In [14]:
```
import pandas as pd
```

In [15]:
```
movies = pd.read_csv(r'C:\Users\RUPA\Downloads\movie.csv')
```

In [16]:
```
movies
```

Out[16]:

| | 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 [17]:
```
import pandas as pd
```

In [18]:
```
rating = pd.read_csv(r'C:\Users\RUPA\Downloads\rating.csv')
```

In [19]:
```
rating
```

Out[19]:

| | 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 [20]:
```
import pandas as pd
```

In [21]:
```
tag = pd.read_csv(r'C:\Users\RUPA\Downloads\tag.csv')
```

In [22]:
```
tag
```

Out[22]:

| | userId | movieId | tag | timestamp |
|---|---|---|---|---|
| **0** | 18 | 4141 | Mark Waters | 24-04-2009 18:19 |
| **1** | 65 | 208 | dark hero | 10-05-2013 01:41 |
| **2** | 65 | 353 | dark hero | 10-05-2013 01:41 |
| **3** | 65 | 521 | noir thriller | 10-05-2013 01:39 |
| **4** | 65 | 592 | dark hero | 10-05-2013 01:41 |
| **...** | ... | ... | ... | ... |
| **465559** | 138446 | 55999 | dragged | 23-01-2013 23:29 |
| **465560** | 138446 | 55999 | Jason Bateman | 23-01-2013 23:29 |
| **465561** | 138446 | 55999 | quirky | 23-01-2013 23:29 |
| **465562** | 138446 | 55999 | sad | 23-01-2013 23:29 |
| **465563** | 138472 | 923 | rise to power | 02-11-2007 21:12 |

465564 rows × 4 columns

In [23]:
```
print(type(movies))
movies.head(20)
```

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

Out[23]:

| | 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 [24]:
```
tag.head()
```

Out[24]:

| | userId | movieId | tag | timestamp |
|---|---|---|---|---|
| **0** | 18 | 4141 | Mark Waters | 24-04-2009 18:19 |
| **1** | 65 | 208 | dark hero | 10-05-2013 01:41 |
| **2** | 65 | 353 | dark hero | 10-05-2013 01:41 |
| **3** | 65 | 521 | noir thriller | 10-05-2013 01:39 |
| **4** | 65 | 592 | dark hero | 10-05-2013 01:41 |

In [25]:
```
se_dates = ['timestamp']
rating.head()
```

Out[25]:

| | 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 [26]:
```
del rating['timestamp']
del tag[timestamp]
```

```
---------------------------------------------------------------------------
-
NameError                                 Traceback (most recent call las
t)
Cell In[26], line 2
      1 del rating['timestamp']
----> 2 del tag[timestamp]

NameError: name 'timestamp' is not defined
```

In [27]:
```
rating.head()
```

Out[27]:

| | 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 [28]:
```
row_0 = tag.iloc[0]
type(row_0)
```

Out[28]:
```
pandas.core.series.Series
```

In [29]:
```
print(row_0)
```

```
userId                    18
movieId                 4141
tag              Mark Waters
timestamp    24-04-2009 18:19
Name: 0, dtype: object
```

In [30]:
```
row_0.index
```

Out[30]:
```
Index(['userId', 'movieId', 'tag', 'timestamp'], dtype='object')
```

In [31]:
```
row_0['userId']
```

Out[31]:
```
18
```

In [32]:
```
'rating' in row_0
```

Out[32]:
```
False
```

In [33]:
```
row_0.name
```

Out[33]:
```
0
```

In [34]:
```
row_0 = row_0.rename('firstRow')
row_0.name
```

Out[34]:
```
'firstRow'
```

In [35]:
```
tag.head()
```

Out[35]:

| | userId | movieId | tag | timestamp |
|---|---|---|---|---|
| **0** | 18 | 4141 | Mark Waters | 24-04-2009 18:19 |
| **1** | 65 | 208 | dark hero | 10-05-2013 01:41 |
| **2** | 65 | 353 | dark hero | 10-05-2013 01:41 |
| **3** | 65 | 521 | noir thriller | 10-05-2013 01:39 |
| **4** | 65 | 592 | dark hero | 10-05-2013 01:41 |

In [36]:
```
tag.index
```

Out[36]:
```
RangeIndex(start=0, stop=465564, step=1)
```

In [37]:
```
tag.columns
```

Out[37]:
```
Index(['userId', 'movieId', 'tag', 'timestamp'], dtype='object')
```

In [38]:
```
tag.iloc[[0,11500]]
```

Out[38]:

| | userId | movieId | tag | timestamp |
|---|---|---|---|---|
| **0** | 18 | 4141 | Mark Waters | 24-04-2009 18:19 |
| **11500** | 2081 | 33679 | explodeytime | 14-01-2006 00:17 |

In [39]:
```
rating['rating'].describe()
```

Out[39]:
```
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 [40]:
```
rating.describe()
```

Out[40]:

| | 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 [41]:
```
rating['rating'].mean()
```

Out[41]:
```
3.5255285642993797
```

In [42]:
```
rating.mean()
```

Out[42]:
```
userId     69045.872583
movieId     9041.567330
rating         3.525529
dtype: float64
```

In [43]:
```
rating['rating'].min()
```

Out[43]:
```
0.5
```

In [44]:
```
rating['rating'].max
```

Out[44]:
```
<bound method NDFrame._add_numeric_operations.<locals>.max 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 [45]:
```
rating['rating'].max()
```

Out[45]:
```
5.0
```

In [46]:
```
rating['rating'].std()
```

Out[46]:
```
1.051988919275684
```

In [47]:
```
rating['rating'].mode()
```

Out[47]:
```
0    4.0
Name: rating, dtype: float64
```

In [48]:
```
rating.corr()
```

Out[48]:

| | 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 [49]:
```
filter1 = rating['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[49]:
```
False
```

In [50]:
```
filter2 = rating['rating'] > 0
filter2.all()
```

Out[50]:
```
True
```

Data Cleaning: Handling Missing Data¶

In [51]:
```
movies.shape
```

Out[51]:
```
(27278, 3)
```

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

Out[52]:
```
False
```

In [53]:
```
movies.isnull().any()
```

Out[53]:
```
movieId    False
title      False
genres     False
dtype: bool
```

In [54]:
```
tag.shape
```

Out[54]:
```
(465564, 4)
```

In [55]:
```
tag.isnull().any().any()
```

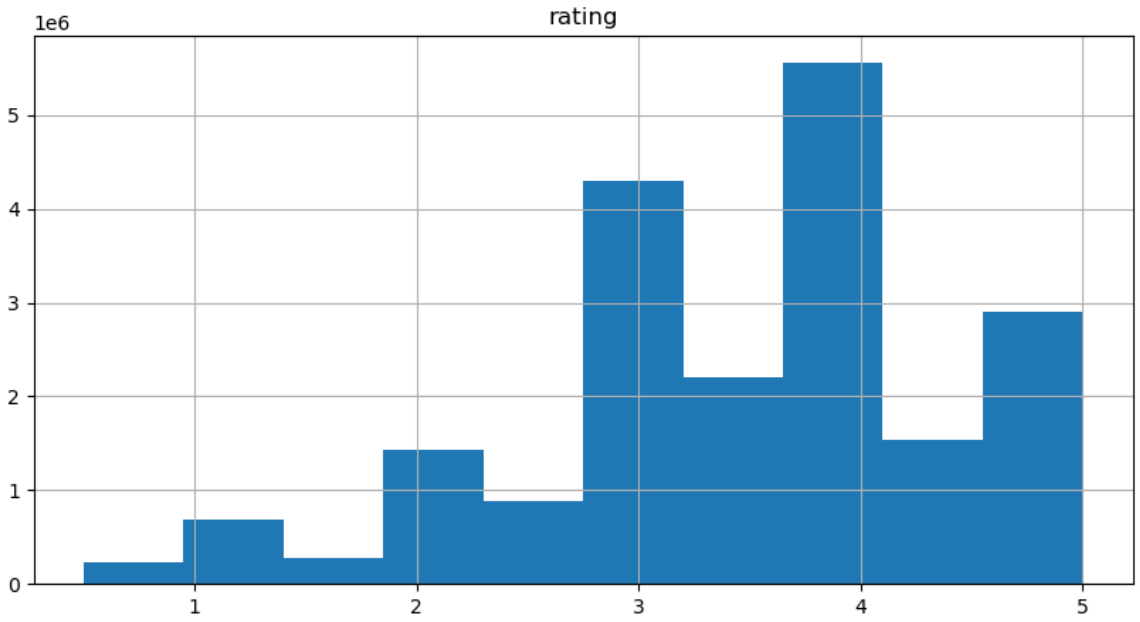
Out[55]:
```
True
```

Data Visualization¶

In [79]:
```
%matplotlib inline
```

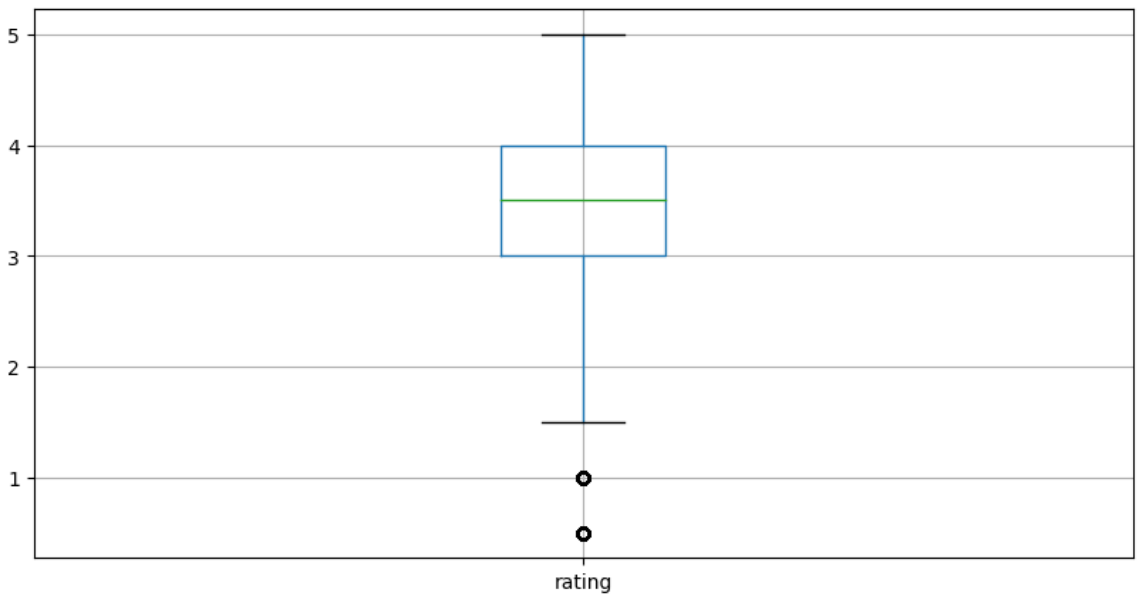
In [74]:
```
rating.hist(column='rating', figsize=(10,5))
```

Out[74]:
```
array([[<Axes: title={'center': 'rating'}>]], dtype=object)
```




In [86]:
```
rating.boxplot(column='rating', figsize=(10,5))
```

Out[86]:
```
<Axes: >
```



Slicing Out Columns

In [81]:
```
tag['tag'].head()
```

Out[81]:
```
0      Mark Waters
1        dark hero
2        dark hero
3    noir thriller
4        dark hero
Name: tag, dtype: object
```

In [82]:
```
movies[['title' , 'genres']].head()
```

Out[82]:

| | 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 [83]:
```
rating[-10:]
```

Out[83]:

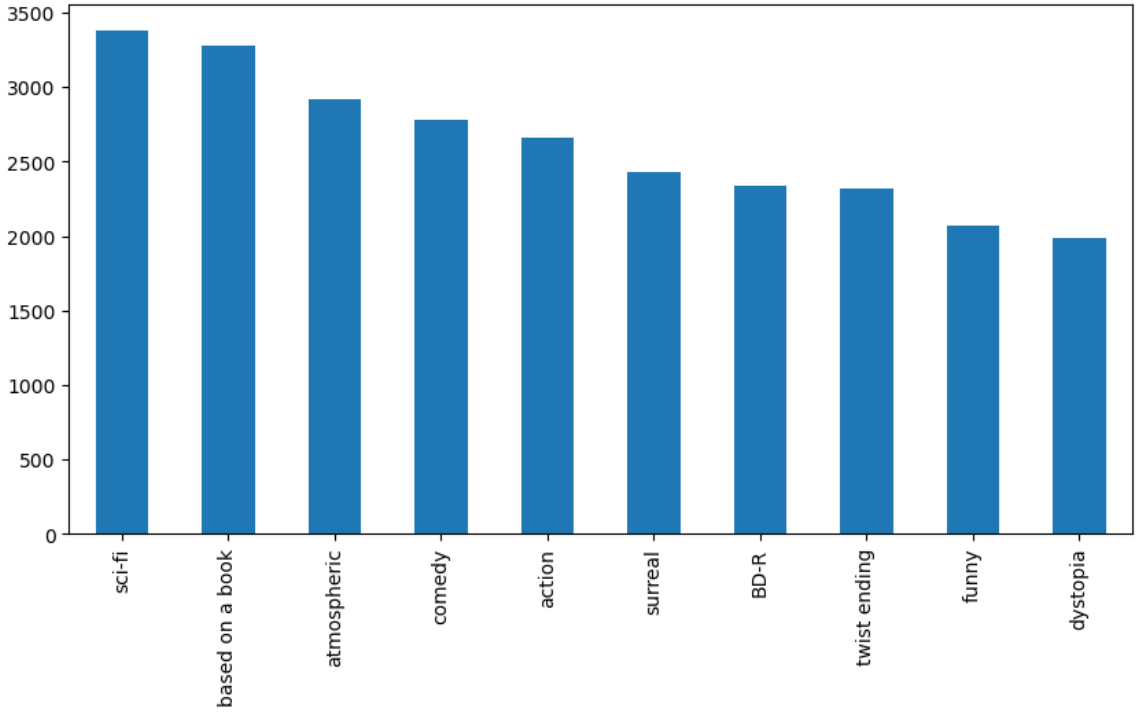
| | userId | movieId | rating |
|---|---|---|---|
| **20000253** | 138493 | 60816 | 4.5 |
| **20000254** | 138493 | 61160 | 4.0 |
| **20000255** | 138493 | 65682 | 4.5 |
| **20000256** | 138493 | 66762 | 4.5 |
| **20000257** | 138493 | 68319 | 4.5 |
| **20000258** | 138493 | 68954 | 4.5 |
| **20000259** | 138493 | 69526 | 4.5 |
| **20000260** | 138493 | 69644 | 3.0 |
| **20000261** | 138493 | 70286 | 5.0 |
| **20000262** | 138493 | 71619 | 2.5 |

In [84]:
```
tag_counts = tag['tag'].value_counts()
tag_counts[-10:]
```

Out[84]:
```
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 [87]:
```
tag_counts[:10].plot(kind='bar',figsize=(10,5))
```

Out[87]: <Axes: >




In [ ]: