IMDB Movie Rating Analysis

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

In [2]:
```
movies = pd.read_csv("movie.csv")
```

In [3]:
```
movies.head()
```

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

In [4]:
```
movies.shape
```

Out[4]:
```
(27278, 3)
```

In [5]:
```
ratings = pd.read_csv("rating.csv")
ratings.head()
```

Out[5]:

	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 [6]:
```
ratings.shape
```

Out[6]:
```
(20000263, 4)
```

In [7]:
```
tags = pd.read_csv("tag.csv")
tags.head()
```

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

In [8]:
```
tags.shape
```
Out[8]:
```
(465564, 4)
```

In [9]:
```
movies.columns
```
Out[9]:
```
Index(['movieId', 'title', 'genres'], dtype='object')
```

In [10]:
```
tags.columns
```
Out[10]:
```
Index(['userId', 'movieId', 'tag', 'timestamp'], dtype='object')
```

In [11]:
```
ratings.columns
```
Out[11]:
```
Index(['userId', 'movieId', 'rating', 'timestamp'], dtype='object')
```

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

In [13]:
```
ratings.head()
```
Out[13]:

	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 [14]:
```
tags.head()
```
Out[14]:

	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 [15]: # Series
         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 [18]: row_0['userId']
```

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

```
In [19]: 'rating' in row_0
```

```
Out[19]: False
```

```
In [20]: row_0.name
```

```
Out[20]: 0
```

```
In [21]: row_0 = row_0.rename('firstRow')
         row_0.name
```

```
Out[21]: 'firstRow'
```

```
In [22]: row_0
```

```
Out[22]: userId               18
         movieId            4141
         tag         Mark Waters
         Name: firstRow, dtype: object
```

Dataframes

```
In [23]: tags.head()
```

Out[23]:

	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 [24]:
```
tags.index
```
Out[24]:
```
RangeIndex(start=0, stop=465564, step=1)
```

In [25]:
```
tags.columns
```
Out[25]:
```
Index(['userId', 'movieId', 'tag'], dtype='object')
```

In [26]:
```
tags.iloc[[0,11,500]]
```
Out[26]:

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

Descriptive Statistics

In [27]:
```
ratings
```
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
...	...	...	...
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 [28]:
```
ratings['rating'].describe()
```

```
Out[28]: 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 [29]: ratings.describe()
```

Out[29]:

	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 [30]: ratings['rating'].mean()

Out[30]: 3.5255285642993797

In [31]: ratings.mean()

Out[31]: userId      69045.872583
         movieId      9041.567330
         rating          3.525529
         dtype: float64

In [32]: ratings['rating'].min()

Out[32]: 0.5

In [33]: ratings['rating'].max()

Out[33]: 5.0

In [34]: ratings['rating'].std()

Out[34]: 1.051988919275684

In [35]: ratings['rating'].mode()

Out[35]: 0    4.0
         Name: rating, dtype: float64

In [36]: ratings.corr()
```

Out[36]:

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

In [38]:
```
filter2 = ratings['rating'] > 0
filter2.all()
```

Out[38]:
```
True
```

Data Cleaning : Handling Missing Data

In [39]:
```
movies.shape
```

Out[39]:
```
(27278, 3)
```

In [40]:
```
movies.isnull().any().any()  #no null values
```

Out[40]:
```
False
```

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

Out[41]:
```
(20000263, 3)
```

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

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

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

Out[43]:
```
(465564, 3)
```

In [44]:
```
tags.isnull().any().any()   # we have some tags which are null
```

```
Out[44]: True
```

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

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

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

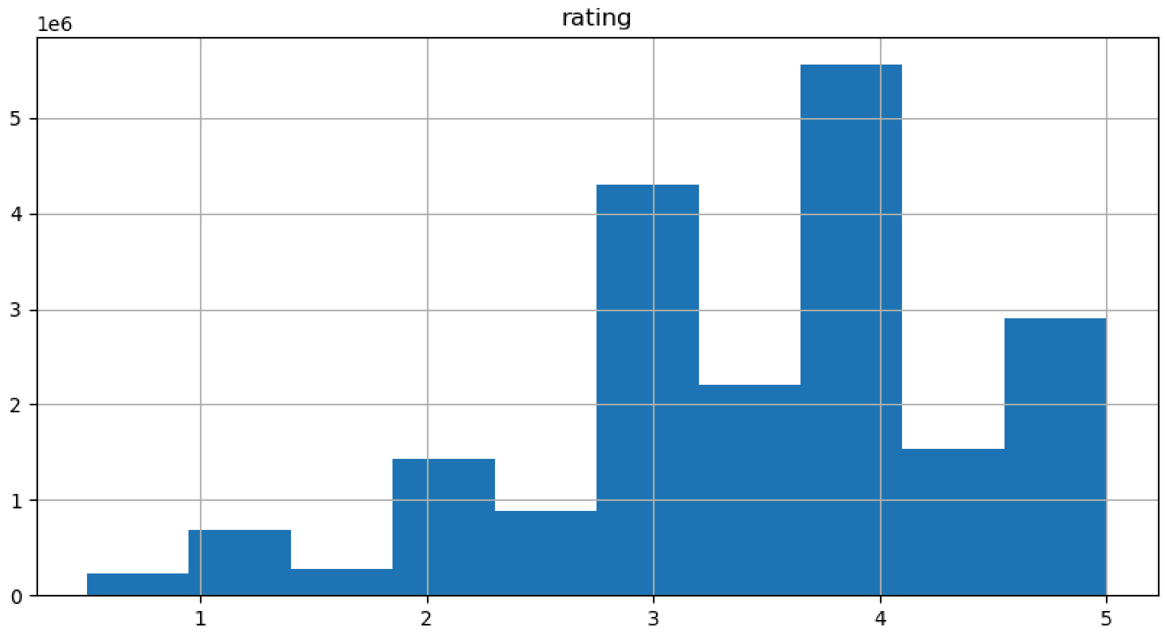
```
In [47]: tags.shape
```

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

Data Visualisation

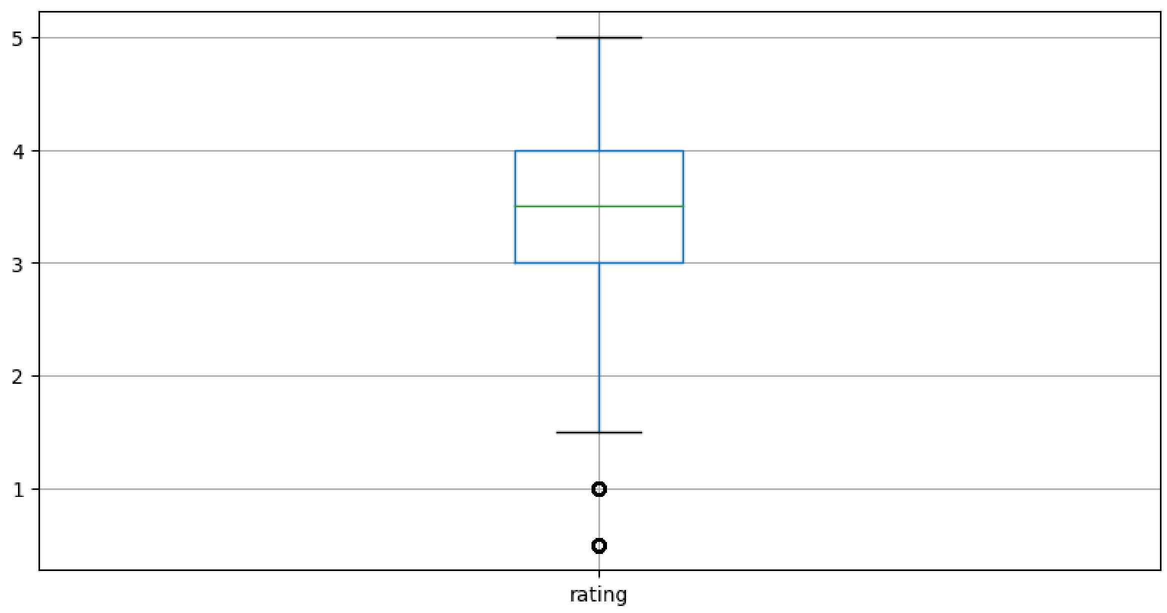
```
In [48]: %matplotlib inline
         ratings.hist(column='rating',figsize=(10,5))
```

```
Out[48]: array([[<Axes: title={'center': 'rating'}>]], dtype=object)
```



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

```
Out[49]: <Axes: >
```



Slicing Out Columns

```
In [50]: tags
```

Out[50]:

	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

465548 rows × 3 columns

```
In [51]: tags['tag'].head()
```

```
Out[51]: 0      Mark Waters
         1        dark hero
         2        dark hero
         3    noir thriller
         4        dark hero
         Name: tag, dtype: object
```

```
In [52]: movies[['title','genres']].head()
```

Out[52]:

	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 [53]:
```
ratings[-10::]
```

Out[53]:

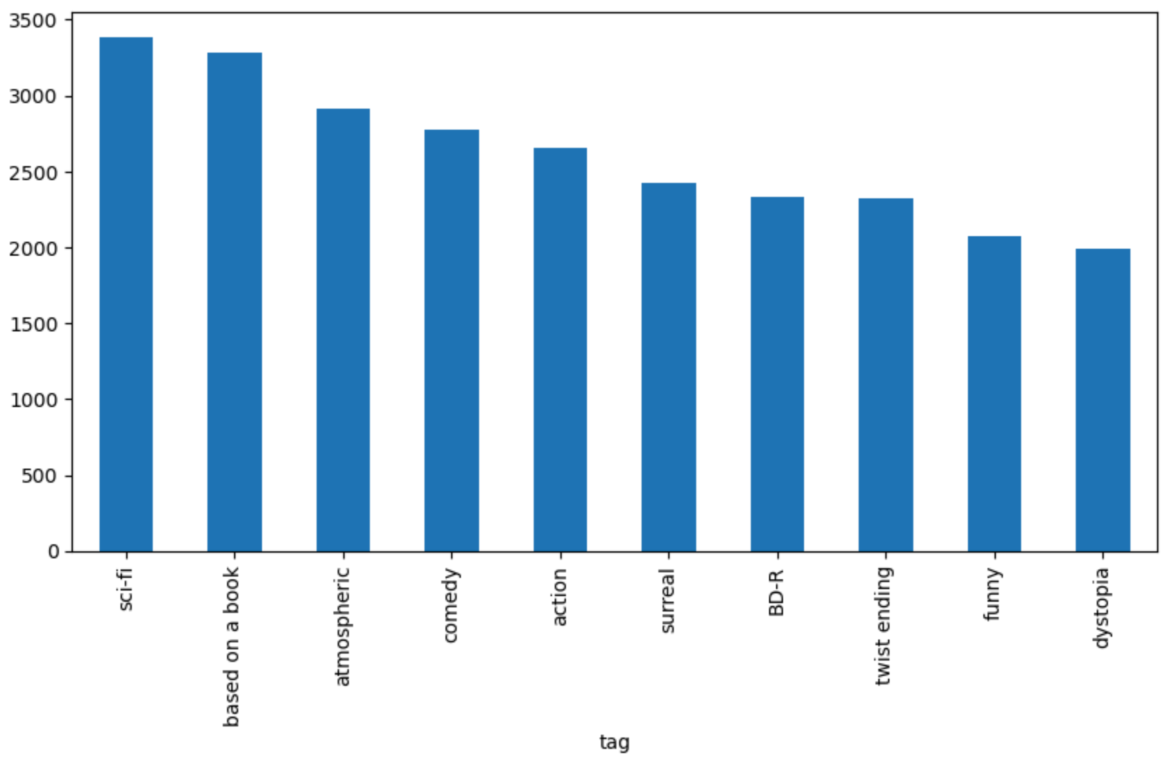
	userId	movieId	rating
20000253	138493	60816	4.5
20000254	138493	61160	4.0
20000255	138493	65682	4.5
20000256	138493	66762	4.5
20000257	138493	68319	4.5
20000258	138493	68954	4.5
20000259	138493	69526	4.5
20000260	138493	69644	3.0
20000261	138493	70286	5.0
20000262	138493	71619	2.5

In [54]:
```
tag_counts = tags['tag'].value_counts()
tag_counts[-10:]
```

Out[54]:
```
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 [55]:
```
tag_counts[:10].plot(kind='bar', figsize=(10,5))
```

Out[55]:
```
<Axes: xlabel='tag'>
```



Filters for Selecting Rows

In [56]:
```
ratings
```

Out[56]:

	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 [57]:
```
is_highly_rated = ratings['rating'] >= 5.0
ratings[is_highly_rated][30:50]
```

Out[57]:

	userId	movieId	rating
239	3	50	5.0
242	3	175	5.0
244	3	223	5.0
245	3	260	5.0
246	3	316	5.0
247	3	318	5.0
248	3	329	5.0
252	3	457	5.0
253	3	480	5.0
254	3	490	5.0
256	3	541	5.0
258	3	593	5.0
263	3	858	5.0
264	3	904	5.0
267	3	924	5.0
268	3	953	5.0
271	3	1060	5.0
272	3	1073	5.0
275	3	1084	5.0
276	3	1089	5.0

In [58]:

```
movies
```

Out[58]:

	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 [59]:
```
is_action = movies['genres'].str.contains('Action')
movies[is_action][5:15]
```

Out[59]:

	movieId	title	genres
22	23	Assassins (1995)	Action|Crime|Thriller
41	42	Dead Presidents (1995)	Action|Crime|Drama
43	44	Mortal Kombat (1995)	Action|Adventure|Fantasy
50	51	Guardian Angel (1994)	Action|Drama|Thriller
65	66	Lawnmower Man 2: Beyond Cyberspace (1996)	Action|Sci-Fi|Thriller
69	70	From Dusk Till Dawn (1996)	Action|Comedy|Horror|Thriller
70	71	Fair Game (1995)	Action
75	76	Screamers (1995)	Action|Sci-Fi|Thriller
77	78	Crossing Guard, The (1995)	Action|Crime|Drama|Thriller
85	86	White Squall (1996)	Action|Adventure|Drama

In [60]:
```
movies[is_action].head(15)
```

Out[60]:

	movieId	title	genres
5	6	Heat (1995)	Action|Crime|Thriller
8	9	Sudden Death (1995)	Action
9	10	GoldenEye (1995)	Action|Adventure|Thriller
14	15	Cutthroat Island (1995)	Action|Adventure|Romance
19	20	Money Train (1995)	Action|Comedy|Crime|Drama|Thriller
22	23	Assassins (1995)	Action|Crime|Thriller
41	42	Dead Presidents (1995)	Action|Crime|Drama
43	44	Mortal Kombat (1995)	Action|Adventure|Fantasy
50	51	Guardian Angel (1994)	Action|Drama|Thriller
65	66	Lawnmower Man 2: Beyond Cyberspace (1996)	Action|Sci-Fi|Thriller
69	70	From Dusk Till Dawn (1996)	Action|Comedy|Horror|Thriller
70	71	Fair Game (1995)	Action
75	76	Screamers (1995)	Action|Sci-Fi|Thriller
77	78	Crossing Guard, The (1995)	Action|Crime|Drama|Thriller
85	86	White Squall (1996)	Action|Adventure|Drama

Group By and Aggregate

In [61]:

```
ratings
```

Out[61]:

	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 [62]:
```
ratings_count = ratings[['movieId','rating']].groupby('rating').count()
ratings_count
```

Out[62]:

	movieId
rating	
0.5	239125
1.0	680732
1.5	279252
2.0	1430997
2.5	883398
3.0	4291193
3.5	2200156
4.0	5561926
4.5	1534824
5.0	2898660

In [63]:
```
average_rating = ratings[['movieId','rating']].groupby('movieId').mean()
average_rating.head()
```

Out[63]:

	rating
movieId	
1	3.921240
2	3.211977
3	3.151040
4	2.861393
5	3.064592

In [64]:
```
movie_count = ratings[['movieId','rating']].groupby('movieId').count()
movie_count.head()
```

Out[64]:

	rating
movieId	
1	49695
2	22243
3	12735
4	2756
5	12161

In [65]:
```
movie_count = ratings[['movieId','rating']].groupby('movieId').count()
movie_count.tail()
```

Out[65]:

	rating
movieId	
131254	1
131256	1
131258	1
131260	1
131262	1

Merge DataFrames

In [66]:
```
tags.head()
```

Out[66]:

	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 [67]:
```
movies.head()
```

Out[67]:

	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 [69]:
```
t = movies.merge(tags, on='movieId', how ='inner')
t.head()
```

Out[69]:

	movieId	title	genres	userId	tag
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy	1644	Watched
1	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy	1741	computer animation
2	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy	1741	Disney animated feature
3	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy	1741	Pixar animation
4	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy	1741	TÃ©a Leoni does not star in this movie