kaggle

In [1]:
```
##
```

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

In [4]:
```
ratings= pd.read_csv(r'/Users/sasidharbhagavatula/Downloads/archive/ratin
```

In [7]:
```
tags=pd.read_csv(r'/Users/sasidharbhagavatula/Downloads/archive/tag.csv')
```

In [8]:
```
movie=pd.read_csv(r'/Users/sasidharbhagavatula/Downloads/archive/movie.cs
```

In [9]:
```
print(tags.columns)
```

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

In [11]:
```
print(ratings.columns)
```

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

In [12]:
```
del tags['timestamp']
```

In [14]:
```
del ratings['timestamp']
```

In [15]:
```
tags.head()
```

Out[15]:

	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 [16]:
```
ratings.head()
```

Out[16]:

	userId	movieId	rating
0	1	2	3.5
1	1	29	3.5
2	1	32	3.5
3	1	47	3.5
4	1	50	3.5

In [17]:
```
row_0 = tags.iloc[0]
type(row_0)
```

Out[17]:
```
pandas.core.series.Series
```

In [18]:
```
print(row_0)
```

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

In [19]:
```
row_0.index
```

Out[19]:
```
Index(['userId', 'movieId', 'tag'], dtype='object')
```

In [20]:
```
row_0['userId']
```

Out[20]:
```
np.int64(18)
```

In [21]:
```
'rating' in row_0
```

Out[21]:
```
False
```

In [22]:
```
row_0.name
```

Out[22]:
```
0
```

In [23]:
```
tags.head()
```

Out[23]:

	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 [24]:
```
'movieId' in row_0
```

Out[24]:
```
True
```

In [25]:
```
row_0 = row_0.rename('firstRow')
row_0.name
```

Out[25]:
```
'firstRow'
```

In [26]:
```
tags.head()
```

Out[26]:

	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 [30]:
```
print(tags.iloc[0])
```

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

In [32]:
```
print(row_0)
```

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

In [33]:
```
tags.index
```

Out[33]:
```
RangeIndex(start=0, stop=465564, step=1)
```

In [34]:
```
tags.columns
```

Out[34]:
```
Index(['userId', 'movieId', 'tag'], dtype='object')
```

In [36]:
```
tags.iloc[[0,11,500]]
```

Out[36]:

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

In [37]:
```
#discriptive statistics
```

In [39]:
```
ratings['rating'].describe()
```

Out[39]:
```
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 [40]:
```
ratings.describe()
```

Out[40]:

	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 [41]:
```
ratings['rating'].mean()
```

Out[41]:
```
np.float64(3.5255285642993797)
```

In [42]:
```
ratings.mean()
```

Out[42]:
```
userId     69045.872583
movieId     9041.567330
rating         3.525529
dtype: float64
```

In [45]:
```
ratings['rating'].min()
```

Out[45]:
```
0.5
```

In [47]:
```
ratings['rating'].max()
```

Out[47]:
```
5.0
```

In [48]:
```
ratings['rating'].std()
```

Out[48]:
```
1.051988919275684
```

In [49]:
```
ratings['rating'].mode()
```

Out[49]:
```
0    4.0
Name: rating, dtype: float64
```

In [51]:
```
ratings.corr()
```

Out[51]:

	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 [54]:
```
print(ratings['rating']>10)
```

```
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
```

In [55]:
```
filter2=ratings['rating']>0
filter2.all()
```

Out[55]:
```
np.True_
```

DATA CLEANING .. HANDELING MISSING DATA

In [56]:
```
movie.shape
```

Out[56]:
```
(27278, 3)
```

In [60]:
```
movie.isnull().any().all()    #so no null values
```

Out[60]:
```
np.False_
```

In [61]:
```
tags.shape
```

Out[61]:
```
(465564, 3)
```

In [63]:
```
tags.isnull().any().any()# we have some tags which are null
```

Out[63]:
```
np.True_
```

In [65]:
```
tags.dropna()
```

Out[65]:

	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

465548 rows × 3 columns

In [66]:
```
tags.isnull().any().any()
```

Out[66]:
```
np.True_
```

In [68]:
```
tags.shape
```

Out[68]:
```
(465564, 3)
```

In [69]:
```
tags.dropna()
```

Out[69]:

	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

465548 rows × 3 columns

In [70]:
```
tags.shape
```

Out[70]:
```
(465564, 3)
```

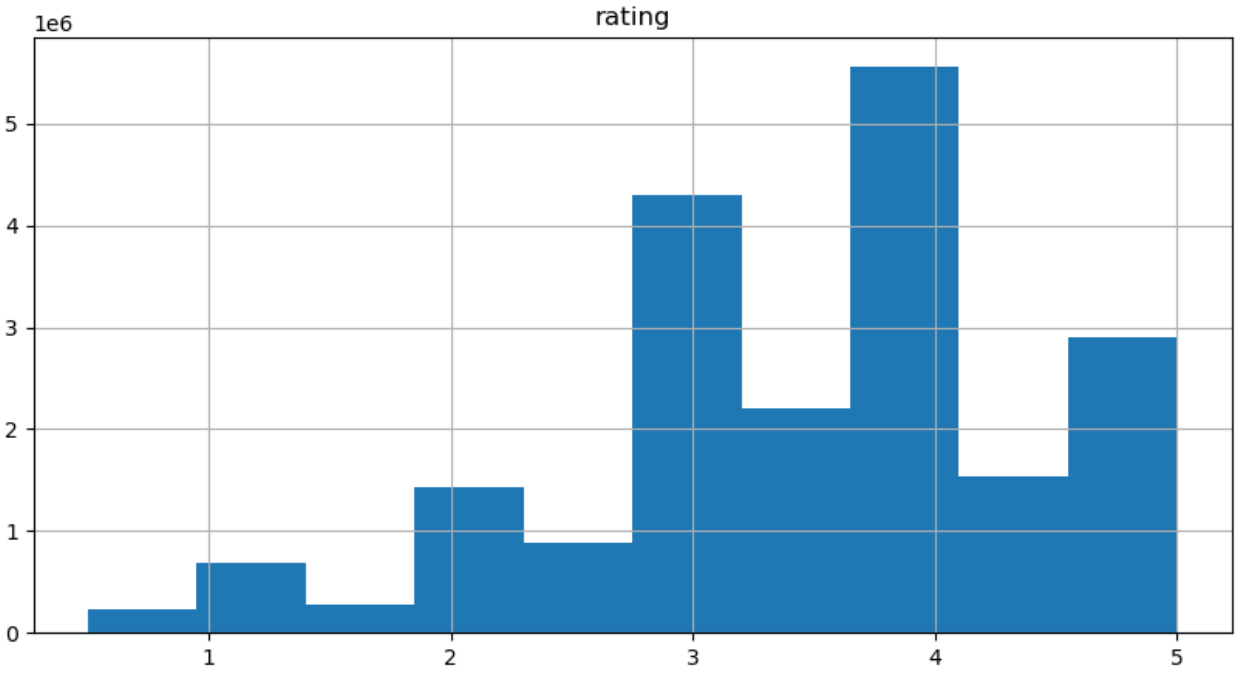
DATA VISUALIZATION

In [78]:
```
import matplotlib
import matplotlib.pyplot as plt
%matplotlib inline
import numpy as np
import pandas as pd
```

In [83]:
```
ratings.hist(column='rating', figsize=(10,5))
```

Out[83]:
```
array([[<Axes: title={'center': 'rating'}>]], dtype=object)
```

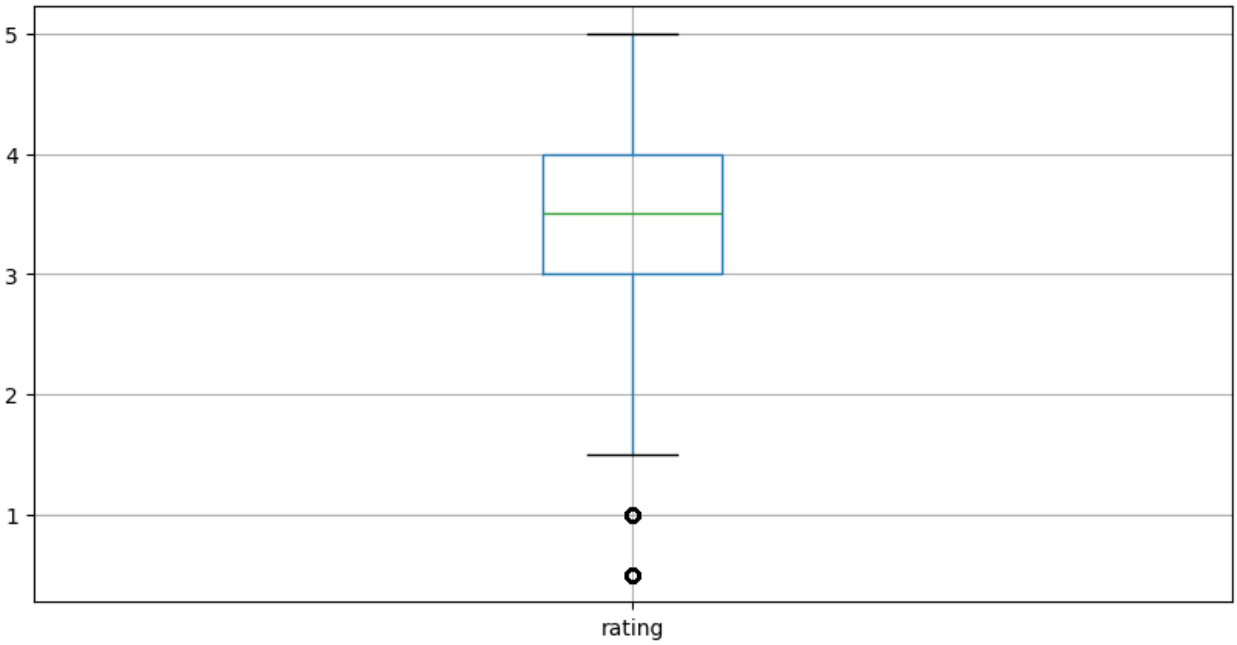
In [84]:
```
plt.show()
```



In [86]:
```
ratings.boxplot(column='rating', figsize=(10,5))
```

Out[86]: <Axes: >

In [87]:
```
plt.show()
```



SLICING OUT COLUMNS

In [89]:
```
movie[['title','genres']].head()
```

Out[89]:

	title	genres
0	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
1	Jumanji (1995)	Adventure|Children|Fantasy
2	Grumpier Old Men (1995)	Comedy|Romance
3	Waiting to Exhale (1995)	Comedy|Drama|Romance
4	Father of the Bride Part II (1995)	Comedy

In [91]:
```
movie[['title']].head()
```

Out[91]:

	title
0	Toy Story (1995)
1	Jumanji (1995)
2	Grumpier Old Men (1995)
3	Waiting to Exhale (1995)
4	Father of the Bride Part II (1995)

In [92]:
```
tags[['tag']].head()
```

Out[92]:

	tag
0	Mark Waters
1	dark hero
2	dark hero
3	noir thriller
4	dark hero

In [94]:
```
ratings[:]
```

Out[94]:

	userId	movieId	rating
0	1	2	3.5
1	1	29	3.5
2	1	32	3.5
3	1	47	3.5
4	1	50	3.5
...	...	...	...
20000258	138493	68954	4.5
20000259	138493	69526	4.5
20000260	138493	69644	3.0
20000261	138493	70286	5.0
20000262	138493	71619	2.5

20000263 rows × 3 columns

In [95]:
```
ratings[-10:]
```

Out[95]:

	userId	movieId	rating
20000253	138493	60816	4.5
20000254	138493	61160	4.0
20000255	138493	65682	4.5
20000256	138493	66762	4.5
20000257	138493	68319	4.5
20000258	138493	68954	4.5
20000259	138493	69526	4.5
20000260	138493	69644	3.0
20000261	138493	70286	5.0
20000262	138493	71619	2.5

In [97]:
```
tags[:]
```

Out[97]:

	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 × 3 columns

In [99]:
```
val=tags['tag']
```

In [100…
```
val[:]
```

Out[100…
```
0           Mark Waters
1             dark hero
2             dark hero
3          noir thriller
4             dark hero
              ...
465559          dragged
465560    Jason Bateman
465561           quirky
465562              sad
465563    rise to power
Name: tag, Length: 465564, dtype: object
```

In [102…
```
val[-10:]
```

Out[102…
```
465554                  visually appealing
465555                     family friendly
465556    Scary Movies To See on Halloween
465557                             Peter Pan
465558                  visually appealing
465559                             dragged
465560                       Jason Bateman
465561                              quirky
465562                                 sad
465563                      rise to power
Name: tag, dtype: object
```

In [ ]:

In [ ]:

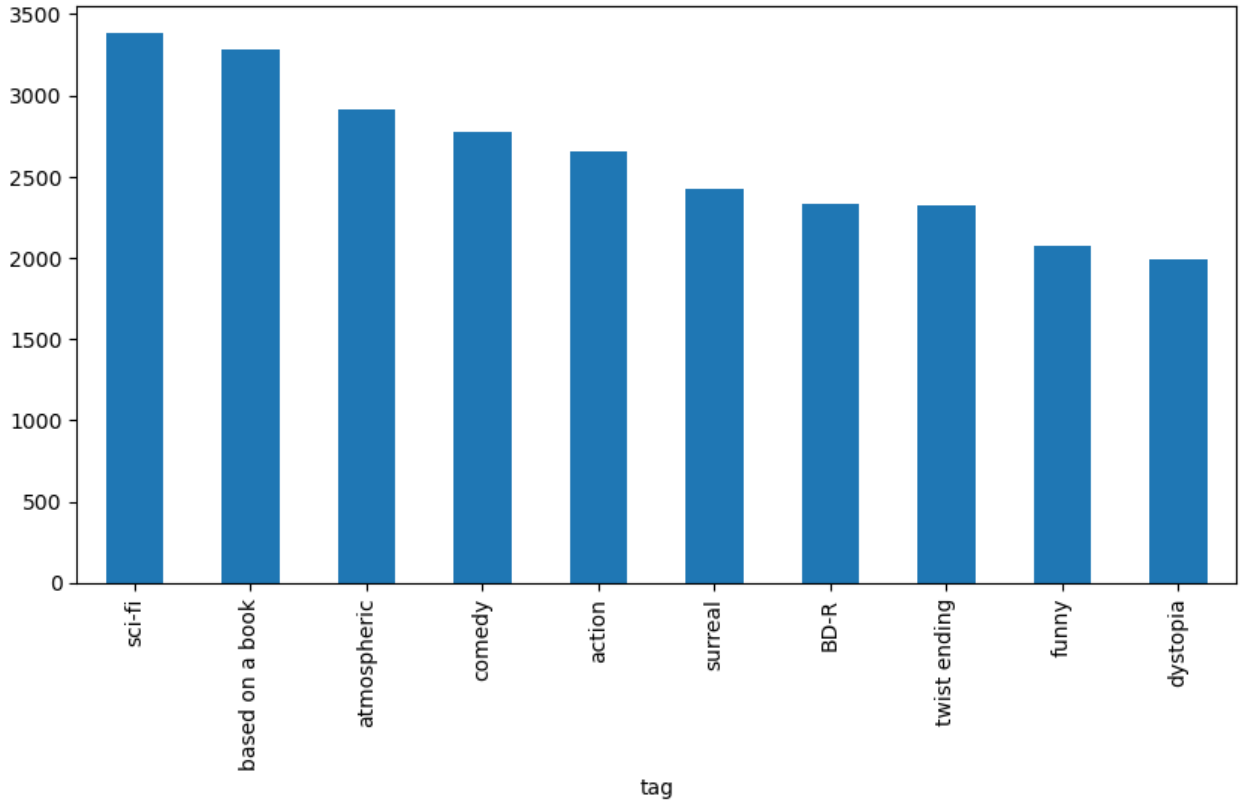
In [105…
```
tag_counts = tags['tag'].value_counts()
tag_counts[-10:]
```

Out[105…
```
tag
Hell naw                      1
This is my happy face         1
I heel toe on Uday's house    1
Why?                          1
Bobo                          1
Diamond Dallas Page           1
I'm Devon Butler!             1
No arguement                  1
Really Bad                    1
Botox                         1
Name: count, dtype: int64
```

In [113…
```
tag_counts[:10].plot(kind='bar','g', figsize=(10,5))
```

```
  Cell In[113], line 1
    tag_counts[:10].plot(kind='bar','g', figsize=(10,5))
                                                        ^
SyntaxError: positional argument follows keyword argument
```

In [111…
```
plt.show()
```


3500
3000
2500
2000
1500
1000
500
0
sci-fi
based on a book
atmospheric
comedy
action
surreal
BD-R
twist ending
funny
dystopia
tag

In [ ]: