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

In [2]:
```
movies=pd.read_csv(r"C:\Users\Admin\Desktop\class\my work\imdb rating project\mo
```

In [3]:
```
movies
```

Out[3]:

	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 [4]:
```
movies.head(20)
```

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
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 [5]:
```
tags=pd.read_csv(r"C:\Users\Admin\Desktop\class\my work\imdb rating project\tag.
```

In [6]:
```
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 [7]:
```
tags.head(20)
```

Out[7]:

	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
5	65	668	bollywood	2013-05-10 01:37:56
6	65	898	screwball comedy	2013-05-10 01:42:40
7	65	1248	noir thriller	2013-05-10 01:39:43
8	65	1391	mars	2013-05-10 01:40:55
9	65	1617	neo-noir	2013-05-10 01:43:37
10	65	1694	jesus	2013-05-10 01:38:45
11	65	1783	noir thriller	2013-05-10 01:39:43
12	65	2022	jesus	2013-05-10 01:38:45
13	65	2193	dragon	2013-05-10 02:01:54
14	65	2353	conspiracy theory	2013-05-10 02:01:06
15	65	2662	mars	2013-05-10 01:40:55
16	65	2726	noir thriller	2013-05-10 01:39:43
17	65	2840	jesus	2013-05-10 01:38:45
18	65	3052	jesus	2013-05-10 01:38:46
19	65	5135	bollywood	2013-05-10 01:37:56

In [8]:
```
ratings=pd.read_csv(r"C:\Users\Admin\Desktop\class\my work\imdb rating project\r
ratings
```

Out[8]:

	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]:
```
ratings.head(20)
```

Out[9]:

	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
5	1	112	3.5	2004-09-10 03:09:00
6	1	151	4.0	2004-09-10 03:08:54
7	1	223	4.0	2005-04-02 23:46:13
8	1	253	4.0	2005-04-02 23:35:40
9	1	260	4.0	2005-04-02 23:33:46
10	1	293	4.0	2005-04-02 23:31:43
11	1	296	4.0	2005-04-02 23:32:47
12	1	318	4.0	2005-04-02 23:33:18
13	1	337	3.5	2004-09-10 03:08:29
14	1	367	3.5	2005-04-02 23:53:00
15	1	541	4.0	2005-04-02 23:30:03
16	1	589	3.5	2005-04-02 23:45:57
17	1	593	3.5	2005-04-02 23:31:01
18	1	653	3.0	2004-09-10 03:08:11
19	1	919	3.5	2004-09-10 03:07:01

In [10]:
```
del(ratings['timestamp'])
del(tags['timestamp'])
```

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

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

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

Out[12]:

	userId	movieId	tag
0	18	4141	Mark Waters

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

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

In [14]:
```
print(type(row_0))
print(row_0.index)
```

```
<class 'pandas.core.series.Series'>
Index(['userId', 'movieId', 'tag'], dtype='object')
```

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

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

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

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

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

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

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

Out[18]:
```
'first row'
```

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

Out[19]:
```
RangeIndex(start=0, stop=465564, step=1)
```

In [20]:
```
tags.shape
```

Out[20]:
```
(465564, 3)
```

In [21]:
```
tags.head()
```

Out[21]:

	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 [22]:
```
tags.iloc[[1,1110,20]]
```

Out[22]:

	userId	movieId	tag
1	65	208	dark hero
1110	409	59315	Jeff Bridges
20	65	6539	treasure

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.describe()
```

Out[24]:

	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 [25]:
```
ratings['rating'].min()
```

Out[25]:
```
0.5
```

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

Out[26]:
```
5.0
```

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

Out[27]:
```
3.5255285642993797
```

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)
```

```
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 [32]:
```
filter2=ratings['rating']>0
filter2
```

Out[32]:
```
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
```

In [34]:
```
movies.shape
```

Out[34]:
```
(27278, 3)
```

In [35]:
```
movies.isnull().any().any()
```

Out[35]:
```
False
```

In [36]:
```
ratings.shape
```

Out[36]:
```
(20000263, 3)
```

In [37]:
```
ratings.isnull().any().any()
```

```
Out[37]: False
```

```
In [38]: tags.shape
```

```
Out[38]: (465564, 3)
```

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

```
Out[43]: True
```

```
In [42]: tags.isnull().sum()
```

```
Out[42]: userId      0
         movieId     0
         tag        16
         dtype: int64
```

```
In [44]: tags=tags.dropna()
```

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

```
Out[45]: False
```

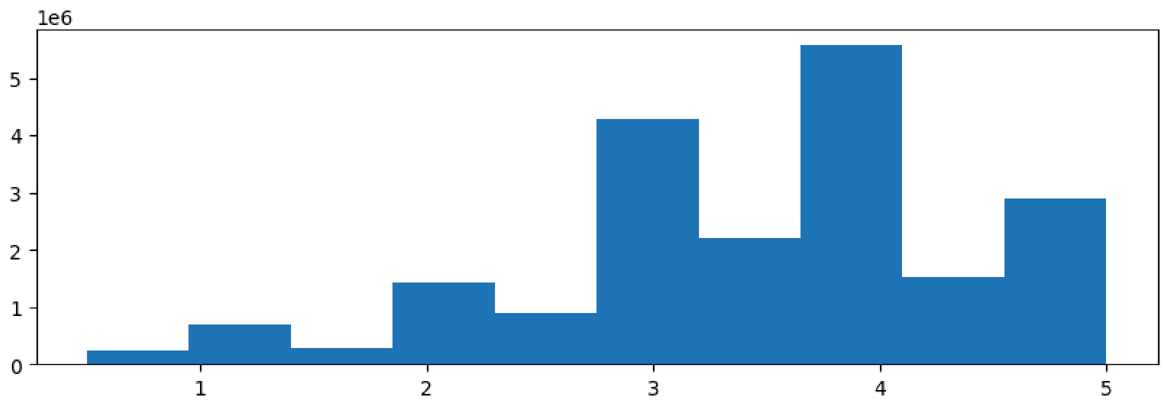
```
In [46]: tags.shape
```

```
Out[46]: (465548, 3)
```

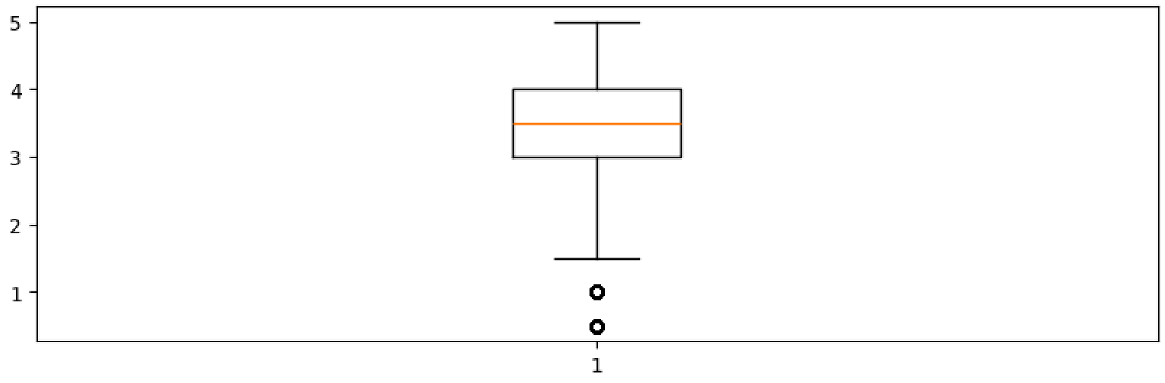
```
In [47]: import matplotlib.pyplot as plt
```

```
In [48]: %matplotlib inline
```

```
In [55]: plt.rcParams['figure.figsize']=10,3
         vis1=plt.hist(ratings['rating'])
```



```
In [56]: vis2=plt.boxplot(ratings['rating'])
```



```
In [57]: movies[['title','genres']]
```

Out[57]:

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

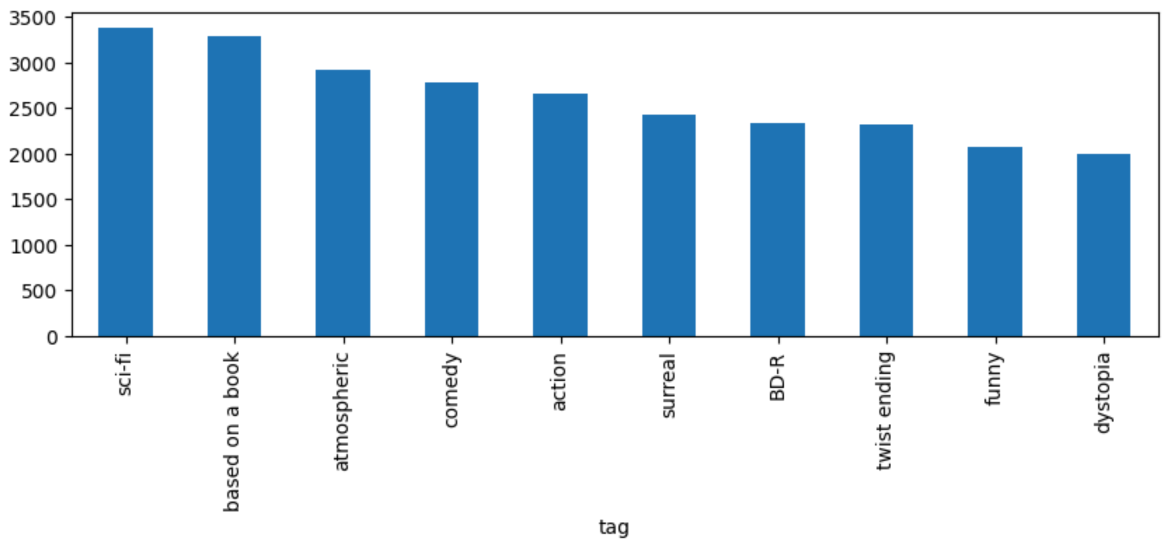
Out[59]:

	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 [65]:
```
tag_count=tags['tag'].value_counts() ####
tag_counts[-10:]
```

Out[65]:
```
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 [70]:
```
vis3=tag_count[:10].plot(kind='bar') ##
```


sci-fi
based on a book
atmospheric
comedy
action
surreal
BD-R
twist ending
funny
dystopia
tag