In [41]:
```
import pandas as pd
```

In [42]:
```
ratings = pd.read_csv(r'C:\Users\lenovo\Desktop\Kaggle project\rating.csv')
```

In [43]:
```
movies = pd.read_csv(r'C:\Users\lenovo\Desktop\Kaggle project\movie.csv')
```

In [44]:
```
tags = pd.read_csv(r'C:\Users\lenovo\Desktop\Kaggle project\tag.csv')
```

In [45]:
```
print(movies.shape)
print(ratings.shape)
print(tags.shape)
```

```
(27278, 3)
(1048575, 4)
(465564, 4)
```

In [46]:
```
print(movies.columns)
print(ratings.columns)
print(tags.columns)
```

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

In [47]:
```
#In output as we can see- movieId(column name/attribute) in 3 of them. So all 3
#From movies excel(dataset) - movieId is considered as a "Foreign key".
#In other Rating dataset(excel) it is called as "Primary key".
#In other Tag dataset(excel) it is called as "Secondary key".
```

In [48]:
```
del ratings['timestamp']
del tags['timestamp']
```

In [49]:
```
print(movies.columns)
print(ratings.columns)
print(tags.columns)
```

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

In [50]:
```
tags.head(2)
```

Out[50]:

	userId	movieId	tag
0	18	4141	Mark Waters
1	65	208	dark hero

In [51]:
```
tags.iloc[0] #iloc gives us index locations, we use iloc in ML
```

Out[51]:
```
userId                18
movieId             4141
tag          Mark Waters
Name: 0, dtype: object
```

In [52]:
```
tags.iloc[1]
```

```
Out[52]:  userId              65
          movieId            208
          tag          dark hero
          Name: 1, dtype: object
```

```
In [53]:  row_0 = tags.iloc[0]
```

```
In [54]:  print(row_0)
```

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

```
In [55]:  row_0.index
```

```
Out[55]:  Index(['userId', 'movieId', 'tag'], dtype='object')
```

```
In [56]:  row_0['userId']
```

```
Out[56]:  18
```

```
In [57]:  'rating' in row_0
```

```
Out[57]:  False
```

```
In [58]:  row_0.name
```

```
Out[58]:  0
```

```
In [59]:  row_0 = row_0.rename('firstRow')
          row_0.name
```

```
Out[59]:  'firstRow'
```

```
In [60]:  tags.head()
```

Out[60]:

	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 [61]:  tags.index
```

```
Out[61]:  RangeIndex(start=0, stop=465564, step=1)
```

```
In [62]:  tags.columns
```

```
Out[62]:  Index(['userId', 'movieId', 'tag'], dtype='object')
```

In [63]:

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

Out[63]:

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

In [64]:

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

Out[64]:

```
count    1.048575e+06
mean     3.529272e+00
std      1.051919e+00
min      5.000000e-01
25%      3.000000e+00
50%      4.000000e+00
75%      4.000000e+00
max      5.000000e+00
Name: rating, dtype: float64
```

In [65]:

```
ratings.describe()
```

Out[65]:

	userId	movieId	rating
count	1.048575e+06	1.048575e+06	1.048575e+06
mean	3.527086e+03	8.648988e+03	3.529272e+00
std	2.018424e+03	1.910014e+04	1.051919e+00
min	1.000000e+00	1.000000e+00	5.000000e-01
25%	1.813000e+03	9.030000e+02	3.000000e+00
50%	3.540000e+03	2.143000e+03	4.000000e+00
75%	5.233000e+03	4.641000e+03	4.000000e+00
max	7.120000e+03	1.306420e+05	5.000000e+00

In [66]:

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

Out[66]:

```
3.5292716305462175
```

In [67]:

```
ratings.mean()
```

Out[67]:

```
userId     3527.086123
movieId    8648.988281
rating        3.529272
dtype: float64
```

In [68]:

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

Out[68]:

```
0.5
```

In [69]:

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

```
Out[69]:  5.0
```

```
In [70]:  ratings['rating'].std()
```

```
Out[70]:  1.0519187535891295
```

```
In [71]:  ratings['rating'].mode()
```

```
Out[71]:  0    4.0
          Name: rating, dtype: float64
```

```
In [73]:  ratings.corr() #correlation
```

Out[73]:

	userId	movieId	rating
userId	1.000000	-0.002837	0.017105
movieId	-0.002837	1.000000	0.002550
rating	0.017105	0.002550	1.000000

```
In [74]:  filter1 = ratings['rating'] > 10
          print(filter1)
          filter1.any()
```

```
0          False
1          False
2          False
3          False
4          False
           ...
1048570    False
1048571    False
1048572    False
1048573    False
1048574    False
Name: rating, Length: 1048575, dtype: bool
```

```
Out[74]:  False
```

```
In [75]:  filter2 = ratings['rating'] > 0
          filter2.all()
```

```
Out[75]:  True
```

```
In [76]:  movies.shape
```

```
Out[76]:  (27278, 3)
```

```
In [77]:  movies.isnull().any().any()
```

```
Out[77]:  False
```

```
In [78]:  ratings.shape
```

```
Out[78]:  (1048575, 3)
```

```
In [79]:  ratings.isnull().any().any()
```

```
Out[79]:  False
```

```
In [80]:  tags.shape
```

```
Out[80]:  (465564, 3)
```

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

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

```
In [83]:  tags=tags.dropna() #removes missing values from rows and columns.
```

```
In [84]:  tags.isnull().any().any()
```

```
Out[84]:  False
```

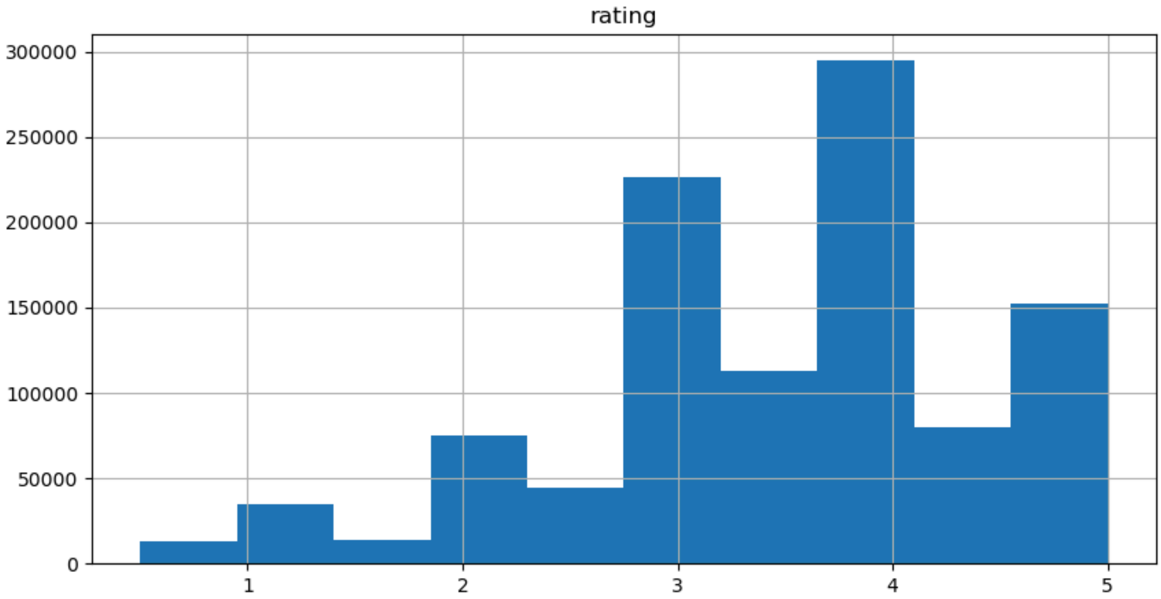
```
In [85]:  tags.shape
```

```
Out[85]:  (465548, 3)
```

```
In [86]:  #Data Visualization
```

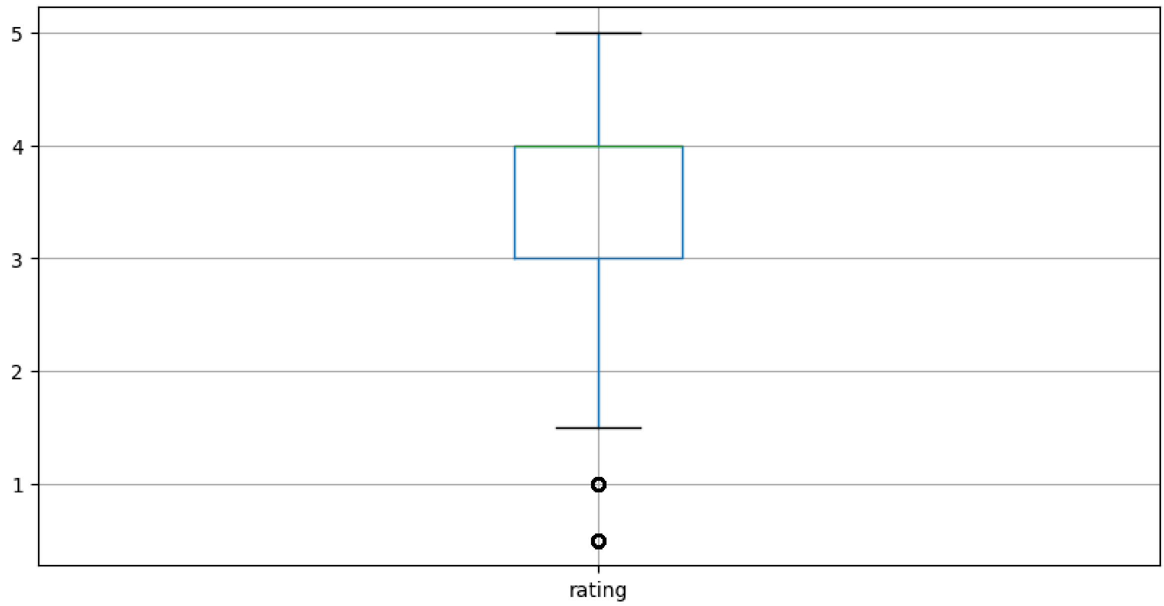
```
In [87]:  %matplotlib inline

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

```
Out[87]:  array([[<Axes: title={'center': 'rating'}>]], dtype=object)
```



```
In [88]:  ratings.boxplot(column='rating', figsize=(10,5))
```

```
Out[88]:  <Axes: >
```



In [90]:
```
#Slicing out column
```

In [91]:
```
tags['tag'].head()
```

Out[91]:
```
0      Mark Waters
1        dark hero
2        dark hero
3    noir thriller
4        dark hero
Name: tag, dtype: object
```

In [92]:
```
movies[['title','genres']].head()
```

Out[92]:

	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 [93]:
```
ratings[-10:]
```

Out[93]:

	userId	movieId	rating
1048565	7120	141	5.0
1048566	7120	151	5.0
1048567	7120	153	0.5
1048568	7120	161	4.0
1048569	7120	163	4.5
1048570	7120	168	5.0
1048571	7120	253	4.0
1048572	7120	260	5.0
1048573	7120	261	4.0
1048574	7120	266	3.5

In [94]:

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

Out[94]:

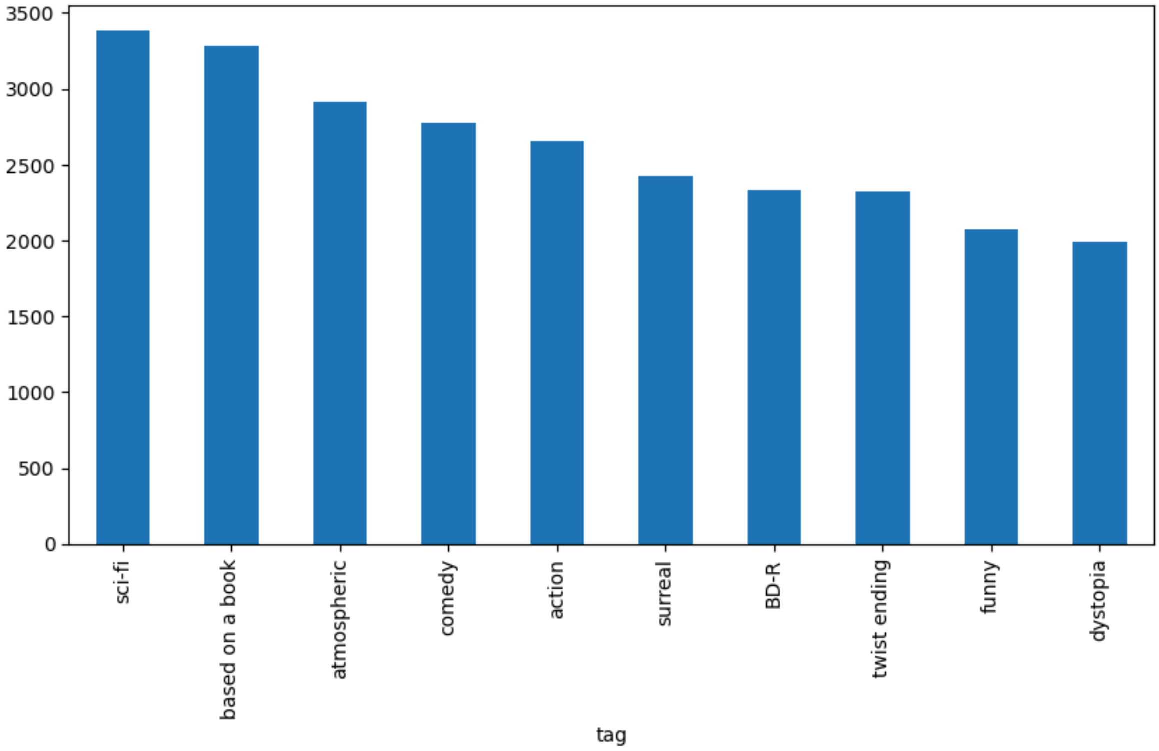
```
tag
missing child                   1
Ron Moore                       1
Citizen Kane                    1
mullet                          1
biker gang                      1
Paul Adelstein                  1
the wig                         1
killer fish                     1
genetically modified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

In [95]:

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

Out[95]:

```
<Axes: xlabel='tag'>
```


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 [ ]:
```