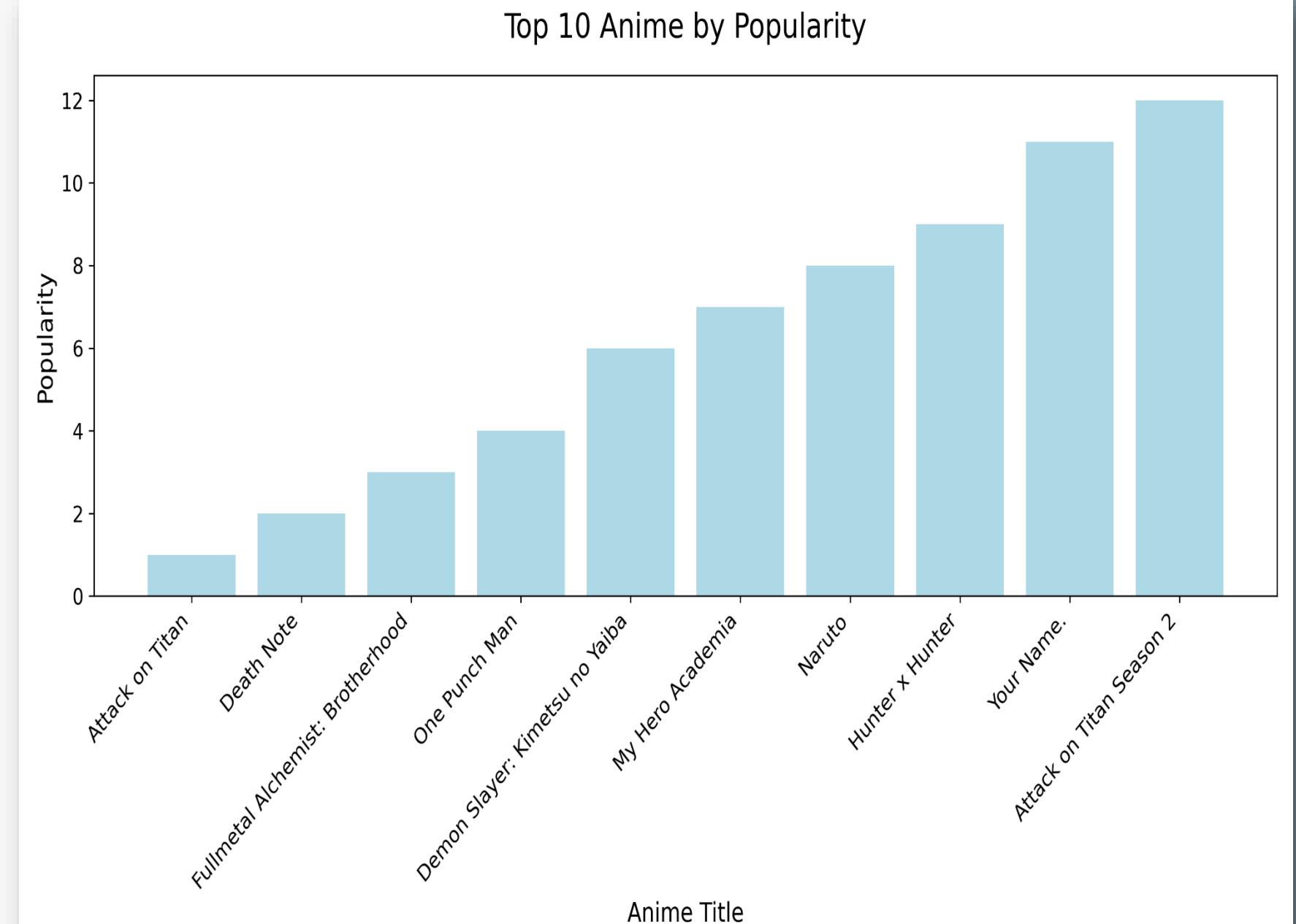
bar plot for Top 10 Anime with x-axis: English and Y-axis: Popularity

```
[22]: # Using bar
English = top_10_popular['English']
Popularity = top_10_popular['Popularity']

plt.figure(figsize=(12, 7))
plt.bar(English, Popularity, color='lightblue')
plt.title('Top 10 Anime by Popularity', fontsize=18, pad=20)
plt.xlabel('Anime Title', fontsize=14)
plt.ylabel('Popularity', fontsize=14)

plt.xticks(rotation=45, ha='right', fontsize=12) # name rotation
plt.yticks(fontsize=12)

plt.tight_layout() # Spacing coordination
plt.savefig('Top 10 Anime by Popularity', dpi=300) # Save chart as pic
plt.show()
```



2) Score distribution

```
[23]: # Score distribution >> Using describe
```

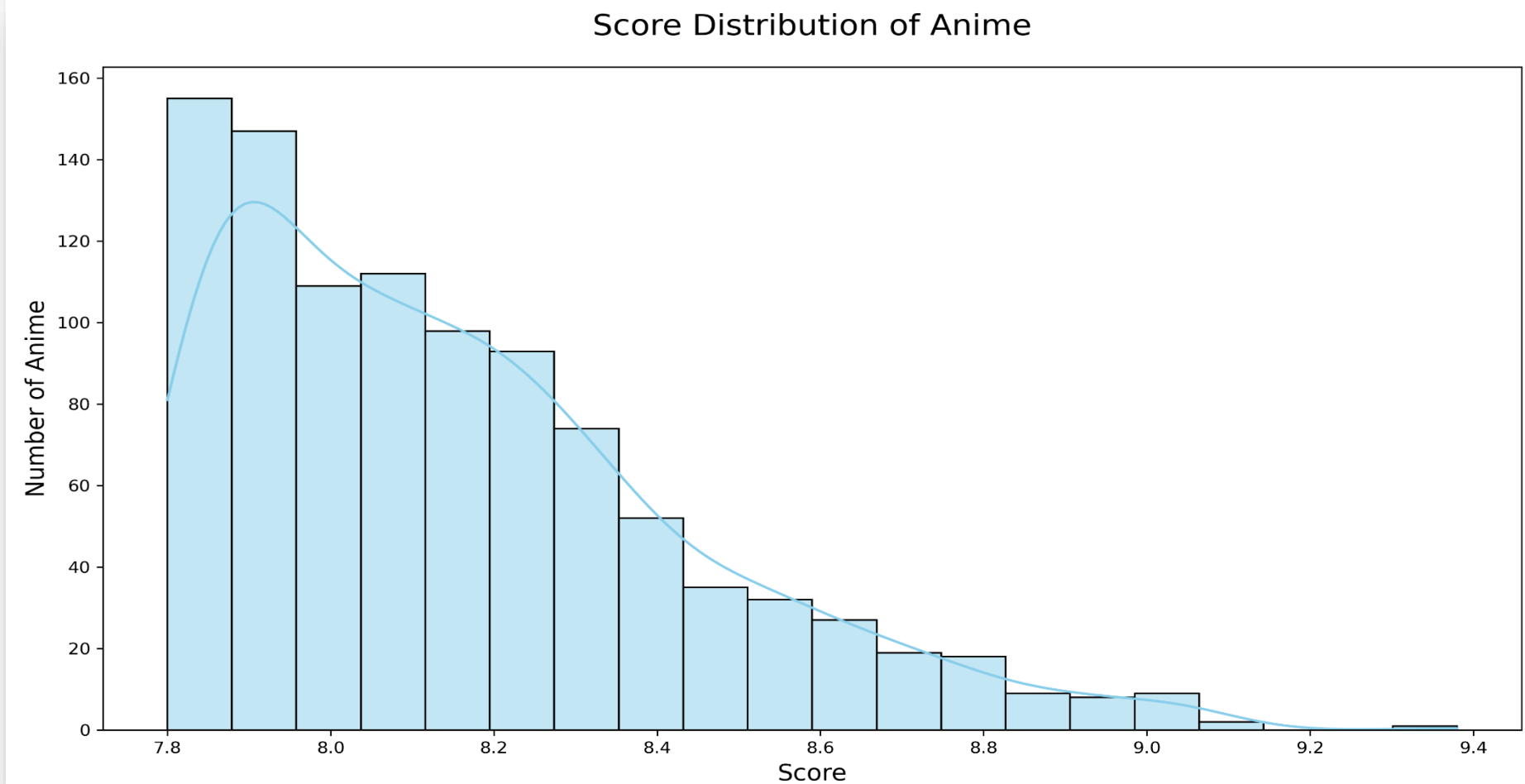
```
print("Basic Score Statistics:")
```

```
Anime_df['Score'].describe()
```

Basic Score Statistics:

```
[23]: count    1000.000000  
      mean      8.156580  
      std       0.284344  
      min       7.800000  
      25%       7.920000  
      50%       8.100000  
      75%       8.312500  
      max       9.380000  
      Name: Score, dtype: float64
```

```
[24]: # Score distribution Using ( histplot )  
plt.figure(figsize=(12, 7))  
sns.histplot(Anime_df['Score'], bins=20, kde=True, color='skyblue')  
  
plt.title('Score Distribution of Anime', fontsize=18, pad=20)  
plt.xlabel('Score', fontsize=14)  
plt.ylabel('Number of Anime', fontsize=14)  
plt.tight_layout()  
plt.savefig('Score Distribution of Anime.png', dpi=300) # Save chart as pic  
plt.show()
```



3) Yearwise Members (TOP 7 years by total numbers)

```
[26]: ##### 3) Yearwise Members #####
# TOP 7 years by total numbers

# Copy from data
df = Anime_df.copy()

# 1) Convert 'Aired' to text
df['Aired'] = df['Aired'].astype(str)

# 2) Extract first date only (before "to")
df['Aired_clean'] = df['Aired'].str.split(' to ').str[0].str.strip()

# 3) Convert to date
df['Aired_clean'] = pd.to_datetime(df['Aired_clean'], errors='coerce') # tells pandas to convert invalid or unrecognized date values
                                                                    # ('Unknown', '?', 'Not yet aired' ) into NaT instead of throwing an error.

# 4) Delete rows that contain invalid dates ( or null).
df = df.dropna(subset=['Aired_clean'])

# 5) Extract year from date (Aired_clean)
df['Aired_Year'] = df['Aired_clean'].dt.year

# 6) Group by year and sort descending
yearly_members = (
    df.groupby('Aired_Year')['Members'].sum().sort_values(ascending=False).head(7)
)

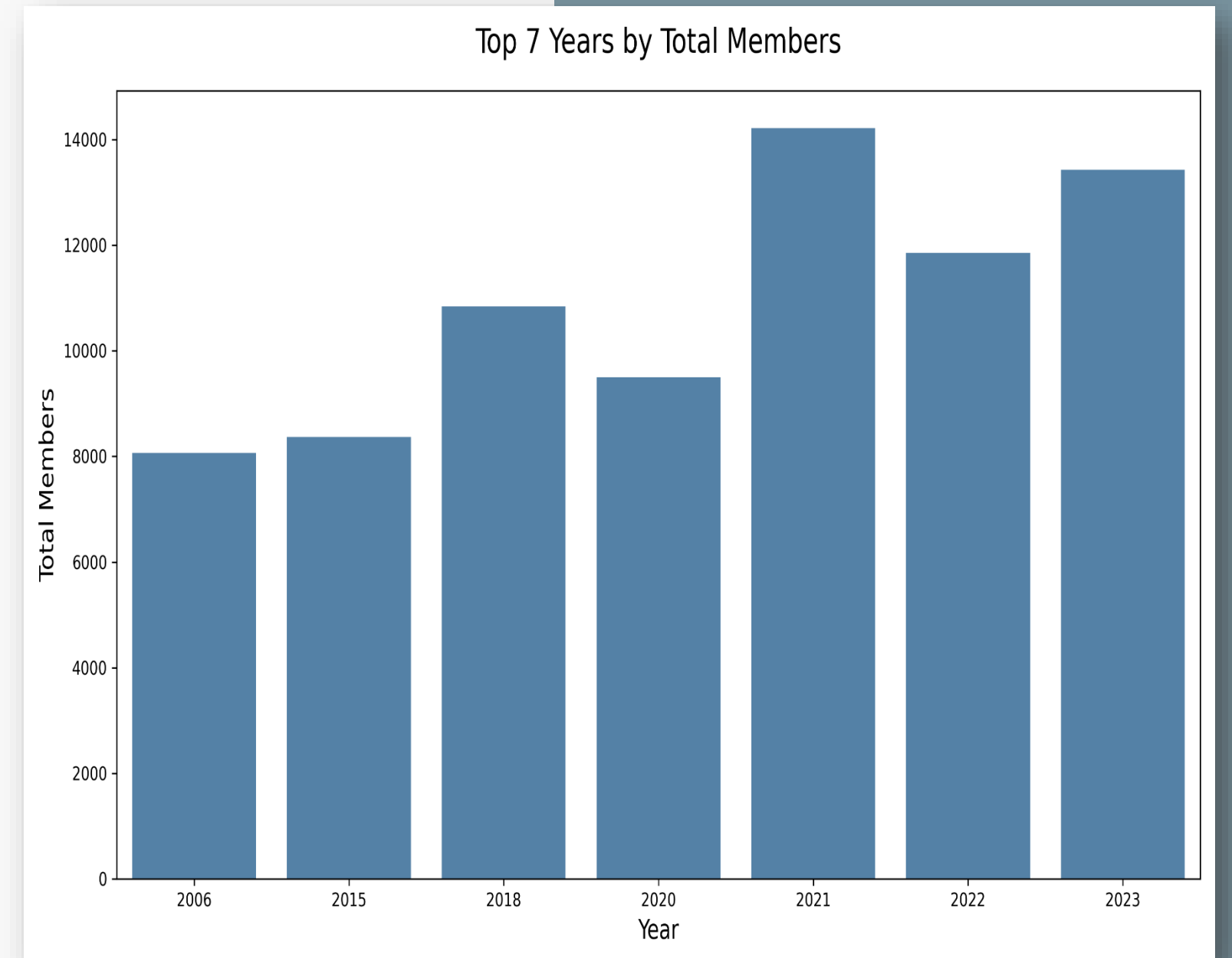
print(yearly_members)
```

Aired_Year	
2021	14211
2023	13428
2022	11855
2018	10844
2020	9499
2015	8365
2006	8060

Name: Members, dtype: int64

Visualization using barplot

```
[28]: # Using barplot
plt.figure(figsize=(12, 7))
sns.barplot(x=yearly_members.index.astype(int),
            y=yearly_members.values, color='steelblue')
plt.title('Top 7 Years by Total Members', fontsize=18, pad=20)
plt.xlabel('Year', fontsize=14)
plt.ylabel('Total Members', fontsize=14)
plt.tight_layout()
plt.savefig('Top 7 Years by Total Members', dpi=300) # Save chart as pic
plt.show()
```



4) Most Common Anime Types

```
[29]: Anime_df.groupby('Type')['English'].count().sort_values(ascending=False).head(3)
```

```
[29]: Type
TV      569
Movie   235
OVA      84
Name: English, dtype: int64
```

*Visualization
Using Pie Plot*

```
[31]: top3_type_counts = Anime_df['Type'].value_counts().head(3)

plt.figure(figsize=(12, 7))
plt.pie(
    top3_type_counts,
    labels = top3_type_counts.index,
    autopct = '%1.1f%%',          # calculate %
    startangle = 210,             # angle
    colors = sns.color_palette("Blues", n_colors=3),
    textprops = {'fontsize': 14}
)
plt.title('Most Common Anime Types', fontsize=18, pad=20)
plt.axis('equal')

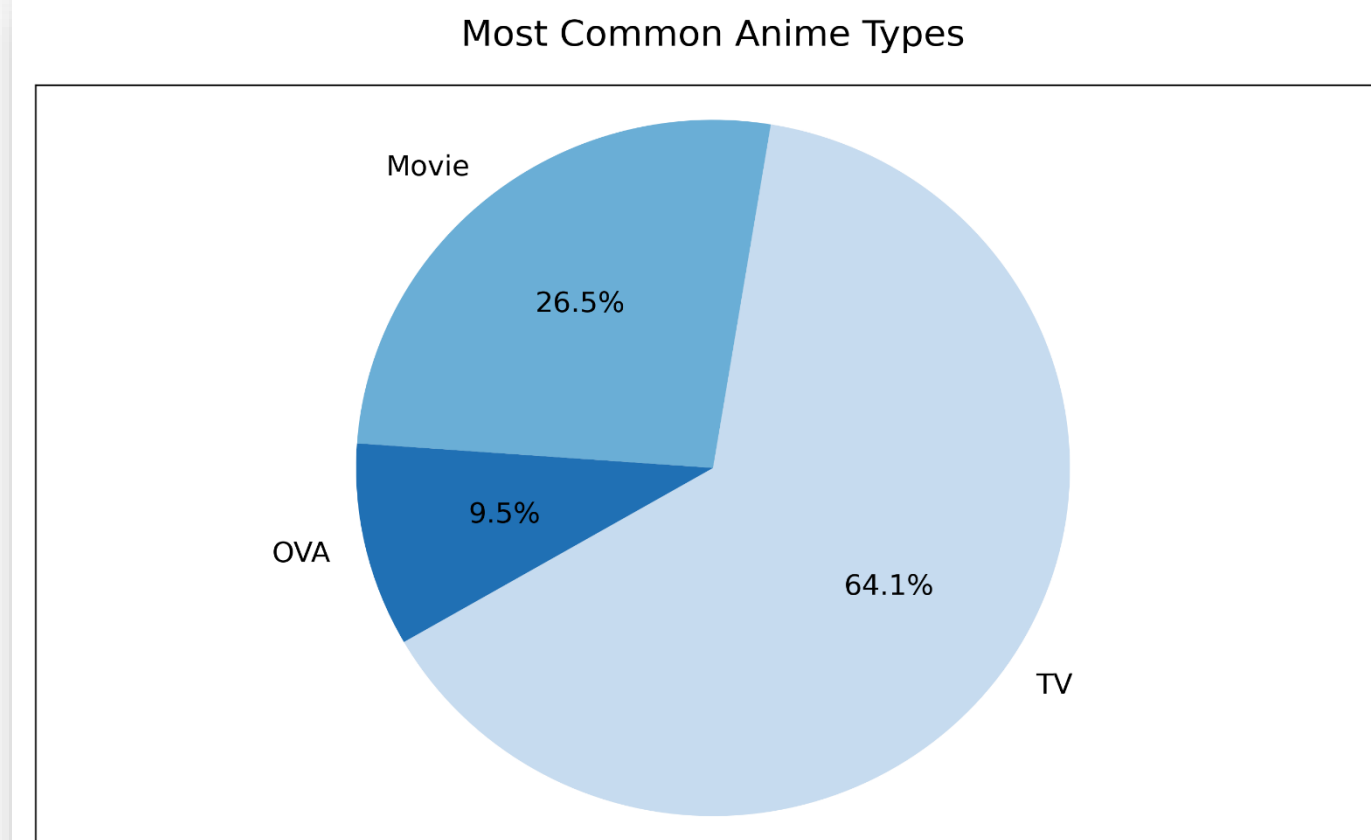
# Add border (frame) around the plot
plt.gca().set_frame_on(True)

plt.savefig('Most Common Anime Types.png', dpi=300) # Save chart as pic
plt.show()
```

■ Another Solution

```
[30]: # or
top3_type_counts = Anime_df['Type'].value_counts().head(3)
print(top3_type_counts)
```

```
Type
TV      569
Movie   235
OVA      84
Name: count, dtype: int64
```



5) Top 5 Genres

```
[32]: # Exclude Unknown
      filtered_genres = Anime_df[Anime_df['Genres'] != 'Unknown']

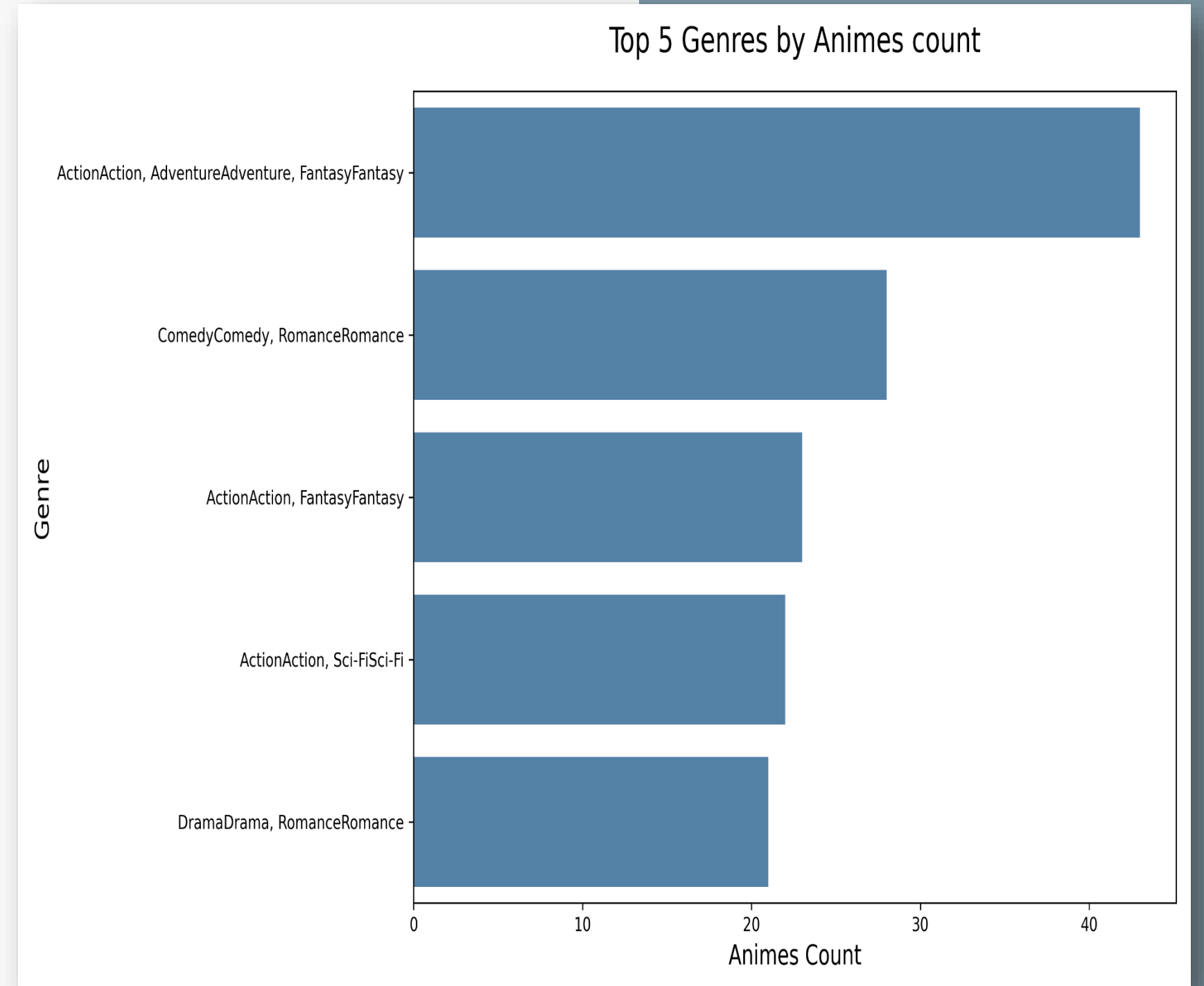
      # Top 5 Genres
      top5_genres = filtered_genres['Genres'].value_counts().head()

      print(top5_genres)
```

```
Genres
ActionAction, AdventureAdventure, FantasyFantasy    43
ComedyComedy, RomanceRomance                        28
ActionAction, FantasyFantasy                        23
ActionAction, Sci-FiSci-Fi                          22
DramaDrama, RomanceRomance                          21
Name: count, dtype: int64
```


Visualization Using bar Plot

```
[33]: #bar Plot
plt.figure(figsize=(12, 7))
sns.barplot(x=top5_genres.values, y=top5_genres.index, color='steelblue')
plt.title('Top 5 Genres by Animes count', fontsize=18, pad=20)
plt.xlabel('Animes Count', fontsize=14)
plt.ylabel('Genre', fontsize=14)
plt.tight_layout()
plt.savefig('Top 5 Genres.png', dpi=300) # Save chart as pic
plt.show()
```



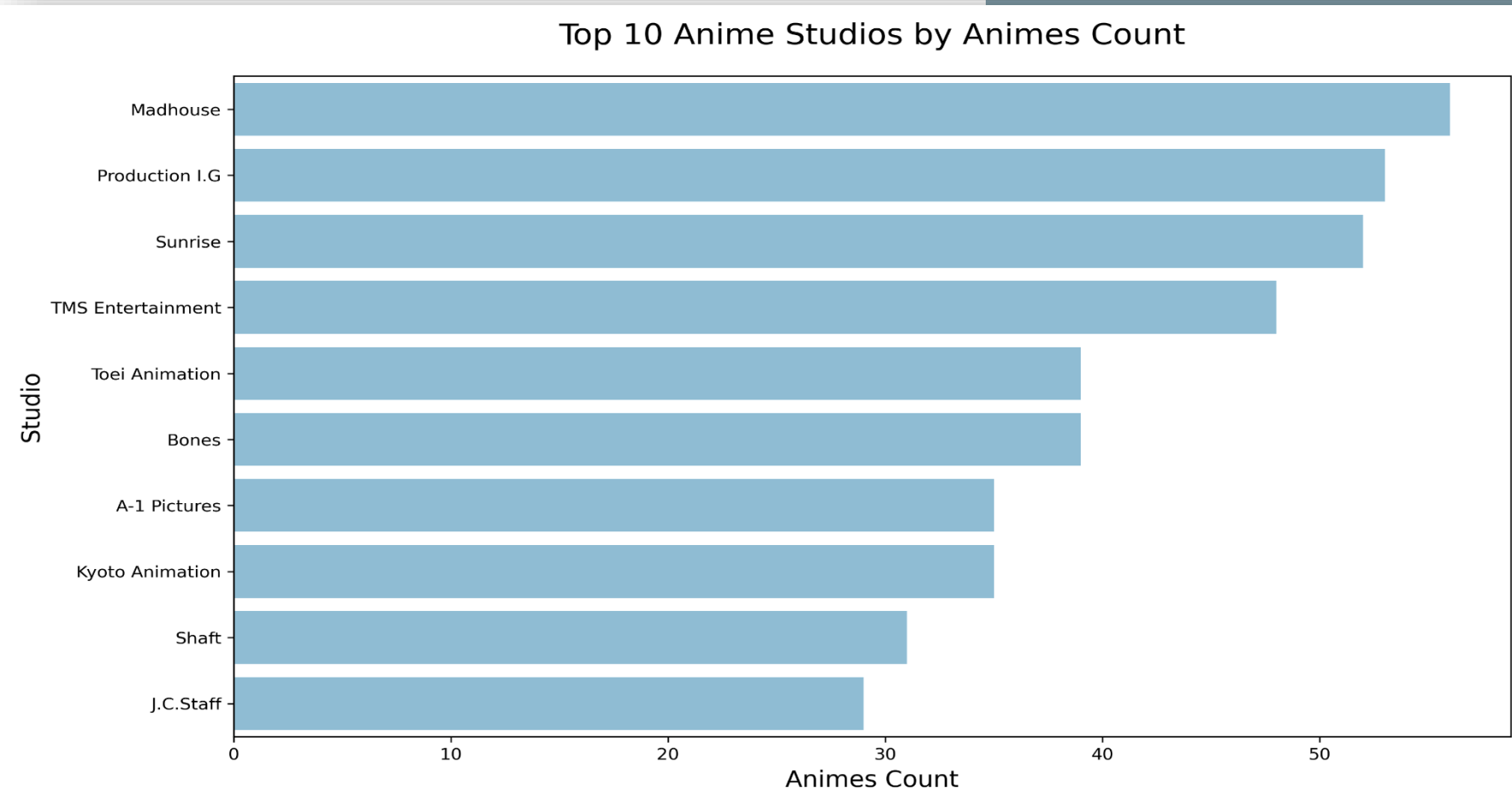
6) Top Studios

```
[34]: # 6) Top Studios
# Top 10
top_studios = Anime_df['Studios'].value_counts().head(10)
top_studios
```

```
[34]: Studios
      Madhouse      56
      Production I.G  53
      Sunrise      52
      TMS Entertainment 48
      Toei Animation  39
      Bones        39
      A-1 Pictures   35
      Kyoto Animation 35
      Shaft        31
      J.C.Staff     29
      Name: count, dtype: int64
```

■ Visualization Using bar Plot

```
[35]: #barplot
plt.figure(figsize=(12, 7))
sns.barplot(
    x=top_studios.values,
    y=top_studios.index,
    color='skyblue'
)
plt.title('Top 10 Anime Studios by Animes Count', fontsize=18, pad=20)
plt.xlabel('Animes Count', fontsize=14)
plt.ylabel('Studio', fontsize=14)
plt.tight_layout()
plt.savefig('Top 10 Anime Studios.png', dpi=300) # Save chart as pic
plt.show()
```



Top 4 Anime Sources

```
[36]: Source = Anime_df['Source'].value_counts().head(4)
Source
```

```
[36]: Source
Manga      547
Original   160
Light novel 110
Novel       63
Name: count, dtype: int64
```

■ Another Solution

```
[37]: source_counts = Anime_df.groupby('Source').size().sort_values(ascending=False).head(4)
print(source_counts)
```

```
Source
Manga      547
Original   160
Light novel 110
Novel       63
dtype: int64
```

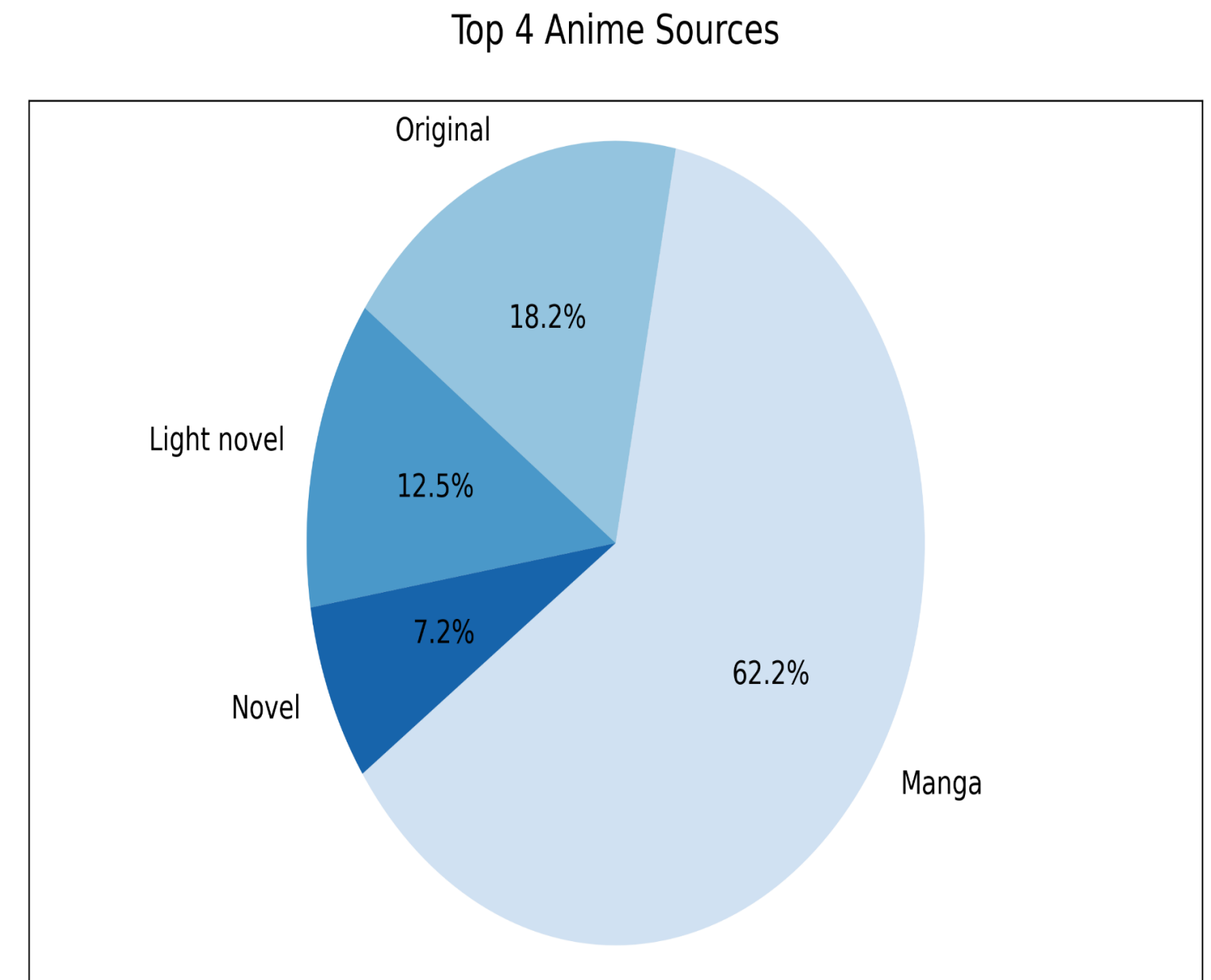

Visualization Using Pie Plot

```
[38]: #Pie Plot
plt.figure(figsize=(12, 7))

plt.pie(
    source_counts,
    labels=source_counts.index,
    autopct='%1.1f%%',
    startangle=215,
    colors=sns.color_palette("Blues", n_colors=4),
    textprops = {'fontsize': 14}
)
plt.title("Top 4 Anime Sources", fontsize=18, pad=25)
plt.axis('equal')

# Add border (frame) around the plot
plt.gca().set_frame_on(True)

plt.savefig('Top 4 Anime Sources.png', dpi=300) # Save chart as pic
plt.show()
```



Anime Dashboard

```
[39]: from PIL import Image

# 1. List of image file paths
image_paths = [
    "Top 10 Anime by Popularity.png",
    "Score Distribution of Anime.png",
    "Top 7 Years by Total Members.png",
    "Most Common Anime Types.png",
    "Top 5 Genres.png",
    "Top 10 Anime Studios.png",
    "Top 4 Anime Sources.png"
]
logo_path = "HS Logo.png"          # Logo filename

# 2. Load all main images + Logo
images = [Image.open(path) for path in image_paths]
logo = Image.open(logo_path)

# 3. Create a 2 x 4 grid
fig, axs = plt.subplots(2, 4, figsize=(32, 12))

# 4. Place each chart
for i in range(7):
    row = 0 if i < 4 else 1
    col = i if i < 4 else i - 4
    axs[row, col].imshow(images[i])
    axs[row, col].axis('off')

# 5. Insert Logo in the empty cell (row 1, col 3)
logo_resized = logo.resize((300, 300))
axs[1, 3].imshow(logo_resized)
axs[1, 3].axis('off')          # keep axes hidden

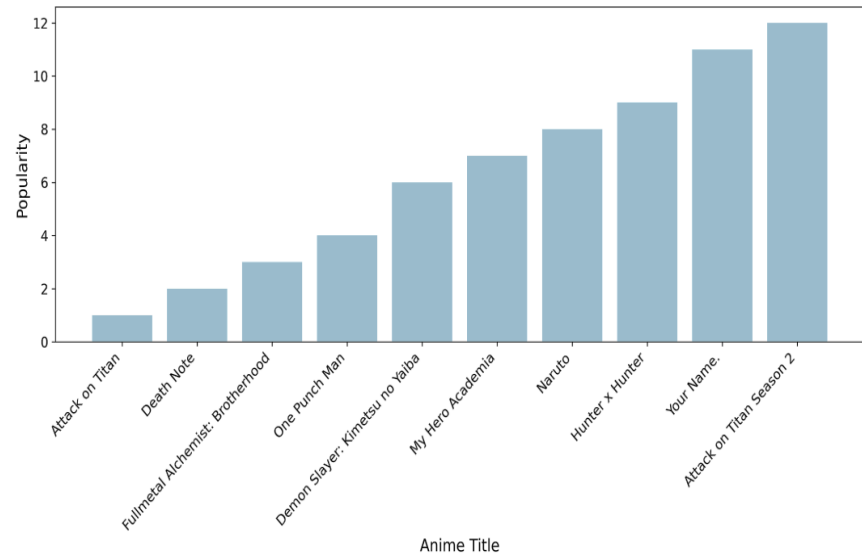
# 6. Global title
fig.suptitle("Anime Dashboard", fontsize=40, weight='bold', y=0.98)

# 7. Show & save
plt.tight_layout()
plt.show()
fig.savefig("Anime_Full_Dashboard.png", dpi=300, bbox_inches='tight')
```

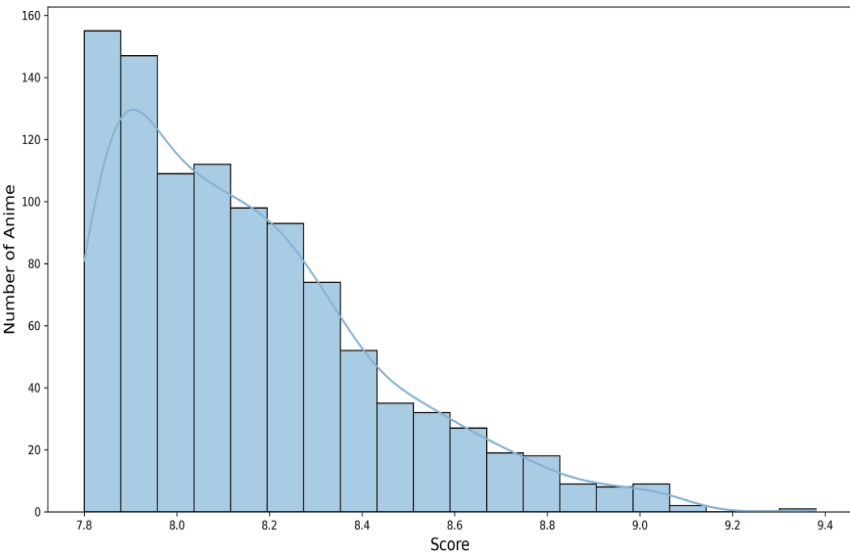
Anime Dashboard

Anime Dashboard

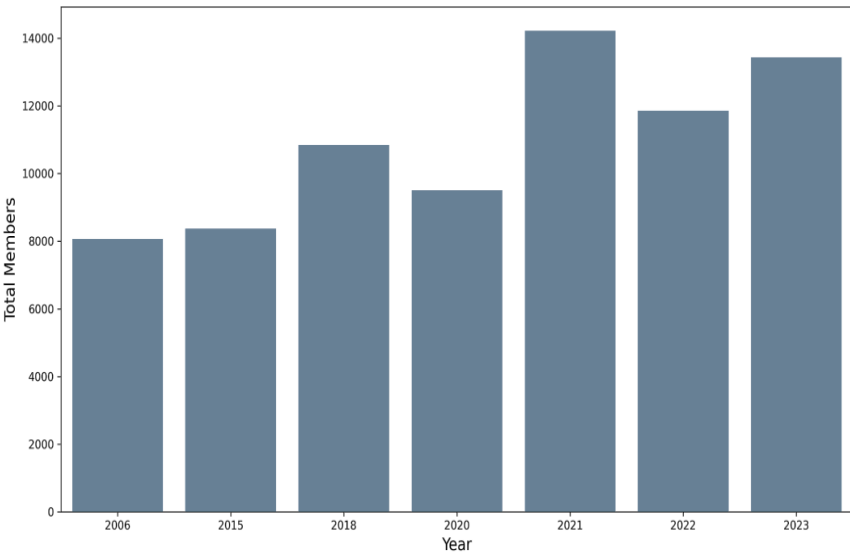
Top 10 Anime by Popularity



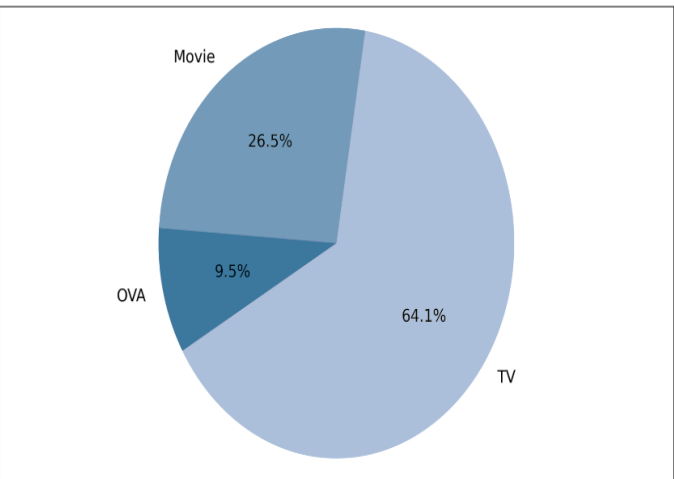
Score Distribution of Anime



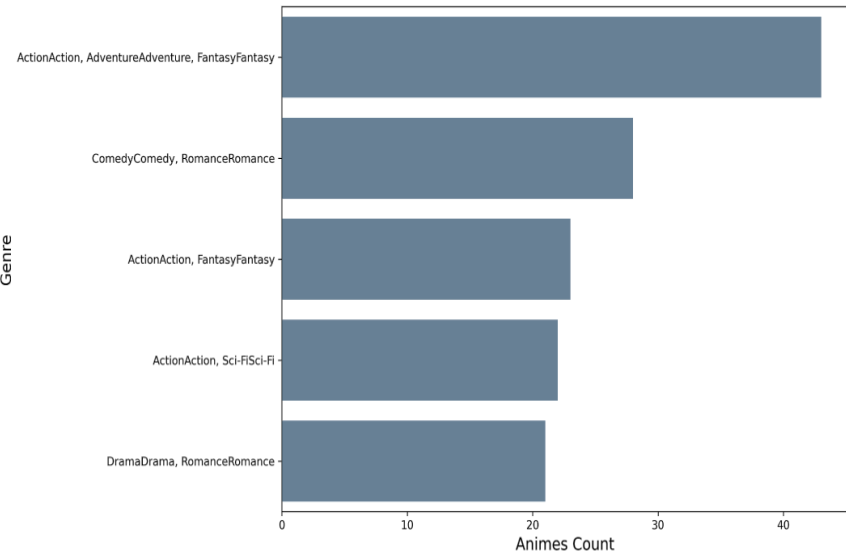
Top 7 Years by Total Members



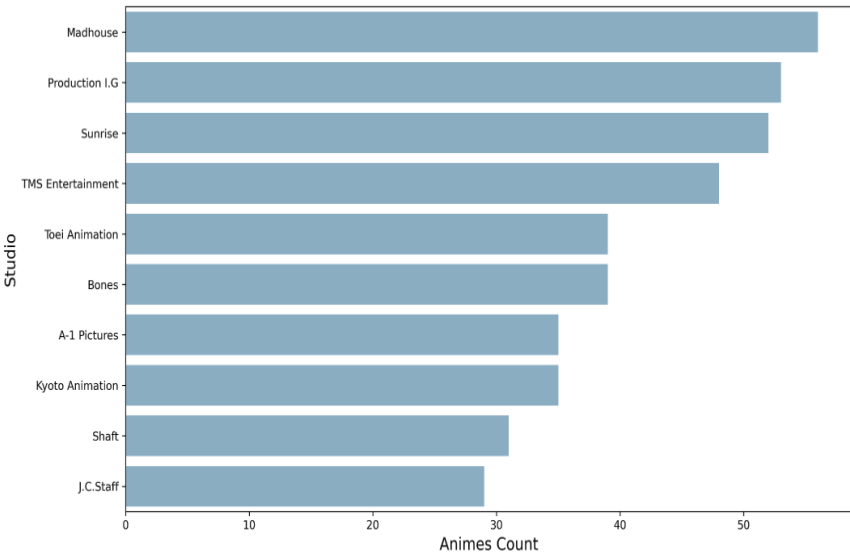
Most Common Anime Types



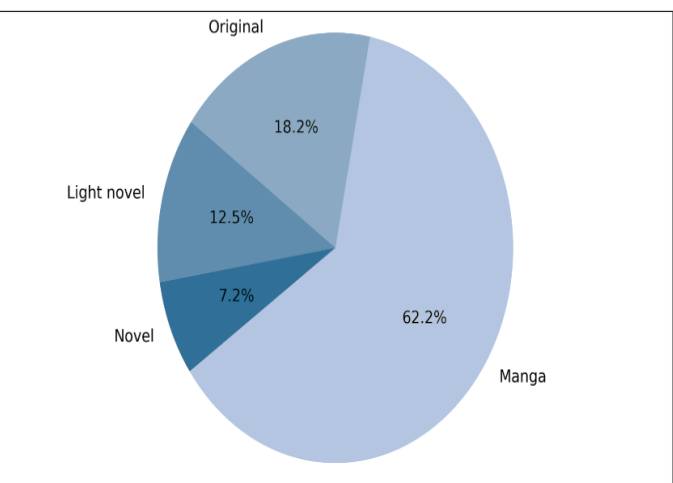
Top 5 Genres by Animes count



Top 10 Anime Studios by Animes Count



Top 4 Anime Sources



Insights

Top Anime:

- Attack on Titan and Death Note lead in popularity. Action and Drama dominate audience interest.

Scores:

- Most anime score between 7.5 – 8.5, with very few below 7.

Yearly Member Engagement:

- Highest member counts in 2021 & 2023, showing rising interest post-2018.

Anime Types:

- TV series make up 60%+ of titles. Movies and OVAs follow.

Top Genres:

- Popular genres: Action, Adventure, Fantasy, with Comedy/Romance also strong.

Top Studios:

- Madhouse, Production I.G and Sunrise rank as top producers.
- Their frequent appearance indicates their leading role in the anime industry.

Anime Sources:

- Over 60% of anime are adapted from Manga, highlighting its importance.
- Original stories, Light Novels, and Novels are also notable contributors.



Recommendations

Invest More in Popular Genres

- Since Action, Adventure, and Fantasy appear frequently in top genres and most popular anime, studios and platforms should focus more on producing and promoting anime within these categories.

TV Format Dominance

- With TV shows making up over 60% of anime types, it's clear that viewers prefer series-based formats. This suggests new anime projects should prioritize serialized content over standalone movies or OVAs.

Manga as Primary Source

- Over half of the anime in the dataset are adapted from Manga. This shows the strength of manga as a content pipeline—making it crucial for studios to continue sourcing stories from successful manga series

Capitalize on Growth Years

- The peak in member engagement from 2018 to 2023 reflects a growing global interest. This is a good opportunity for international distribution, streaming partnerships, and broader marketing.

Support Top Studios for Quality Production

- Studios like Madhouse , Production I.G and Sunrise repeatedly appear in top positions. Collaboration with or learning from such studios could help new producers maintain quality and popularity.





Thank you

