Anime Popularity & Genre Analysis

Project Goal:

- Use real-world anime data to analyse:
- What genres are most popular?
- Does rating correlate with number of episodes?
- Which studios are dominating?
- What time of year do top-rated anime release?
- Is there a formula for a "hit" anime?

Prepare the Dataset:

- Create (or collect) sample data

Tools:

- Python (Pandas, NumPy, Matplotlib, Seaborn, Plotly)
- VS Code

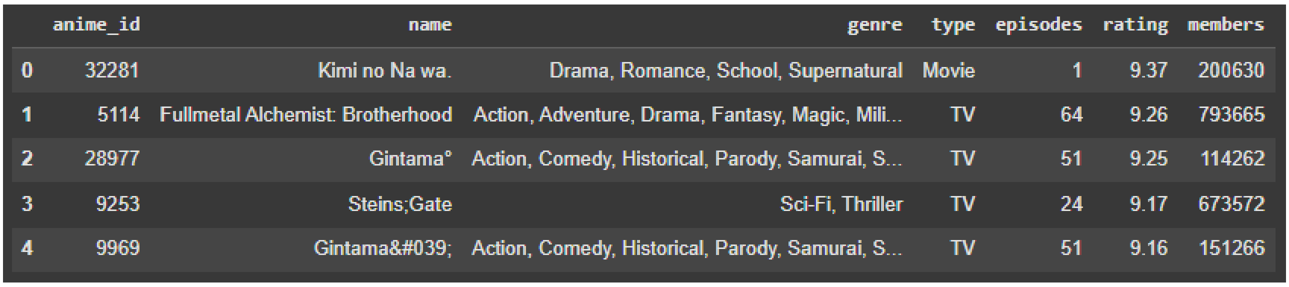
Solutions:

- #Loading data

```
import pandas as pd
df = pd.read_csv('/content/anime.csv')
```

- #shows the first 5 rows of your DataFrame

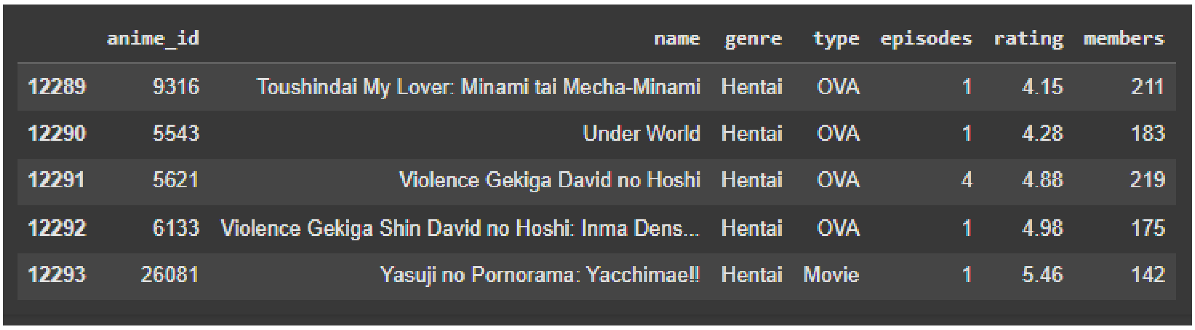
```
df.head()
```



	anime_id	name	genre	type	episodes	rating	members
0	32281	Kimi no Na wa.	Drama, Romance, School, Supernatural	Movie	1	9.37	200630
1	5114	Fullmetal Alchemist: Brotherhood	Action, Adventure, Drama, Fantasy, Magic, Mili...	TV	64	9.26	793665
2	28977	Gintama°	Action, Comedy, Historical, Parody, Samurai, S...	TV	51	9.25	114262
3	9253	Steins;Gate	Sci-Fi, Thriller	TV	24	9.17	673572
4	9969	Gintama&#039;	Action, Comedy, Historical, Parody, Samurai, S...	TV	51	9.16	151266

- #shows the bottom 5 rows of your DataFrame

```
df.tail()
```



	anime_id	name	genre	type	episodes	rating	members
12289	9316	Toushindai My Lover: Minami tai Mecha-Minami	Hentai	OVA	1	4.15	211
12290	5543	Under World	Hentai	OVA	1	4.28	183
12291	5621	Violence Gekiga David no Hoshi	Hentai	OVA	4	4.88	219
12292	6133	Violence Gekiga Shin David no Hoshi: Inma Dens...	Hentai	OVA	1	4.98	175
12293	26081	Yasuji no Pornorama: Yacchimae!!	Hentai	Movie	1	5.46	142

- #Fill missing ratings with average rating.

```
df['rating'] = df['rating'].fillna(df['rating'].mean())
```

Explanation:

So if a row has:

name	rating
Naruto	8.0
Death Note	NaN
Bleach	7.2

After running this line, it becomes:

name	rating
Naruto	8.0
Death Note	**7.6** (avg)
Bleach	7.2

- #Filling missing genres with "Unknown"

```
df['genre'] = df['genre'].fillna('Unknown')
```

Explanation:

This just saves the updated column back into the DataFrame.

You're basically replacing the original "genre" column with a cleaner version that doesn't have missing values anymore.

- #Fill missing Episodes with 0 (after converting to numeric)

```
df['episodes'] = pd.to_numeric(df['episodes'], errors='coerce').fillna(0).astype(int) # converts non- numeric to NaN
```

Explanation:

To **make sure** the episodes column is full of **clean, actual numbers** (not text like "Unknown", not blank cells), and to replace anything broken with a 0.

- #Parse genres into lists.

```
df['genre_list'] = df['genre'].apply(lambda x: x.split(', ') if x!="unknown" else[])
```

Explanation:

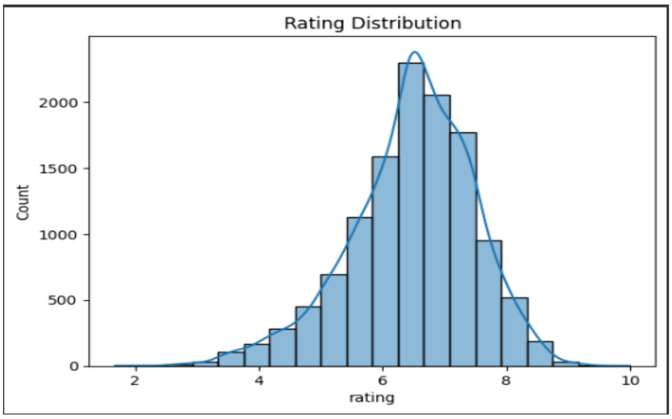
To turn the genre column (which is just a string like "Action, Comedy") into an actual **Python list** like ["Action", "Comedy"].

- #a. Top genres by frequency

```
from collections import Counter
genre_counts = Counter(genre for sublist in df['genre'].str.split(', ') for genre in sublist)
```

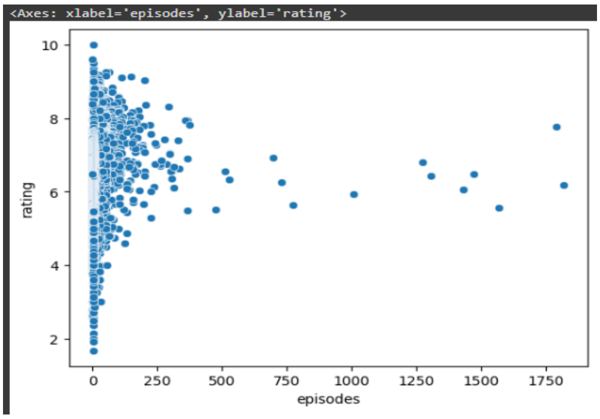
- # b. Rating distribution

```
import seaborn as sns
import matplotlib.pyplot as plt
sns.histplot(df['rating'], bins=20, kde=True)
plt.title("Rating Distribution")
plt.show()
```



- #c. Episodes vs Rating (Does binge-length affect score?)

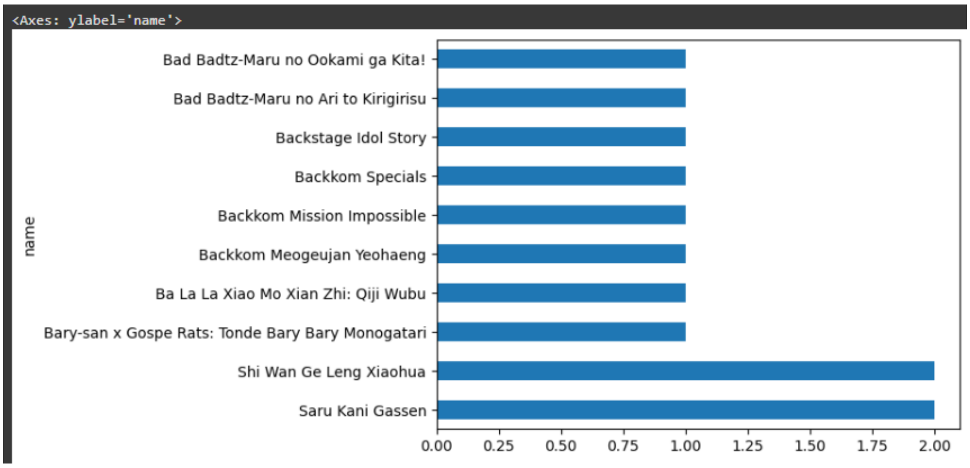
```
sns.scatterplot(data=df, x='episodes', y='rating')
```



```
<Axes: xlabel='episodes', ylabel='rating'>
```

- #d. Most popular anime

```
df['name'].value_counts().head(10).plot(kind='barh')
```



```
<Axes: ylabel='name'>
```

name
Bad Badtz-Maru no Ookami ga Kita!
Bad Badtz-Maru no Ari to Kirigirisu
Backstage Idol Story
Backkom Specials
Backkom Mission Impossible
Backkom Meogeujan Yeohaeng
Ba La La Xiao Mo Xian Zhi: Qiji Wubu
Bary-san x Gospe Rats: Tonde Bary Bary Monogatari
Shi Wan Ge Leng Xiaohua
Saru Kani Gassen
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00