Rinex-Education Research Center

Rinex, 823, 2nd floor, 27th Main, HSR Layout, Sector 1, Bangalore - 560102, Karnataka, India

Exploratory Data Analysis

Mini Project report submitted in partial fulfillment of the requirement for the course of

DATA SCIENCE

Submitted By

NAME	**SUMAN K**
COLLEGE NAME	**KNS Institute of Technology**
BRANCH	**Information Science & Engineering**
YEAR	**4th-year**

Under Guidance

Ameen Manna

November-2022

Source Code with Snapshots

#dataset :/content/anime_movie.csv

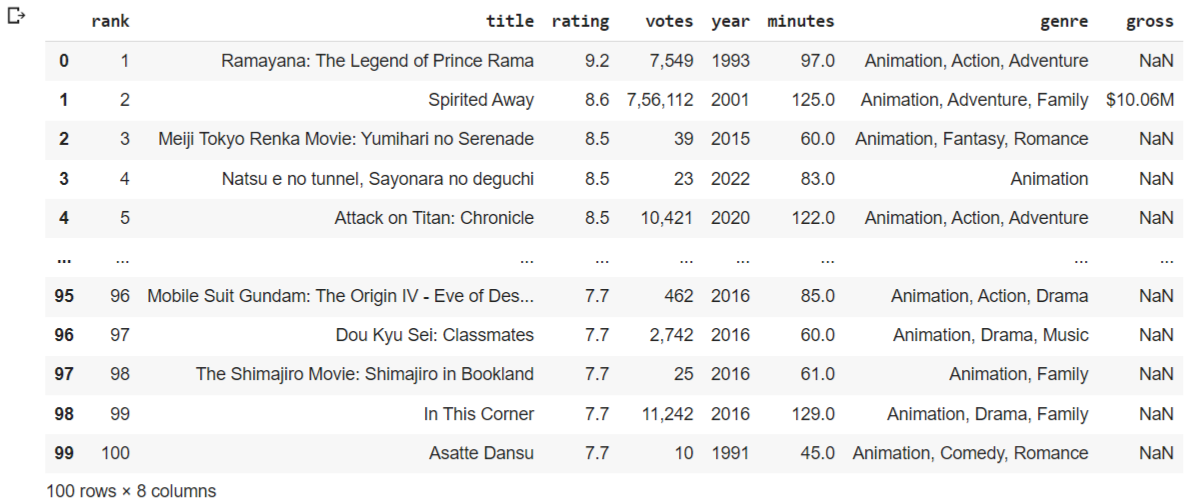
#This dataset is about the anime movies released from 1970s to the present year

# create dataframe

import pandas as pd

df=pd.read_csv('/content/anime_movie.csv')

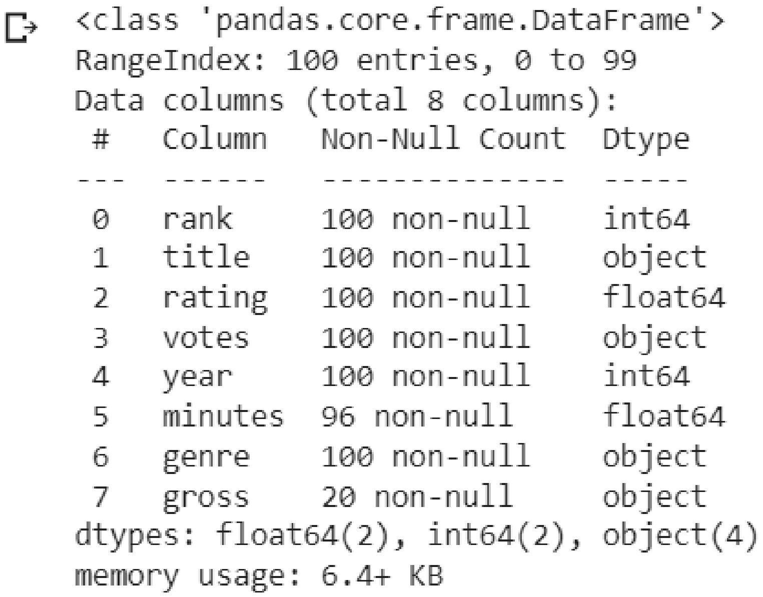
df



	rank	title	rating	votes	year	minutes	genre	gross
0	1	Ramayana: The Legend of Prince Rama	9.2	7,549	1993	97.0	Animation, Action, Adventure	NaN
1	2	Spirited Away	8.6	7,56,112	2001	125.0	Animation, Adventure, Family	$10.06M
2	3	Meiji Tokyo Renka Movie: Yumihari no Serenade	8.5	39	2015	60.0	Animation, Fantasy, Romance	NaN
3	4	Natsu e no tunnel, Sayonara no deguchi	8.5	23	2022	83.0	Animation	NaN
4	5	Attack on Titan: Chronicle	8.5	10,421	2020	122.0	Animation, Action, Adventure	NaN
...	...	...	...	...	...	...	...	...
95	96	Mobile Suit Gundam: The Origin IV - Eve of Des...	7.7	462	2016	85.0	Animation, Action, Drama	NaN
96	97	Dou Kyu Sei: Classmates	7.7	2,742	2016	60.0	Animation, Drama, Music	NaN
97	98	The Shimajiro Movie: Shimajiro in Bookland	7.7	25	2016	61.0	Animation, Family	NaN
98	99	In This Corner	7.7	11,242	2016	129.0	Animation, Drama, Family	NaN
99	100	Asatte Dansu	7.7	10	1991	45.0	Animation, Comedy, Romance	NaN

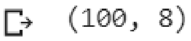
100 rows × 8 columns

df.info() #gives the complete infromation abt our dataframe



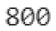
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 8 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   rank     100 non-null    int64
 1   title    100 non-null    object
 2   rating   100 non-null    float64
 3   votes    100 non-null    object
 4   year     100 non-null    int64
 5   minutes  96 non-null     float64
 6   genre    100 non-null    object
 7   gross    20 non-null     object
dtypes: float64(2), int64(2), object(4)
memory usage: 6.4+ KB
```

df.shape #displays rows n cols



```
(100, 8)
```

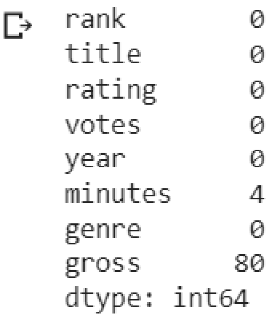
df.size #total no. of elements in dataframe



```
800
```

#to check the null values or missing values offically

df.isnull().sum()

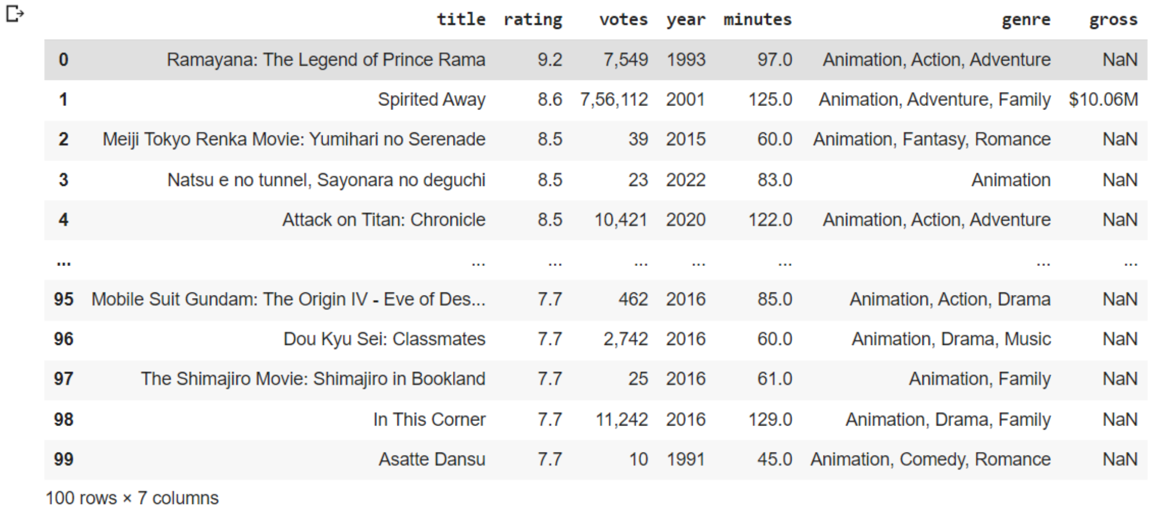


```
rank        0
title       0
rating      0
votes       0
year        0
minutes     4
genre       0
gross      80
dtype: int64
```

#as rank column is not required we will drop it

df1=df.drop(['rank'],axis=1)

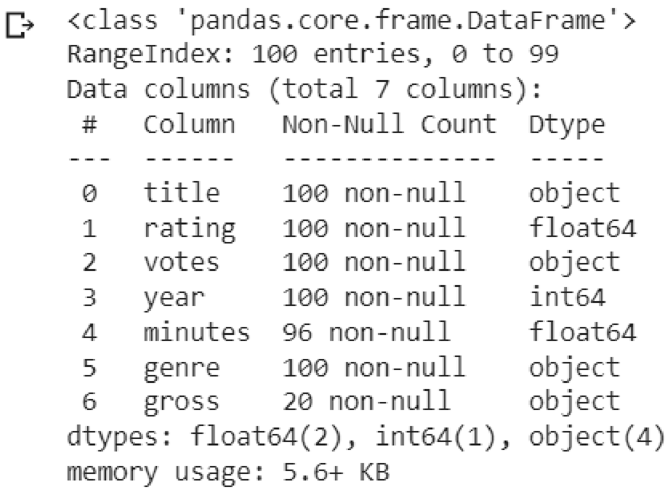
df1



	title	rating	votes	year	minutes	genre	gross
0	Ramayana: The Legend of Prince Rama	9.2	7,549	1993	97.0	Animation, Action, Adventure	NaN
1	Spirited Away	8.6	7,56,112	2001	125.0	Animation, Adventure, Family	$10.06M
2	Meiji Tokyo Renka Movie: Yumihari no Serenade	8.5	39	2015	60.0	Animation, Fantasy, Romance	NaN
3	Natsu e no tunnel, Sayonara no deguchi	8.5	23	2022	83.0	Animation	NaN
4	Attack on Titan: Chronicle	8.5	10,421	2020	122.0	Animation, Action, Adventure	NaN
...	...	...	...	...	...	...	...
95	Mobile Suit Gundam: The Origin IV - Eve of Des...	7.7	462	2016	85.0	Animation, Action, Drama	NaN
96	Dou Kyu Sei: Classmates	7.7	2,742	2016	60.0	Animation, Drama, Music	NaN
97	The Shimajiro Movie: Shimajiro in Bookland	7.7	25	2016	61.0	Animation, Family	NaN
98	In This Corner	7.7	11,242	2016	129.0	Animation, Drama, Family	NaN
99	Asatte Dansu	7.7	10	1991	45.0	Animation, Comedy, Romance	NaN

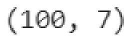
100 rows × 7 columns

```
df1.info()
```



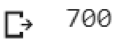
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 7 columns):
 #   Column   Non-Null Count  Dtype  
---  ------   --------------  -----  
 0   title    100 non-null    object 
 1   rating   100 non-null    float64
 2   votes    100 non-null    object 
 3   year     100 non-null    int64  
 4   minutes  96 non-null     float64
 5   genre    100 non-null    object 
 6   gross    20 non-null     object 
dtypes: float64(2), int64(1), object(4)
memory usage: 5.6+ KB
```

```
df1.shape
```



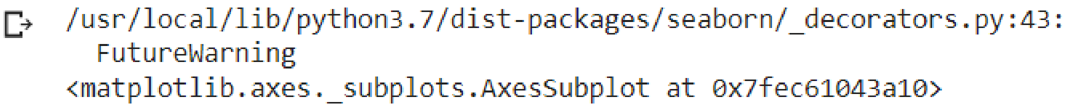
```
(100, 7)
```

```
df1.size
```

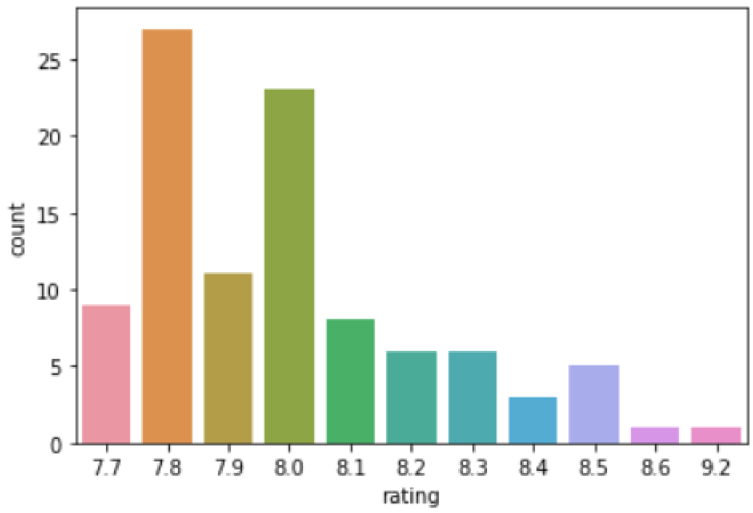


```
700
```

```
#visualisation
import seaborn as sns
sns.countplot(df1['rating'])
```

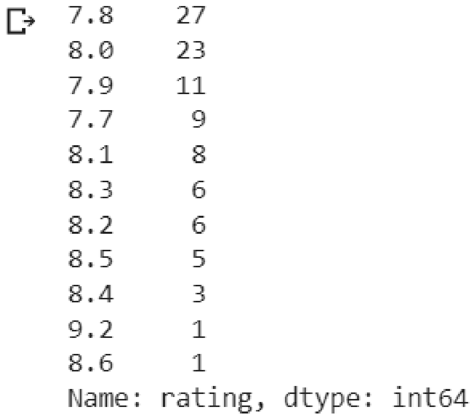


```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43:
  FutureWarning
<matplotlib.axes._subplots.AxesSubplot at 0x7fec61043a10>
```



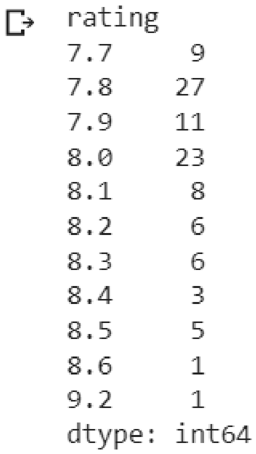
#to know exact count of movies with same rating

df1.rating.value_counts()



```
7.8    27
8.0    23
7.9    11
7.7     9
8.1     8
8.3     6
8.2     6
8.5     5
8.4     3
9.2     1
8.6     1
Name: rating, dtype: int64
```

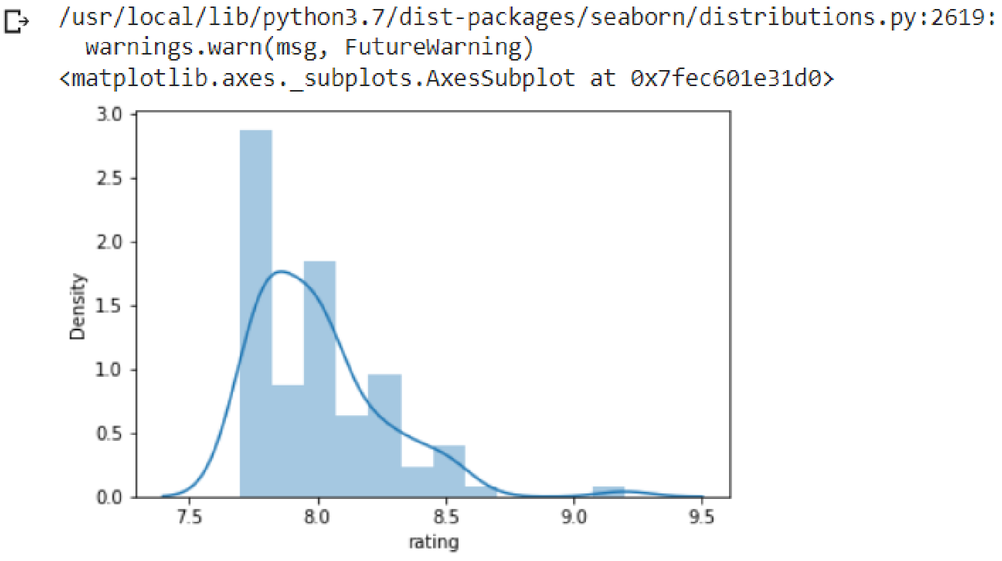
df1.groupby('rating').size()



```
rating
7.7     9
7.8    27
7.9    11
8.0    23
8.1     8
8.2     6
8.3     6
8.4     3
8.5     5
8.6     1
9.2     1
dtype: int64
```

#visualisation of rating column

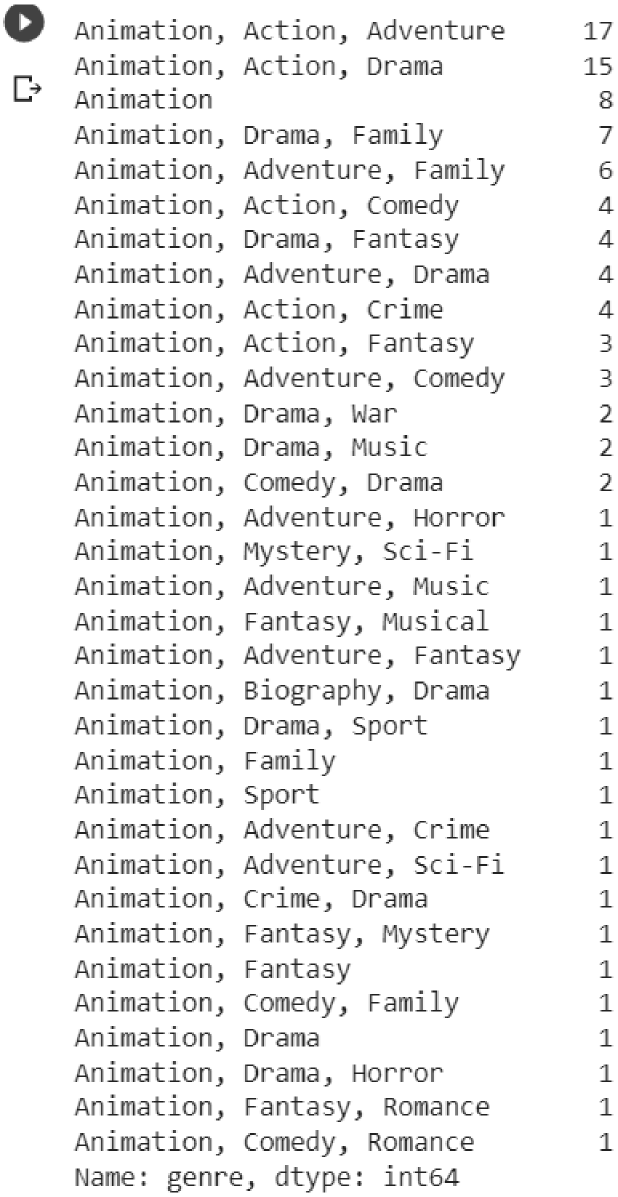
sns.distplot(df1['rating'])



```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619:
  warnings.warn(msg, FutureWarning)
<matplotlib.axes._subplots.AxesSubplot at 0x7fec601e31d0>
```

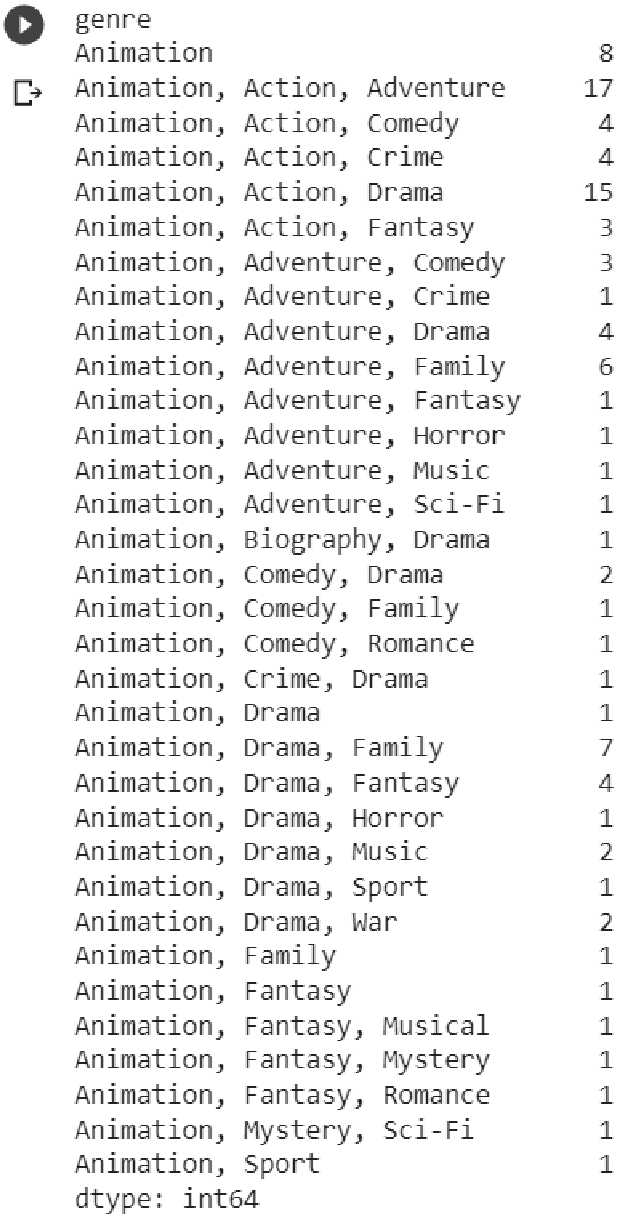
#to know exact count of movies with same genres

df1.genre.value_counts()



```
Animation, Action, Adventure       17
Animation, Action, Drama           15
Animation                           8
Animation, Drama, Family            7
Animation, Adventure, Family        6
Animation, Action, Comedy           4
Animation, Drama, Fantasy           4
Animation, Adventure, Drama         4
Animation, Action, Crime            4
Animation, Action, Fantasy          3
Animation, Adventure, Comedy        3
Animation, Drama, War               2
Animation, Drama, Music             2
Animation, Comedy, Drama            2
Animation, Adventure, Horror        1
Animation, Mystery, Sci-Fi          1
Animation, Adventure, Music         1
Animation, Fantasy, Musical         1
Animation, Adventure, Fantasy       1
Animation, Biography, Drama         1
Animation, Drama, Sport             1
Animation, Family                   1
Animation, Sport                    1
Animation, Adventure, Crime         1
Animation, Adventure, Sci-Fi        1
Animation, Crime, Drama             1
Animation, Fantasy, Mystery         1
Animation, Fantasy                  1
Animation, Comedy, Family           1
Animation, Drama                    1
Animation, Drama, Horror            1
Animation, Fantasy, Romance         1
Animation, Comedy, Romance          1
Name: genre, dtype: int64
```

df1.groupby('genre').size()

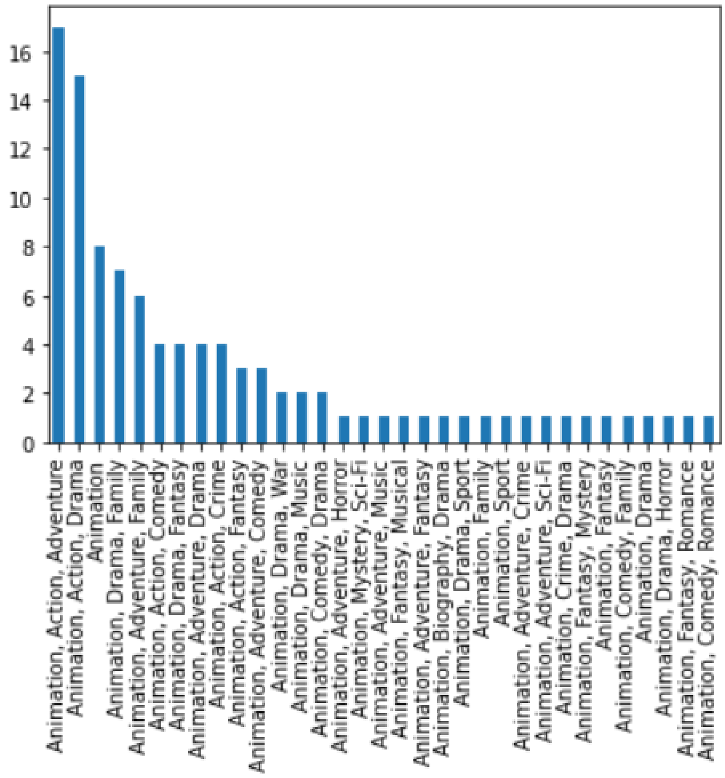


```
genre
Animation                          8
Animation, Action, Adventure      17
Animation, Action, Comedy          4
Animation, Action, Crime           4
Animation, Action, Drama          15
Animation, Action, Fantasy         3
Animation, Adventure, Comedy       3
Animation, Adventure, Crime        1
Animation, Adventure, Drama        4
Animation, Adventure, Family       6
Animation, Adventure, Fantasy      1
Animation, Adventure, Horror       1
Animation, Adventure, Music        1
Animation, Adventure, Sci-Fi       1
Animation, Biography, Drama        1
Animation, Comedy, Drama           2
Animation, Comedy, Family          1
Animation, Comedy, Romance         1
Animation, Crime, Drama            1
Animation, Drama                   1
Animation, Drama, Family           7
Animation, Drama, Fantasy          4
Animation, Drama, Horror           1
Animation, Drama, Music            2
Animation, Drama, Sport            1
Animation, Drama, War              2
Animation, Family                  1
Animation, Fantasy                 1
Animation, Fantasy, Musical        1
Animation, Fantasy, Mystery        1
Animation, Fantasy, Romance        1
Animation, Mystery, Sci-Fi         1
Animation, Sport                   1
dtype: int64
```

#visualisation of genre column

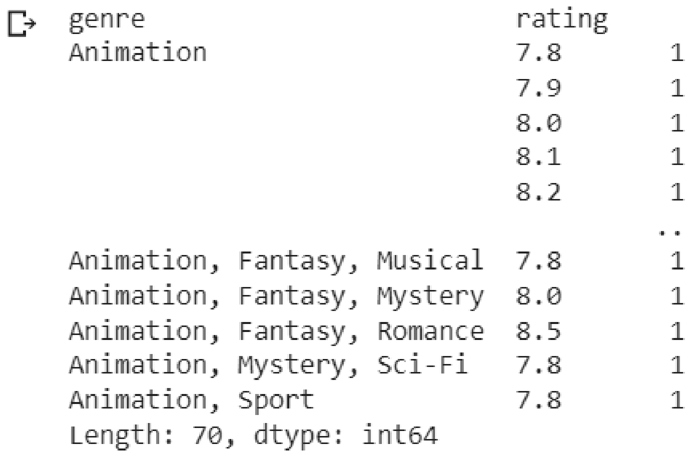
df1['genre'].value_counts().plot(kind='bar')

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fec60122750>
```



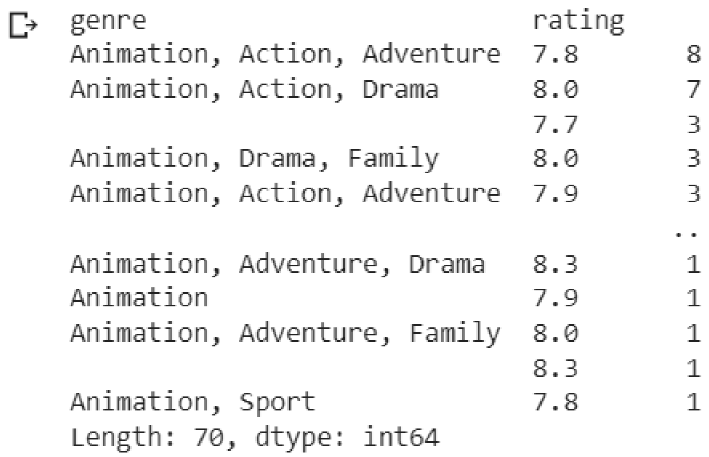
#grouping two columns ('genre' & 'rating')

df1.groupby(['genre','rating']).size()



```
genre                         rating
Animation                     7.8       1
                              7.9       1
                              8.0       1
                              8.1       1
                              8.2       1
                                       ..
Animation, Fantasy, Musical   7.8       1
Animation, Fantasy, Mystery   8.0       1
Animation, Fantasy, Romance   8.5       1
Animation, Mystery, Sci-Fi    7.8       1
Animation, Sport              7.8       1
Length: 70, dtype: int64
```

df1[['genre','rating']].value_counts()

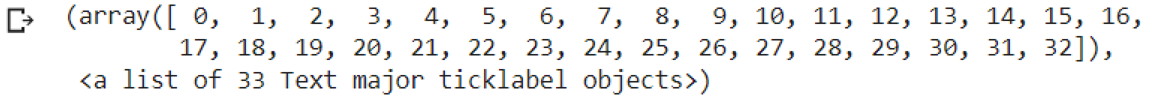


```
genre                          rating
Animation, Action, Adventure   7.8       8
Animation, Action, Drama       8.0       7
                               7.7       3
Animation, Drama, Family       8.0       3
Animation, Action, Adventure   7.9       3
                                        ..
Animation, Adventure, Drama    8.3       1
Animation                      7.9       1
Animation, Adventure, Family   8.0       1
                               8.3       1
Animation, Sport               7.8       1
Length: 70, dtype: int64
```

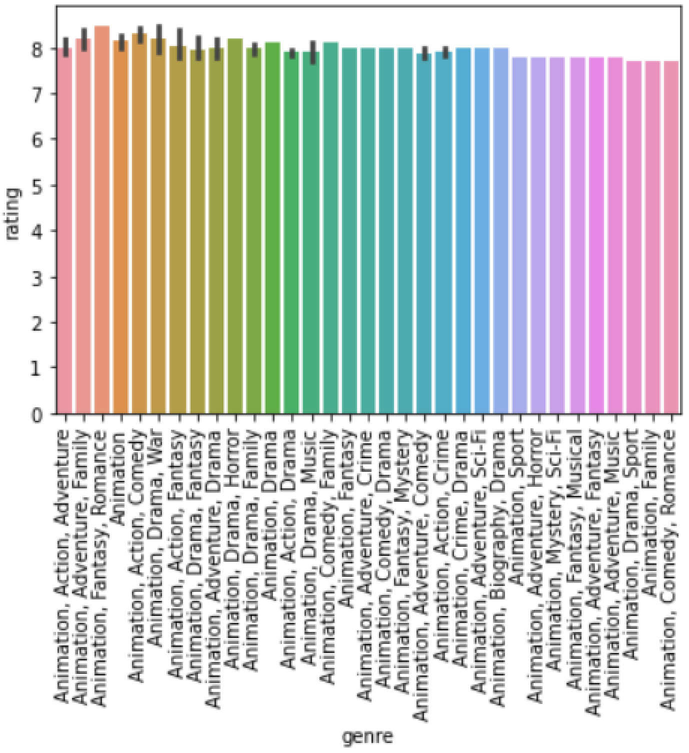
#visualisation of genre & rating columns

sns.barplot(x=df1['genre'],y=df1['rating'])

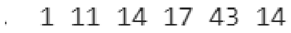
plt.xticks(rotation=90)



```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
        17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32]),
 <a list of 33 Text major ticklabel objects>)
```



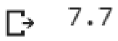
```
#now let us divide the year column into decades
import numpy as np
dec_1=np.sum((df1['year']>=1970)&(df1['year']<1980))
dec_2=np.sum((df1['year']>=1980)&(df1['year']<1990))
dec_3=np.sum((df1['year']>=1990)&(df1['year']<2000))
dec_4=np.sum((df1['year']>=2000)&(df1['year']<2010))
dec_5=np.sum((df1['year']>=2010)&(df1['year']<2020))
dec_6=np.sum((df1['year']>=2020)&(df1['year']<2030))
print(dec_1,dec_2,dec_3,dec_4,dec_5,dec_6)
```



```
1 11 14 17 43 14
```

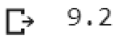
```
#from output we come to know that
#in dec_1 only 1 movie was released
#in dec_2 11 movies was released
#in dec_3 14 movies was released
#in dec_4 17 movies was released
#in dec_5 43 movies was released
#in dec_6 14 movies was released
```

```
#to find out lowest rating
np.min(df1['rating'])
```



```
7.7
```

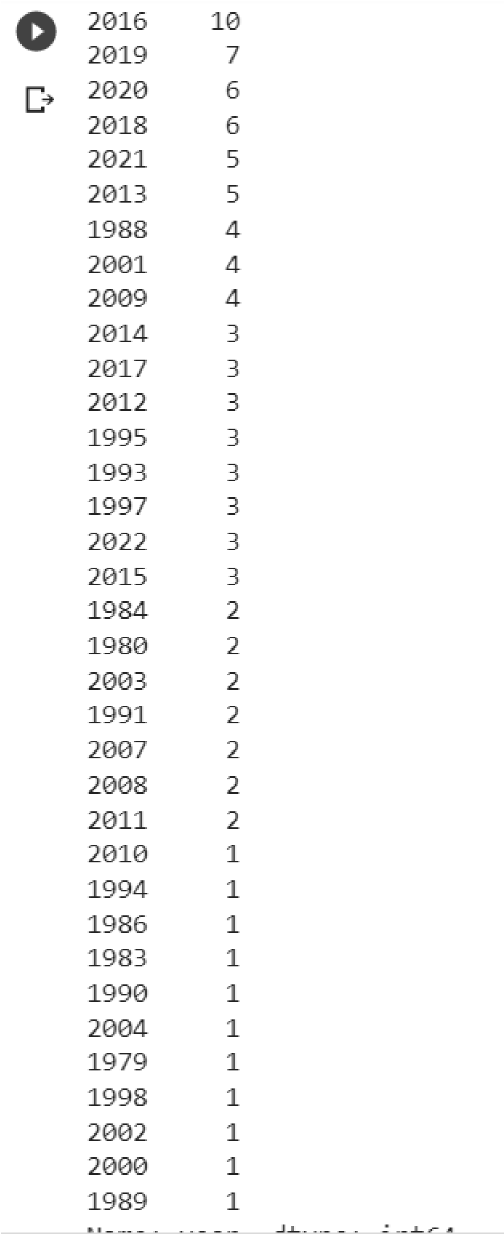
```
#to find the highest rating
np.max(df1['rating'])
```



```
9.2
```

#to know which year released more movies

df1['year'].value_counts()



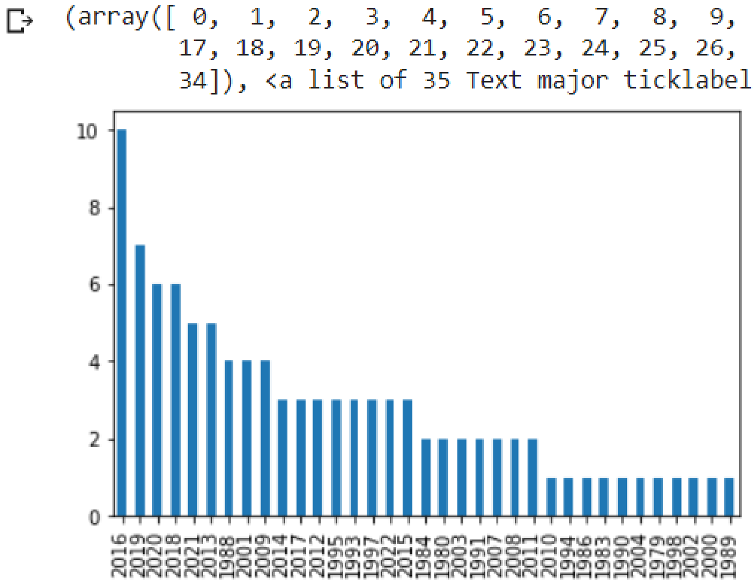
```
2016    10
2019     7
2020     6
2018     6
2021     5
2013     5
1988     4
2001     4
2009     4
2014     3
2017     3
2012     3
1995     3
1993     3
1997     3
2022     3
2015     3
1984     2
1980     2
2003     2
1991     2
2007     2
2008     2
2011     2
2010     1
1994     1
1986     1
1983     1
1990     1
2004     1
1979     1
1998     1
2002     1
2000     1
1989     1
Name: year, dtype: int64
```

#visualisation of year column

import matplotlib.pyplot as plt

df1['year'].value_counts().plot(kind='bar')

plt.xticks(rotation='90')



```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
        17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
        34]), <a list of 35 Text major ticklabel objects>)
```

Dataset URL: https://raw.githubusercontent.com/Sumank02/datasets/main/anime_movies.csv