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

In [3]:
```
movies=pd.read_csv(r"C:\Users\siddharth.bose\830 am In Class Docs\archive\movie.
```

In [4]:
```
movies
```

Out[4]:

	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 [11]:
```
ratings=pd.read_csv(r"C:\Users\siddharth.bose\830 am In Class Docs\archive\ratin
ratings
```

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
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
...	...	...	...	...
20000258	138493	68954	4.5	2009-11-13 15:42:00
20000259	138493	69526	4.5	2009-12-03 18:31:48
20000260	138493	69644	3.0	2009-12-07 18:10:57
20000261	138493	70286	5.0	2009-11-13 15:42:24
20000262	138493	71619	2.5	2009-10-17 20:25:36

20000263 rows × 4 columns

In [9]:
```
tags=pd.read_csv(r"C:\Users\siddharth.bose\830 am In Class Docs\archive\tag.csv"
tags
```

Out[9]:

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

Datastructures

Series

In [13]:
```
row_0=tags.iloc[0]
print(row_0)
```

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

In [14]:
```
row_0.index
```

Out[14]:
```
Index(['userId', 'movieId', 'tag'], dtype='object')
```

In [15]:
```
row_0['userId']
```

Out[15]:
```
18
```

In [16]:
```
'rating' in row_0
```

Out[16]:
```
False
```

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

Out[17]:
```
0
```

In [18]:
```
row_0 = row_0.rename('firstRow')
row_0.name
```

Out[18]:
```
'firstRow'
```

DataFrames

In [19]:
```
tags.head()
```

Out[19]:

	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 [20]:
```
tags.index
```

Out[20]:
```
RangeIndex(start=0, stop=465564, step=1)
```

In [21]:
```
tags.columns
```

Out[21]:
```
Index(['userId', 'movieId', 'tag'], dtype='object')
```

In [22]:
```
tags.iloc[[0,11,500]]
```

Out[22]:

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

Descriptive Statistics

In [23]:
```
ratings['rating'].describe()
```

Out[23]:
```
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 [24]:
```
ratings['rating'].mean()
```

Out[24]:
```
3.5255285642993797
```

In [25]:
```
ratings.mean()
```

Out[25]:
```
userId     69045.872583
movieId     9041.567330
rating         3.525529
dtype: float64
```

In [26]:
```
ratings['rating'].min()
```

Out[26]:
```
0.5
```

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

Out[27]:
```
5.0
```

In [28]:
```
ratings['rating'].std()
```

Out[28]:
```
1.051988919275684
```

In [29]:
```
ratings['rating'].mode()
```

Out[29]:
```
0    4.0
Name: rating, dtype: float64
```

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

Out[30]:

	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 [31]:
```
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[31]:
```
False
```

In [32]:
```
filter2 = ratings['rating'] > 0
print(filter2)
filter2.all()
```

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

Out[32]:
```
True
```

Data Cleaning: Handling Missing Data

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

Out[33]:
```
(27278, 3)
```

In [37]:
```
movies.isnull()
```

Out[37]:

	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 [38]:
```
movies.isnull().any()
```
Out[38]:
```
movieId    False
title      False
genres     False
dtype: bool
```

In [39]:
```
movies.isnull().any().any()
```
Out[39]:
```
False
```

In [40]:
```
ratings.shape
```
Out[40]:
```
(20000263, 3)
```

In [41]:
```
ratings.isnull().any().any()
```
Out[41]:
```
False
```

In [42]:
```
tags.shape
```
Out[42]:
```
(465564, 3)
```

In [43]:
```
tags.isnull().any().any()
```
Out[43]:
```
True
```

In [44]:
```
tags.isnull().any()
```
Out[44]:
```
userId     False
movieId    False
tag         True
dtype: bool
```

```
In [45]: tags=tags.dropna()
```

```
In [46]: tags.isnull().any().any()
```

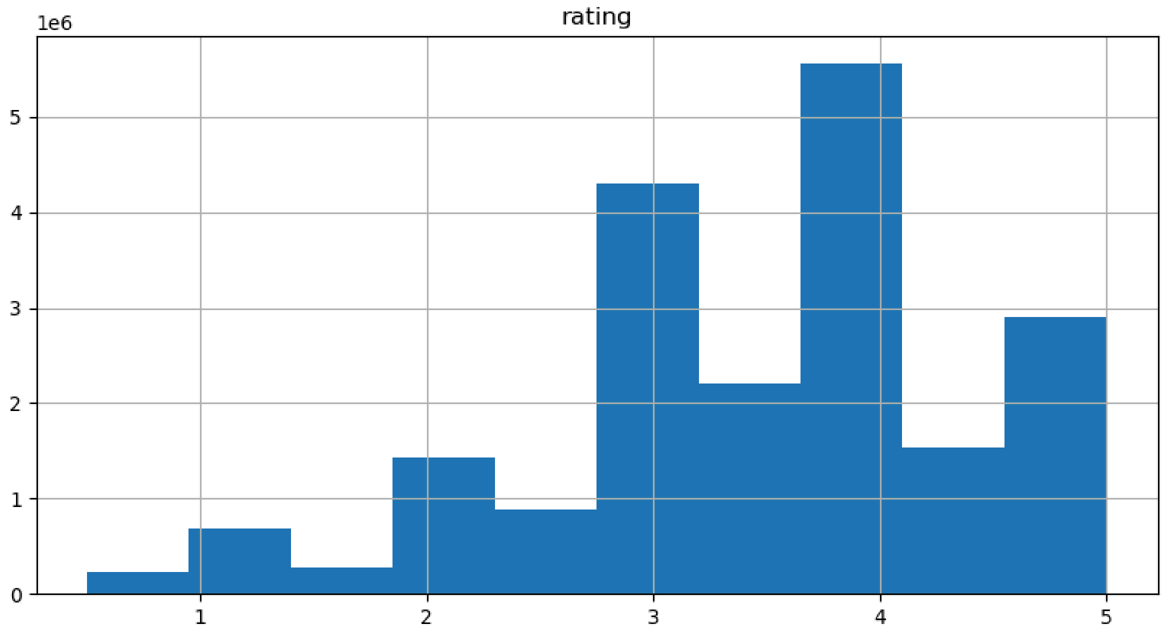
```
Out[46]: False
```

```
In [47]: tags.shape
```

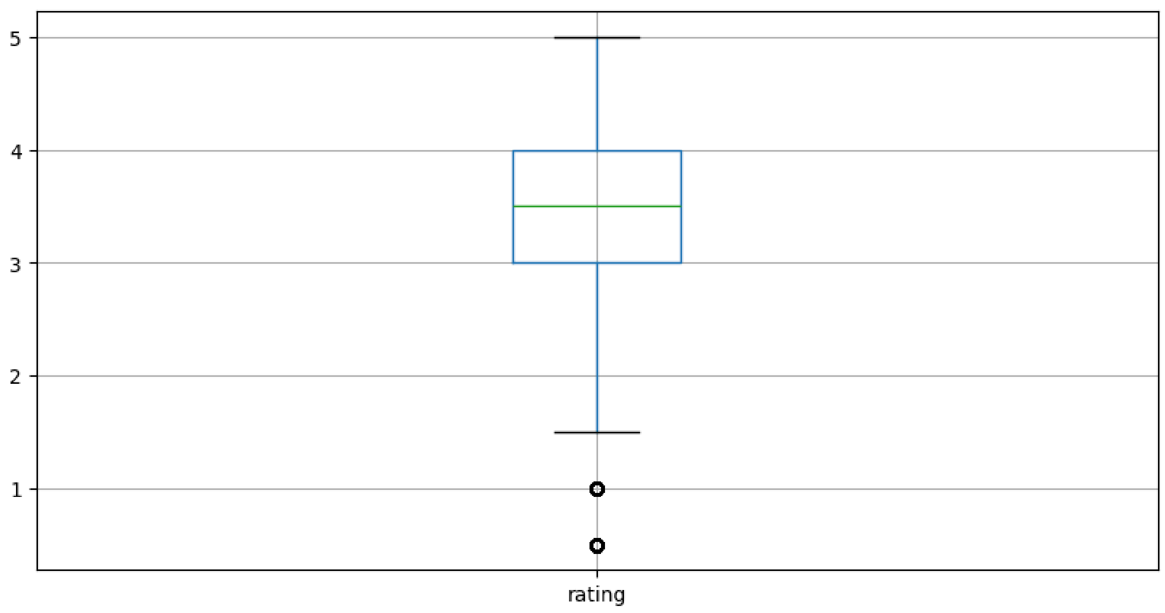
```
Out[47]: (465548, 3)
```

Data Visualization

```
In [48]: import matplotlib.pyplot as plt
         %matplotlib inline
         x=ratings.hist(column='rating', figsize=(10,5))
         plt.show(x)
```



```
In [49]: y=ratings.boxplot(column='rating', figsize=(10,5))
         plt.show(y)
```



Slicing Out Columns

In [50]:
```
tags['tag'].head()
```

Out[50]:
```
0      Mark Waters
1        dark hero
2        dark hero
3    noir thriller
4        dark hero
Name: tag, dtype: object
```

In [51]:
```
movies[['title','genres']].head()
```

Out[51]:

	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 [52]:
```
ratings[-10:]
```

Out[52]:

	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

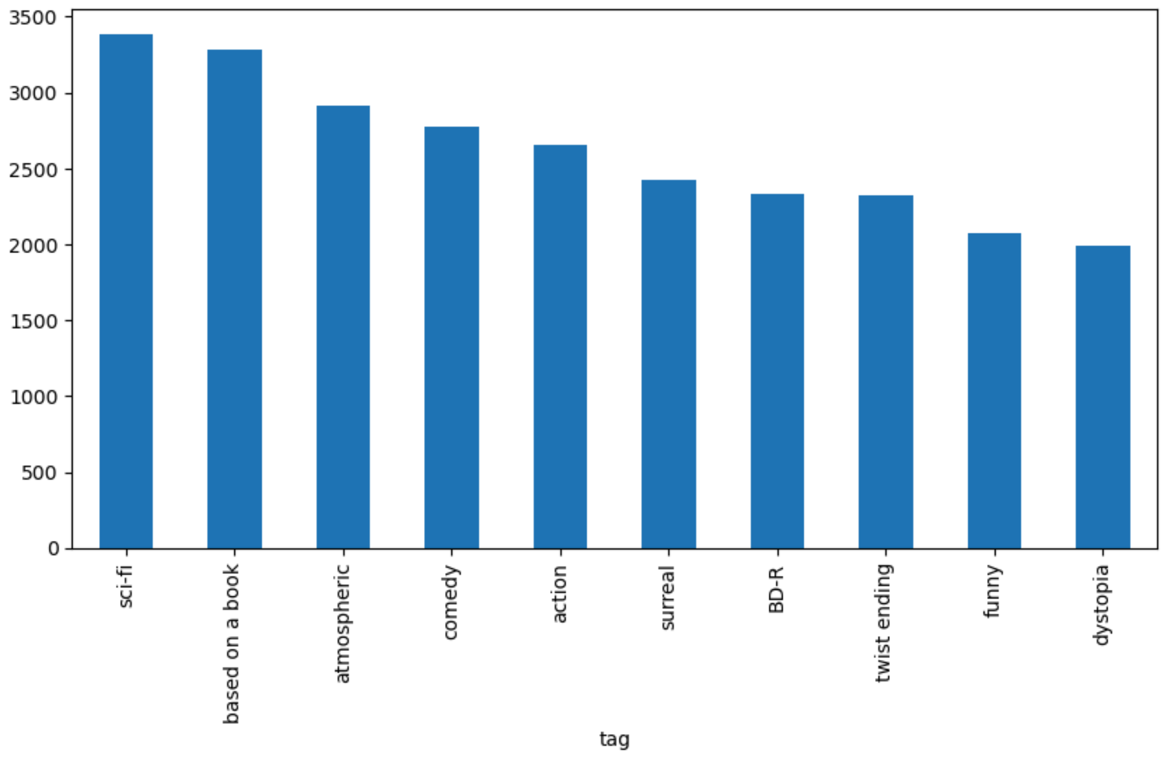
In [53]:
```
tag_counts = tags['tag'].value_counts()
tag_counts
```

Out[53]:
```
tag
sci-fi                                3384
based on a book                       3281
atmospheric                           2917
comedy                                2779
action                                2657
                                      ... 
Paul Adelstein                           1
the wig                                  1
killer fish                              1
genetically modified monsters            1
topless scene                            1
Name: count, Length: 38643, dtype: int64
```

In [54]:
```
tag_counts[:10]
```

Out[54]:
```
tag
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: count, dtype: int64
```

In [55]:
```
z=tag_counts[:10].plot(kind='bar', figsize=(10,5))
plt.show(z)
```



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