In [1]:
```
import pandas as pd
```

In [2]:
```
movies = pd.read_csv(r"C:\Users\chait\Downloads\archive\movie.csv")
```

In [3]:
```
tags = pd.read_csv(r"C:\Users\chait\Downloads\archive\tag.csv")
```

In [4]:
```
ratings = pd.read_csv(r"C:\Users\chait\Downloads\archive\rating.csv")
```

In [5]:
```
movies.head()
```

Out[5]:

	movieId	title	genres
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
1	2	Jumanji (1995)	Adventure|Children|Fantasy
2	3	Grumpier Old Men (1995)	Comedy|Romance
3	4	Waiting to Exhale (1995)	Comedy|Drama|Romance
4	5	Father of the Bride Part II (1995)	Comedy

In [6]:
```
tags.head()
```

Out[6]:

	userId	movieId	tag	timestamp
0	18	4141	Mark Waters	2009-04-24 18:19:40
1	65	208	dark hero	2013-05-10 01:41:18
2	65	353	dark hero	2013-05-10 01:41:19
3	65	521	noir thriller	2013-05-10 01:39:43
4	65	592	dark hero	2013-05-10 01:41:18

In [7]:
```
ratings.head()
```

Out[7]:

	userId	movieId	rating	timestamp
0	1	2	3.5	2005-04-02 23:53:47
1	1	29	3.5	2005-04-02 23:31:16
2	1	32	3.5	2005-04-02 23:33:39
3	1	47	3.5	2005-04-02 23:32:07
4	1	50	3.5	2005-04-02 23:29:40

In [8]:
```
movies.head(20)
```

Out[8]:

	movieId	title	genres
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
1	2	Jumanji (1995)	Adventure|Children|Fantasy
2	3	Grumpier Old Men (1995)	Comedy|Romance
3	4	Waiting to Exhale (1995)	Comedy|Drama|Romance
4	5	Father of the Bride Part II (1995)	Comedy
5	6	Heat (1995)	Action|Crime|Thriller
6	7	Sabrina (1995)	Comedy|Romance
7	8	Tom and Huck (1995)	Adventure|Children
8	9	Sudden Death (1995)	Action
9	10	GoldenEye (1995)	Action|Adventure|Thriller
10	11	American President, The (1995)	Comedy|Drama|Romance
11	12	Dracula: Dead and Loving It (1995)	Comedy|Horror
12	13	Balto (1995)	Adventure|Animation|Children
13	14	Nixon (1995)	Drama
14	15	Cutthroat Island (1995)	Action|Adventure|Romance
15	16	Casino (1995)	Crime|Drama
16	17	Sense and Sensibility (1995)	Drama|Romance
17	18	Four Rooms (1995)	Comedy
18	19	Ace Ventura: When Nature Calls (1995)	Comedy
19	20	Money Train (1995)	Action|Comedy|Crime|Drama|Thriller

In [9]:
```
del ratings["timestamp"]
```

In [10]:
```
ratings.head(1)
```

Out[10]:

	userId	movieId	rating
0	1	2	3.5

In [11]:
```
del tags["timestamp"]
```

In [12]:
```
tags.head(1)
```

Out[12]:

	userId	movieId	tag
0	18	4141	Mark Waters

In [13]:
```
tags.iloc[0]
```

Out[13]:
```
userId              18
movieId           4141
tag        Mark Waters
Name: 0, dtype: object
```

In [14]:
```
type(tags.iloc[0])
```

Out[14]:
```
pandas.core.series.Series
```

In [15]:
```
row_0 = tags.iloc[0]
```

In [16]:
```
row_0
```

Out[16]:
```
userId              18
movieId           4141
tag        Mark Waters
Name: 0, dtype: object
```

In [17]:
```
row_0.index
```

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

In [18]:
```
row_0.userId
```

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

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

Out[19]:
```
4141
```

In [20]:
```
"rating" in row_0
```

Out[20]:
```
False
```

In [21]:
```
"movieId" in row_0
```

Out[21]:
```
True
```

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

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

In [23]:
```
row_0 = row_0.rename("First Row")
```

In [24]:
```
row_0
```

Out[24]:
```
userId             18
movieId          4141
tag        Mark Waters
Name: First Row, dtype: object
```

In [25]:
```
row_0.dtype
```

Out[25]:
```
dtype('O')
```

In [26]:
```
row_0.name
```

Out[26]:
```
'First Row'
```

Data Frames

In [27]:
```
tags.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 [28]:
```
tags.index
```

Out[28]:
```
RangeIndex(start=0, stop=465564, step=1)
```

In [29]:
```
movies.index
```

Out[29]:
```
RangeIndex(start=0, stop=27278, step=1)
```

In [30]:
```
ratings.index
```

Out[30]:
```
RangeIndex(start=0, stop=20000263, step=1)
```

In [31]:
```
tags.columns
```

Out[31]:
```
Index(['userId', 'movieId', 'tag'], dtype='object')
```

In [32]:
```
ratings.columns
```

Out[32]:
```
Index(['userId', 'movieId', 'rating'], dtype='object')
```

In [33]:
```
movies.columns
```

Out[33]:
```
Index(['movieId', 'title', 'genres'], dtype='object')
```

In [34]:
```
movies.head(10)
```

Out[34]:

	movieId	title	genres
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
1	2	Jumanji (1995)	Adventure|Children|Fantasy
2	3	Grumpier Old Men (1995)	Comedy|Romance
3	4	Waiting to Exhale (1995)	Comedy|Drama|Romance
4	5	Father of the Bride Part II (1995)	Comedy
5	6	Heat (1995)	Action|Crime|Thriller
6	7	Sabrina (1995)	Comedy|Romance
7	8	Tom and Huck (1995)	Adventure|Children
8	9	Sudden Death (1995)	Action
9	10	GoldenEye (1995)	Action|Adventure|Thriller

In [35]:
```
movies[:2]
```

Out[35]:

	movieId	title	genres
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
1	2	Jumanji (1995)	Adventure|Children|Fantasy

In [36]:
```
movies.iloc[2,1]
```

Out[36]:
```
'Grumpier Old Men (1995)'
```

In [37]:
```
movies.iloc[:2] == movies[:2]
```

Out[37]:

	movieId	title	genres
0	True	True	True
1	True	True	True

In [38]:
```
movies.iloc[-1,1]
```

Out[38]:
```
'Innocence (2014)'
```

In [39]:
```
movies.iloc[::-1].head()
```

Out[39]:

	movieId	title	genres
27277	131262	Innocence (2014)	Adventure|Fantasy|Horror
27276	131260	Rentun Ruusu (2001)	(no genres listed)
27275	131258	The Pirates (2014)	Adventure
27274	131256	Feuer, Eis & Dosenbier (2002)	Comedy
27273	131254	Kein Bund für's Leben (2007)	Comedy

In [40]:
```
movies.iloc[1:5]
```

Out[40]:

	movieId	title	genres
1	2	Jumanji (1995)	Adventure|Children|Fantasy
2	3	Grumpier Old Men (1995)	Comedy|Romance
3	4	Waiting to Exhale (1995)	Comedy|Drama|Romance
4	5	Father of the Bride Part II (1995)	Comedy

In [41]:
```
movies.sample()
```

Out[41]:

	movieId	title	genres
27151	130628	Boys (2014)	Drama

In [42]:
```
movies.sample(4)
```

Out[42]:

	movieId	title	genres
7796	8399	We're Not Married! (1952)	Comedy|Romance
19586	96960	Tobor the Great (1954)	Children|Sci-Fi
14774	73879	Motocrossed (2001)	Action|Children|Comedy|Romance
18869	93894	Unknown Woman, The (Tuntematon emäntä) (2011)	Documentary

In [43]:
```
condition = movies.loc[movies["genres"] == "Drama"]
```

In [44]:
```
condition.head()
```

Out[44]:

	movieId	title	genres
13	14	Nixon (1995)	Drama
25	26	Othello (1995)	Drama
30	31	Dangerous Minds (1995)	Drama
39	40	Cry, the Beloved Country (1995)	Drama
42	43	Restoration (1995)	Drama

In [45]:
```
condition.iloc[4,1]
```

Out[45]:
```
'Restoration (1995)'
```

In [46]:
```
condition2 = movies.loc[movies["genres"]=="Action"]
```

In [47]:
```
condition2
```

Out[47]:

	movieId	title	genres
8	9	Sudden Death (1995)	Action
70	71	Fair Game (1995)	Action
202	204	Under Siege 2: Dark Territory (1995)	Action
248	251	Hunted, The (1995)	Action
659	667	Bloodsport 2 (a.k.a. Bloodsport II: The Next K...	Action
...	...	...	...
26876	129239	The Monkey Hustle (1976)	Action
26899	129346	The Package (2012)	Action
26965	129657	Tracers (2015)	Action
27139	130526	The Detective 2 (2011)	Action
27198	131025	The Brass Legend (1956)	Action

178 rows × 3 columns

In [48]:
```
ratings
```

Out[48]:

	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 [49]:
```
cond = ratings[ratings ["rating"]<3]
```

In [50]:
```
cond.count()
```

Out[50]:
```
userId     3513504
movieId    3513504
rating     3513504
dtype: int64
```

In [51]:
```
cond.describe()
```

Out[51]:

	userId	movieId	rating
count	3.513504e+06	3.513504e+06	3.513504e+06
mean	6.893548e+04	9.167238e+03	1.790139e+00
std	4.005873e+04	1.968869e+04	6.209392e-01
min	2.000000e+00	1.000000e+00	5.000000e-01
25%	3.410300e+04	1.037000e+03	1.000000e+00
50%	6.913300e+04	2.485000e+03	2.000000e+00
75%	1.034920e+05	4.919000e+03	2.500000e+00
max	1.384930e+05	1.312580e+05	2.500000e+00

In [52]:
```
cond.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 3513504 entries, 186 to 20000262
Data columns (total 3 columns):
 #   Column   Dtype  
---  ------   -----  
 0   userId   int64  
 1   movieId  int64  
 2   rating   float64
dtypes: float64(1), int64(2)
memory usage: 107.2 MB
```

In [53]:
```
movies.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 27278 entries, 0 to 27277
Data columns (total 3 columns):
 #   Column   Non-Null Count  Dtype 
---  ------   --------------  ----- 
 0   movieId  27278 non-null  int64 
 1   title    27278 non-null  object
 2   genres   27278 non-null  object
dtypes: int64(1), object(2)
memory usage: 639.5+ KB
```

In [54]:
```
train = movies.sample(10)
genre = input("Enter the genre you wish to search: ")
condi =train[train["genres"]==f"{genre}"]
condi
```

```
Enter the genre you wish to search: Comedy
```

Out[54]:

	movieId	title	genres
18042	90432	Lentsu (1990)	Comedy
3682	3773	House Party (1990)	Comedy
26826	128912	Mike Birbiglia: My Girlfriend's Boyfriend (2013)	Comedy

In [55]:
```
cond_genre = input("Enter the genre you wish to see: ")
condition_genre = movies[movies["genres"]==f"{cond_genre}"]
condition_genre
```

```
Enter the genre you wish to see: Action
```

Out[55]:

	movieId	title	genres
8	9	Sudden Death (1995)	Action
70	71	Fair Game (1995)	Action
202	204	Under Siege 2: Dark Territory (1995)	Action
248	251	Hunted, The (1995)	Action
659	667	Bloodsport 2 (a.k.a. Bloodsport II: The Next K...	Action
...	...	...	...
26876	129239	The Monkey Hustle (1976)	Action
26899	129346	The Package (2012)	Action
26965	129657	Tracers (2015)	Action
27139	130526	The Detective 2 (2011)	Action
27198	131025	The Brass Legend (1956)	Action

178 rows × 3 columns

In [56]:
```
tags.iloc[[0,11,100]]
```

Out[56]:

	userId	movieId	tag
0	18	4141	Mark Waters
11	65	1783	noir thriller
100	121	52973	drugs

In [57]:
```
ratings["rating"].describe()
```

Out[57]:
```
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 [58]:
```
ratings.describe()
```

Out[58]:

	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 [59]:
```
ratings["rating"].mean()
```

Out[59]:
```
3.5255285642993797
```

In [60]:
```
ratings["rating"].min()
```

Out[60]:
```
0.5
```

In [61]:
```
ratings["rating"].max()
```

Out[61]:
```
5.0
```

In [62]:
```
ratings["rating"].std()
```

Out[62]:
```
1.051988919275684
```

In [63]:
```
ratings["rating"].mode()
```

Out[63]:
```
0    4.0
Name: rating, dtype: float64
```

In [64]:
```
tags["movieId"].mode()
```

Out[64]:
```
0    296
Name: movieId, dtype: int64
```

In [65]:
```
ratings.corr(method="spearman")#need to ask
```

Out[65]:

	userId	movieId	rating
userId	1.000000	-0.002742	0.001302
movieId	-0.002742	1.000000	-0.022152
rating	0.001302	-0.022152	1.000000

In [68]:
```
ratings.corr(method="kendall")
```

```
C:\Users\chait\anaconda3\lib\site-packages\scipy\stats\stats.py:4812: RuntimeWarning: overflow encountered i
n longlong_scalars
  (2 * xtie * ytie) / m + x0 * y0 / (9 * m * (size - 2)))
```

Out[68]:

	userId	movieId	rating
userId	1.000000	-0.001829	0.000937
movieId	-0.001829	1.000000	-0.015970
rating	0.000937	-0.015970	1.000000

In [69]:
```
movies.isnull()
```

Out[69]:

	movieId	title	genres
0	False	False	False
1	False	False	False
2	False	False	False
3	False	False	False
4	False	False	False
...	...	...	...
27273	False	False	False
27274	False	False	False
27275	False	False	False
27276	False	False	False
27277	False	False	False

27278 rows × 3 columns

In [70]:
```
movies.isnull().any()
```

Out[70]:
```
movieId    False
title      False
genres     False
dtype: bool
```

In [71]:
```
movies.isnull().any().any()
```

Out[71]:
```
False
```

In [72]:
```
tags.isnull().any().any()
```

Out[72]:
```
True
```

In [73]:
```
ratings.isnull().any().any()
```

Out[73]:
```
False
```

In [74]:
```
tags[tags.isnull().any(axis=1)].index
```

Out[74]:
```
Int64Index([373276, 373277, 373281, 373288, 373289, 373291, 373299, 373301,
            373303, 373319, 373325, 373332, 373334, 373339, 373340, 454615],
           dtype='int64')
```

In [76]:
```
tags.iloc[373277]
```

Out[76]:
```
userId     116460
movieId       346
tag           NaN
Name: 373277, dtype: object
```

In [77]:
```
tags.dropna(axis='columns')
```

Out[77]:

	userId	movieId
0	18	4141
1	65	208
2	65	353
3	65	521
4	65	592
...	...	...
465559	138446	55999
465560	138446	55999
465561	138446	55999
465562	138446	55999
465563	138472	923

465564 rows × 2 columns

In [78]:
```
tags.isnull().any().any()
```

Out[78]:
```
True
```

In [79]:
```
tags[tags.isnull().any(axis=1)].index
```

Out[79]:
```
Int64Index([373276, 373277, 373281, 373288, 373289, 373291, 373299, 373301,
            373303, 373319, 373325, 373332, 373334, 373339, 373340, 454615],
           dtype='int64')
```

In [80]:
```
tags.dropna(axis='rows')
```

Out[80]:

	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 [81]:
```
tags.isnull().any().any()
```

Out[81]:
```
True
```

In [82]:
```
tags.dropna()
```

Out[82]:

	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 [83]:
```
tags.isnull().any().any()
```

Out[83]:
```
True
```

In [86]:
```
tags[tags.isnull().any(axis=1)].index
```

Out[86]:
```
Int64Index([373276, 373277, 373281, 373288, 373289, 373291, 373299, 373301,
            373303, 373319, 373325, 373332, 373334, 373339, 373340, 454615],
           dtype='int64')
```

In [87]:
```
tags.shape
```

Out[87]:
```
(465564, 3)
```

In [88]:
```
tags.index
```

Out[88]:
```
RangeIndex(start=0, stop=465564, step=1)
```

In [89]:
```
tags.shape
```

Out[89]:
```
(465564, 3)
```

In [90]:
```
%matplotlib inline
```

In [91]:
```
import matplotlib.pyplot as plt
```

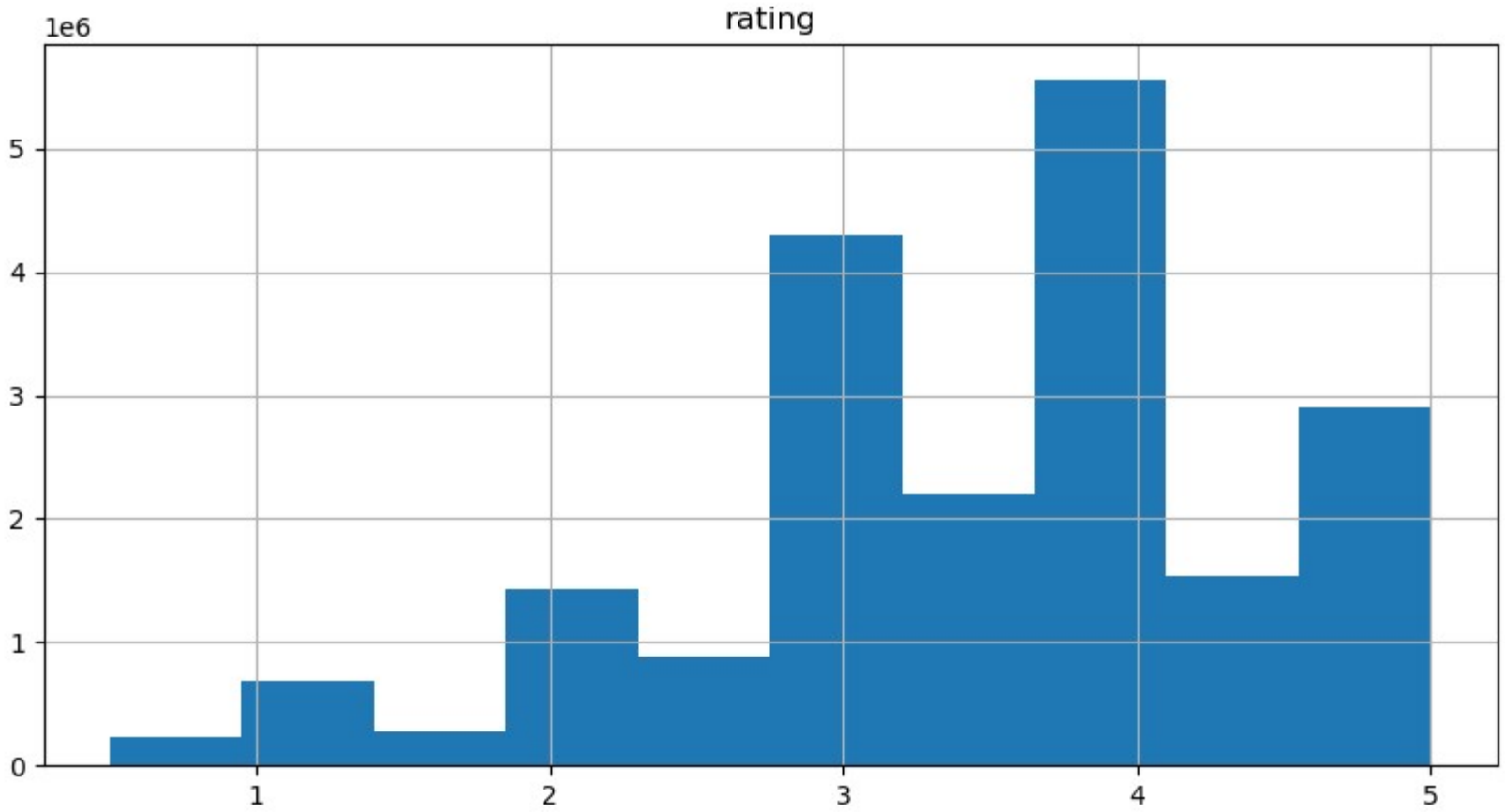
In [92]:
```
ratings.head()
```

Out[92]:

	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 [84]:
```
ratings.hist(column="rating", figsize=(10,5))
```

Out[84]:
```
array([[<AxesSubplot:title={'center':'rating'}>]], dtype=object)
```



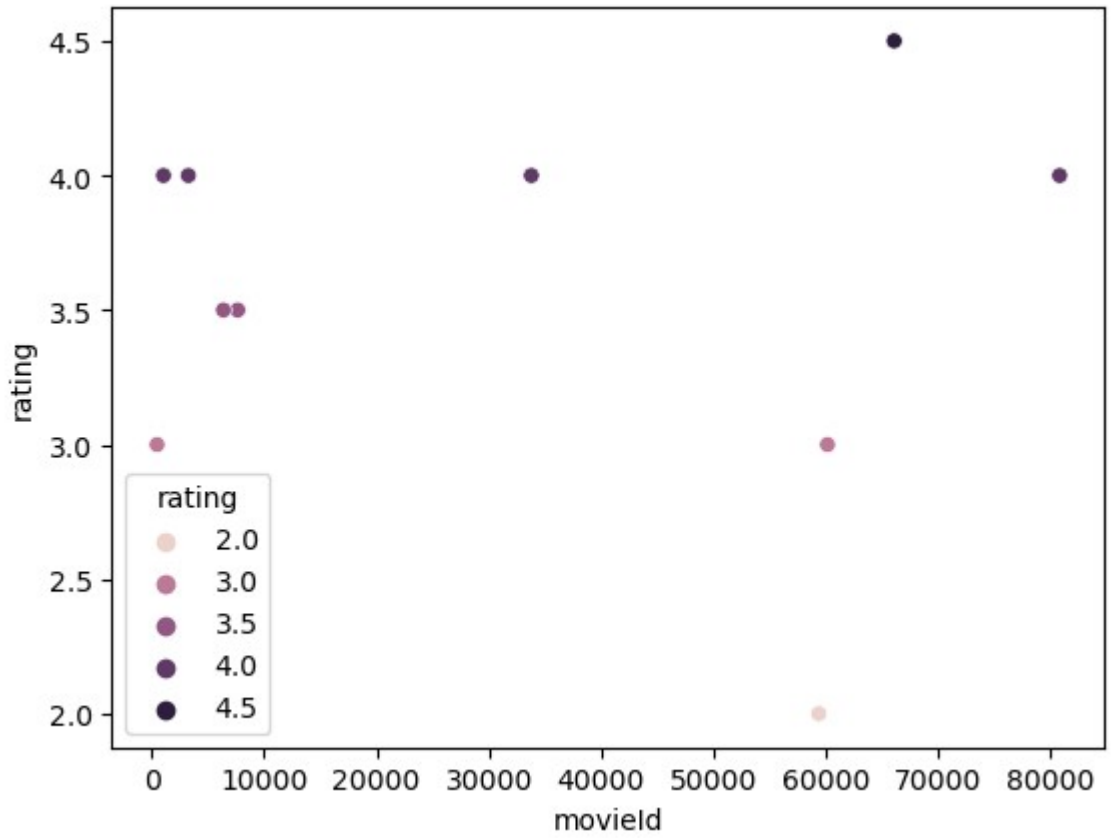
In [93]:
```
import seaborn as sns
```

In [120]:
```
sns.lmplot(data = ratings.sample(10), x="movieId", y="rating", hue="rating", fit_reg=False)
```

Out[120]:
```
<seaborn.axisgrid.FacetGrid at 0x182820ed670>
```

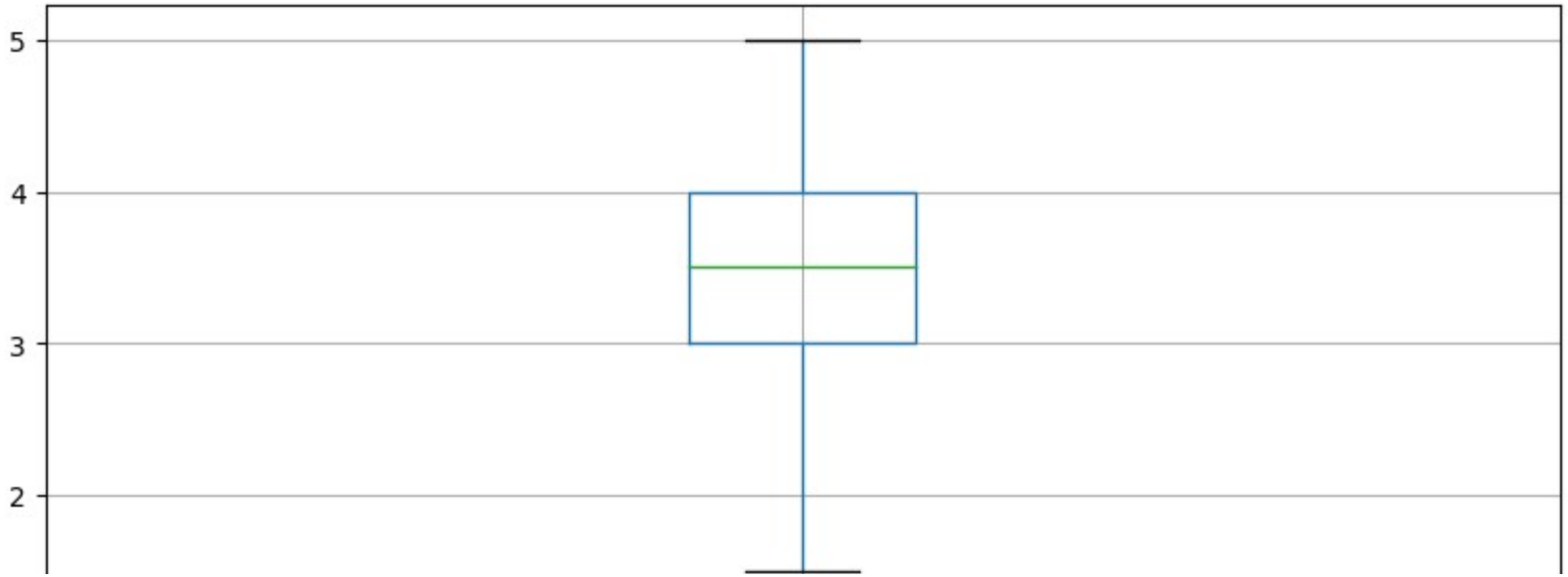
In [122]:
```
sns.scatterplot(data = ratings.sample(10), x="movieId", y="rating", hue="rating")
```

Out[122]:
```
<AxesSubplot:xlabel='movieId', ylabel='rating'>
```



In [107]:
```
ratings.boxplot(column="rating", figsize = (10,5))
```

Out[107]: <AxesSubplot:>



In [95]:
```
movies
```

Out[95]:

	movieId	title	genres
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
1	2	Jumanji (1995)	Adventure|Children|Fantasy
2	3	Grumpier Old Men (1995)	Comedy|Romance
3	4	Waiting to Exhale (1995)	Comedy|Drama|Romance
4	5	Father of the Bride Part II (1995)	Comedy
...	...	...	...
27273	131254	Kein Bund für's Leben (2007)	Comedy
27274	131256	Feuer, Eis & Dosenbier (2002)	Comedy
27275	131258	The Pirates (2014)	Adventure
27276	131260	Rentun Ruusu (2001)	(no genres listed)
27277	131262	Innocence (2014)	Adventure|Fantasy|Horror

27278 rows × 3 columns

In [96]:
```
movies[["title","genres"]]
```

Out[96]:

	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
...	...	...
27273	Kein Bund für's Leben (2007)	Comedy
27274	Feuer, Eis & Dosenbier (2002)	Comedy
27275	The Pirates (2014)	Adventure
27276	Rentun Ruusu (2001)	(no genres listed)
27277	Innocence (2014)	Adventure|Fantasy|Horror

27278 rows × 2 columns

In [97]:
```
ratings
```

Out[97]:

	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 [99]:
```
ratings[["movieId","rating"]].head(10)
```

Out[99]:

	movieId	rating
0	2	3.5
1	29	3.5
2	32	3.5
3	47	3.5
4	50	3.5
5	112	3.5
6	151	4.0
7	223	4.0
8	253	4.0
9	260	4.0

In [100]:
```
movies
```

Out[100]:

	movieId	title	genres
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
1	2	Jumanji (1995)	Adventure|Children|Fantasy
2	3	Grumpier Old Men (1995)	Comedy|Romance
3	4	Waiting to Exhale (1995)	Comedy|Drama|Romance
4	5	Father of the Bride Part II (1995)	Comedy
...	...	...	...
27273	131254	Kein Bund für's Leben (2007)	Comedy
27274	131256	Feuer, Eis & Dosenbier (2002)	Comedy
27275	131258	The Pirates (2014)	Adventure
27276	131260	Rentun Ruusu (2001)	(no genres listed)
27277	131262	Innocence (2014)	Adventure|Fantasy|Horror

27278 rows × 3 columns

In [101]:
```
movies[["title","genres"]].sample(10)
```

Out[101]:

	title	genres
26879	Loaf and Camouflage (1984)	Comedy
20885	Band Called Death, A (2012)	Documentary
21957	Alien Space Avenger (1989)	Comedy|Horror|Sci-Fi
23915	Osaka Elegy (Naniwa erejî) (1936)	Drama
5946	Bullet (1996)	Action|Crime|Drama
2404	Spanish Fly (1998)	Drama
16566	Southern District (Zona Sur) (2009)	Drama
24896	Santa Who? (2000)	Comedy|Fantasy
8248	Born Rich (2003)	Documentary
13454	AmericanEast (2008)	Drama

In [102]:
```
tags
```

Out[102]:

	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

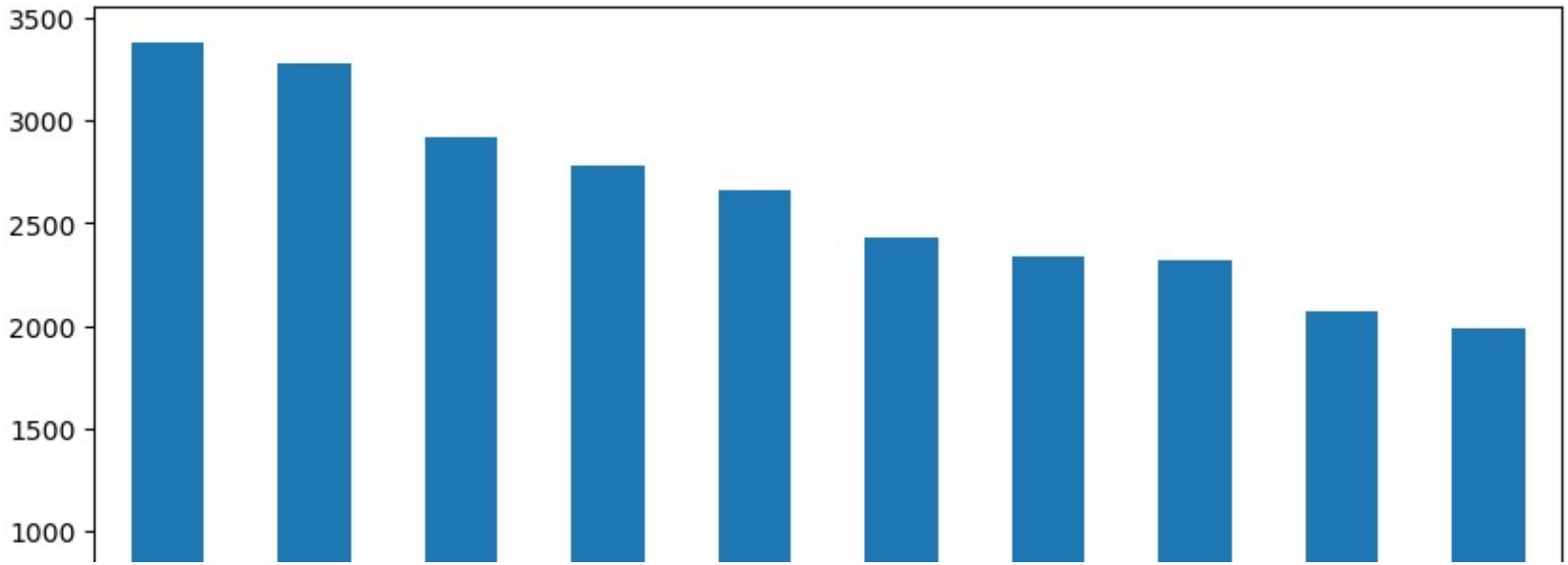
In [103]:
```
tag_count=tags["tag"].value_counts()
```

In [104]:
```
tag_count.head(10)
```

Out[104]:
```
sci-fi             3384
based on a book    3281
atmospheric        2917
comedy             2779
action             2657
surreal            2427
BD-R               2334
twist ending       2323
funny              2072
dystopia           1991
Name: tag, dtype: int64
```

In [105]:
```
tag_count.head(10).plot(kind="bar",figsize=(10,5))
```

Out[105]: <AxesSubplot:>



In [*]:
```
tag_count[:-10].plot(kind="bar",figsize=(10,5))
```

Out[123]: <AxesSubplot:>

```
C:\Users\chait\anaconda3\lib\site-packages\IPython\core\pylabtools.py:152: UserWarning: Glyph 131 (\x83) mis
sing from current font.
  fig.canvas.print_figure(bytes_io, **kw)
C:\Users\chait\anaconda3\lib\site-packages\IPython\core\pylabtools.py:152: UserWarning: Glyph 9 (       ) mi
ssing from current font.
  fig.canvas.print_figure(bytes_io, **kw)
C:\Users\chait\anaconda3\lib\site-packages\IPython\core\pylabtools.py:152: UserWarning: Glyph 143 (\x8f) mis
sing from current font.
  fig.canvas.print_figure(bytes_io, **kw)
C:\Users\chait\anaconda3\lib\site-packages\IPython\core\pylabtools.py:152: UserWarning: Glyph 130 (\x82) mis
sing from current font.
  fig.canvas.print_figure(bytes_io, **kw)
C:\Users\chait\anaconda3\lib\site-packages\IPython\core\pylabtools.py:152: UserWarning: Glyph 129 (\x81) mis
sing from current font.
  fig.canvas.print_figure(bytes_io, **kw)
```