In [1]:

```
import pandas as pd
```

In [5]:

```
movies = pd.read_csv(r'C:\Users\Rachana Jena\Downloads\movies\movie.csv')
movies
```

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
...	...	...	...
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 [6]:

```
tags = pd.read_csv(r'C:\Users\Rachana Jena\Downloads\movies\tag.csv')
tags
```

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
...	...	...	...	...
465559	138446	55999	dragged	2013-01-23 23:29:32
465560	138446	55999	Jason Bateman	2013-01-23 23:29:38
465561	138446	55999	quirky	2013-01-23 23:29:38
465562	138446	55999	sad	2013-01-23 23:29:32
465563	138472	923	rise to power	2007-11-02 21:12:47

465564 rows × 4 columns

In [ ]:

```
ratings = pd.read_csv(r'C:\Users\Rachana Jena\Downloads\movies\rating.csv')
ratings
```

In [11]:

```
ratings.head(2)
```

Out[11]:

	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

In [12]:

```
del ratings['timestamp']
del tags['timestamp']
```

Data Structures:

- series

In [13]:

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

Out[13]:

```
pandas.core.series.Series
```

In [14]:

```
print(row_0)
```

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

In [15]:
```
row_0.index
```

Out[15]:
```
Index(['userId', 'movieId', 'tag'], dtype='object')
```

In [19]:
```
row_0['userId']
```

Out[19]:
```
18
```

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

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

In [21]:
```
row_0.name
```

Out[21]:
```
0
```

In [22]:
```
row_0 = row_0.rename('firstRow')
row_0.name
```

Out[22]:
```
'firstRow'
```

DataFrames

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]:
```
tags.index
```

Out[24]:
```
RangeIndex(start=0, stop=465564, step=1)
```

In [25]:
```
tags.columns
```

Out[25]:
```
Index(['userId', 'movieId', 'tag'], dtype='object')
```

In [26]:
```
tags.iloc[ [0,11,500] ]
```

Out[26]:

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

Descriptive Statistics

In [27]:
```
ratings['rating'].describe()
```

Out[27]:
```
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 [28]:
```
ratings.describe()
```

Out[28]:

	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 [29]:
```
ratings['rating'].mean()
```

Out[29]:
```
3.5255285642993797
```

In [30]:
```
ratings.mean()
```

Out[30]:
```
userId     69045.872583
movieId     9041.567330
rating         3.525529
dtype: float64
```

In [31]:
```
ratings['rating'].min()
```

Out[31]:
```
0.5
```

In [32]:
```
ratings['rating'].max()
```

Out[32]:
```
5.0
```

In [34]:
```
ratings['rating'].std()
```

Out[34]:
```
1.051988919275684
```

In [35]:
```
ratings['rating'].mode
```

Out[35]:
```
<bound method Series.mode of 0           3.5
1           3.5
2           3.5
3           3.5
4           3.5
           ...
20000258    4.5
20000259    4.5
20000260    3.0
20000261    5.0
20000262    2.5
Name: rating, Length: 20000263, dtype: float64>
```

In [36]:
```
ratings.corr()
```

Out[36]:

	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 [38]:
```
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[38]:
```
False
```

In [39]:
```
filter2 = ratings['rating'] > 0
filter2.all()
```

Out[39]:
```
True
```

Data Cleaning:Handling Missing Data

In [40]:
```
movies.shape
```

Out[40]:
```
(27278, 3)
```

In [41]:
```
movies.isnull().any().any()
```

Out[41]:
```
False
```

- Thats nice ! No Null values

In [42]:
```
ratings.shape
```

Out[42]:
```
(20000263, 3)
```

In [43]:
```
ratings.isnull().any().any()
```

Out[43]:
```
False
```

In [44]:
```
tags.shape
```

Out[44]:
```
(465564, 3)
```

In [45]:
```
tags.isnull().any().any()
```

Out[45]:
```
True
```

- We have some tags which are NULL.

In [47]:
```
tags=tags.dropna()
```

In [48]:
```
tags.isnull().any().any()
```

Out[48]:
```
False
```

In [49]:
```
tags.shape
```

Out[49]:
```
(465548, 3)
```

Data Visualization

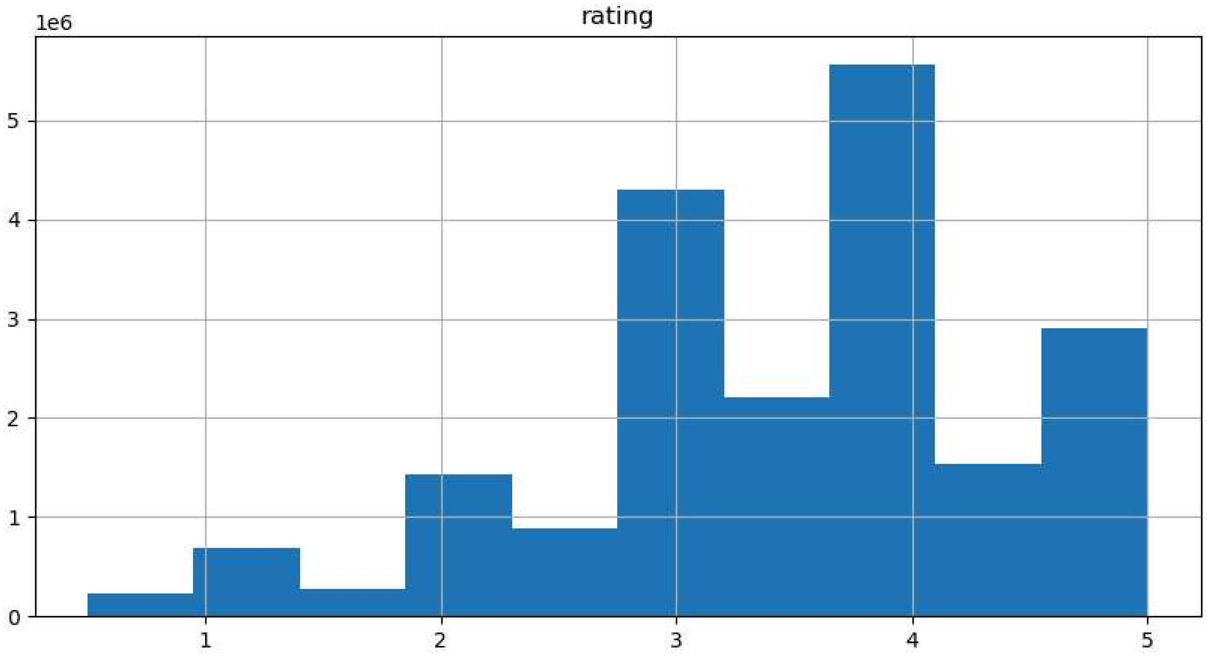
In [50]:

```
%matplotlib inline

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

Out[50]:

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

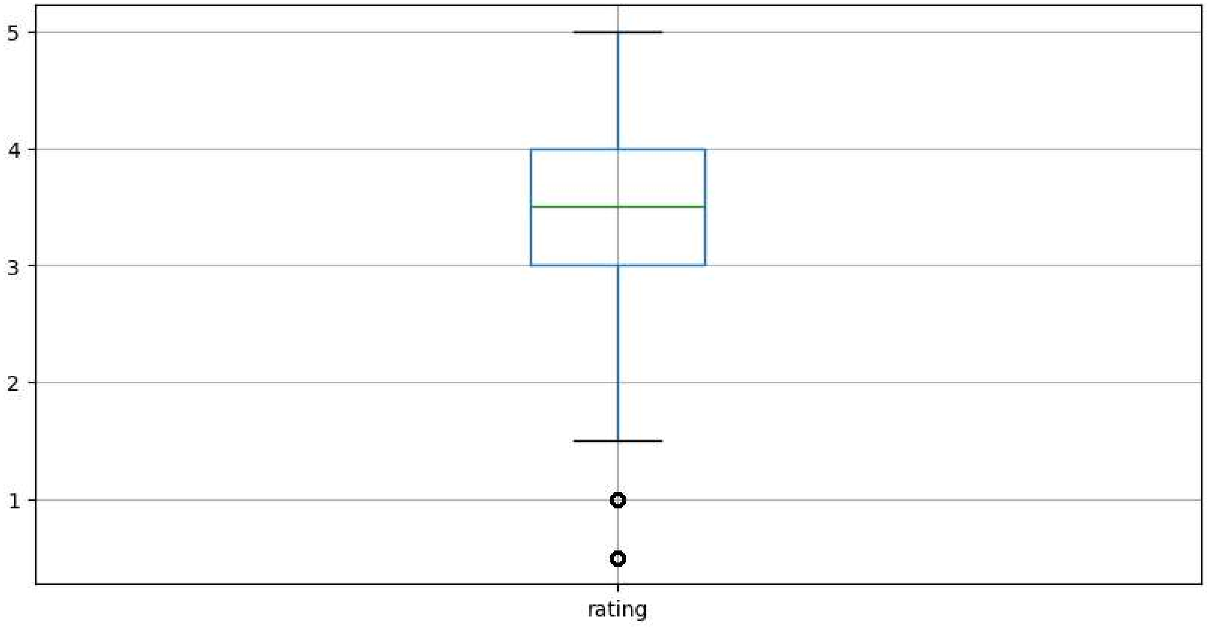


In [54]:

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

Out[54]:

```
<Axes: >
```



Slicing Out Columns

In [55]:

```
tags['tag'].head()
```

Out[55]:

```
0      Mark Waters
1        dark hero
2        dark hero
3    noir thriller
4        dark hero
Name: tag, dtype: object
```

In [56]:

```
movies[['title','genres']].head()
```

Out[56]:

	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 [57]:

```
ratings[-10:]
```

Out[57]:

	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

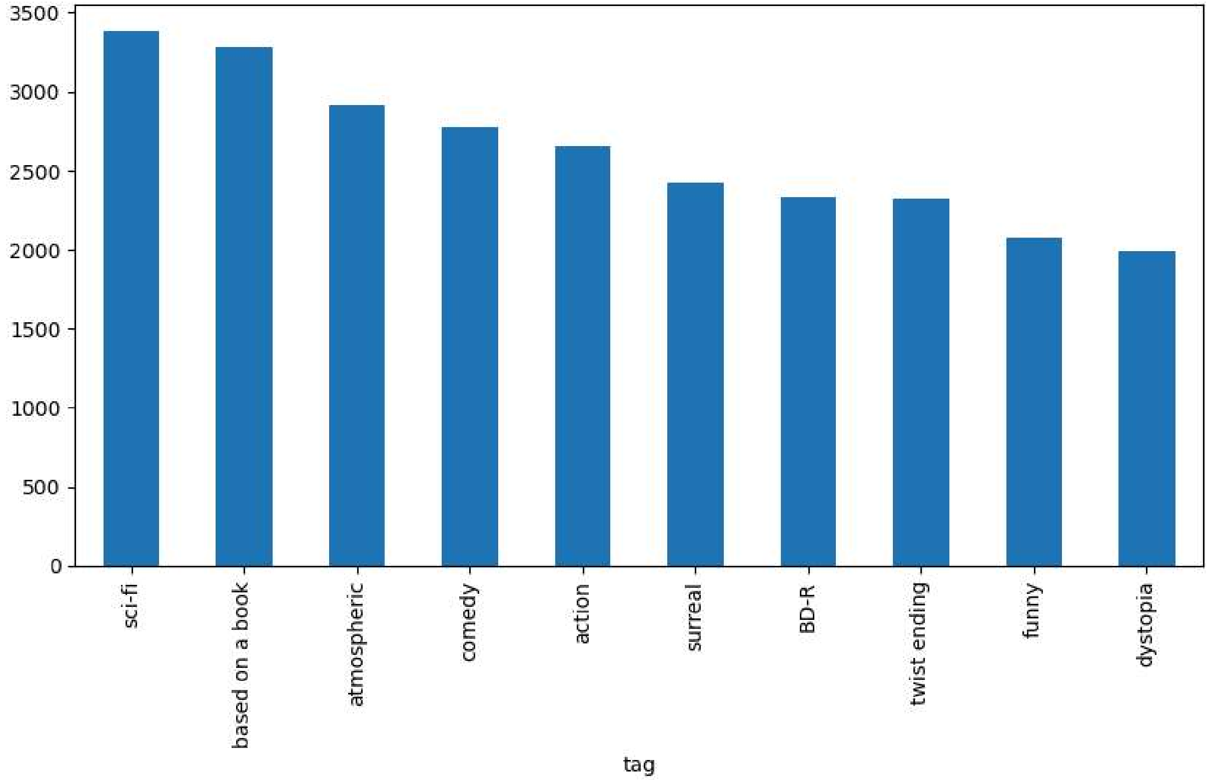
In [58]:

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

```
Out[58]: tag
         missing child                    1
         Ron Moore                        1
         Citizen Kane                     1
         mullet                           1
         biker gang                       1
         Paul Adelstein                   1
         the wig                          1
         killer fish                      1
         genetically modified monsters    1
         topless scene                    1
         Name: count, dtype: int64
```

```
In [60]: tag_counts[:10].plot(kind='bar', figsize=(10,5))
```

```
Out[60]: <Axes: xlabel='tag'>
```



Filters for Selecting Rows

```
In [ ]:
```