Data visualisation lab 2

Made by: Paulius Lapienis

- Data set link: https://www.kaggle.com/datasets/azathoth42/myanimelist
- All of the tasks were performed using the Python programming language.

Task 1: Describing data types.

- The features used for the visualisations:
    - episodes: ratio, quantitative
    - studio: nominal, quantitative
    - score: ratio, quantitative
    - source: nominal, quantitative

In [ ]:

```
from labs.definitions import DATA_DIR
import pandas as pd

DATA_PATH = DATA_DIR / "anime_filtered.csv"
df = pd.read_csv(DATA_PATH)
df
```

Out[ ]:

	anime_id	title	title_english	title_japanese	title_synonyms	ima
0	11013	Inu x Boku SS	Inu X Boku Secret Service	妖狐×僕SS	Youko x Boku SS	https://myanimeli dena.com/images/ar
1	2104	Seto no Hanayome	My Bride is a Mermaid	瀬戸の花嫁	The Inland Sea Bride	https://myanimeli dena.com/images/ar
2	5262	Shugo Chara!! Doki	Shugo Chara!! Doki	しゅごキャラ！！どきっ	Shugo Chara Ninenme, Shugo Chara! Second Year	https://myanimeli dena.com/images/ar
3	721	Princess Tutu	Princess Tutu	プリンセスチュチュ	NaN	https://myanimeli dena.com/images/ar
4	12365	Bakuman. 3rd Season	Bakuman.	バクマン。	Bakuman Season 3	https://myanimeli dena.com/images/ar
...	...	...	...	...	...	
14469	26089	Gutchonpa Omoshiro Hanashi	NaN	グッチョンパおもしろ話	NaN	https://myanimeli dena.com/images/ar
14470	21525	Geba Geba Shou Time!	NaN	ゲバゲバ笑タイム!	NaN	https://myanimeli dena.com/images/ar
14471	37897	Godzilla: Hoshi wo Kuu Mono	NaN	GODZILLA -星を喰う者-	Godzilla Part 3, Godzilla: Eater of Stars	https://myanimeli dena.com/images/ar
14472	34193	Nippon Mukashibanashi: Sannen Netarou	NaN	日本昔ばなし 三ねん寝太郎	NaN	https://myanimeli dena.com/images/ar
14473	37908	Senjou no Valkyria Special	NaN	戦場のヴァルキュリア Valkyria Chronicles	Senjou no Valkyria Fake Movie Promo	https://myanimeli dena.com/images/ar

14474 rows × 31 columns

Task 2: Statistcs (mean, min, max, etc. depending on the data types). Use box plots and other similar plots to illustrate it

The table below shows the statistics of the data set for each feature.

In [ ]:

```python
df.drop(columns=['anime_id']).describe()
```

Out[ ]:

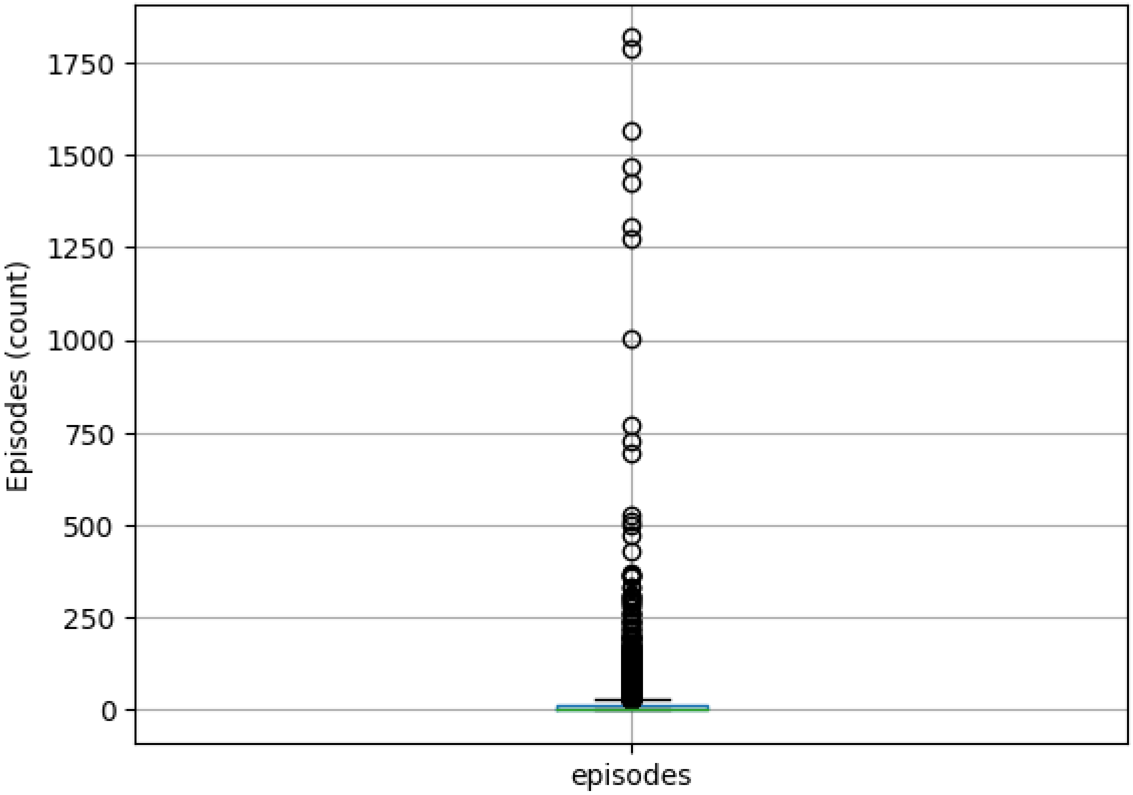
	episodes	score	scored_by	rank	popularity	members	
count	14474.000000	14474.000000	1.447400e+04	12901.000000	14474.000000	1.447400e+04	1
mean	11.310971	6.144179	1.146319e+04	6439.625068	7220.277256	2.297275e+04	
std	43.449161	1.460617	4.311072e+04	3719.462602	4168.959000	7.499075e+04	1
min	0.000000	0.000000	0.000000e+00	0.000000	0.000000	0.000000e+00	
25%	1.000000	5.550000	4.600000e+01	3218.000000	3613.250000	2.450000e+02	
50%	1.000000	6.370000	5.010000e+02	6442.000000	7225.500000	1.682500e+03	
75%	12.000000	7.060000	3.947250e+03	9664.000000	10826.750000	1.038050e+04	
max	1818.000000	10.000000	1.009477e+06	12919.000000	14487.000000	1.456378e+06	10

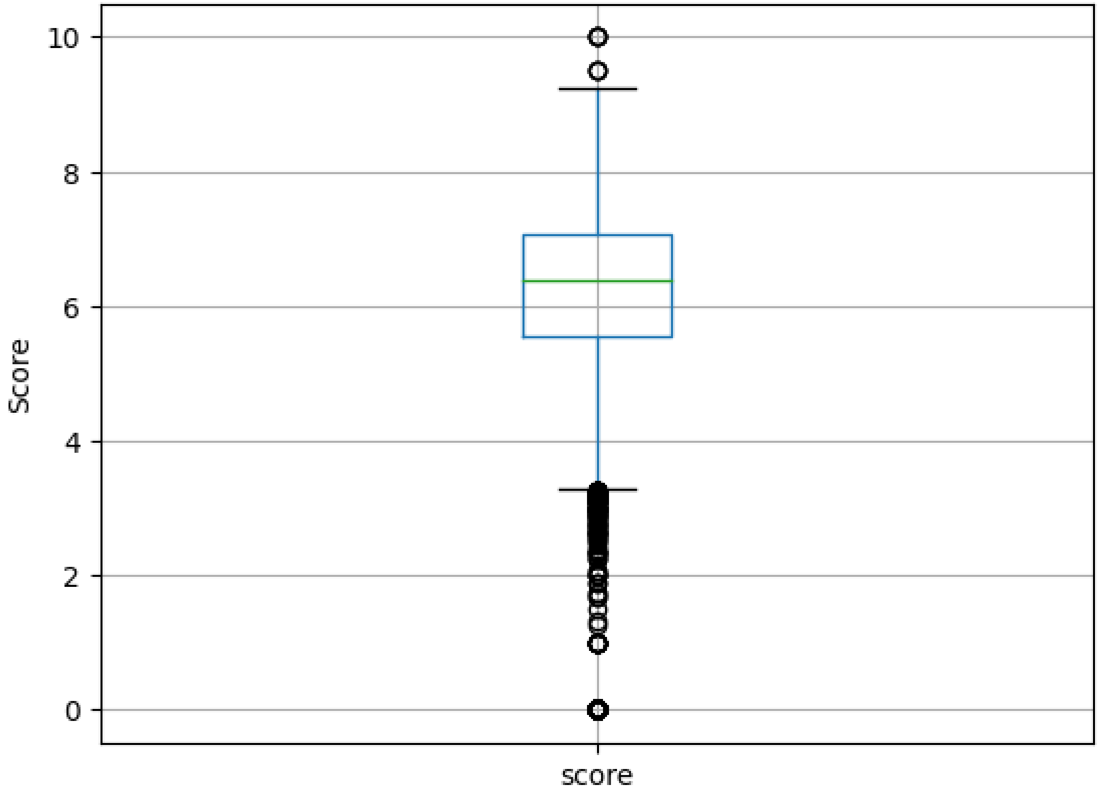
Bellow are the box plots for each of the features. The box plots show the distribution of the data, the median, the interquartile range, the minimum and maximum values, and the outliers. Wind speed and pressure have the most outliers, with pressure having the highest outlier.

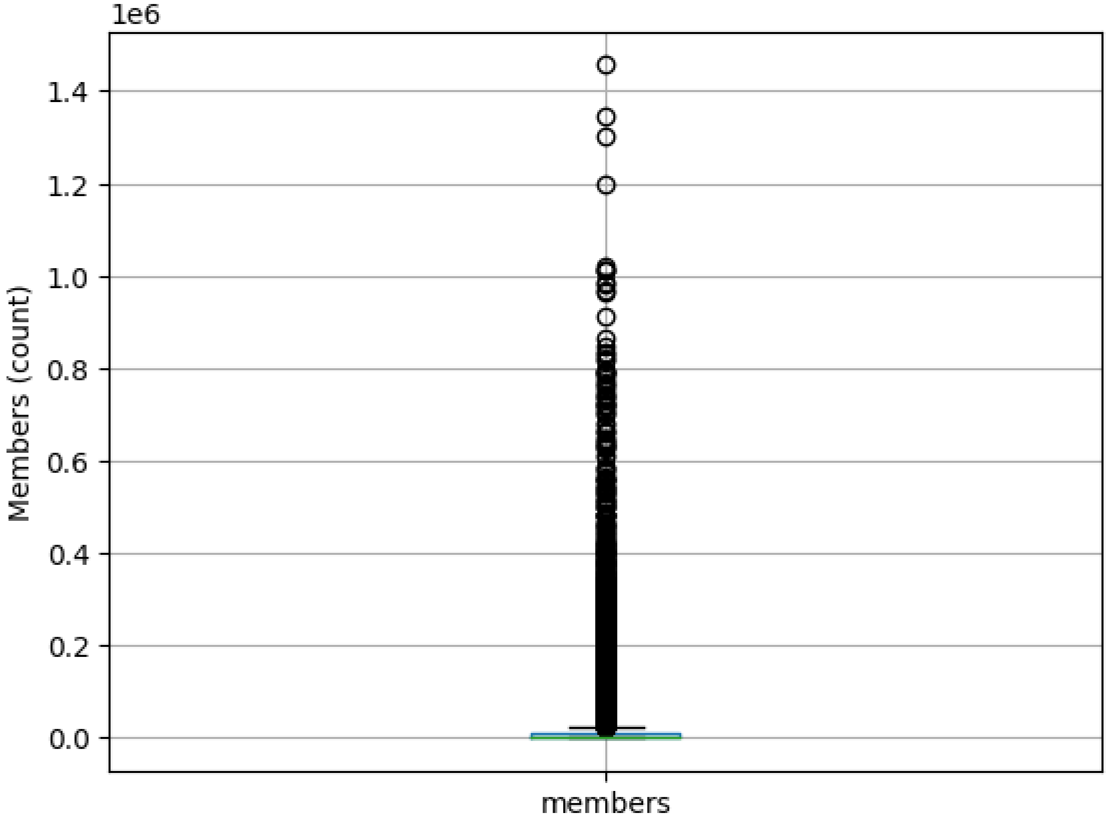
In [ ]:

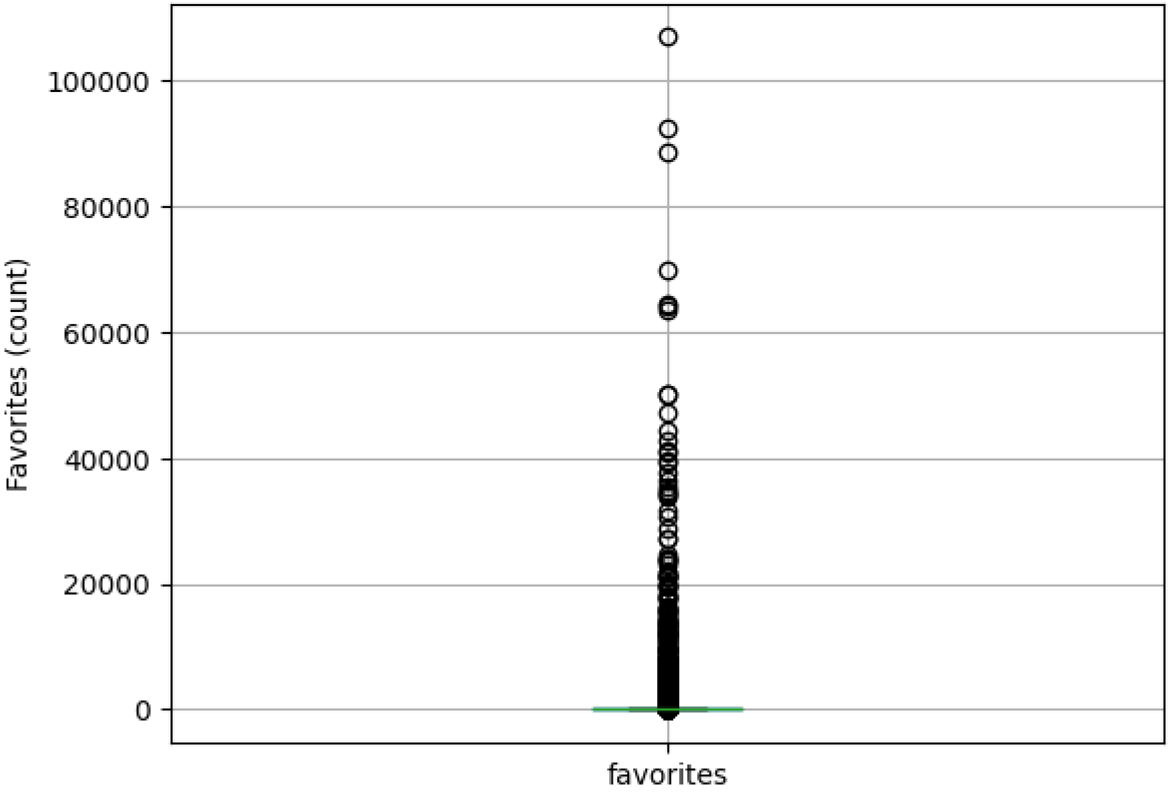
```python
import matplotlib.pyplot as plt

for column in df:
    match column:
        case "score":
            plt.figure()
            ax = df.boxplot([column])
            ax.set_ylabel("Score")
        case "favorites":
            plt.figure()
            ax = df.boxplot([column])
            ax.set_ylabel("Favorites (count)")
        case "episodes":
            plt.figure()
            ax = df.boxplot([column])
            ax.set_ylabel("Episodes (count)")
        case "members":
            plt.figure()
            ax = df.boxplot([column])
            ax.set_ylabel("Members (count)")
```


Episodes (count)
0
250
500
750
1000
1250
1500
1750
episodes


Score
0
2
4
6
8
10
score





Task 3: Create basic visualizations of your data.

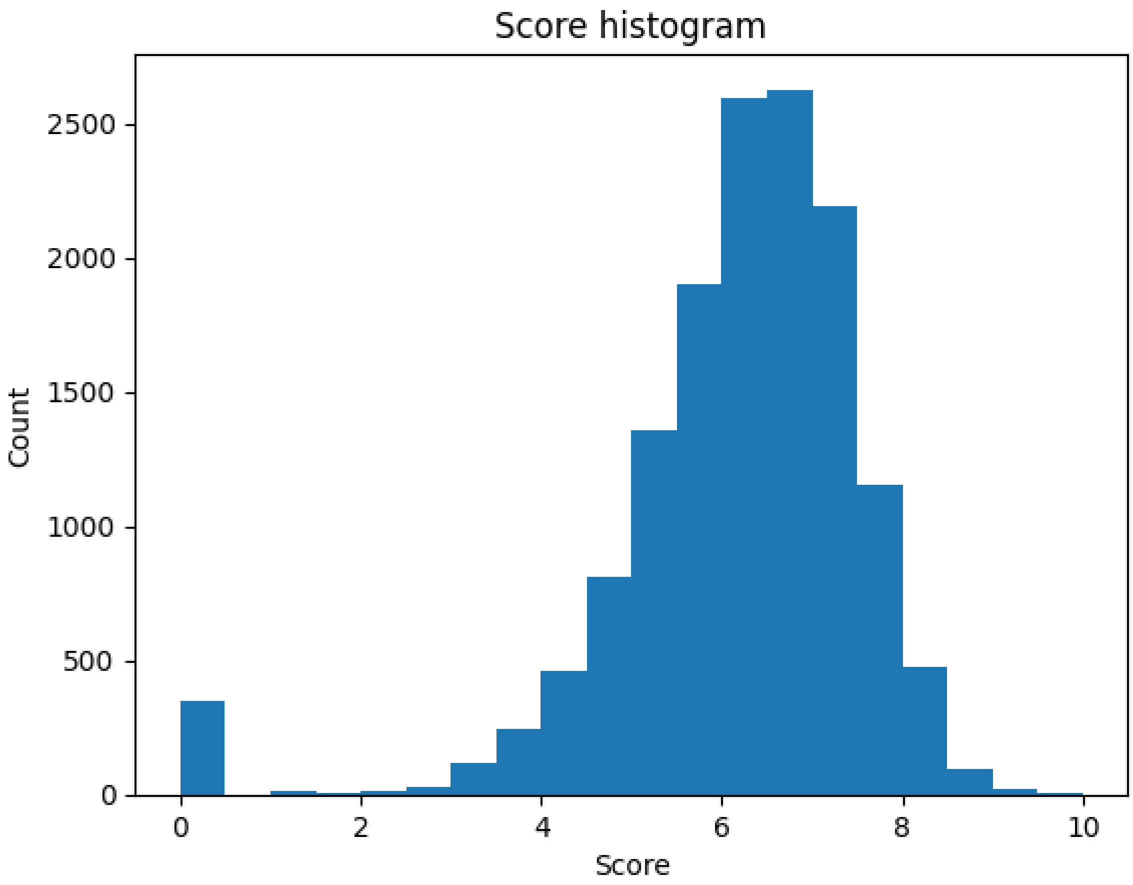
Bellow is the score histogram.

In [ ]:

```python
ax = df['score'].plot.hist(bins=20)
ax.set_ylabel("Count")
ax.set_xlabel("Score")
ax.set_title("Score histogram")
```

Out[ ]:

```
Text(0.5, 1.0, 'Score histogram')
```



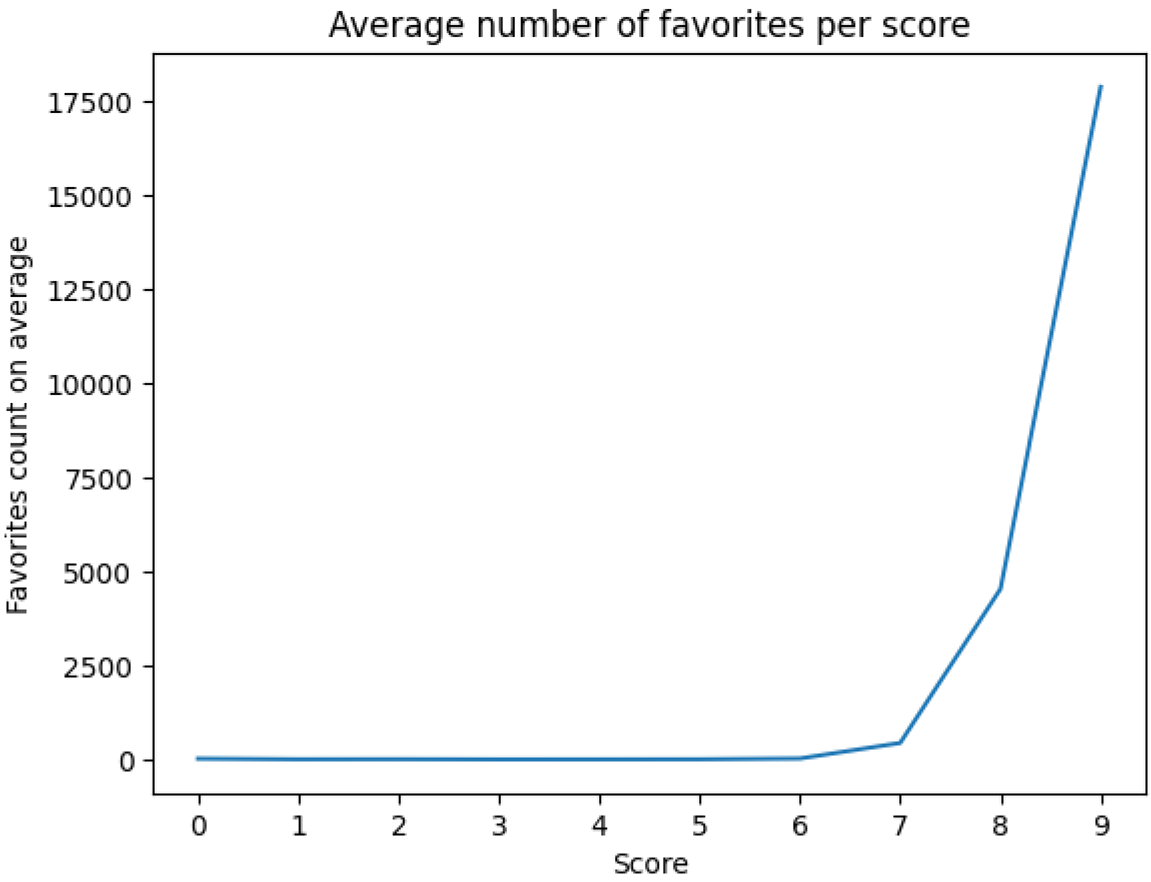
Bellow is the score relationship with the amount of favorites.

In [ ]:

```python
import numpy as np
ax = df.groupby(pd.cut(df["score"], bins=10))["favorites"].mean().plot()
ax.set_xlabel("Score")
ax.set_ylabel("Favorites count on average")
ax.set_xticks(np.arange(0, 10, 1))
ax.set_xticklabels(np.arange(0, 10, 1))
ax.set_title("Average number of favorites per score")
```

Out[ ]:

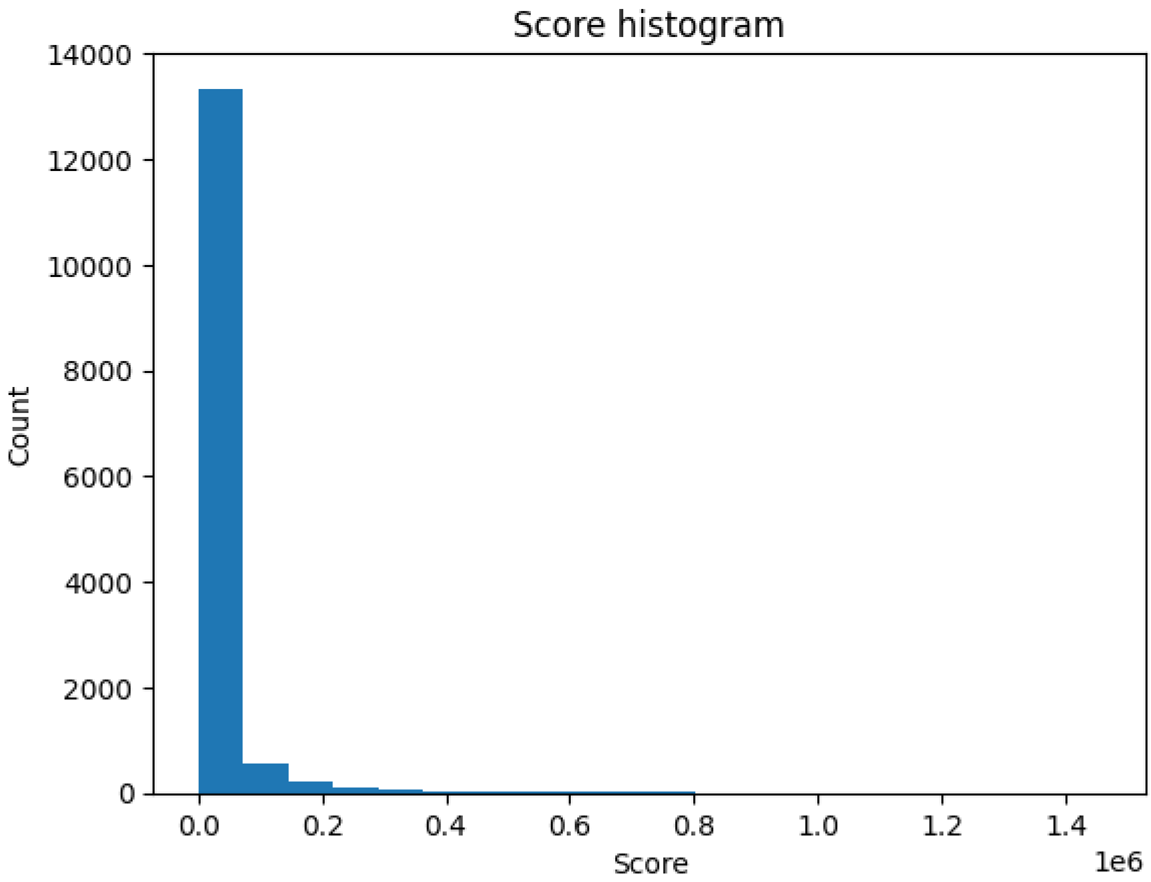
```
Text(0.5, 1.0, 'Average number of favorites per score')
```



Bellow is the histogram for the number of members.

```
In [ ]: ax = df['members'].plot.hist(bins=20)
        ax.set_ylabel("Members")
        ax.set_xlabel("Count")
        ax.set_title("Members histogram")
```

```
Out[ ]: Text(0.5, 1.0, 'Score histogram')
```



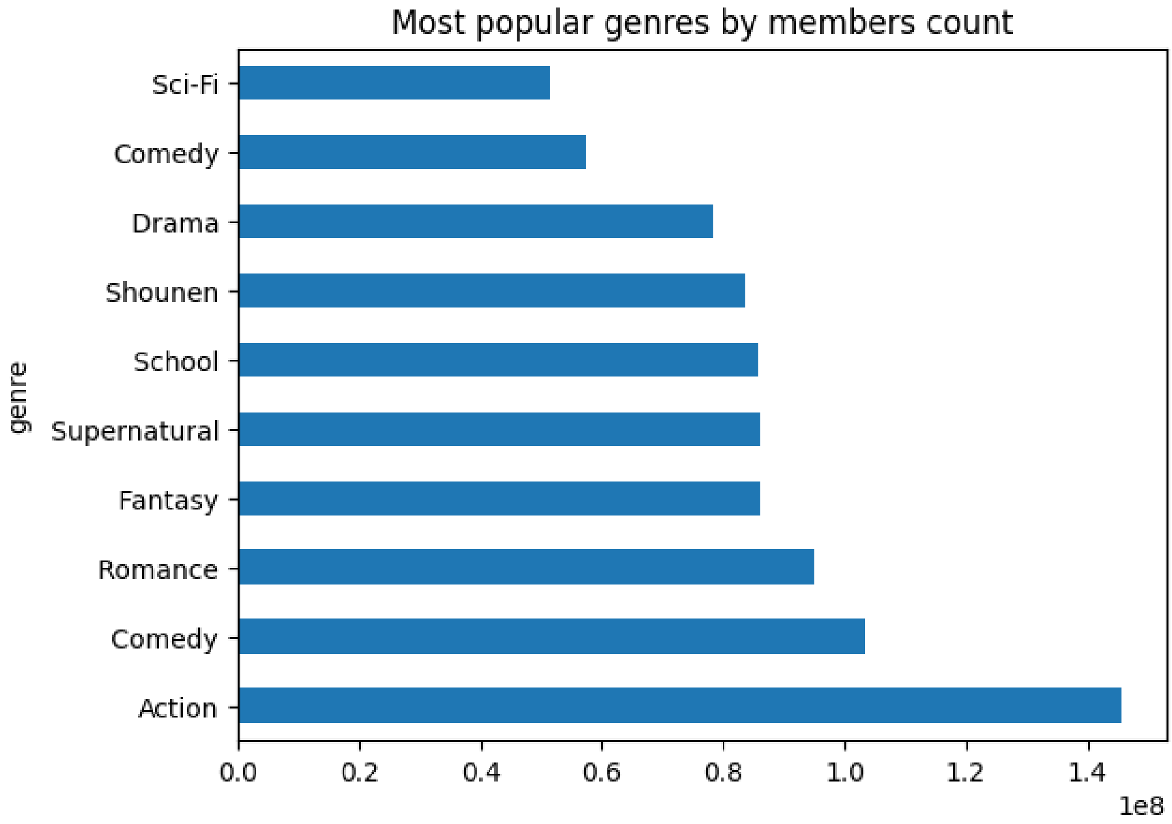
Bellow is a bar graph representing the most popular genres.

In [ ]:

```
(
    df[["members", "genre"]]
    .dropna()
    .assign(genre=df["genre"].str.split(","))
    .explode(["genre"])
    .groupby("genre")
    .agg([np.sum])
    .sort_values(by=("members", "sum"), ascending=False)
    .head(10)
    .plot(
        kind="barh",
        y="members",
        legend=False,
        title="Most popular genres by members count",
    )
)
```

Out[ ]:

```
<Axes: title={'center': 'Most popular genres by members count'}, ylabel='ge
nre'>
```



Task 4: Check for periodicity in your data, show it (if there is no seasonality, show that there is no seasonality).

This task is impossible to do with my chosen dataset since there is no time dimention.