import the data set

In [2]:
```
import pandas as pd
```

import dataset

In [5]:
```
ratings = pd.read_csv(r"C:\Users\jayes\OneDrive\Desktop\NareshIT\kaggle\archive\rating.csv")
```

In [9]:
```
ratings.shape
```

Out[9]:
```
(20000263, 4)
```

In [11]:
```
movies = pd.read_csv(r"C:\Users\jayes\OneDrive\Desktop\NareshIT\kaggle\archive\movie.csv")
```

In [13]:
```
movies.shape
```

Out[13]:
```
(27278, 3)
```

In [15]:
```
tag = pd.read_csv(r"C:\Users\jayes\OneDrive\Desktop\NareshIT\kaggle\archive\tag.csv")
tag.shape
```

Out[15]:
```
(465564, 4)
```

In [17]:
```
ratings.columns
```

Out[17]:
```
Index(['userId', 'movieId', 'rating', 'timestamp'], dtype='object')
```

In [21]:
```
del ratings['timestamp']
del tag['timestamp']
```

In [23]:
```
print(ratings.columns)
print(ratings.columns)
```

```
Index(['userId', 'movieId', 'rating'], dtype='object')
Index(['userId', 'movieId', 'rating'], dtype='object')
```

data frames

In [27]:

```
tag.head()
```

Out[27]:

	userId	movieId	tag
0	18	4141	Mark Waters
1	65	208	dark hero
2	65	353	dark hero
3	65	521	noir thriller
4	65	592	dark hero

In [25]:

```
raw_0 = tag.iloc[0]
type[raw_0]
```

Out[25]:

```
type[userId              18
movieId            4141
tag         Mark Waters
Name: 0, dtype: object]
```

In [31]:

```
print(raw_0)
```

```
userId              18
movieId           4141
tag        Mark Waters
Name: 0, dtype: object
```

In [33]:

```
raw_0.index
```

Out[33]:

```
Index(['userId', 'movieId', 'tag'], dtype='object')
```

In [35]:

```
raw_0.userId
```

```
Out[35]: 18
```

```
In [39]: 'rating' in raw_0
```

```
Out[39]: False
```

```
In [41]: raw_0.name
```

```
Out[41]: 0
```

```
In [43]: raw_0 = raw_0.rename('first raw')
         raw_0.name
```

```
Out[43]: 'first raw'
```

```
In [47]: tag.head
```

```
Out[47]: <bound method NDFrame.head of          userId  movieId             tag
         0            18     4141     Mark Waters
         1            65      208       dark hero
         2            65      353       dark hero
         3            65      521   noir thriller
         4            65      592       dark hero
         ...         ...      ...             ...
         465559  138446    55999         dragged
         465560  138446    55999  Jason Bateman
         465561  138446    55999          quirky
         465562  138446    55999             sad
         465563  138472      923  rise to power

         [465564 rows x 3 columns]>
```

```
In [51]: tag.index
```

```
Out[51]: RangeIndex(start=0, stop=465564, step=1)
```

```
In [53]: tag.columns
```

```
Out[53]: Index(['userId', 'movieId', 'tag'], dtype='object')
```

In [57]:

```
tag.iloc[[0,11,500]]
```

Out[57]:

	userId	movieId	tag
0	18	4141	Mark Waters
11	65	1783	noir thriller
500	342	55908	entirely dialogue

descriptive statitics

In [65]:

```
ratings['rating'].describe()
```

Out[65]:

```
count    2.000026e+07
mean     3.525529e+00
std      1.051989e+00
min      5.000000e-01
25%      3.000000e+00
50%      3.500000e+00
75%      4.000000e+00
max      5.000000e+00
Name: rating, dtype: float64
```

In [67]:

```
ratings.describe()
```

Out[67]:

	userId	movieId	rating
count	2.000026e+07	2.000026e+07	2.000026e+07
mean	6.904587e+04	9.041567e+03	3.525529e+00
std	4.003863e+04	1.978948e+04	1.051989e+00
min	1.000000e+00	1.000000e+00	5.000000e-01
25%	3.439500e+04	9.020000e+02	3.000000e+00
50%	6.914100e+04	2.167000e+03	3.500000e+00
75%	1.036370e+05	4.770000e+03	4.000000e+00
max	1.384930e+05	1.312620e+05	5.000000e+00

In [71]:

```
ratings['rating'].mean()
```

Out[71]:

```
3.5255285642993797
```

In [73]:

```
ratings.mean()
```

Out[73]:

```
userId     69045.872583
movieId     9041.567330
rating         3.525529
dtype: float64
```

In [75]:

```
ratings['rating'].min()
```

Out[75]:

```
0.5
```

In [77]:

```
ratings['rating'].max()
```

Out[77]:

```
5.0
```

In [81]:

```
ratings['rating'].std()
```

Out[81]:

```
1.051988919275684
```

In [83]:

```
ratings['rating'].mode()
```

Out[83]:

```
0    4.0
Name: rating, dtype: float64
```

In [85]:

```
ratings.corr()
```

Out[85]:

	userId	movieId	rating
userId	1.000000	-0.000850	0.001175
movieId	-0.000850	1.000000	0.002606
rating	0.001175	0.002606	1.000000

In [89]:

```
filter1 = ratings['rating'] > 10
print(filter1)
filter1.any()
```

```
0           False
1           False
2           False
3           False
4           False
            ...
20000258    False
20000259    False
20000260    False
20000261    False
20000262    False
Name: rating, Length: 20000263, dtype: bool
```

Out[89]:

```
False
```

In [91]:

```
filter2 = ratings['rating'] > 0
filter2.all()
```

Out[91]:

```
True
```

handling missing data - data cleaning

In [109…

```
movies.shape
```

Out[109…

```
(27278, 3)
```

In [119…

```
movies.isnull().any().any()
```

Out[119…

```
False
```

- that's nice no null values

In [123…

```
ratings.shape
```

Out[123…

```
(20000263, 3)
```

In [125…

```
ratings.isnull().any().any()
```

Out[125…

```
False
```

- that's nice no null values

In [131…

```
tag.shape
```

Out[131…

```
(465564, 3)
```

In [135…

```
tag.isnull().any().any()
```

Out[135…

```
True
```

- there is some tags, which are NULL

In [140…

```
tag = tag.dropna()
```

In [142...

```
tag.isnull().any().any()
```

Out[142...

```
False
```

In [146...

```
tag.shape
```

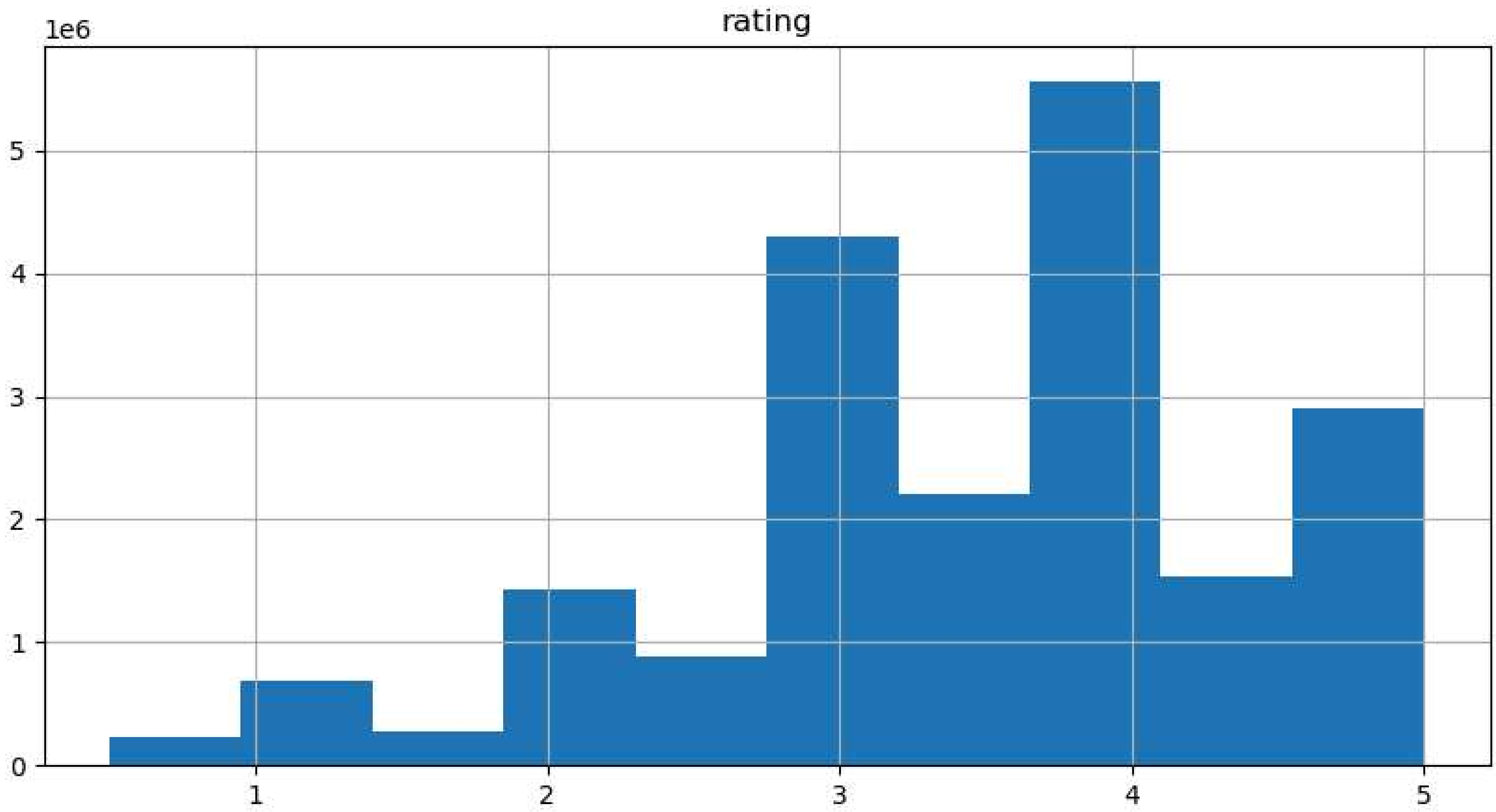
Out[146...

```
(465548, 3)
```

- that's nice, no null values. no of lines have reduced

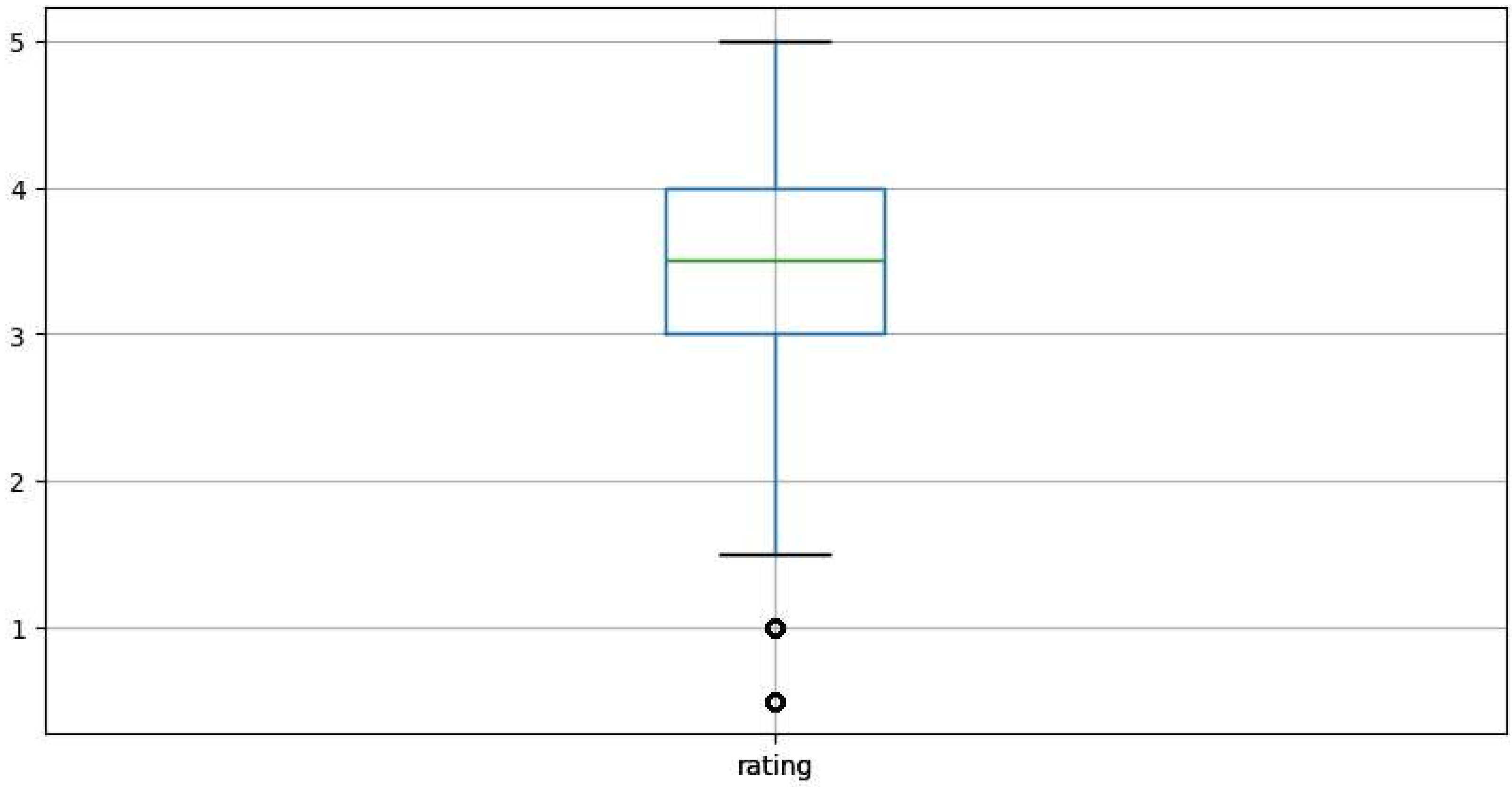
In [163...

```
import matplotlib.pyplot as plt

%matplotlib inline

ratings.hist(column='rating', figsize=(10,5))

plt.show()
```



In [167…

```
ratings.boxplot(column='rating', figsize=(10,5))

plt.show()
```



```
In [ ]:
```