IMDB Movie Rating Analysis Using Pandas & Matplotlib

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

In [2]:
```
movies=pd.read_csv(r'C:\Users\yogay\OneDrive\Documents\python\movie.csv')
```

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]:
```
print(type(movies))
movies.head()
```

```
<class 'pandas.core.frame.DataFrame'>
```

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

In [5]:
```
tags=pd.read_csv(r'C:\Users\yogay\OneDrive\Documents\python\tag.csv')
```

In [10]:
```
print(type(tags))
tags.head()
```

```
<class 'pandas.core.frame.DataFrame'>
```

Out[10]:

	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 [8]:
```
ratings=pd.read_csv(r'C:\Users\yogay\OneDrive\Documents\python\rating.csv')
```

In [11]:
```
print(type(ratings))
ratings.head()
```

```
<class 'pandas.core.frame.DataFrame'>
```

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

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

In [13]:
```
tags.head()
```

Out[13]:

	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 [14]:
```
del ratings['timestamp']
ratings.head()
```

Out[14]:

	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 [15]:
```
row_0 = tags.iloc[0]
type(row_0)
```

Out[15]:
```
pandas.core.series.Series
```

In [16]:
```
print(row_0)
```

```
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 [20]:
```
row_0['userId']  #count of userId
```

Out[20]:
```
18
```

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

Out[21]:
```
(465564, 3)
```

In [22]:
```
'rating' in row_0
```

Out[22]:
```
False
```

In [24]:
```
row_0.name   #name in not present in tag file
```

Out[24]:
```
0
```

In [25]:
```
row_0 = row_0.rename('firstRow')
row_0.name
```

Out[25]:
```
'firstRow'
```

In [26]:
```
tags.head
```

Out[26]:
```
<bound method NDFrame.head of         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 x 3 columns]>
```

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.columns
```

Out[28]:
```
Index(['userId', 'movieId', 'tag'], dtype='object')
```

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

Out[29]:
```
RangeIndex(start=0, stop=465564, step=1)
```

In [30]:
```
tags.iloc[ [0,11,500] ]
```

Out[30]:

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

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

Out[31]:
```
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 [32]:
```
ratings.describe()
```

Out[32]:

	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 [33]:
```
ratings['rating'].mean()
```

Out[33]:
```
3.5255285642993797
```

In [34]:
```
ratings.mean()
```

Out[34]:
```
userId     69045.872583
movieId     9041.567330
rating         3.525529
dtype: float64
```

In [35]:
```
ratings['rating'].min()
```

Out[35]:
```
0.5
```

In [36]:
```
ratings['rating'].max()
```

Out[36]:
```
5.0
```

In [37]:
```
ratings['rating'].std()
```

Out[37]:
```
1.051988919275684
```

In [38]:
```
ratings['rating'].mode()
```

Out[38]:
```
0    4.0
Name: rating, dtype: float64
```

In [39]:
```
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
False
```

Out[39]:

In [40]:
```
filter2 = ratings['rating'] > 0
filter2.all()
```

Out[40]:
```
True
```

In [41]:
```
movies.shape
```

Out[41]:
```
(27278, 3)
```

In [42]:
```
movies.isnull().any().any()
```

Out[42]:
```
False
```

In [43]:
```
ratings.shape
```

Out[43]:
```
(20000263, 3)
```

In [44]:
```
ratings.isnull().any().any()
```

Out[44]:
```
False
```

In [45]:
```
tags.shape
```

Out[45]:
```
(465564, 3)
```

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

Out[46]:
```
True
```

In [47]:
```
tags.isnull().any().all()
```

Out[47]:
```
False
```

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

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

In [49]:
```
tags.isnull().any()
```

Out[49]:
```
userId     False
movieId    False
tag         True
dtype: bool
```

In [50]:
```
tags=tags.dropna()
```

In [51]:
```
tags.isnull().any().any()
```

Out[51]:
```
False
```

In [52]:
```
tags.isnull().any()
```

Out[52]:
```
userId     False
movieId    False
tag        False
dtype: bool
```

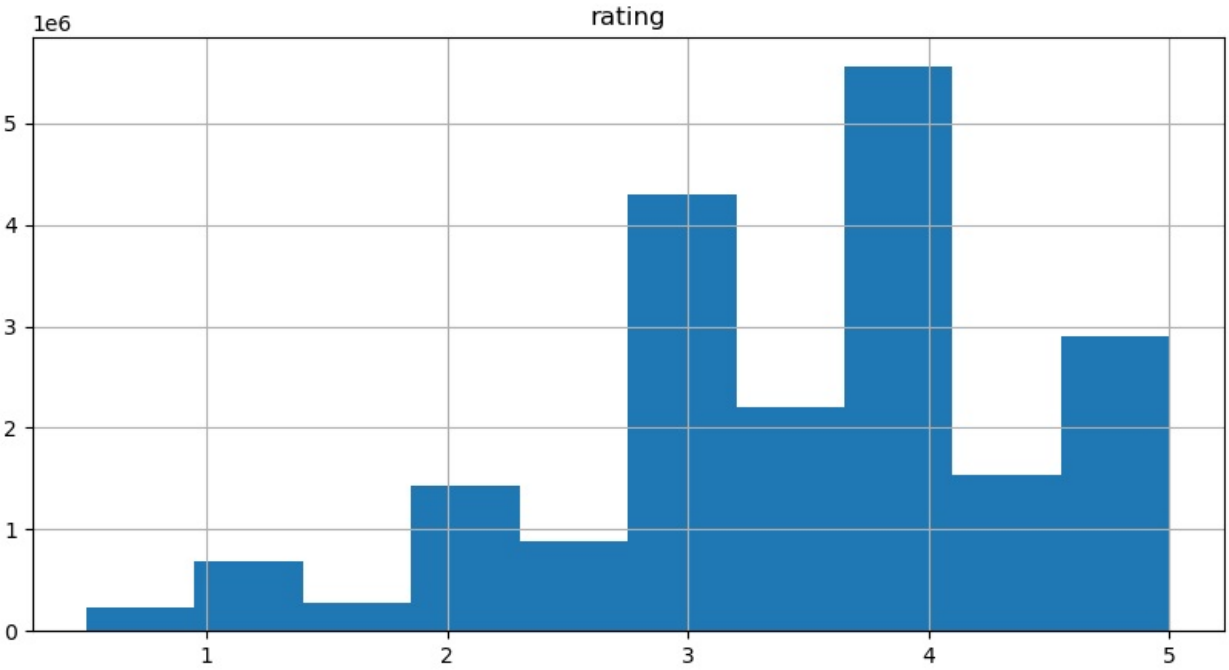
In [53]:
```
tags.shape
```

Out[53]:
```
(465548, 3)
```

In [54]:
```
%matplotlib inline
```

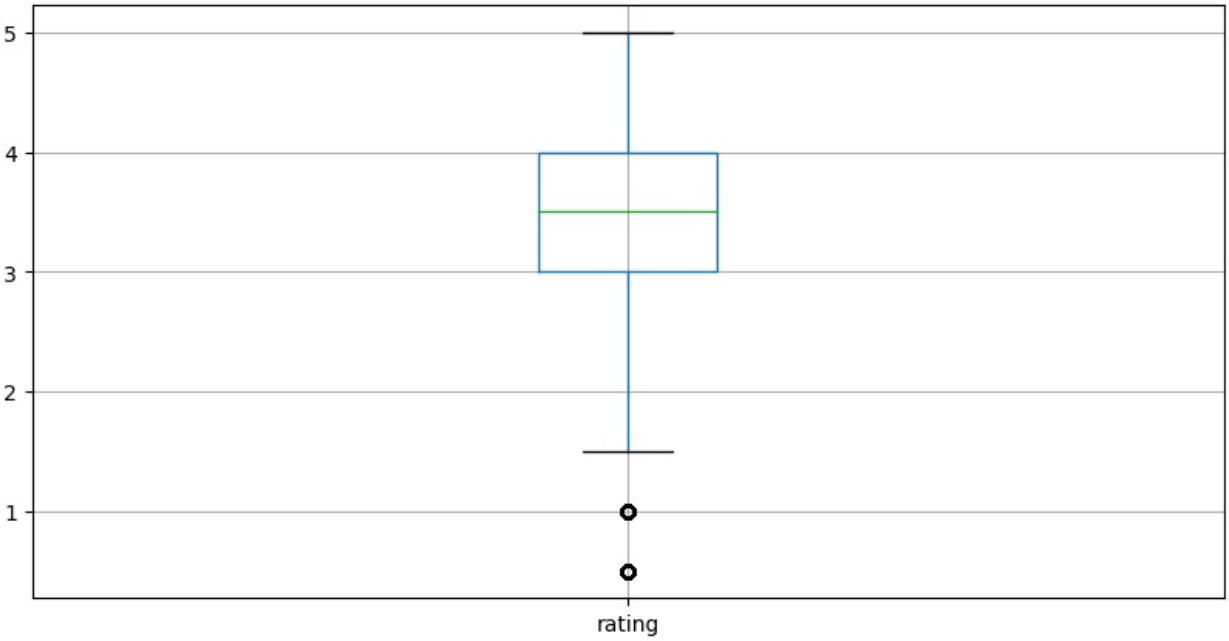
In [54]:
```
%matplotlib inline
ratings.hist(column='rating', figsize=(10,5))
```

Out[54]:
```
array([[<Axes: title={'center': 'rating'}>]], dtype=object)
```



In [55]:
```
ratings.boxplot(column='rating', figsize=(10,5))
```

Out[55]:
```
<Axes: >
```



In [56]:
```
ratings['rating'].head()
```

Out[56]:
```
0    3.5
1    3.5
2    3.5
3    3.5
4    3.5
Name: rating, dtype: float64
```

In [57]:
```
ratings['rating']
```

Out[57]:
```
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 [58]:
```
tags['tag'].head()
```

Out[58]:
```
0      Mark Waters
1        dark hero
2        dark hero
3    noir thriller
4        dark hero
Name: tag, dtype: object
```

In [59]:
```
movies[['title','genres']].head()
```

Out[59]:

	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 [60]:
```
ratings[-10:]
```

Out[60]:

	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 [61]:
```
ratings.shape
```

Out[61]:
```
(20000263, 3)
```

In [62]:
```
ratings
```

Out[62]:

	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

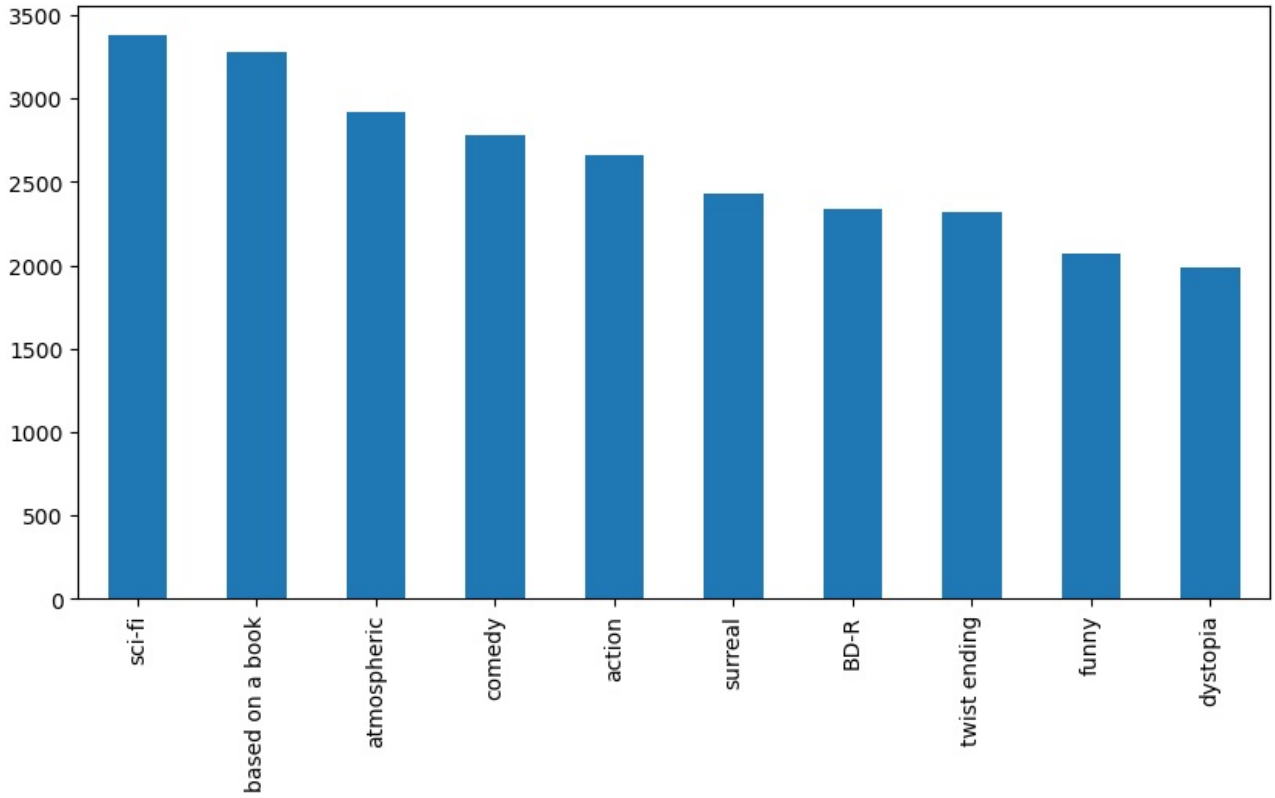
20000263 rows × 3 columns

In [63]:
```
tag_counts = tags['tag'].value_counts()
tag_counts[-10:]
```

Out[63]:
```
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: tag, dtype: int64
```

In [64]:
```
tag_counts[:10].plot(kind='bar', figsize=(10,5))
```

Out[64]: <Axes: >



In [ ]: