data_understanding_explore_data

August 11, 2025

0.1 Explore Data

0.1.1 Anime Recommendations Database

Anime Listing

[11]:
```
import pandas as pd
```

[12]:
```
cuAnime = pd.read_csv("E:\\applied data science␣
↪capstone\\data\\CooperUnion\\archive\\anime.csv")
cuAnime.head()
```

[12]:

	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'	Action, Comedy, Historical, Parody, Samurai, S…	TV	51	9.16	151266

Name attribute will need to be cleaned to remove special characters.
Rating is what is called score in the other datasets and will be removed.

Revisiting the statistics for the anime.

[13]:
```
cuAnime.describe()
```

[13]:

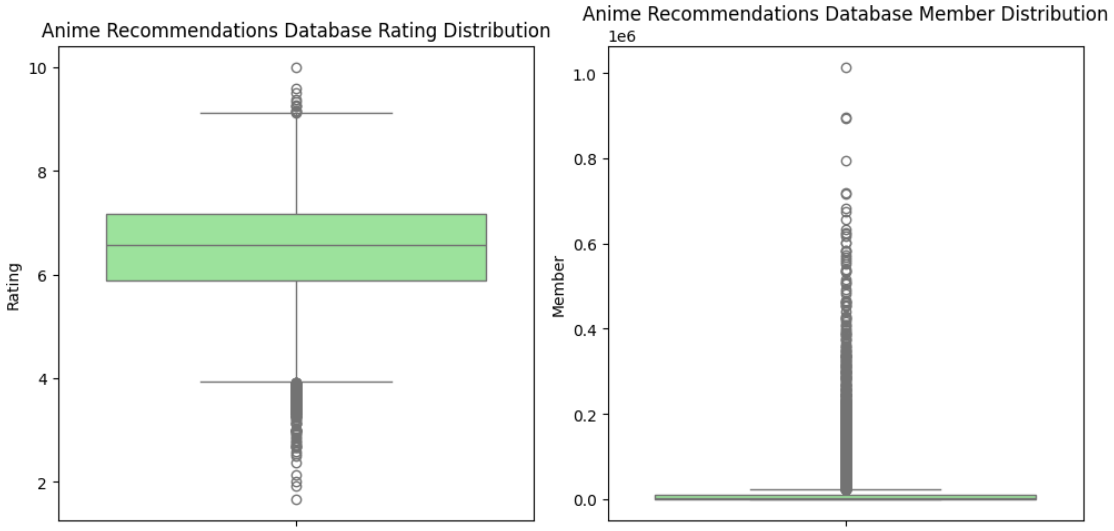
	anime_id	rating	members
count	12294.000000	12064.000000	1.229400e+04
mean	14058.221653	6.473902	1.807134e+04
std	11455.294701	1.026746	5.482068e+04
min	1.000000	1.670000	5.000000e+00
25%	3484.250000	5.880000	2.250000e+02
50%	10260.500000	6.570000	1.550000e+03
75%	24794.500000	7.180000	9.437000e+03
max	34527.000000	10.000000	1.013917e+06

[14]:
```
import matplotlib.pyplot as plt
import seaborn as sns
```

The members ditribution is heavily skewed because some anime have significantly large members which result in extreme outliers.
The ratings distribution is as expected, with most anime being average or slightly better than average.

[15]:
```
plots_location = "E:\\applied data science␣
↪capstone\\anime-recommendation\\exploration\\plots"
```

[16]:
```
plt.figure(figsize=(10, 5))

plt.subplot(1, 2, 1)
sns.boxplot(data=cuAnime["rating"], color="lightgreen")
plt.title('Anime Recommendations Database Rating Distribution')
plt.ylabel('Rating')

plt.subplot(1, 2, 2)
sns.boxplot(data=cuAnime["members"], color="lightgreen")
plt.title('Anime Recommendations Database Member Distribution')
plt.ylabel('Member')

plt.tight_layout()
plt.
↪savefig(f"{plots_location}\\anime_recommendations_database\\rating_member_distribution.
↪png")
plt.show()
```



[17]:
```
cuAnime.describe(include=object)
```

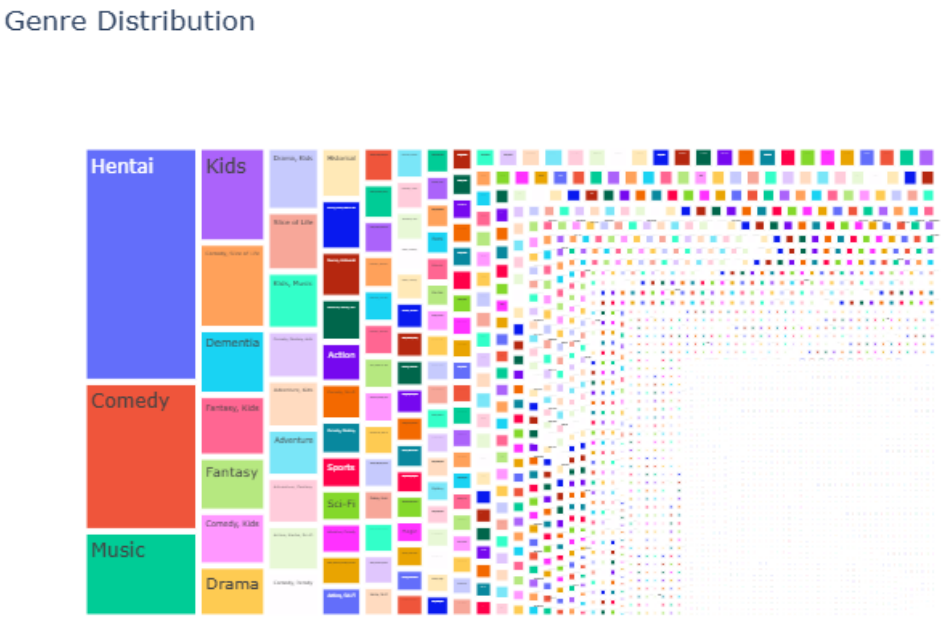
[17]:

	name	genre	type	episodes
count	12294	12232	12269	12294
unique	12292	3264	6	187
top	Saru Kani Gassen	Hentai	TV	1
freq	2	823	3787	5677

[18]:
```
import plotly.express as px
from IPython.display import Image
```

[19]:
```
genreDf = cuAnime["genre"].value_counts().to_frame()
genreDf = genreDf.reset_index()
fig = px.treemap(genreDf, path=['genre'], values='count', title="Genre␣
 ↪Distribution")
fig.
 ↪write_image(f"{plots_location}\\anime_recommendations_database\\genre_distribution.
 ↪png")
Image(filename=f"{plots_location}\\anime_recommendations_database\\genre_distribution.
 ↪png")
```

[19]:



The majority of genres are unique to a single anime. We'll filter to remove these to get a better distribution of the genres. But, we can already see that the most popular genre is Hentai and music is 3rd.

[20]:
```
genreDf.describe()
```

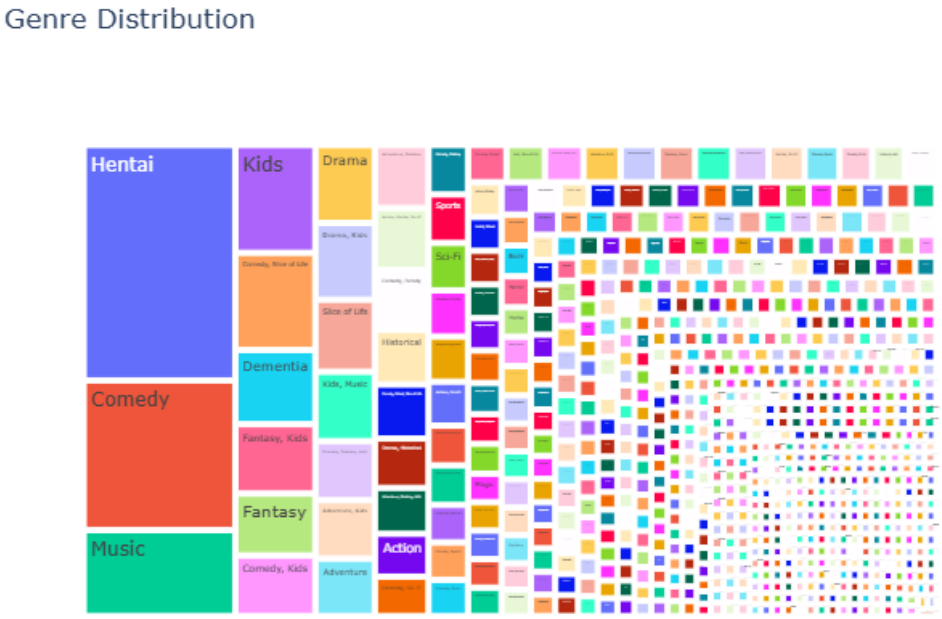
[20]:

	count
count	3264.000000
mean	3.747549
std	20.004397
min	1.000000
25%	1.000000
50%	1.000000
75%	2.000000
max	823.000000

[21]:
```
genreDf = genreDf[genreDf["count"] > 2]
fig = px.treemap(genreDf, path=['genre'], values='count', title="Genre␣
↪Distribution")
fig.
↪write_image(f"{plots_location}\\anime_recommendations_database\\genre_distribution_count_gt
↪png")
```

```
Image(filename=f"{plots_location}\\anime_recommendations_database\\genre_distribution_count_gt
 ↪png")
```

[21]:



Hentai is an undesirable genre, so it will be removed. Music may also be undesirable, myanimelist.net also includes music videos and songs in their anime dataset. Neither of which are within the scope of the recommendation system.

[22]:

```
typeDf = cuAnime["type"].value_counts().to_frame()
typeDf = typeDf.reset_index()
fig = px.treemap(typeDf, path=['type'], values='count', title="Type␣
 ↪Distribution")
fig.
 ↪write_image(f"{plots_location}\\anime_recommendations_database\\type_distribution.
 ↪png")
Image(filename=f"{plots_location}\\anime_recommendations_database\\type_distribution.
 ↪png")
```

[22]:

Looking at the type distribution, once more we can see music is a large chunk of the dataset. In fact, the only useful types are TV and Movie.
OVA(Original Video Animation) and ONA(Original Net Animation) as well as specials are typically alternate forms of the anime. So, we'll focus on TV and Movie.

[23]:
```
cuAnime[cuAnime["type"] == "OVA"].head()
```

[23]:

	anime_id	name	genre	type	episodes	rating	members
7	820	Ginga Eiyuu Densetsu	Drama, Military, Sci-Fi, Space	OVA	110	9.11	
21	44	Rurouni Kenshin: Meiji Kenkaku Romantan - Tsui…	Action, Drama, Historical, Martial Arts, Roman…	OVA	4	8.83	
41	32366	Gintama°: Aizome Kaori-hen	Comedy, Parody	OVA	2	8.69	
52	30709	Kamisama Hajimemashita: Kako-hen	Comedy, Demons, Fantasy, Shoujo, Supernatural	OVA	4	8.64	
66	777	Hellsing Ultimate	Action, Horror, Military, Seinen, Supernatural…	OVA	10	8.59	

```
7     80679
21   129307
41    16947
52    33422
66   297454
```

[24]:
```
cuAnime[cuAnime["type"] == "ONA"].head()
```

[24]:
```
     anime_id                              name  \
246      3167                       Eve no Jikan
409      6505                     There She Is!!
421     32613                  Elsword: El Lady
532     31973  Mobile Suit Gundam Thunderbolt
547     15195    Hetalia: The Beautiful World

                                        genre type episodes  rating  \
246                         Sci-Fi, Slice of Life  ONA        6    8.26
409                               Comedy, Romance  ONA        5    8.11
421                               Action, Fantasy  ONA       12    8.11
532  Action, Drama, Mecha, Military, Sci-Fi, Space  ONA        4    8.00
547                     Comedy, Historical, Parody  ONA       20    7.98

     members
246    99074
409    13935
421     3846
532    14419
547    34960
```

Similar anime for some of the ONAs

[25]:
```
cuAnime[cuAnime["name"].str.contains("Mobile Suit Gundam", case=False)].head()
```

[25]:
```
     anime_id                                              name  \
140     10937                     Mobile Suit Gundam: The Origin
154      6336                         Mobile Suit Gundam Unicorn
269      2581                              Mobile Suit Gundam 00
298      3927                  Mobile Suit Gundam 00 Second Season
308     33051  Mobile Suit Gundam: Iron-Blooded Orphans 2nd S…

                                          genre type episodes  rating  \
140  Action, Mecha, Military, Sci-Fi, Shounen, Space  OVA        6    8.42
154    Action, Drama, Mecha, Military, Sci-Fi, Space  OVA        7    8.40
269    Action, Drama, Mecha, Military, Sci-Fi, Space   TV       25    8.24
298    Action, Drama, Mecha, Military, Sci-Fi, Space   TV       25    8.21
308              Action, Drama, Mecha, Sci-Fi, Space   TV       25    8.20

     members
```

```
140    15420
154    42076
269   120351
298    86972
308    21210
```

[26]:
```
cuAnime[cuAnime["name"].str.contains("Hetalia", case=False)].head()
```

[26]:
```
     anime_id                                  name  \
527     17273  Hetalia: The Beautiful World Specials
547     15195           Hetalia: The Beautiful World
589      8479                  Hetalia World Series
710     10497          Hetalia World Series Specials
934      5060                   Hetalia Axis Powers

                          genre     type episodes  rating  members
527  Comedy, Historical, Parody  Special        4    8.00     8644
547  Comedy, Historical, Parody      ONA       20    7.98    34960
589  Comedy, Historical, Parody      ONA       48    7.95    64416
710  Comedy, Historical, Parody  Special        4    7.88    14348
934  Comedy, Historical, Parody      ONA       52    7.76   144898
```

[27]:
```
cuAnime[cuAnime["type"] == "Special"].head()
```

[27]:
```
     anime_id                                  name  \
48      21329           Mushishi Special: Hihamukage
85      24687     Mushishi Zoku Shou: Odoro no Michi
126     10863  Steins;Gate: Oukoubakko no Poriomania
162       264          Hajime no Ippo: Champion Road
171      6945             Gintama: Shiroyasha Koutan

                                              genre     type episodes  \
48   Adventure, Fantasy, Historical, Mystery, Seine…  Special        1
85   Adventure, Fantasy, Historical, Mystery, Seine…  Special        1
126                                Sci-Fi, Thriller  Special        1
162                           Comedy, Shounen, Sports  Special        1
171         Action, Comedy, Historical, Parody, Sci-Fi  Special        1

     rating  members
48     8.66    49036
85     8.54    34011
126    8.46   159548
162    8.39    47840
171    8.37    27213
```

Similar anime for some of the Specials

[28]:
```
cuAnime[cuAnime["name"].str.contains("Gintama", case=False)].head()
```

[28]:
```
    anime_id                                                name  \
2      28977                                            Gintama°
4       9969                                  Gintama&amp;#039;
8      15335  Gintama Movie: Kanketsu-hen - Yorozuya yo Eien…
9      15417                       Gintama&amp;#039;: Enchousen
12       918                                            Gintama

                                                 genre   type episodes  rating  \
2   Action, Comedy, Historical, Parody, Samurai, S…     TV       51    9.25
4   Action, Comedy, Historical, Parody, Samurai, S…     TV       51    9.16
8   Action, Comedy, Historical, Parody, Samurai, S…  Movie        1    9.10
9   Action, Comedy, Historical, Parody, Samurai, S…     TV       13    9.11
12  Action, Comedy, Historical, Parody, Samurai, S…     TV      201    9.04

    members
2    114262
4    151266
8     72534
9     81109
12   336376
```

[29]:
```
cuAnime[cuAnime["name"].str.contains("Hajime", case=False)].head()
```

[29]:
```
    anime_id                                  name  \
20       263                        Hajime no Ippo
32      5258         Hajime no Ippo: New Challenger
44     19647                 Hajime no Ippo: Rising
52     30709       Kamisama Hajimemashita: Kako-hen
54     31240  Re:Zero kara Hajimeru Isekai Seikatsu

                                        genre type episodes  rating  \
20                Comedy, Drama, Shounen, Sports   TV       75    8.83
32                Comedy, Drama, Shounen, Sports   TV       26    8.75
44                Comedy, Drama, Shounen, Sports   TV       25    8.68
52  Comedy, Demons, Fantasy, Shoujo, Supernatural  OVA        4    8.64
54       Drama, Fantasy, Psychological, Thriller   TV       25    8.64

    members
20   157670
32    88995
44    66756
52    33422
54   355839
```

There may be some anime that were solely released as ONA or OVA, but their numbers are negligible compared to the ones that are variations of a popular anime.
With the assumption that an anime consumer that watches an anime is very likely to find its variations easily, the decision has been taken to remove these variations from the dataset.

[30]:
```
cuAnime[cuAnime["type"] == "Music"].head()
```

[30]:

	anime_id	name	genre	type	episodes	rating	members
169	34240	Shelter	Music, Sci-Fi	Music	1	8.38	71136
336	731	Interstella5555: The 5tory of The 5ecret 5tar …	Adventure, Drama, Music, Sci-Fi	Music	1	8.17	31464
533	17949	The Everlasting Guilty Crown	Music	Music	1	8.00	11663
1178	2768	CLAMP in Wonderland 2	Action, Comedy, Drama, Fantasy, Magic, Music, …	Music	1	7.64	13985
1267	9930	Snow Halation	Music	Music	1	7.61	8731

The anime of type music all have 1 episode, which is an indicator that it is likely referring to a music video or song.

Anime ratings for each user

[31]:
```
cuRating = pd.read_csv("E:\\applied data science␣
↪capstone\\data\\CooperUnion\\archive\\rating.csv")
cuRating.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7813737 entries, 0 to 7813736
Data columns (total 3 columns):
 #   Column    Dtype
---  ------    -----
 0   user_id   int64
 1   anime_id  int64
 2   rating    int64
dtypes: int64(3)
memory usage: 178.8 MB
```

There are about 8 million rating entries

[32]:
```
cuRating.describe()
```

[32]:

	user_id	anime_id	rating
count	7.813737e+06	7.813737e+06	7.813737e+06

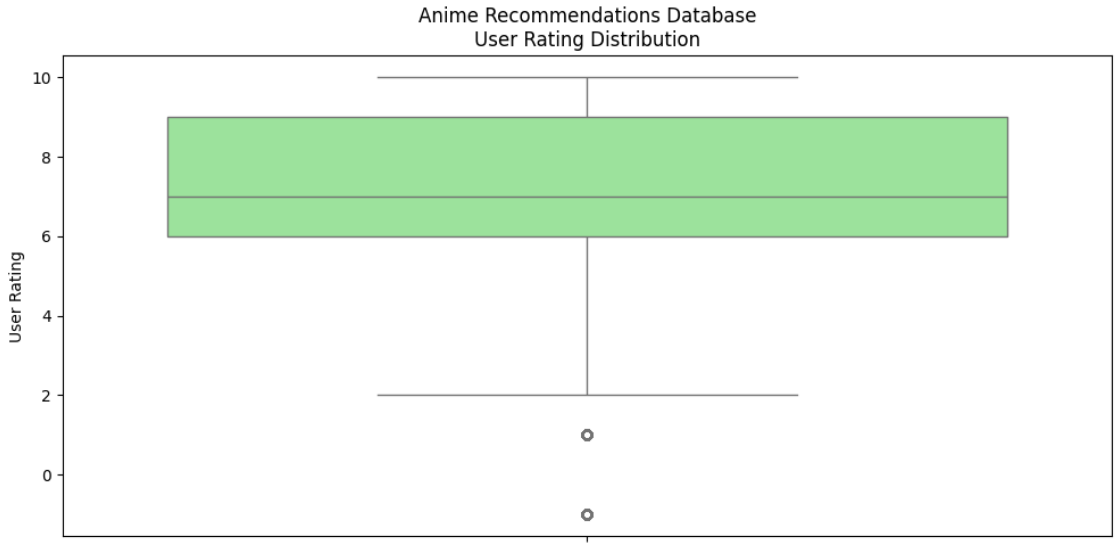
```
mean   3.672796e+04  8.909072e+03  6.144030e+00
std    2.099795e+04  8.883950e+03  3.727800e+00
min    1.000000e+00  1.000000e+00 -1.000000e+00
25%    1.897400e+04  1.240000e+03  6.000000e+00
50%    3.679100e+04  6.213000e+03  7.000000e+00
75%    5.475700e+04  1.409300e+04  9.000000e+00
max    7.351600e+04  3.451900e+04  1.000000e+01
```

[33]:
```
cuRating.isna().sum()
```

[33]:
```
user_id     0
anime_id    0
rating      0
dtype: int64
```

The only invalid values are from the rating field and these are -1. The assumption is made that -1 is a placeholder for missing values.

[34]:
```
plt.figure(figsize=(10, 5))

sns.boxplot(data=cuRating["rating"], color="lightgreen")
plt.title('Anime Recommendations Database\nUser Rating Distribution')
plt.ylabel('User Rating')

plt.tight_layout()
plt.
  ↪savefig(f"{plots_location}\\anime_recommendations_database\\user_rating_distribution.
  ↪png")
plt.show()
```



[35]:
```
cuRating[cuRating["rating"] < 0].count()/cuRating.shape[0]
```

[35]:
```
user_id     0.188962
anime_id    0.188962
rating      0.188962
dtype: float64
```

[36]:
```
cuRating[cuRating["rating"] < 0].count()
```

[36]:
```
user_id     1476496
anime_id    1476496
rating      1476496
dtype: int64
```

So approximately 1 out of every 5 rating is missing. Removing the data is an option, but the missing values could also be replaced with the median values seeing as the disribution is skewed. The recommendation system to be built is one based on collaborative filtering, so dropping the missing values is an acceptable solution.

We could use clustering to visualize the missing data to see if there are similarities for the anime not rated

0.1.2 Anime Recommendations Database 2020

Anime Listing

[37]:
```
hernanAnime = pd.read_csv("E:\\applied data science␣
↪capstone\\data\\hernan4444\\archive\\anime-hernan.csv")
hernanAnime.head()
```

[37]:

	MAL_ID	Name	Score	Genres	English name	Japanese name	Type	Episodes
0	1	Cowboy Bebop	8.78	Action, Adventure, Comedy, Drama, Sci-Fi, Space	Cowboy Bebop		TV	26
1	5	Cowboy Bebop: Tengoku no Tobira	8.39	Action, Drama, Mystery, Sci-Fi, Space	Cowboy Bebop:The Movie		Movie	1
2	6	Trigun	8.24	Action, Sci-Fi, Adventure, Comedy, Drama, Shounen	Trigun		TV	26
3	7	Witch Hunter Robin	7.27	Action, Mystery, Police, Supernatural, Drama, …	Witch Hunter Robin	Witch Hunter ROBIN ( )	TV	26
4	8	Bouken Ou Beet	6.98	Adventure, Fantasy, Shounen, Supernatural	Beet the Vandel Buster		TV	52

```
                          Aired    Premiered  …  Score-10   Score-9  \
0   Apr 3, 1998 to Apr 24, 1999  Spring 1998  …  229170.0  182126.0
1                   Sep 1, 2001      Unknown  …   30043.0   49201.0
2   Apr 1, 1998 to Sep 30, 1998  Spring 1998  …   50229.0   75651.0
3   Jul 2, 2002 to Dec 24, 2002  Summer 2002  …    2182.0    4806.0
4  Sep 30, 2004 to Sep 29, 2005    Fall 2004  …     312.0     529.0

    Score-8  Score-7  Score-6 Score-5 Score-4  Score-3  Score-2  Score-1
0  131625.0  62330.0  20688.0  8904.0  3184.0   1357.0    741.0   1580.0
1   49505.0  22632.0   5805.0  1877.0   577.0    221.0    109.0    379.0
2   86142.0  49432.0  15376.0  5838.0  1965.0    664.0    316.0    533.0
3   10128.0  11618.0   5709.0  2920.0  1083.0    353.0    164.0    131.0
4    1242.0   1713.0   1068.0   634.0   265.0     83.0     50.0     27.0

[5 rows x 35 columns]
```

The scores attributes will not be particularly useful for the models that will be built as recommendations will be based on user ratings.

[38]:
```
hernanAnime.describe()
```

[38]:
```
             MAL_ID    Popularity       Members      Favorites       Watching  \
count  17562.000000  17562.000000  1.756200e+04   17562.000000   17562.000000
mean   21477.192347   8763.452340  3.465854e+04     457.746270    2231.487758
std    14900.093170   5059.327278  1.252821e+05    4063.473313   14046.688133
min        1.000000      0.000000  1.000000e+00       0.000000       0.000000
25%     5953.500000   4383.500000  3.360000e+02       0.000000      13.000000
50%    22820.000000   8762.500000  2.065000e+03       3.000000      73.000000
75%    35624.750000  13145.000000  1.322325e+04      31.000000     522.000000
max    48492.000000  17565.000000  2.589552e+06  183914.000000  887333.000000

          Completed        On-Hold        Dropped  Plan to Watch
count  1.756200e+04   17562.000000   17562.000000   17562.000000
mean   2.209557e+04     955.049653    1176.599533    8199.831227
std    9.100919e+04    4275.675096    4740.348653   23777.691963
min    0.000000e+00       0.000000       0.000000       1.000000
25%    1.110000e+02       6.000000      37.000000     112.000000
50%    8.175000e+02      45.000000      77.000000     752.500000
75%    6.478000e+03     291.750000     271.000000    4135.500000
max    2.182587e+06  187919.000000  174710.000000  425531.000000
```

None of these attributes are particularly useful for the recommendation system to be built. They will likely be dropped with the exception of MAL_ID. This will need to be renamed to anime_id for consistency across the datasets.
In addition, the other datasets do not track all these fields.

[39]:
```
hernanAnime.describe(include=object)
```

[39]:

	Name	Score	Genres	\
count	17562	17562	17562	
unique	17558	533	5034	
top	Maou Gakuin no Futekigousha: Shijou Saikyou no…	Unknown	Hentai	
freq	3	5141	969	

	English name	Japanese name	Type	Episodes	Aired	Premiered	\
count	17562	17562	17562	17562	17562	17562	
unique	6831	16679	7	201	11947	231	
top	Unknown	Unknown	TV	1	Unknown	Unknown	
freq	10565	48	4996	8381	309	12817	

	Producers	…	Score-10	Score-9	Score-8	Score-7	Score-6	Score-5	\
count	17562	…	17562	17562	17562	17562	17562	17562	
unique	3783	…	3379	3645	4515	4933	4236	3288	
top	Unknown	…	4.0	Unknown	Unknown	2.0	Unknown	Unknown	
freq	7794	…	936	3167	1371	610	511	584	

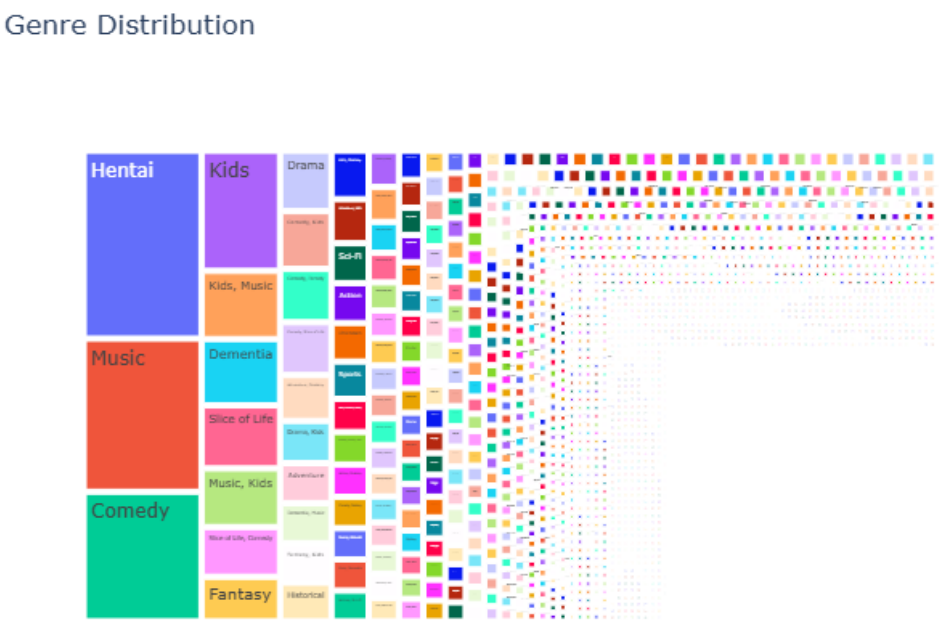
	Score-4	Score-3	Score-2	Score-1
count	17562	17562	17562	17562
unique	2235	1506	1110	1084
top	Unknown	Unknown	Unknown	4.0
freq	977	1307	1597	955

[4 rows x 26 columns]

There is an anime that is repeated 3 times.
There are many unknown values across the attributes.
Many of the anime do not have a value for english name. This will be a problem as the audience is for english speakers. Will need to be able to reconcile these english names.
Once more hentai is a popular genre.
About half the anime have only 1 episode, could be another case where many of the anime are music.
Aired, Premiered, Producers, Licensors and Studios have a lot of unknown values.
Duration is expressed as a category, but this will be need to be converted to integers and possibly apply some binning.
As mentioned, the other scores attributes will be disregarded.

[40]:
```
genre_df = hernanAnime["Genres"].value_counts().to_frame()
genre_df = genre_df.reset_index()
fig = px.treemap(genre_df, path=['Genres'], values='count', title="Genre␣
 ↪Distribution")
fig.
 ↪write_image(f"{plots_location}\\anime_recommendations_database_2020\\genre_distribution.
 ↪png")
Image(filename=f"{plots_location}\\anime_recommendations_database_2020\\genre_distribution.
 ↪png")
```

[40]:



Once more Hentai and music take top spot.
Both of these will need to be removed.

[41]:
```
genre_df.describe()
```

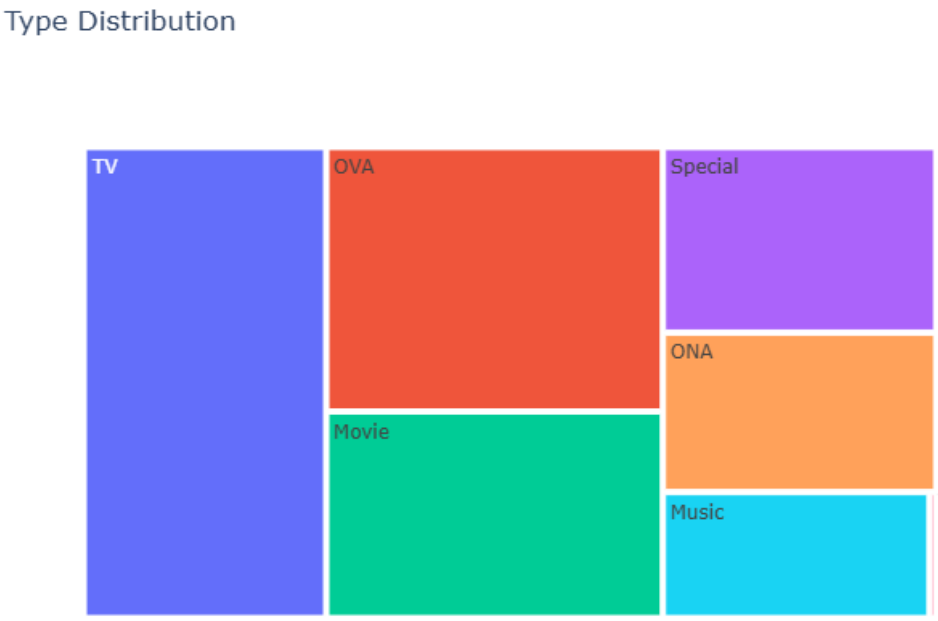
[41]:

	count
count	5034.000000
mean	3.488677
std	22.824897
min	1.000000
25%	1.000000
50%	1.000000
75%	2.000000
max	969.000000

[42]:
```
type_df = hernanAnime["Type"].value_counts().to_frame()
type_df = type_df.reset_index()
fig = px.treemap(type_df, path=['Type'], values='count', title="Type␣
↪Distribution")
fig.
↪write_image(f"{plots_location}\\anime_recommendations_database_2020\\type_distribution.
↪png")
```

```
Image(filename=f"{plots_location}\\anime_recommendations_database_2020\\type_distribution.
↪png")
```

[42]:



Much like the previous dataset, we will only focus on Tv and movie. There seems to be an additional unknown category.

[43]:
```
hernanAnime[hernanAnime["Type"] == "Unknown"].head()
```

[43]:

	MAL_ID	Name	Score	Genres	English name
9074	24023	Project758	Unknown	Drama	Unknown
10552	30448	Mirai Arise	Unknown	Sci-Fi	Unknown
10557	30455	Kantai Collection: KanColle Zoku-hen	Unknown	Action, Military, Sci-Fi, Slice of Life, School	Unknown
12107	33852	Mekakucity Reload	Unknown	Sci-Fi, Comedy, Super Power, Supernatural, Rom…	Unknown
13241	35737	Pluto	Unknown	Action, Sci-Fi, Mystery, Psychological, Mecha,…	Pluto

```
          Japanese name     Type Episodes      Aired Premiered  … Score-10  \
9074          PROJECT758  Unknown  Unknown    Unknown   Unknown  …  Unknown
10552                    Unknown  Unknown    Unknown   Unknown  …  Unknown
10557         - -    Unknown  Unknown  2022 to ?   Unknown  …  Unknown
12107  MEKAKUCITY RELOAD  Unknown  Unknown    Unknown   Unknown  …      1.0
13241                    Unknown  Unknown    Unknown   Unknown  …  Unknown

       Score-9  Score-8  Score-7  Score-6  Score-5  Score-4  Score-3  Score-2  \
9074   Unknown  Unknown  Unknown  Unknown  Unknown  Unknown  Unknown  Unknown
10552  Unknown  Unknown  Unknown  Unknown  Unknown  Unknown  Unknown  Unknown
10557  Unknown  Unknown  Unknown  Unknown  Unknown  Unknown  Unknown  Unknown
12107  Unknown  Unknown  Unknown  Unknown  Unknown  Unknown  Unknown  Unknown
13241  Unknown  Unknown  Unknown  Unknown  Unknown  Unknown  Unknown  Unknown

       Score-1
9074   Unknown
10552  Unknown
10557  Unknown
12107  Unknown
13241  Unknown

[5 rows x 35 columns]
```

The unknown types seem to have a lot of missing data, but as long as they have corresponding user ratings they could still prove to be useful.

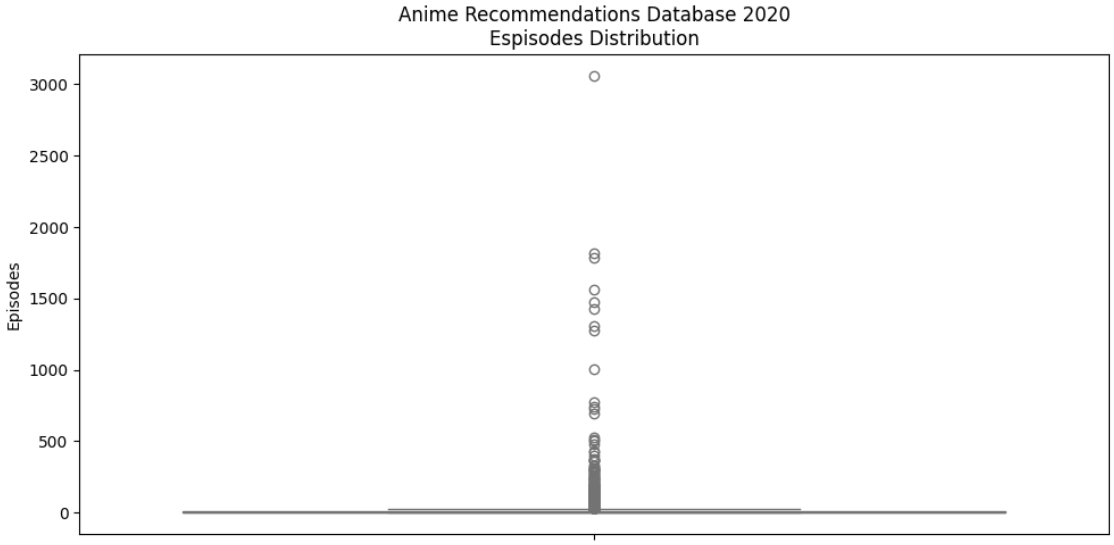
Episodes contain some unknown values which seems to be why it is not recognized as a numeric type.

[44]:
```
hernanAnime["Episodes"] = pd.to_numeric(hernanAnime["Episodes"],␣
 ↪errors='coerce')
hernanAnime["Episodes"].describe()
```

[44]:
```
count    17046.000000
mean        11.525519
std         47.348640
min          1.000000
25%          1.000000
50%          2.000000
75%         12.000000
max       3057.000000
Name: Episodes, dtype: float64
```

[45]:
```
plt.figure(figsize=(10, 5))

sns.boxplot(data=hernanAnime["Episodes"], color="lightgreen")
plt.title('Anime Recommendations Database 2020\nEspisodes Distribution')
plt.ylabel('Episodes')
```

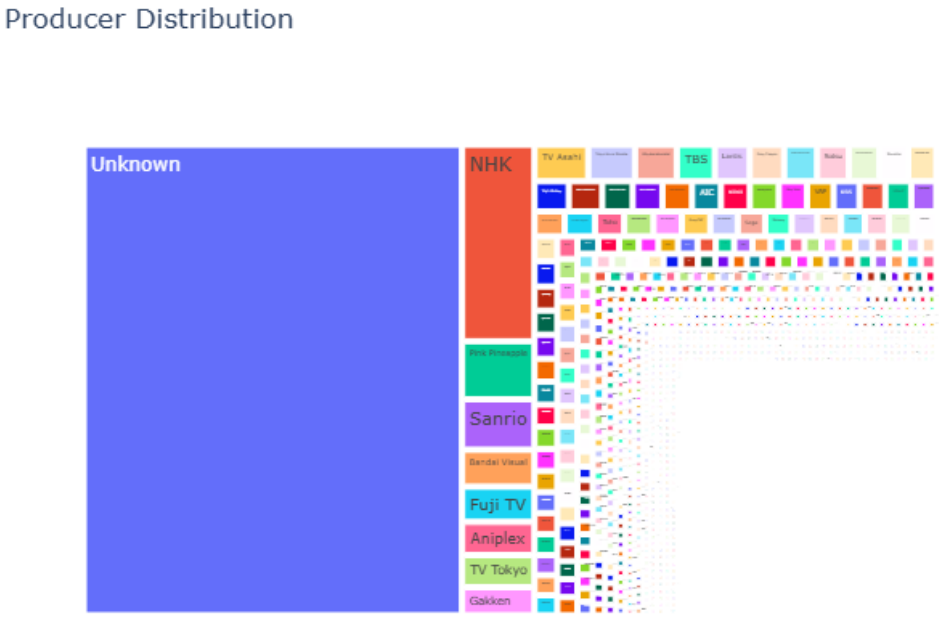
```
plt.tight_layout()
plt.
 ↪savefig(f"{plots_location}\\anime_recommendations_database_2020\\episodes_distribution.
 ↪png")
plt.show()
```



Episodes have some extreme outliers that skew the distribution. It will probably be useful to bin the episodes count.

[46]:
```
producer_df = hernanAnime["Producers"].value_counts().to_frame()
producer_df = producer_df.reset_index()
fig = px.treemap(producer_df, path=['Producers'], values='count',␣
 ↪title="Producer Distribution")
fig.
 ↪write_image(f"{plots_location}\\anime_recommendations_database_2020\\producer_distribution.
 ↪png")
Image(filename=f"{plots_location}\\anime_recommendations_database_2020\\producer_distribution.
 ↪png")
```

[46]:

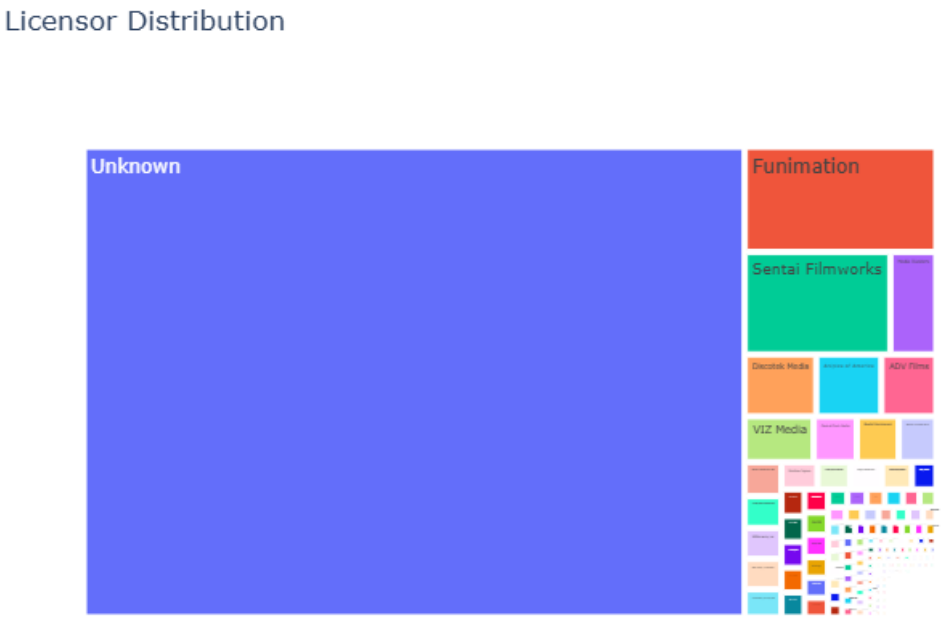


[47]:
```
hernanAnime.loc[hernanAnime["Producers"] == "Unknown", "Producers"].count() /␣
↪hernanAnime.shape[0]
```

[47]:
```
np.float64(0.4437991117184831)
```

[48]:
```
licensor_df = hernanAnime["Licensors"].value_counts().to_frame()
licensor_df = licensor_df.reset_index()
fig = px.treemap(licensor_df, path=['Licensors'], values='count',␣
↪title="Licensor Distribution")
fig.
↪write_image(f"{plots_location}\\anime_recommendations_database_2020\\licensor_distribution.
↪png")
Image(filename=f"{plots_location}\\anime_recommendations_database_2020\\licensor_distribution.
↪png")
```

[48]:

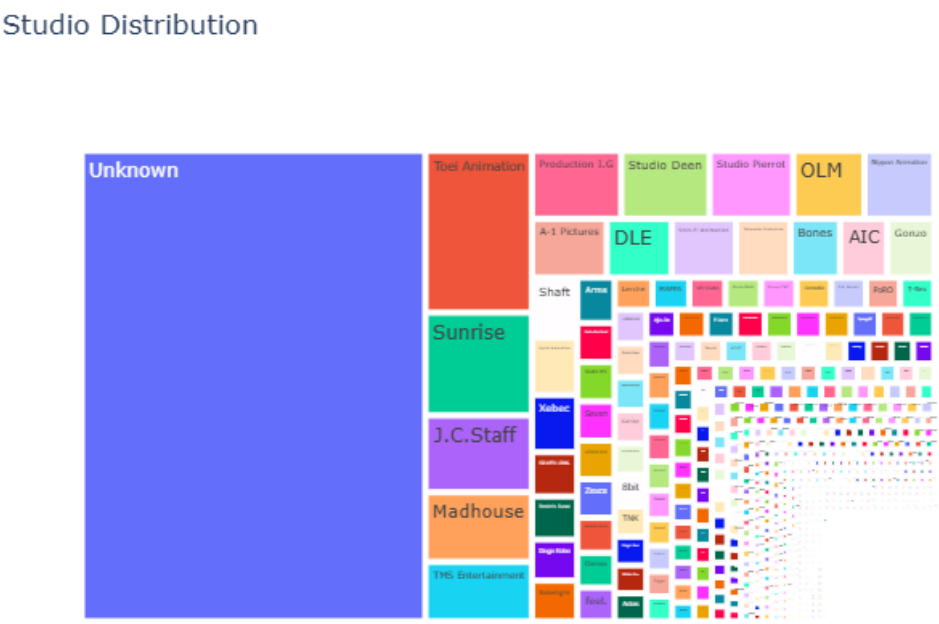


[49]:
```
hernanAnime.loc[hernanAnime["Licensors"] == "Unknown", "Licensors"].count() /␣
↪hernanAnime.shape[0]
```

[49]:
```
np.float64(0.7753103291196902)
```

[50]:
```
studio_df = hernanAnime["Studios"].value_counts().to_frame()
studio_df = studio_df.reset_index()
fig = px.treemap(studio_df, path=['Studios'], values='count', title="Studio␣
↪Distribution")
fig.
↪write_image(f"{plots_location}\\anime_recommendations_database_2020\\studio_distribution.
↪png")
Image(filename=f"{plots_location}\\anime_recommendations_database_2020\\studio_distribution.
↪png")
```

[50]:



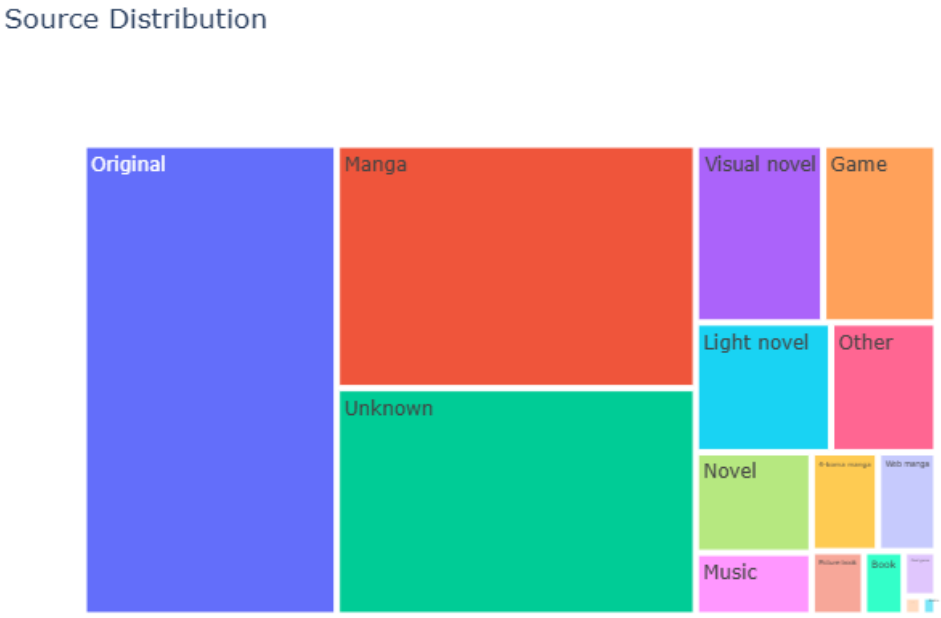
[51]:
```
hernanAnime.loc[hernanAnime["Studios"] == "Unknown", "Studios"].count() /␣
↪hernanAnime.shape[0]
```

[51]:
```
np.float64(0.4030862088600387)
```

The vast majority of Studio (40%), Licensor (78%) and Producer (44%) are unknown. This will impact the usability of these features in the models to be developed.
They cannot be filled as myanimelist.net does not have the information either and neither can the offending rows be dropped as that would reduce the size of the dataset considerably.

[52]:
```
source_df = hernanAnime["Source"].value_counts().to_frame()
source_df = source_df.reset_index()
fig = px.treemap(source_df, path=['Source'], values='count', title="Source␣
↪Distribution")
fig.
↪write_image(f"{plots_location}\\anime_recommendations_database_2020\\source_distribution.
↪png")
Image(filename=f"{plots_location}\\anime_recommendations_database_2020\\source_distribution.
↪png")
```

[52]:



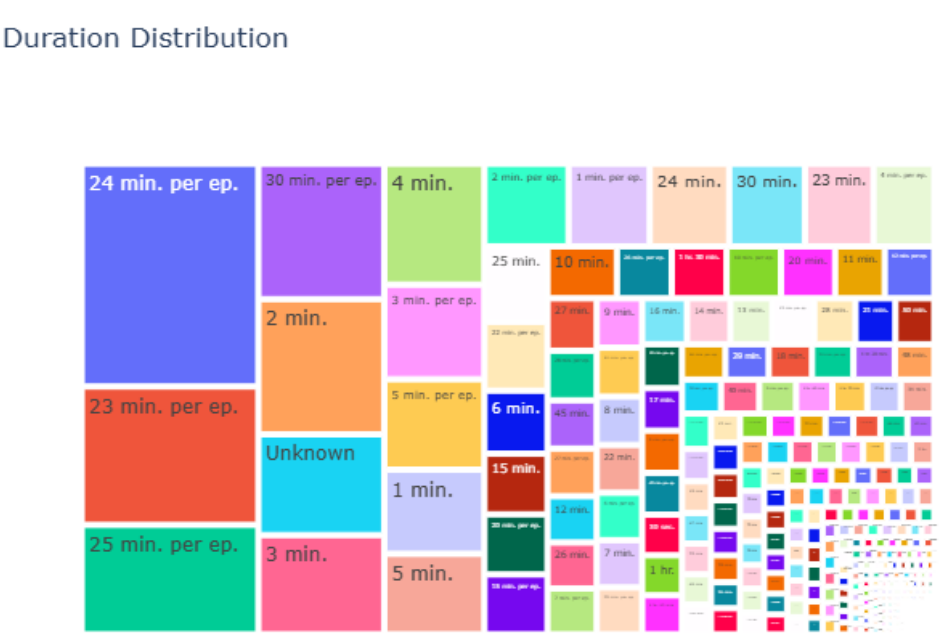
[53]:
```
hernanAnime.loc[hernanAnime["Source"] == "Unknown", "Source"].count() /␣
↪hernanAnime.shape[0]
```

[53]:
```
np.float64(0.2031089853091903)
```

Source also has a lot of missing values, but they account for only 20% of the dataset.

[54]:
```
duration_df = hernanAnime["Duration"].value_counts().to_frame()
duration_df = duration_df.reset_index()
fig = px.treemap(duration_df, path=['Duration'], values='count',␣
↪title="Duration Distribution")
fig.
↪write_image(f"{plots_location}\\anime_recommendations_database_2020\\duration_distribution.
↪png")
Image(filename=f"{plots_location}\\anime_recommendations_database_2020\\duration_distribution.
↪png")
```

[54]:



There is some inconsistency in the representations for the duration which makes there seem to be more categories than there actually are. These will be reconciled and then converted to an integer representing duration in minutes. At which point maybe some binning can be applied.
There are also some unknown values, but not a lot in comparison to some of the other features.

[55]:
```
duration_df.loc[duration_df["Duration"].str.contains("sec"), "Duration"].
↪unique()
```

[55]:
```
array(['30 sec.', '30 sec. per ep.', '15 sec.', '15 sec. per ep.',
       '40 sec. per ep.', '31 sec.', '45 sec. per ep.', '45 sec.',
       '41 sec. per ep.', '39 sec. per ep.', '52 sec.', '20 sec. per ep.',
       '35 sec.', '28 sec.', '10 sec.', '44 sec. per ep.', '16 sec.',
       '42 sec. per ep.', '12 sec.', '35 sec. per ep.', '54 sec.',
       '55 sec.', '53 sec. per ep.', '50 sec.', '58 sec.', '32 sec.',
       '29 sec.', '41 sec.', '42 sec.', '37 sec.', '20 sec.',
       '26 sec. per ep.', '50 sec. per ep.', '46 sec. per ep.',
       '38 sec. per ep.', '34 sec. per ep.', '38 sec.', '39 sec.',
       '51 sec.', '40 sec.', '16 sec. per ep.', '33 sec. per ep.',
       '44 sec.', '33 sec.', '25 sec. per ep.', '24 sec.',
       '51 sec. per ep.', '57 sec. per ep.', '25 sec.', '58 sec. per ep.',
       '57 sec.', '32 sec. per ep.', '36 sec.', '7 sec.', '46 sec.',
       '21 sec. per ep.', '47 sec.', '22 sec. per ep.', '37 sec. per ep.',
```

```
       '24 sec. per ep.', '49 sec.', '31 sec. per ep.', '14 sec.',
       '36 sec. per ep.', '34 sec.', '21 sec.', '14 sec. per ep.',
       '6 sec. per ep.', '23 sec. per ep.', '12 sec. per ep.',
       '54 sec. per ep.', '23 sec.', '22 sec.', '13 sec.',
       '29 sec. per ep.', '17 sec.', '49 sec. per ep.', '27 sec.',
       '3 sec.', '10 sec. per ep.', '19 sec.', '56 sec. per ep.',
       '18 sec. per ep.', '28 sec. per ep.', '43 sec. per ep.', '48 sec.',
       '6 sec.', '43 sec.', '55 sec. per ep.'], dtype=object)
```

[56]:
```
duration_df.loc[duration_df["Duration"].str.contains("min"), "Duration"].
↪unique()
```

[56]:
```
array(['24 min. per ep.', '23 min. per ep.', '25 min. per ep.',
       '30 min. per ep.', '2 min.', '3 min.', '4 min.', '3 min. per ep.',
       '5 min. per ep.', '1 min.', '5 min.', '2 min. per ep.',
       '1 min. per ep.', '24 min.', '30 min.', '23 min.',
       '4 min. per ep.', '25 min.', '22 min. per ep.', '6 min.',
       '15 min.', '20 min. per ep.', '15 min. per ep.', '10 min.',
       '26 min. per ep.', '1 hr. 30 min.', '10 min. per ep.', '20 min.',
       '11 min.', '12 min. per ep.', '27 min.', '28 min. per ep.',
       '45 min.', '27 min. per ep.', '12 min.', '26 min.',
       '7 min. per ep.', '9 min.', '11 min. per ep.', '8 min.', '22 min.',
       '6 min. per ep.', '7 min.', '29 min. per ep.', '16 min.',
       '14 min.', '13 min.', '13 min. per ep.', '28 min.', '21 min.',
       '50 min.', '16 min. per ep.', '17 min.', '8 min. per ep.',
       '45 min. per ep.', '1 hr. 10 min.', '14 min. per ep.', '29 min.',
       '18 min.', '21 min. per ep.', '1 hr. 20 min.', '48 min.',
       '40 min.', '9 min. per ep.', '1 hr. 40 min.', '1 hr. 35 min.',
       '17 min. per ep.', '31 min.', '1 hr. 15 min.', '1 hr. 25 min.',
       '19 min.', '47 min.', '55 min.', '46 min.', '19 min. per ep.',
       '18 min. per ep.', '50 min. per ep.', '1 hr. 33 min.',
       '1 hr. 45 min.', '59 min.', '51 min.', '1 hr. 38 min.',
       '1 hr. 34 min.', '1 hr. 32 min.', '32 min.', '1 hr. 36 min.',
       '1 hr. 22 min.', '44 min.', '42 min.', '1 hr. 50 min.',
       '1 hr. 28 min.', '1 hr. 12 min.', '1 hr. 5 min.', '1 hr. 27 min.',
       '1 hr. 29 min.', '43 min.', '40 min. per ep.', '35 min.',
       '52 min.', '56 min.', '1 hr. 31 min.', '1 hr. 23 min.',
       '1 hr. 11 min.', '1 hr. 26 min.', '1 hr. 39 min.', '54 min.',
       '57 min.', '49 min.', '1 hr. 37 min.', '58 min.', '1 hr. 18 min.',
       '1 hr. 21 min.', '1 hr. 24 min.', '53 min.', '1 hr. 42 min.',
       '1 hr. 13 min.', '33 min.', '1 hr. 14 min.', '1 hr. 46 min.',
       '1 hr. 44 min.', '41 min.', '1 hr. 7 min.', '1 hr. 51 min.',
       '34 min.', '1 hr. 55 min.', '1 hr. 1 min.', '1 hr. 48 min.',
       '38 min.', '1 hr. 16 min.', '1 hr. 59 min.', '32 min. per ep.',
       '1 hr. 41 min.', '1 hr. 19 min.', '35 min. per ep.', '39 min.',
       '33 min. per ep.', '37 min.', '1 hr. 2 min.', '1 hr. 6 min.',
       '37 min. per ep.', '1 hr. 9 min.', '1 hr. 57 min.',
```

```
       '1 hr. 17 min.', '1 hr. 8 min.', '43 min. per ep.',
       '46 min. per ep.', '1 hr. 54 min.', '1 hr. 43 min.',
       '1 hr. 49 min.', '1 hr. 3 min.', '1 hr. 47 min.', '1 hr. 52 min.',
       '2 hr. 10 min.', '36 min. per ep.', '42 min. per ep.',
       '1 hr. 4 min.', '47 min. per ep.', '31 min. per ep.',
       '1 hr. 30 min. per ep.', '1 hr. 56 min.', '55 min. per ep.',
       '34 min. per ep.', '2 hr. 15 min.', '36 min.', '54 min. per ep.',
       '52 min. per ep.', '51 min. per ep.', '48 min. per ep.',
       '1 hr. 58 min.', '57 min. per ep.', '2 hr. 20 min.',
       '2 hr. 5 min.', '39 min. per ep.', '49 min. per ep.',
       '2 hr. 1 min.', '2 hr. 2 min.', '2 hr. 16 min.', '44 min. per ep.',
       '41 min. per ep.', '1 hr. 53 min.', '58 min. per ep.',
       '2 hr. 8 min.', '2 hr. 3 min.', '2 hr. 30 min.', '2 hr. 6 min.',
       '2 hr. 14 min.', '1 hr. 5 min. per ep.', '38 min. per ep.',
       '2 hr. 12 min.', '2 hr. 11 min.', '1 hr. 17 min. per ep.',
       '53 min. per ep.', '1 hr. 35 min. per ep.',
       '1 hr. 36 min. per ep.', '2 hr. 4 min.', '2 hr. 19 min.',
       '2 hr. 41 min.', '2 hr. 40 min.', '1 hr. 7 min. per ep.',
       '1 hr. 38 min. per ep.', '2 hr. 33 min.', '2 hr. 32 min.',
       '2 hr. 43 min.', '2 hr. 17 min.', '1 hr. 15 min. per ep.',
       '2 hr. 36 min.', '2 hr. 28 min.', '1 hr. 11 min. per ep.',
       '2 hr. 7 min.', '2 hr. 27 min.', '1 hr. 16 min. per ep.',
       '2 hr. 42 min.', '1 hr. 2 min. per ep.', '1 hr. 14 min. per ep.',
       '2 hr. 21 min.', '1 hr. 52 min. per ep.', '1 hr. 24 min. per ep.',
       '2 hr. 47 min.', '1 hr. 9 min. per ep.', '1 hr. 27 min. per ep.'],
      dtype=object)
```

[57]:
```
duration_df.loc[duration_df["Duration"].str.contains("hr"), "Duration"].unique()
```

[57]:
```
array(['1 hr. 30 min.', '1 hr.', '1 hr. 10 min.', '1 hr. 20 min.',
       '1 hr. 40 min.', '1 hr. 35 min.', '1 hr. 15 min.', '1 hr. 25 min.',
       '1 hr. 33 min.', '1 hr. 45 min.', '1 hr. 38 min.', '1 hr. 34 min.',
       '1 hr. 32 min.', '1 hr. 36 min.', '1 hr. 22 min.', '1 hr. 50 min.',
       '1 hr. 28 min.', '1 hr. 12 min.', '1 hr. 5 min.', '1 hr. 27 min.',
       '1 hr. 29 min.', '2 hr.', '1 hr. 31 min.', '1 hr. 23 min.',
       '1 hr. 11 min.', '1 hr. 26 min.', '1 hr. 39 min.', '1 hr. 37 min.',
       '1 hr. 18 min.', '1 hr. 21 min.', '1 hr. 24 min.', '1 hr. 42 min.',
       '1 hr. 13 min.', '1 hr. 14 min.', '1 hr. 46 min.', '1 hr. 44 min.',
       '1 hr. 7 min.', '1 hr. 51 min.', '1 hr. 55 min.', '1 hr. 1 min.',
       '1 hr. 48 min.', '1 hr. 16 min.', '1 hr. 59 min.', '1 hr. 41 min.',
       '1 hr. 19 min.', '1 hr. 2 min.', '1 hr. 6 min.', '1 hr. 9 min.',
       '1 hr. 57 min.', '1 hr. 17 min.', '1 hr. 8 min.', '1 hr. 54 min.',
       '1 hr. 43 min.', '1 hr. 49 min.', '1 hr. 3 min.', '1 hr. 47 min.',
       '1 hr. 52 min.', '2 hr. 10 min.', '1 hr. 4 min.',
       '1 hr. 30 min. per ep.', '1 hr. 56 min.', '2 hr. 15 min.',
       '1 hr. 58 min.', '2 hr. 20 min.', '1 hr. per ep.', '2 hr. 5 min.',
       '2 hr. 1 min.', '2 hr. 2 min.', '2 hr. 16 min.', '1 hr. 53 min.',
```

```
       '2 hr. 8 min.', '2 hr. 3 min.', '2 hr. 30 min.', '2 hr. 6 min.',
       '2 hr. 14 min.', '1 hr. 5 min. per ep.', '2 hr. 12 min.',
       '2 hr. 11 min.', '1 hr. 17 min. per ep.', '1 hr. 35 min. per ep.',
       '1 hr. 36 min. per ep.', '2 hr. 4 min.', '2 hr. 19 min.',
       '2 hr. 41 min.', '2 hr. 40 min.', '1 hr. 7 min. per ep.',
       '1 hr. 38 min. per ep.', '2 hr. 33 min.', '2 hr. 32 min.',
       '2 hr. 43 min.', '2 hr. 17 min.', '1 hr. 15 min. per ep.',
       '2 hr. 36 min.', '2 hr. 28 min.', '1 hr. 11 min. per ep.',
       '2 hr. 7 min.', '2 hr. 27 min.', '1 hr. 16 min. per ep.',
       '2 hr. 42 min.', '1 hr. 2 min. per ep.', '1 hr. 14 min. per ep.',
       '2 hr. 21 min.', '1 hr. 52 min. per ep.', '1 hr. 24 min. per ep.',
       '2 hr. 47 min.', '1 hr. 9 min. per ep.', '1 hr. 27 min. per ep.'],
      dtype=object)
```

[58]:
```
duration_df.loc[~duration_df["Duration"].str.contains("(hr|min|sec)"),␣
↪"Duration"].unique()
```

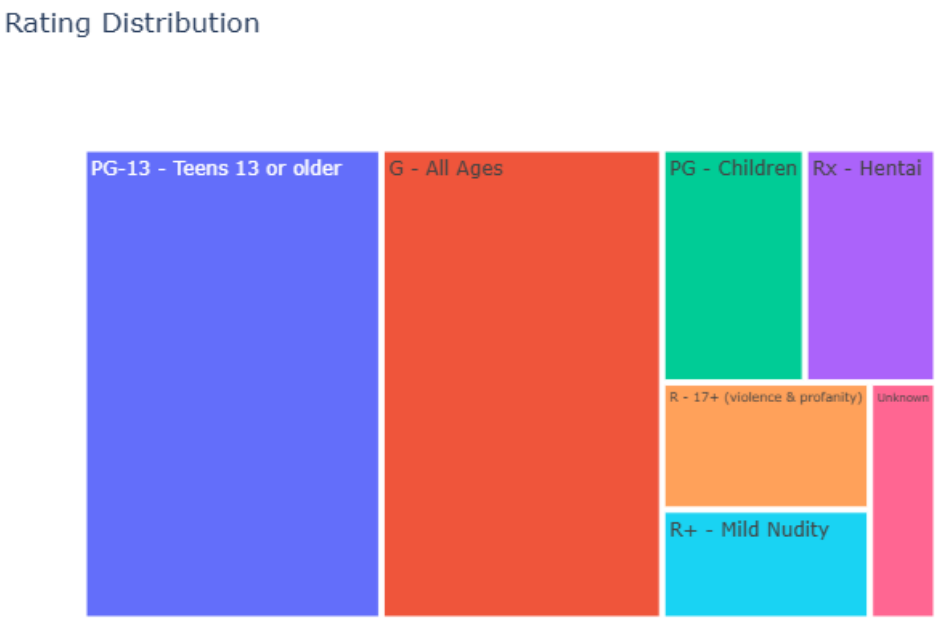
```
C:\Users\Asus-Home\AppData\Local\Temp\ipykernel_22652\2277963426.py:1:
UserWarning:

This pattern is interpreted as a regular expression, and has match groups. To
actually get the groups, use str.extract.
```

[58]:
```
array(['Unknown'], dtype=object)
```

Duration only contains the values hr, min, sec and unknown. The unknown will have to be replaced with Nan in order for the duration to be converted to a numeric value.

[59]:
```
rating_df = hernanAnime["Rating"].value_counts().to_frame()
rating_df = rating_df.reset_index()
fig = px.treemap(rating_df, path=['Rating'], values='count', title="Rating␣
↪Distribution")
fig.
↪write_image(f"{plots_location}\\anime_recommendations_database_2020\\rating_distribution.
↪png")
Image(filename=f"{plots_location}\\anime_recommendations_database_2020\\rating_distribution.
↪png")
```

[59]:



Hentai appears yet again.

As we can see, many of the categories for each of the attributes are unknown.

None of the other features seem useful at this stage.

Anime ratings for each user

[60]:
```
hernanRatings = pd.read_csv("E:\\applied data science␣
  ↪capstone\\data\\hernan4444\\archive\\rating_complete.csv")
hernanRatings.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 57633278 entries, 0 to 57633277
Data columns (total 3 columns):
 #   Column    Dtype
---  ------    -----
 0   user_id   int64
 1   anime_id  int64
 2   rating    int64
dtypes: int64(3)
memory usage: 1.3 GB
```

There are 58 million records

[61]:
```
hernanRatings.describe()
```

[61]:

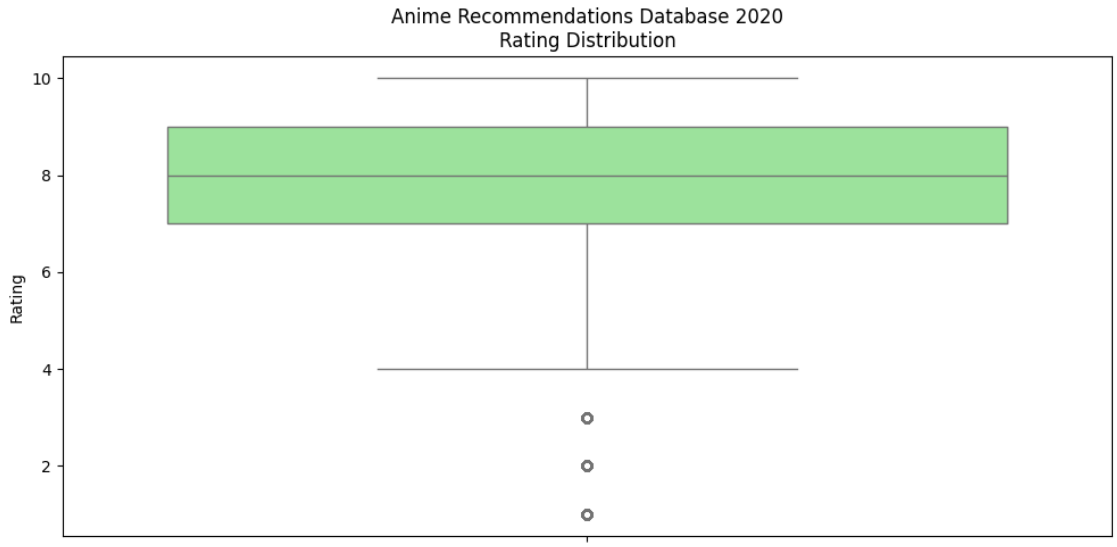
	user_id	anime_id	rating
count	5.763328e+07	5.763328e+07	5.763328e+07
mean	1.768878e+05	1.583147e+04	7.510789e+00
std	1.020117e+05	1.326114e+04	1.697722e+00
min	0.000000e+00	1.000000e+00	1.000000e+00
25%	8.827800e+04	3.091000e+03	7.000000e+00
50%	1.772910e+05	1.188700e+04	8.000000e+00
75%	2.654190e+05	2.899900e+04	9.000000e+00
max	3.534040e+05	4.845600e+04	1.000000e+01

[62]:
```
hernanRatings.isna().sum()
```

[62]:
```
user_id     0
anime_id    0
rating      0
dtype: int64
```

There are no missing values for this dataset which makes it even more inconsequential if we remove the missing values from the previous dataset.

[63]:
```
plt.figure(figsize=(10, 5))

sns.boxplot(data=hernanRatings["rating"], color="lightgreen")
plt.title('Anime Recommendations Database 2020\nRating Distribution')
plt.ylabel('Rating')

plt.tight_layout()
plt.
 ↪savefig(f"{plots_location}\\anime_recommendations_database_2020\\user_score_dsitribution.
 ↪png")
plt.show()
```



Synopsis Data

[64]:
```
synopsisDf = pd.read_csv("E:\\applied data science␣
↪capstone\\data\\hernan4444\\archive\\anime_with_synopsis.csv")
synopsisDf.describe(include=object)
```

[64]:
```
                                                  Name    Score Genres  \
count                                            16214    16214  16214
unique                                           16210      532   4857
top     Maou Gakuin no Futekigousha: Shijou Saikyou no…  Unknown  Music
freq                                                 3     5123    790

                                                sypnopsis
count                                               16206
unique                                              15221
top     No synopsis information has been added to this…
freq                                                  709
```

[65]:
```
synopsisDf[~synopsisDf["sypnopsis"].str.contains("No synopsis", na=False)].
↪describe(include=object)
```

[65]:
```
                                                  Name    Score Genres  \
count                                            15469    15469  15469
unique                                           15465      530   4780
top     Maou Gakuin no Futekigousha: Shijou Saikyou no…  Unknown  Music
freq                                                 3     4559    768

                sypnopsis
count               15461
```

```
unique                15218
top     Furukawa Taku film.
freq                     13
```

[66]:
```
synopsisDf["sypnopsis"].value_counts().head(10)
```

[66]:
```
sypnopsis
No synopsis information has been added to this title. Help improve our database
by adding a synopsis here .    709
No synopsis has been added for this series yet. Click here to update this
information.                          35
Furukawa Taku film.
13
Film by Takashi Ito.
13
short animation by Taku Furukawa.
10
short film by Okamoto Tadanari.
8
Short animation by Rapparu.
8
short puppet animation movie by Tadahito Mochinaga.
6
Short film by Hirano Ryou.
6
Short film by Kurosaka Keita.
6
Name: count, dtype: int64
```

The No synopsis synopsis will need to be replaced with na or Unknown

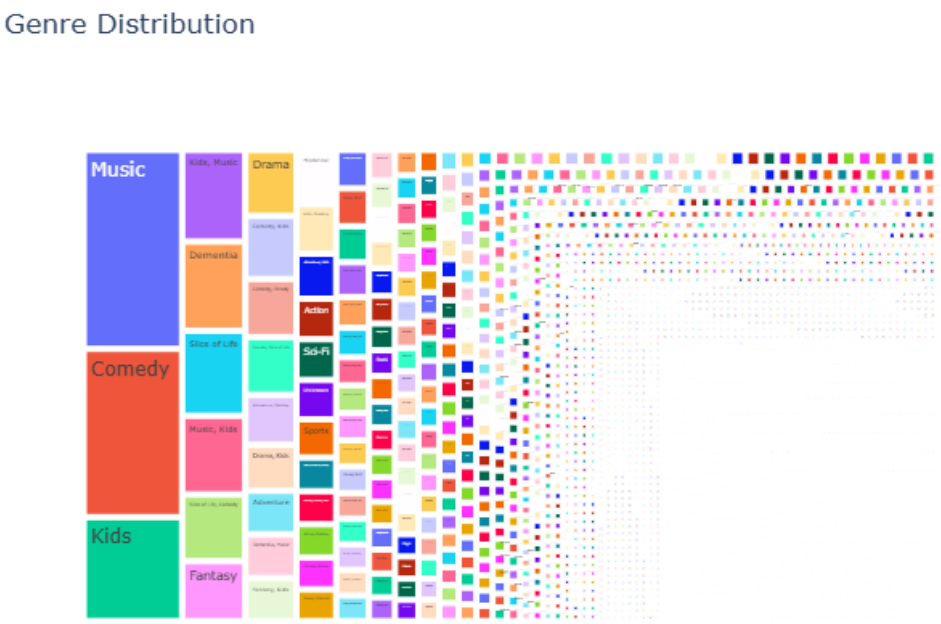
[67]:
```
synopsisDf.isna().sum()
```

[67]:
```
MAL_ID       0
Name         0
Score        0
Genres       0
sypnopsis    8
dtype: int64
```

There doesn't seem to be any missing data for the genres field. This can be used to fill in any missing genres in the other datasets.
Sypnopsis needs to be renamed to synopsis.
While the synopsis count only has 8 missing values, the most frequent description seems to be stating that there is no missing synopsis. This will also be counted as missing data.
MAL_ID is simple the anime_id.
Name could also be used to fill in any missing values.

[68]:
```
genre_df = synopsisDf["Genres"].value_counts().to_frame()
genre_df = genre_df.reset_index()
fig = px.treemap(genre_df, path=['Genres'], values='count', title="Genre␣
↪Distribution")
fig.
↪write_image(f"{plots_location}\\anime_recommendations_database_2020\\genre_distribution.
↪png")
Image(filename=f"{plots_location}\\anime_recommendations_database_2020\\genre_distribution.
↪png")
```

[68]:



Music does seem to be a large part of the dataset, but these will be removed from the other datasets. So no need to remove it here as this dataset will only be used for the synopsis and missing genres and names.

[69]:
```
synopsisDf.head()
```

[69]:

	MAL_ID	Name	Score	\
0	1	Cowboy Bebop	8.78	
1	5	Cowboy Bebop: Tengoku no Tobira	8.39	
2	6	Trigun	8.24	
3	7	Witch Hunter Robin	7.27	
4	8	Bouken Ou Beet	6.98	

```
                                                  Genres  \
0    Action, Adventure, Comedy, Drama, Sci-Fi, Space
1              Action, Drama, Mystery, Sci-Fi, Space
2  Action, Sci-Fi, Adventure, Comedy, Drama, Shounen
3  Action, Mystery, Police, Supernatural, Drama, …
4          Adventure, Fantasy, Shounen, Supernatural

                                             sypnopsis
0  In the year 2071, humanity has colonized sever…
1  other day, another bounty-such is the life of …
2  Vash the Stampede is the man with a $$60,000,0…
3  ches are individuals with special powers like …
4  It is the dark century and the people are suff…
```

The synopsis field might need some cleaning done on it.

0.1.3 MyAnimeList Comment Dataset V2

Anime Listing

[70]:
```
natleeAnime = pd.read_csv("E:\\applied data science␣
  ↪capstone\\data\\natlee\\archive\\anime_list.csv")
natleeAnime.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 24594 entries, 0 to 24593
Data columns (total 28 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   id              24594 non-null  int64
 1   workId          24594 non-null  int64
 2   url             24594 non-null  object
 3   jpName          24504 non-null  object
 4   engName         10181 non-null  object
 5   synonymsName    12821 non-null  object
 6   workType        21988 non-null  object
 7   episodes        24594 non-null  object
 8   status          24594 non-null  object
 9   aired           24594 non-null  object
 10  premiered       5498 non-null   object
 11  producer        24594 non-null  object
 12  broadcast       7576 non-null   object
 13  licensors       24594 non-null  object
 14  studios         24594 non-null  object
 15  genres          19795 non-null  object
 16  themes          13612 non-null  object
 17  demographic     9392 non-null   object
 18  source          24594 non-null  object
```

```
 19  duration        24594 non-null  object
 20  rating          24013 non-null  object
 21  score           15647 non-null  float64
 22  allRank         20073 non-null  object
 23  popularityRank  24594 non-null  object
 24  members         24594 non-null  object
 25  favorites       24594 non-null  object
 26  scoredByUser    15647 non-null  float64
 27  lastUpdate      24594 non-null  object
dtypes: float64(2), int64(2), object(24)
memory usage: 5.3+ MB
```

[71]:
```
natleeAnime.head()
```

[71]:

	id	workId	url	jpName	engName	synonymsName	workType	episodes	status	aired	…	source	duration
0	1	47917	https://myanimelist.net/anime/47917/Bocchi_the…		Bocchi the Rock!	NaN	TV	12	Finished Airing	Oct 9, 2022 to Dec 25, 2022	…	4-koma manga	23 min. per ep.
1	2	19	https://myanimelist.net/anime/19/Monster		Monster	NaN	TV	74	Finished Airing	Apr 7, 2004 to Sep 28, 2005	…	Manga	24 min. per ep.
2	3	35247	https://myanimelist.net/anime/35247/Owarimonog…		Owarimonogatari Second Season	End Story 2nd Season	TV	7	Finished Airing	Aug 12, 2017 to Aug 13, 2017	…	Light novel	22 min. per ep.
3	4	37491	https://myanimelist.net/anime/37491/Gintama__S…	.	Gintama.: Silver Soul Arc - Second Half War	Gintama.: Silver Soul Arc 2	TV	14	Finished Airing	Jul 9, 2018 to Oct 8, 2018	…	Manga	24 min. per ep.
4	5	52198	https://myanimelist.net/anime/52198/Kaguya-sam…	-	Kaguya-sama: Love is War - The First Kiss That…	NaN	Movie	1	Finished Airing		…		

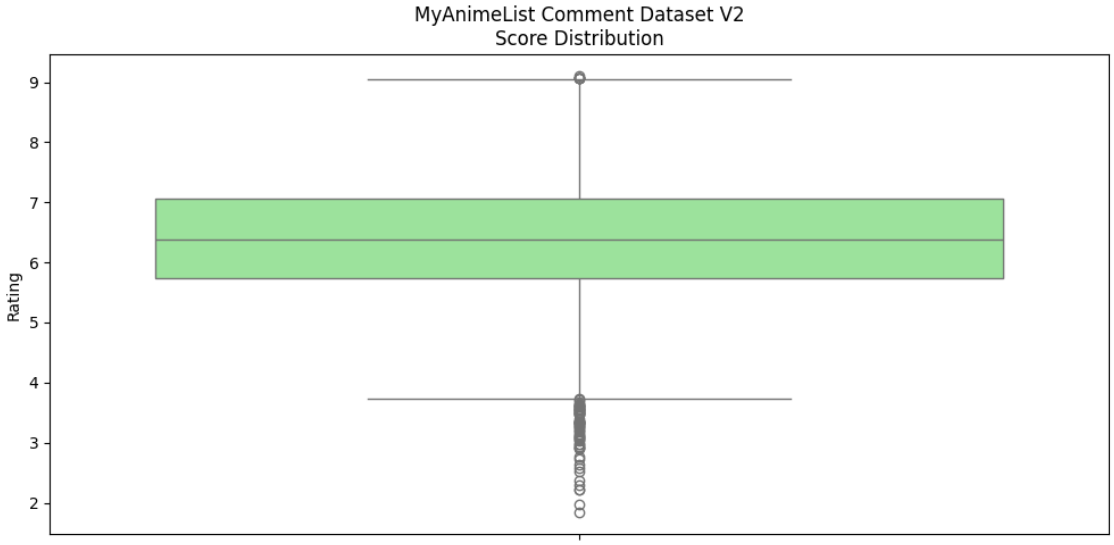
```
4                  Dec 17, 2022  …         Manga    1 hr. 36 min.

                          rating score allRank popularityRank    members  \
0       PG-13 - Teens 13 or older  8.86     #27           #509    415,475
1                R+ - Mild Nudity  8.87     #26           #142  1,006,450
2  R - 17+ (violence & profanity)  8.88     #25           #567    379,309
3       PG-13 - Teens 13 or older  8.88     #24          #1186    184,986
4       PG-13 - Teens 13 or older  8.89     #23          #1270    173,642

  favorites scoredByUser           lastUpdate
0    20,435     241578.0  2023-06-02 00:11:01
1    46,856     365316.0  2023-06-02 00:11:10
2     8,444     182679.0  2023-06-02 00:11:19
3     1,018      89095.0  2023-06-02 00:11:30
4     1,305      77107.0  2023-06-02 00:11:35

[5 rows x 28 columns]
```

The id attribute will be dropped as it provides no value.
Url also provides no real value to the data, but might be useful for a product page.
jpName will be dropped as the japanese title will not provide much value for the english users.

[72]:
```
natleeAnime.describe()
```

[72]:
```
                id        workId         score  scoredByUser
count  24594.000000  24594.000000  15647.000000  1.564700e+04
mean   12297.500000  29456.666057      6.382660  2.992858e+04
std     7099.820596  17860.097090      0.928238  1.167218e+05
min        1.000000      1.000000      1.840000  1.010000e+02
25%     6149.250000  10335.250000      5.730000  3.835000e+02
50%    12297.500000  34330.000000      6.390000  1.768000e+03
75%    18445.750000  44896.750000      7.060000  1.084700e+04
max    24594.000000  55566.000000      9.100000  2.654325e+06
```

[73]:
```
plt.figure(figsize=(10, 5))

sns.boxplot(data=natleeAnime["score"], color="lightgreen")
plt.title('MyAnimeList Comment Dataset V2\nScore Distribution')
plt.ylabel('Rating')

plt.tight_layout()
plt.
 ↪savefig(f"{plots_location}\\myanimelist_comments_dataset_v2\\user_score_distribution.
 ↪png")
plt.show()
```



[74]:
```
natleeAnime.describe(include=object)
```

[74]:

```
                                                  url       jpName  \
count                                           24594        24504
unique                                          24594        23530
top     https://myanimelist.net/anime/47917/Bocchi_the…
freq                                                1            8

                engName  synonymsName workType episodes           status  \
count             10181         12821    21988    24594            24594
unique             9992         12128        5      252                3
top     Spirit Guardians  Minna no Uta       TV        1  Finished Airing
freq                   5           386     7576    11314            23810

                aired    premiered  producer  … themes demographic  \
count           24594         5498     24594  …  13612        9392
unique          15095          243      4402  …    831           5
top     Not available  Spring 2017  add some  …  Music        Kids
freq              887           88     13075  …   2674        5846

          source         duration                     rating allRank  \
count      24594            24594                      24013   20073
unique        17              328                          6   18499
top     Original  24 min. per ep.  PG-13 - Teens 13 or older  #10290
freq        9446             1957                       8372       3

       popularityRank members favorites           lastUpdate
count           24594   24594     24594                24594
```

```
unique          18253   10990      1787                 24593
top            #18698      38         0  2023-06-02 11:06:59
freq                6     165     10546                     2

[4 rows x 24 columns]
```

EngName and synonymsName have some repeated info.

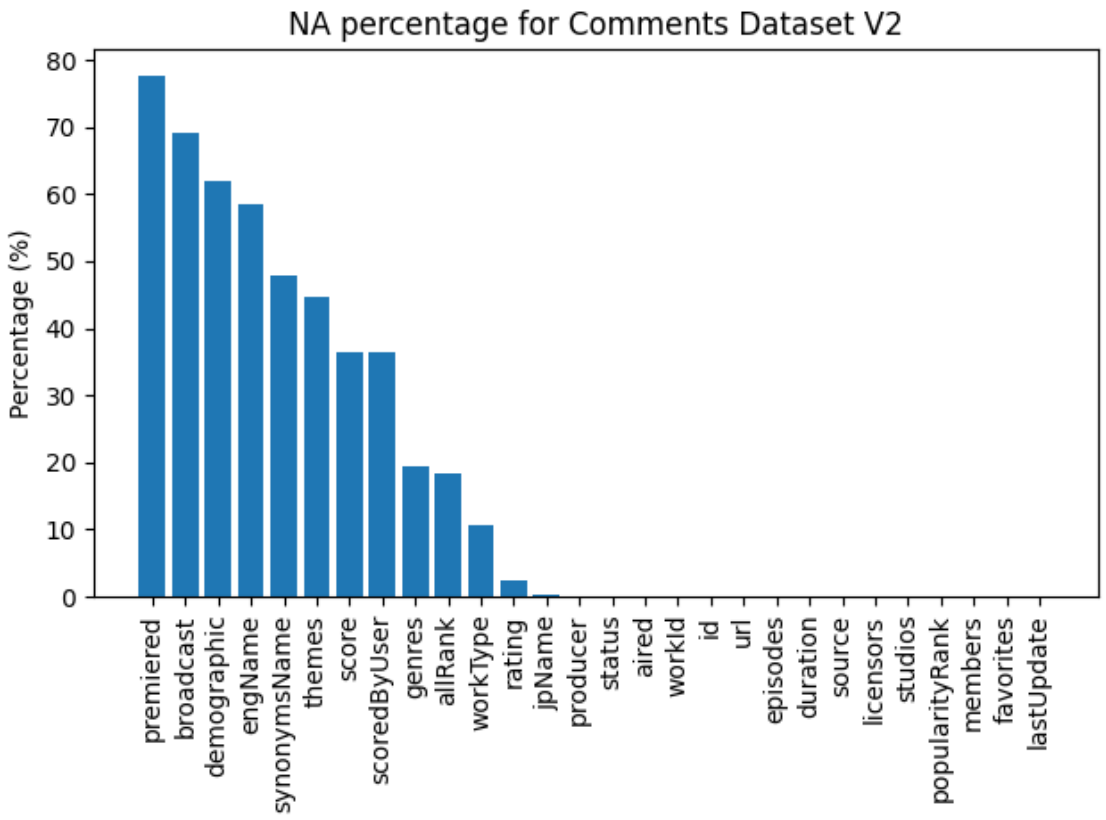
[75]:
```
natleeAnime[natleeAnime["engName"] == "Spirit Guardians"]
```

[75]:
```
          id  workId                                                 url  \
5020    5021   52378  https://myanimelist.net/anime/52378/Dou_Hun_We…
5021    5022   45448  https://myanimelist.net/anime/45448/Dou_Hun_We…
5022    5023   41081  https://myanimelist.net/anime/41081/Dou_Hun_We…
5023    5024   41079  https://myanimelist.net/anime/41079/Dou_Hun_We…
12117  12118   41078  https://myanimelist.net/anime/41078/Dou_Hun_We…

             jpName           engName  \
5020          V  Spirit Guardians
5021         IV  Spirit Guardians
5022        III  Spirit Guardians
5023         II  Spirit Guardians
12117            Spirit Guardians

                                          synonymsName workType episodes  \
5020   Dou Hun Wei Zhi Xuan Yue Qi Yuan 5, The Dream …       ONA       13
5021   Dou Hun Wei Zhi Xuan Yue Qiyuan 4th Season, Th…       ONA       13
5022   Dou Hun Wei Zhi Xuan Yue Qi Yuan 3, The Dream …       ONA       13
5023   Dou Hun Wei Zhi Xuan Yue Qi Yuan 2, Spirit Gua…       ONA       13
12117  Dou Hun Wei Zhi Xuan Yue Qi Yuan, The Dream of…       ONA       12

                status                          aired  …    source  \
5020   Finished Airing  Dec 28, 2021 to Mar 15, 2022  …  Original
5021   Finished Airing   May 19, 2020 to Aug 4, 2020  …  Original
5022   Finished Airing  May 27, 2019 to Aug 19, 2019  …  Original
5023   Finished Airing  Nov 27, 2018 to Feb 19, 2019  …  Original
12117  Finished Airing   Jun 19, 2018 to Sep 4, 2018  …  Original

              duration                      rating score allRank  \
5020   19 min. per ep.  PG-13 - Teens 13 or older   NaN  #15638
5021   20 min. per ep.  PG-13 - Teens 13 or older   NaN  #15637
5022   20 min. per ep.  PG-13 - Teens 13 or older   NaN  #15636
5023   21 min. per ep.  PG-13 - Teens 13 or older   NaN  #15635
12117  19 min. per ep.  PG-13 - Teens 13 or older  5.77   #9805

      popularityRank members favorites scoredByUser           lastUpdate
5020          #20229     116         0          NaN  2023-06-02 12:48:20
5021          #17413     282         0          NaN  2023-06-02 12:48:25
```

```
5022          #16775     331         0          NaN  2023-06-02 12:48:33
5023          #15792     406         0          NaN  2023-06-02 12:48:43
12117         #13073     868         3        215.0  2023-06-03 07:02:12

[5 rows x 28 columns]
```

These repetitions refer to different seasons of the same anime.
There are a lot of NaNs as well so an na count could be useful.

[76]:
```
na_percentage_df = (natleeAnime.isna().sum() / natleeAnime.shape[0] * 100).
 ↪to_frame()
na_percentage_df = na_percentage_df.reset_index()
na_percentage_df.columns = ["feature", "percentage"]
na_percentage_df = na_percentage_df.sort_values('percentage', ascending=False)

fig, ax = plt.subplots()

ax.bar(na_percentage_df["feature"], na_percentage_df["percentage"],␣
 ↪label=na_percentage_df["feature"])

ax.set_ylabel('Percentage (%)')
ax.set_title('NA percentage for Comments Dataset V2')

plt.xticks(rotation=90)
plt.tight_layout()
plt.
 ↪savefig(f"{plots_location}\\myanimelist_comments_dataset_v2\\na_percentage_distribution.
 ↪png")
plt.show()
```



In addition to all these na values we have unknown values that carry the label *add some* or *unknown*

[77]:
```
natleeAnime[natleeAnime["synonymsName"] == "Minna no Uta"].head()
```

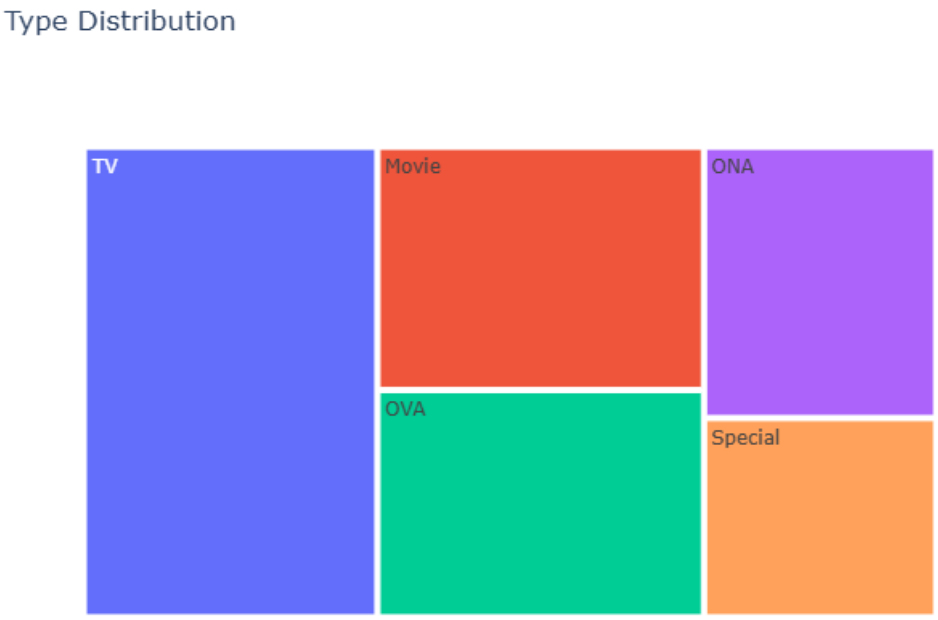
[77]:
```
          id  workId                                                url  \
19577  19578   37607  https://myanimelist.net/anime/37607/Shounen_to…
19638  19639   30199  https://myanimelist.net/anime/30199/Hakimono_t…
19656  19657   37067  https://myanimelist.net/anime/37067/Yume_no_To…
19662  19663   37690  https://myanimelist.net/anime/37690/Fure_Fure_…
19673  19674   37692  https://myanimelist.net/anime/37692/Tousan_no_…

            jpName                  engName  synonymsName workType episodes  \
19577         The Boy and Magic Robot  Minna no Uta      NaN        1
19638                             NaN  Minna no Uta      NaN        1
19656                             NaN  Minna no Uta      NaN        1
19662                             NaN  Minna no Uta      NaN        1
19673                             NaN  Minna no Uta      NaN        1

               status        aired  …    source duration        rating  \
19577  Finished Airing  Aug 1, 2013  …  Original   5 min.  G - All Ages
19638  Finished Airing  Feb 1, 2015  …  Original   4 min.  G - All Ages
```

```
19656  Finished Airing     Aug 2017  …  Original   4 min.  G - All Ages
19662  Finished Airing  Oct 1, 2010  …  Original   4 min.  G - All Ages
19673  Finished Airing  Apr 2, 2018  …  Original   2 min.  G - All Ages

      score allRank popularityRank members favorites scoredByUser  \
19577  5.95     NaN         #16544     346         0        169.0
19638  5.74     NaN         #16035     385         0        191.0
19656  5.59     NaN         #17625     270         0        121.0
19662  5.61     NaN         #18317     230         0        112.0
19673  5.55     NaN         #18018     246         0        105.0

               lastUpdate
19577  2023-06-04 01:45:07
19638  2023-06-04 01:54:04
19656  2023-06-04 01:57:04
19662  2023-06-04 01:57:59
19673  2023-06-04 01:59:44

[5 rows x 28 columns]
```

These also seem to refer to different seasons of the same anime.

[78]:
```
work_type_df = natleeAnime["workType"].value_counts().to_frame()
work_type_df = work_type_df.reset_index()
fig = px.treemap(work_type_df, path=['workType'], values='count', title="Type␣
 ↪Distribution")
fig.
 ↪write_image(f"{plots_location}\\myanimelist_comments_dataset_v2\\type_distribution.
 ↪png")
Image(filename=f"{plots_location}\\myanimelist_comments_dataset_v2\\type_distribution.
 ↪png")
```

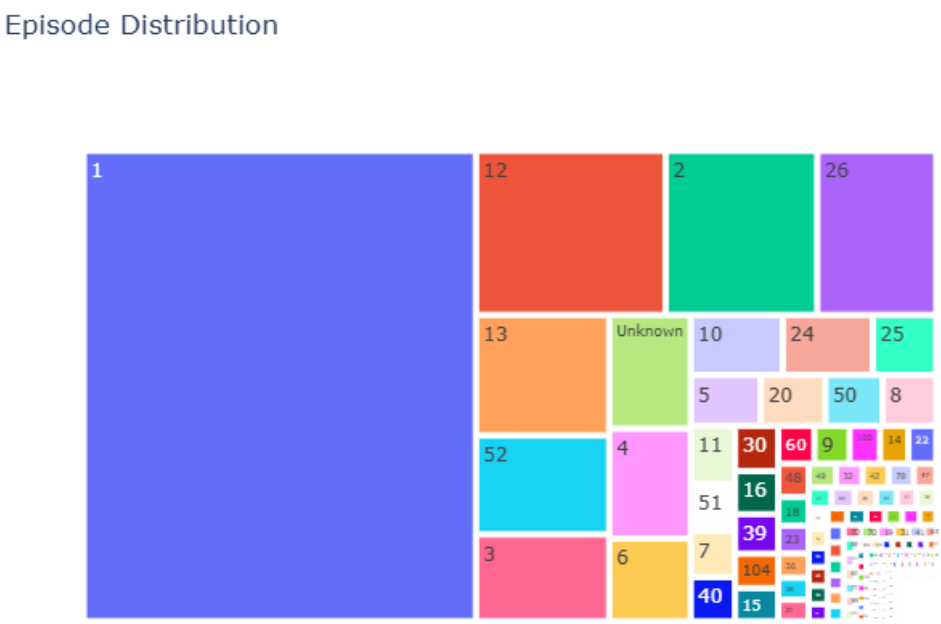
[78]:



Like the others we're only interested in Tv and movie

[79]:
```
episodes_df = natleeAnime["episodes"].value_counts().to_frame()
episodes_df = episodes_df.reset_index()
fig = px.treemap(episodes_df, path=['episodes'], values='count', title="Episode␣
↪Distribution")
fig.
↪write_image(f"{plots_location}\\myanimelist_comments_dataset_v2\\episode_distribution.
↪png")
Image(filename=f"{plots_location}\\myanimelist_comments_dataset_v2\\episode_distribution.
↪png")
```

[79]:



There are some unknowns which will need to be accounted for. Having the majority of the episodes being 1 points to many of the entries being movies or music videos and songs. The music videos and songs will need to be removed.

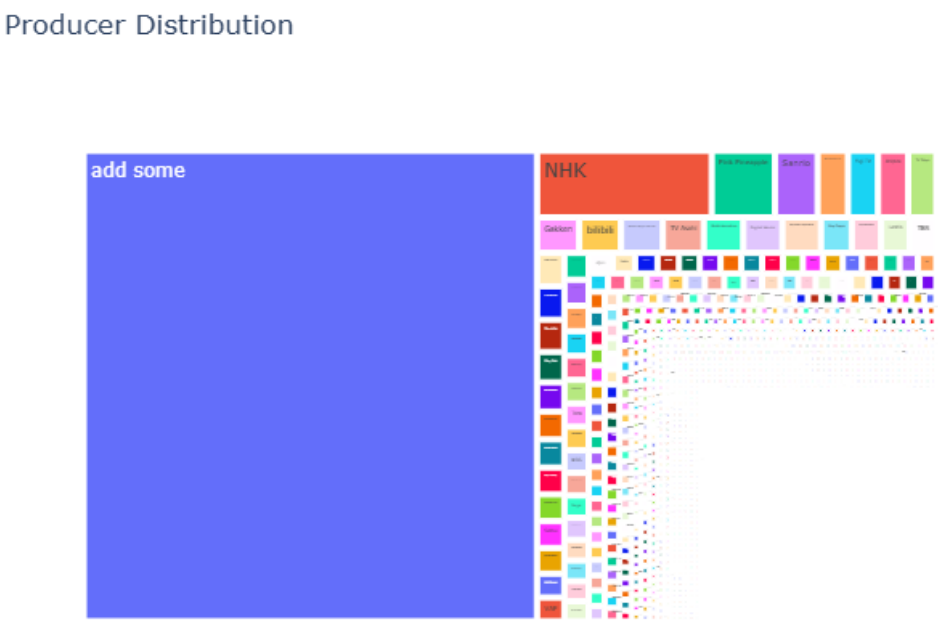
[80]:
```
status_df = natleeAnime["status"].value_counts().to_frame()
status_df = status_df.reset_index()
fig = px.treemap(status_df, path=['status'], values='count', title="Status␣
↪Distribution")
fig.
↪write_image(f"{plots_location}\\myanimelist_comments_dataset_v2\\status_distribution.
↪png")
Image(filename=f"{plots_location}\\myanimelist_comments_dataset_v2\\status_distribution.
↪png")
```

[80]:

Most of the anime has finished airing, which is good for the ratings provided for the anime assuming the users completed the anime before leaving a rating.

[81]:
```
producer_df = natleeAnime["producer"].value_counts().to_frame()
producer_df = producer_df.reset_index()
fig = px.treemap(producer_df, path=['producer'], values='count',␣
 ↪title="Producer Distribution")
fig.
 ↪write_image(f"{plots_location}\\myanimelist_comments_dataset_v2\\producer_distribution.
 ↪png")
Image(filename=f"{plots_location}\\myanimelist_comments_dataset_v2\\producer_distribution.
 ↪png")
```

[81]:



A lot of the information for producers is missing as noted by the label *add some*

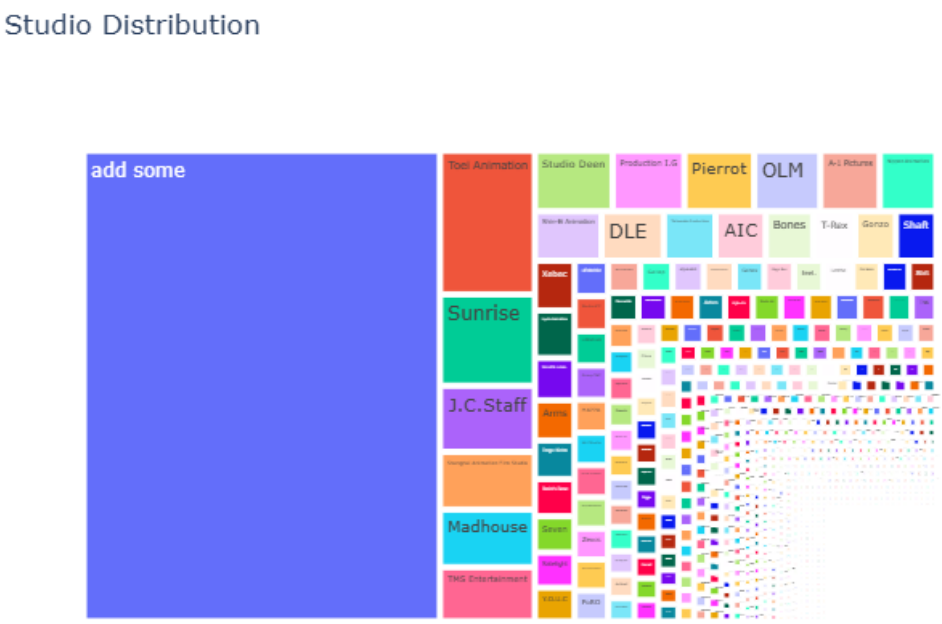
[82]:
```
licensor_df = natleeAnime["licensors"].value_counts().to_frame()
licensor_df = licensor_df.reset_index()
fig = px.treemap(licensor_df, path=['licensors'], values='count',␣
 ↪title="Licensor Distribution")
fig.
 ↪write_image(f"{plots_location}\\myanimelist_comments_dataset_v2\\licensor_distribution.
 ↪png")
Image(filename=f"{plots_location}\\myanimelist_comments_dataset_v2\\licensor_distribution.
 ↪png")
```

[82]:

Licensors is also missing a lot of data

[83]:
```
studio_df = natleeAnime["studios"].value_counts().to_frame()
studio_df = studio_df.reset_index()
fig = px.treemap(studio_df, path=['studios'], values='count', title="Studio␣
↪Distribution")
fig.
↪write_image(f"{plots_location}\\myanimelist_comments_dataset_v2\\studio_distribution.
↪png")
Image(filename=f"{plots_location}\\myanimelist_comments_dataset_v2\\studio_distribution.
↪png")
```

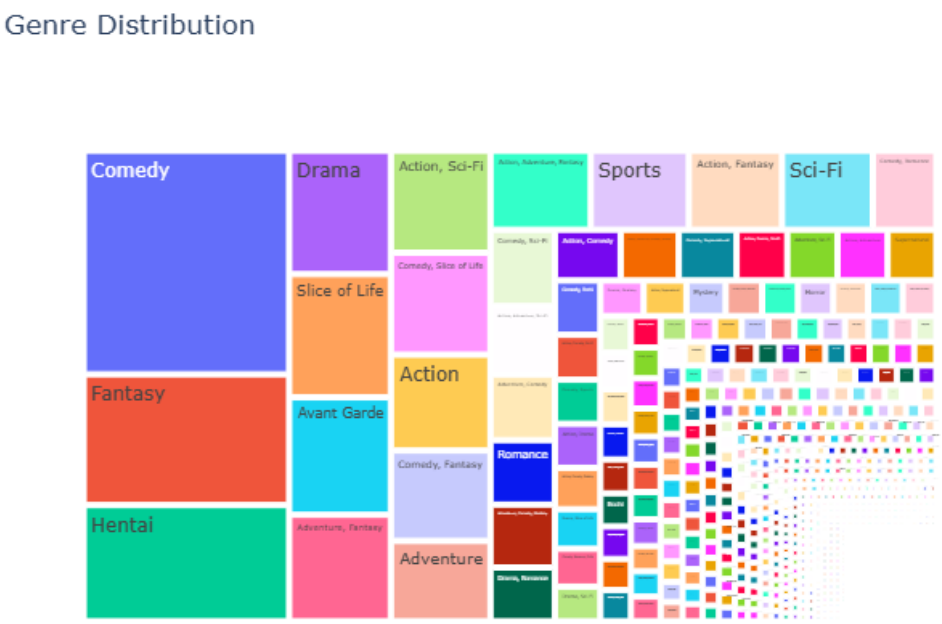
[83]:



Studios is also missing data.

[84]:
```
genre_df = natleeAnime["genres"].value_counts().to_frame()
genre_df = genre_df.reset_index()
fig = px.treemap(genre_df, path=['genres'], values='count', title="Genre␣
↪Distribution")
fig.
↪write_image(f"{plots_location}\\myanimelist_comments_dataset_v2\\genre_distribution.
↪png")
Image(filename=f"{plots_location}\\myanimelist_comments_dataset_v2\\genre_distribution.
↪png")
```

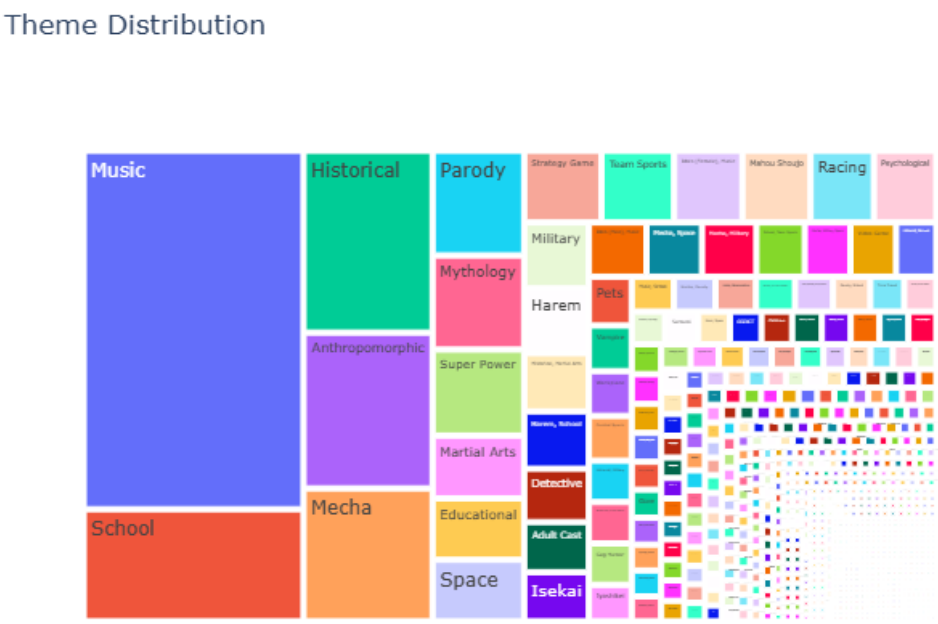
[84]:



Hentai once more needs to be removed.

[85]:
```
theme_df = natleeAnime["themes"].value_counts().to_frame()
theme_df = theme_df.reset_index()
fig = px.treemap(theme_df, path=['themes'], values='count', title="Theme␣
↪Distribution")
fig.
↪write_image(f"{plots_location}\\myanimelist_comments_dataset_v2\\theme_distribution.
↪png")
Image(filename=f"{plots_location}\\myanimelist_comments_dataset_v2\\theme_distribution.
↪png")
```

[85]:



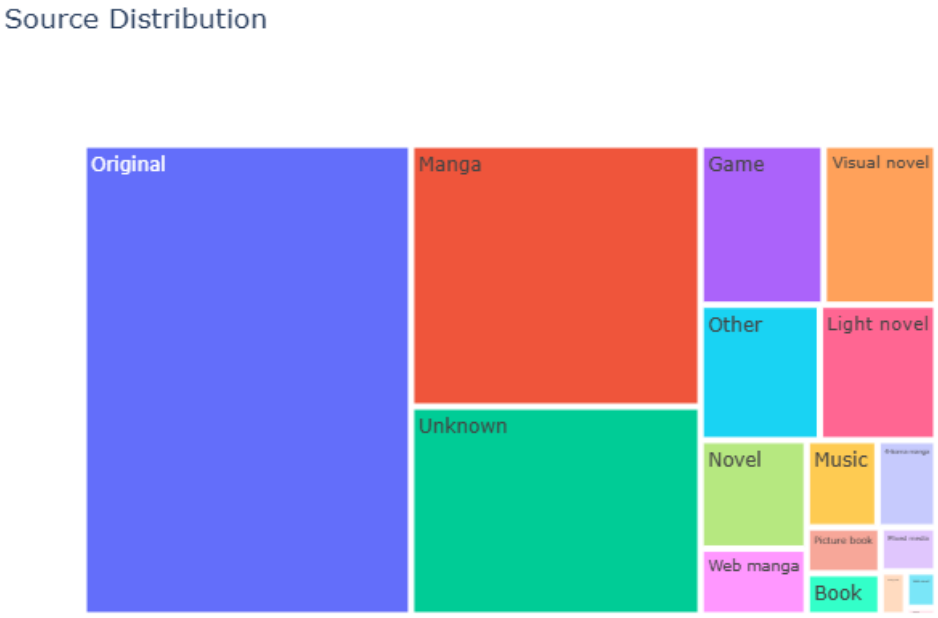
[86]:
```
demographic_df = natleeAnime["demographic"].value_counts().to_frame()
demographic_df = demographic_df.reset_index()
fig = px.treemap(demographic_df, path=['demographic'], values='count',␣
 ↪title="Demographic Distribution")
fig.
 ↪write_image(f"{plots_location}\\myanimelist_comments_dataset_v2\\demographic_distribution.
 ↪png")
Image(filename=f"{plots_location}\\myanimelist_comments_dataset_v2\\demographic_distribution.
 ↪png")
```

[86]:

A lot of the anime seems to be for kids. Kids referring to anyone younger than teenage years

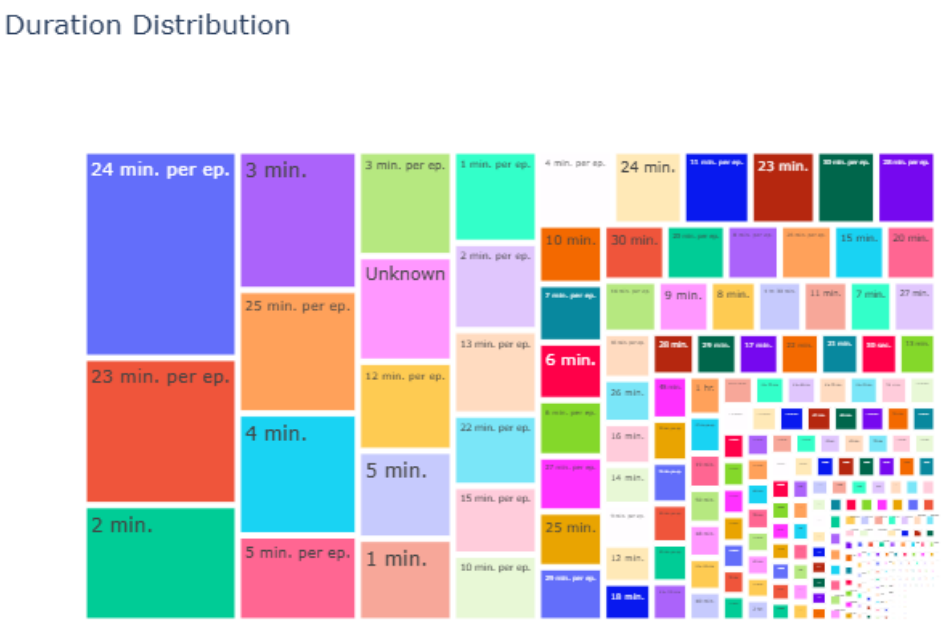
[87]:
```
source_df = natleeAnime["source"].value_counts().to_frame()
source_df = source_df.reset_index()
fig = px.treemap(source_df, path=['source'], values='count', title="Source␣
↪Distribution")
fig.
↪write_image(f"{plots_location}\\myanimelist_comments_dataset_v2\\source_distribution.
↪png")
Image(filename=f"{plots_location}\\myanimelist_comments_dataset_v2\\source_distribution.
↪png")
```

[87]:



[88]:
```
duration_df = natleeAnime["duration"].value_counts().to_frame()
duration_df = duration_df.reset_index()
fig = px.treemap(duration_df, path=['duration'], values='count',␣
 ↪title="Duration Distribution")
fig.
 ↪write_image(f"{plots_location}\\myanimelist_comments_dataset_v2\\duration_distribution.
 ↪png")
Image(filename=f"{plots_location}\\myanimelist_comments_dataset_v2\\duration_distribution.
 ↪png")
```

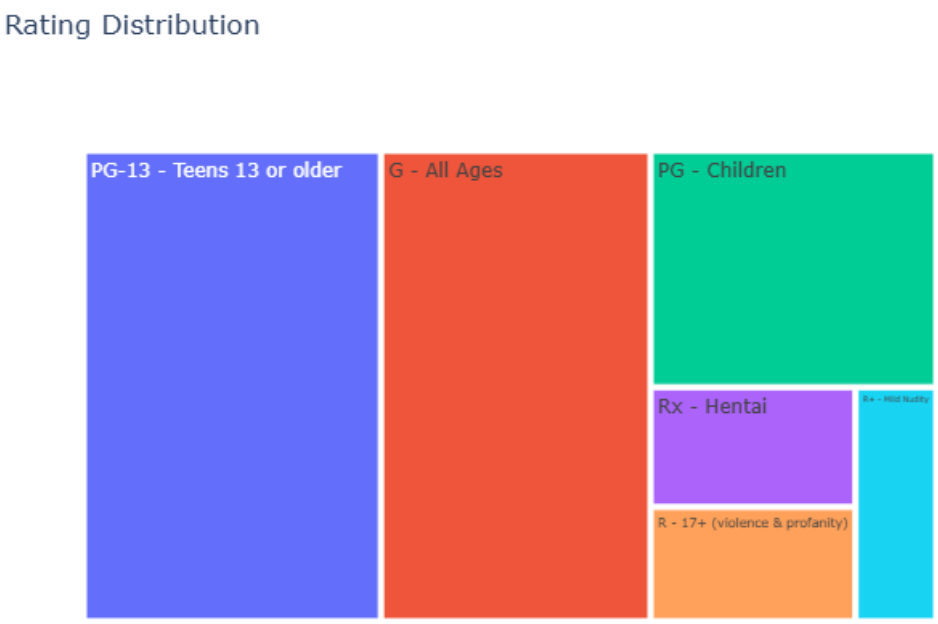
[88]:



Once more there is some inconsitency when it comes to the categorization of durations.

[89]:
```
rating_df = natleeAnime["rating"].value_counts().to_frame()
rating_df = rating_df.reset_index()
fig = px.treemap(rating_df, path=['rating'], values='count', title="Rating␣
↪Distribution")
fig.
↪write_image(f"{plots_location}\\myanimelist_comments_dataset_v2\\rating_distribution.
↪png")
Image(filename=f"{plots_location}\\myanimelist_comments_dataset_v2\\rating_distribution.
↪png")
```

[89]:



Once more hentai needs to be removed.