IMDB Movie Rating Analysis

Kaggle Project: Pandas with DataScience AI

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

Read the movie dataset

In [2]:
```
movies=pd.read_csv(r'C:\Users\world\Desktop\FullStackDSandAI\Day25-14July2025\movie
```

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

shape gives number of rows and columns in dataset. output(rows,columns)

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

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

Read the rating dataset

In [5]:
```
ratings=pd.read_csv(r'C:\Users\world\Desktop\FullStackDSandAI\Day25-14July2025\rati
```

In [6]:
```
ratings.head()
```

Out[6]:

	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 [7]:
```
ratings.shape
```

Out[7]:
```
(20000263, 4)
```

Read the tag dataset

In [8]:
```
tags=pd.read_csv(r'C:\Users\world\Desktop\FullStackDSandAI\Day25-14July2025\tag.csv
```

In [9]:
```
tags.head()
```

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

In [10]:
```
tags.shape
```

Out[10]:
```
(465564, 4)
```

In [11]:
```
tags.columns
```

Out[11]:
```
Index(['userId', 'movieId', 'tag', 'timestamp'], dtype='object')
```

In [12]:
```
ratings.columns
```

Out[12]:
```
Index(['userId', 'movieId', 'rating', 'timestamp'], dtype='object')
```

Delete a column 'timestamp' from data sets 'ratings' and 'tags'

In [13]:
```
del ratings['timestamp']
del tags['timestamp']
```

In [14]:
```
ratings.columns
```

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

In [15]:
```
tags.columns
```

Out[15]:
```
Index(['userId', 'movieId', 'tag'], dtype='object')
```

In [16]:
```
tags.head()
```

Out[16]:

	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

iloc[index] ----> it returns given indexed row

In [17]:
```
tags.iloc[0]
```

Out[17]:
```
userId              18
movieId           4141
tag        Mark Waters
Name: 0, dtype: object
```

In [18]:
```
tags.iloc[2]
```

Out[18]:
```
userId            65
movieId          353
tag        dark hero
Name: 2, dtype: object
```

In [19]:
```
row_0=tags.iloc[0]
```

In [20]:
```
print(row_0)
```

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

To get the column names of above row_0

In [21]:
```
row_0.index
```

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

In [22]:
```
row_0['userId']
```

Out[22]: 18

In [23]:
```
'rating' in row_0
```

Out[23]: False

In [24]:
```
4141 in row_0
```

Out[24]: False

In [25]:
```
'tag' in row_0
```

Out[25]: True

In [26]:
```
'Mark Waters' in row_0
```

Out[26]: False

In [27]:
```
row_0
```

Out[27]:
```
userId             18
movieId          4141
tag        Mark Waters
Name: 0, dtype: object
```

In [28]:
```
row_0.name
```

Out[28]: 0

rename()

In [29]:
```
row_0=row_0.rename('firstRow')
row_0.name
```

Out[29]: 'firstRow'

In [30]:
```
row_0
```

Out[30]:
```
userId             18
movieId          4141
tag        Mark Waters
Name: firstRow, dtype: object
```

head() ---> gives first 5 records

In [31]:
```
tags.head()
```

Out[31]:

	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

index

In [32]:
```
tags.index
```

Out[32]:
```
RangeIndex(start=0, stop=465564, step=1)
```

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

Out[33]:
```
(465564, 3)
```

columns

In [34]:
```
tags.columns
```

Out[34]:
```
Index(['userId', 'movieId', 'tag'], dtype='object')
```

to get specified records using iloc

In [35]:
```
tags.iloc[[0,11,500]]
```

Out[35]:

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

Descriptive Statistics

lets look how the reatings are distributed

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

Out[36]:

	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

describe()

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

Out[37]:
```
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 [38]:
```
ratings.describe()
```

Out[38]:

	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

mean()

In [39]:
```
ratings['rating'].mean()
```

Out[39]:
```
3.5255285642993797
```

In [40]:
```
ratings.mean()
```

Out[40]:
```
userId     69045.872583
movieId     9041.567330
rating         3.525529
dtype: float64
```

min()

In [41]:
```
ratings['rating'].min()
```

Out[41]:
```
0.5
```

max()

In [42]:
```
ratings['rating'].max()
```

Out[42]:
```
5.0
```

std()

In [43]:
```
ratings['rating'].std()
```

Out[43]:
```
1.051988919275684
```

mode()

In [44]:
```
ratings['rating'].mode()
```

Out[44]:
```
0    4.0
Name: rating, dtype: float64
```

corr()

In [45]:
```
ratings.corr()
```

Out[45]:

	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 [46]:
```
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
```

any()

In [47]:
```
filter1.any()
```

Out[47]: False

In [48]:
```
filter2=ratings['rating']>0
print(filter2)
```

```
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
```

all()

In [49]:
```
filter2.all()
```

Out[49]: True

Data Cleaning: Handling missing Data

In [50]:
```
movies.shape
```

Out[50]: (27278, 3)

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

Out[51]:
```
movieId    False
title      False
genres     False
dtype: bool
```

Checking any null values in 'movies' data set

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

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

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

Out[53]:
```
(20000263, 3)
```

Checking any null values in 'ratings' data set

In [54]:
```
ratings.isnull().any().any()
```

Out[54]:
```
False
```

Checking any null values in 'tags' data set

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

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

In [56]:
```
tags.isnull().any()
```

Out[56]:
```
userId     False
movieId    False
tag         True
dtype: bool
```

dropna() ---> removes missing values

In [57]:
```
tags.shape
```

Out[57]:
```
(465564, 3)
```

In [58]:
```
tags=tags.dropna()
```

In [59]:
```
tags.shape
```

Out[59]:
```
(465548, 3)
```

Data Visualization

In [60]:
```
import matplotlib.pyplot as plt
```

```
%matplotlib inline
```

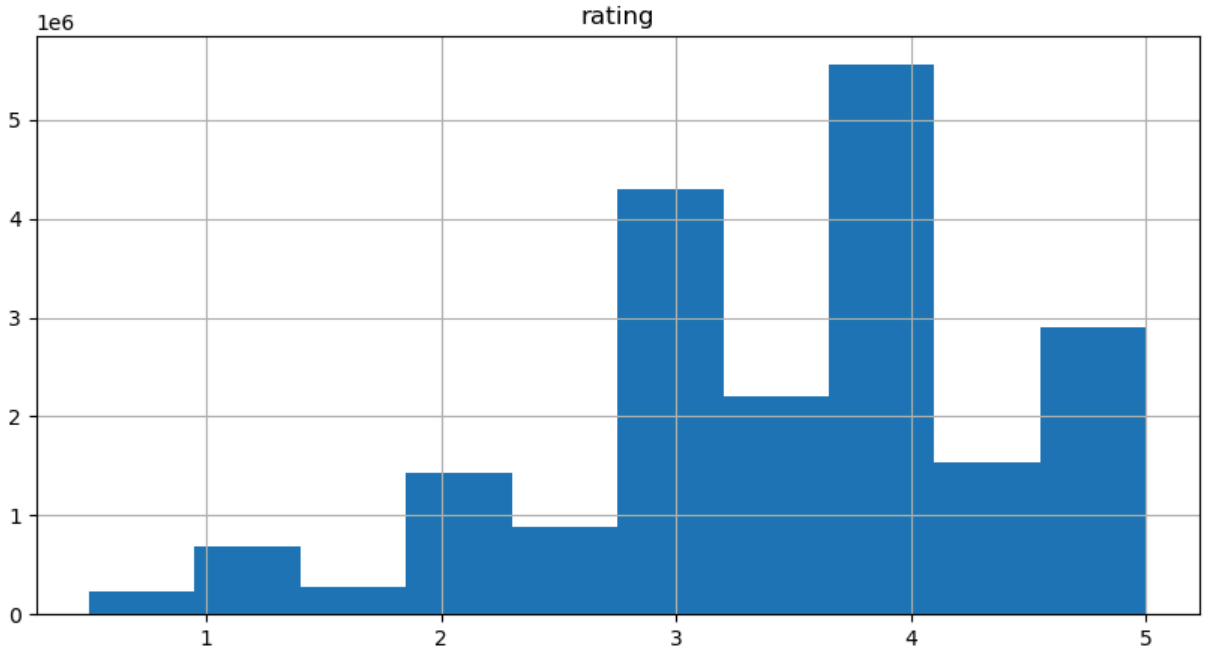
histogram -->hist()

In [61]:
```
ratings.head()
```

Out[61]:

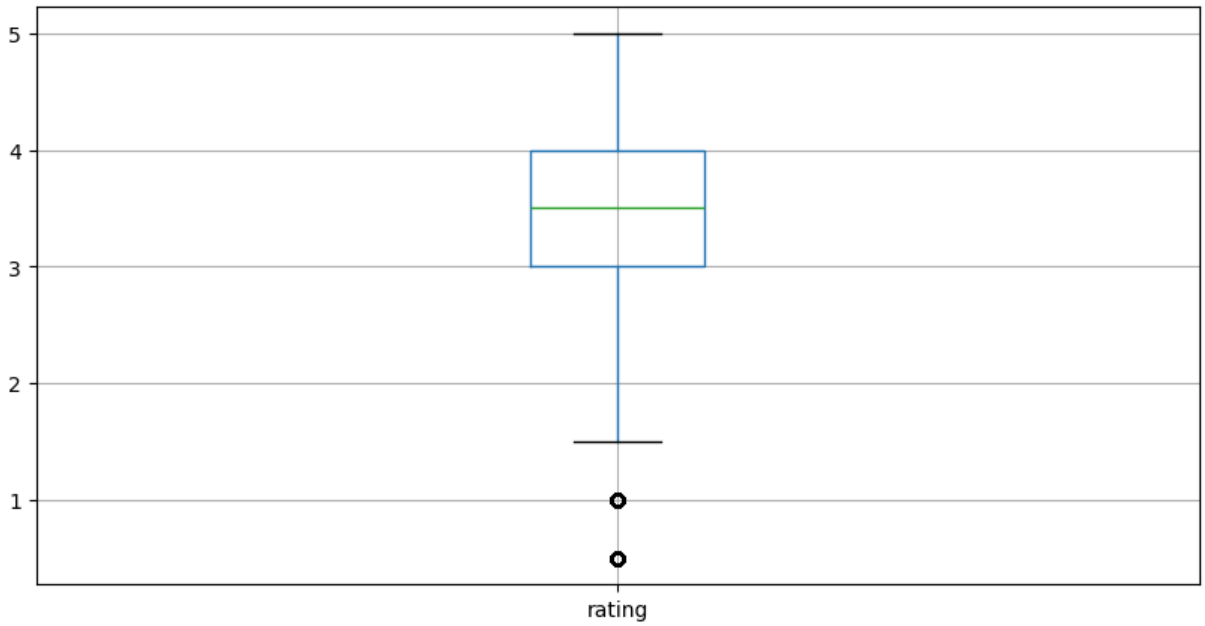
	userId	movieId	rating
0	1	2	3.5
1	1	29	3.5
2	1	32	3.5
3	1	47	3.5
4	1	50	3.5

In [62]:
```
ratings.hist(column='rating', figsize=(10,5))
plt.show()
```



Box plot

In [63]:
```
ratings.boxplot(column='rating', figsize=(10,5))
plt.show()
```



Slicing out columns

In [64]:
```
tags['tag'].head()
```

Out[64]:
```
0      Mark Waters
1        dark hero
2        dark hero
3    noir thriller
4        dark hero
Name: tag, dtype: object
```

In [65]:
```
movies.columns
```

Out[65]:
```
Index(['movieId', 'title', 'genres'], dtype='object')
```

In [66]:
```
movies[['title','genres']].head()
```

Out[66]:

	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

Slicing out Rows

In [67]:
```
ratings[-10:]
```

Out[67]:

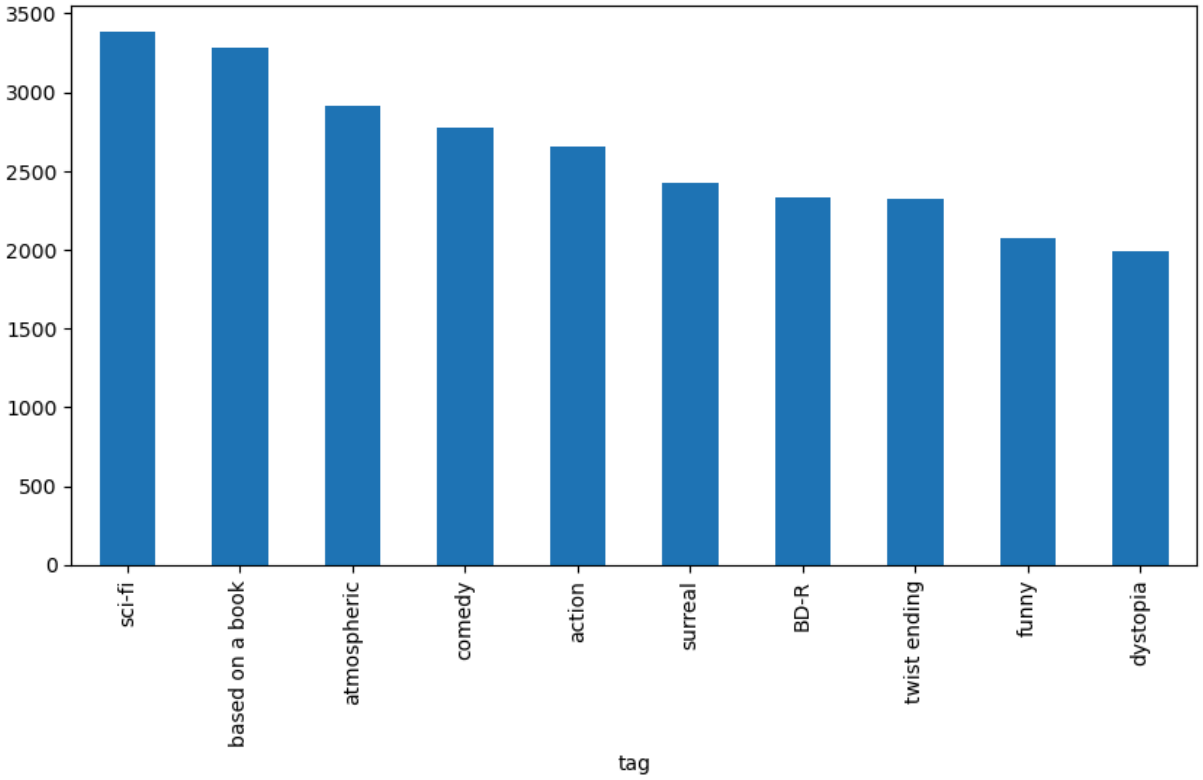
	userId	movieId	rating
20000253	138493	60816	4.5
20000254	138493	61160	4.0
20000255	138493	65682	4.5
20000256	138493	66762	4.5
20000257	138493	68319	4.5
20000258	138493	68954	4.5
20000259	138493	69526	4.5
20000260	138493	69644	3.0
20000261	138493	70286	5.0
20000262	138493	71619	2.5

value_counts() ----> count the unique values in a dataframe

In [68]:
```
tag_counts=tags['tag'].value_counts()
tag_counts
```

Out[68]:
```
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 [69]:
```
tag_counts[:10].plot(kind='bar', figsize=(10,5))
plt.show()
```



Filter for selecting rows

In [70]:
```
is_highly_rated= ratings['rating']>=5
ratings[is_highly_rated]
```

Out[70]:

	userId	movieId	rating
131	1	4993	5.0
142	1	5952	5.0
158	1	7153	5.0
170	1	8507	5.0
176	2	62	5.0
...	...	...	...
20000230	138493	48780	5.0
20000244	138493	55269	5.0
20000245	138493	55814	5.0
20000251	138493	59784	5.0
20000261	138493	70286	5.0

2898660 rows × 3 columns

In [71]:
```
ratings[is_highly_rated][30:50]
```

Out[71]:

	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 [72]:
```
is_action=movies['genres'].str.contains('Action')
movies[is_action]
```

Out[72]:

	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
...	...	...	...
27168	130842	Power/Rangers (2015)	Action|Adventure|Sci-Fi
27187	130984	Santo vs. las lobas (1976)	Action|Fantasy|Horror
27198	131025	The Brass Legend (1956)	Action
27236	131122	Love Exposure (2007)	Action|Comedy|Drama|Romance
27264	131180	Dead Rising: Watchtower (2015)	Action|Horror|Thriller

3520 rows × 3 columns

In [73]:
```
movies[is_action][5:15]
```

Out[73]:

	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 [74]:
```
movies[is_action].head(15)
```

Out[74]:

	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 [75]:
```
rating_count=ratings.groupby('rating').count()
rating_count
```

Out[75]:

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

In [76]:
```
average_rating=ratings[['movieId','rating']].groupby('movieId').mean()
average_rating.head()
```

Out[76]:

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

In [77]:
```
movie_count=ratings[['movieId','rating']].groupby('movieId').count()
movie_count.head()
```

Out[77]:

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

In [78]:
```
movie_count=ratings[['movieId','rating']].groupby('movieId').count()
movie_count.tail()
```

Out[78]:

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

Merge Dataframes

In [79]:
```
tags.head()
```

Out[79]:

	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 [80]:
```
movies.head()
```

Out[80]:

	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

merge()

In [81]:
```
t= movies.merge(tags, on='movieId', how='inner')
t.head()
```

Out[81]:

	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

Combine aggregation, merging and filters

In [82]:
```
# by default as_index is True means movieId acts as index
avg_ratings=ratings.groupby('movieId').mean()
del avg_ratings['userId']
avg_ratings.head()
```

Out[82]:

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

In [83]:
```
# here, as_index=False means movieId doesn't act as index and it gives seperate ind
avg_ratings=ratings.groupby('movieId', as_index=False).mean()
del avg_ratings['userId']
avg_ratings
```

Out[83]:

	movieId	rating
0	1	3.921240
1	2	3.211977
2	3	3.151040
3	4	2.861393
4	5	3.064592
...	...	...
26739	131254	4.000000
26740	131256	4.000000
26741	131258	2.500000
26742	131260	3.000000
26743	131262	4.000000

26744 rows × 2 columns

In [84]:
```
box_office=movies.merge(avg_ratings, on='movieId',how='inner')
box_office.tail()
```

Out[84]:

	movieId	title	genres	rating
26739	131254	Kein Bund für's Leben (2007)	Comedy	4.0
26740	131256	Feuer, Eis & Dosenbier (2002)	Comedy	4.0
26741	131258	The Pirates (2014)	Adventure	2.5
26742	131260	Rentun Ruusu (2001)	(no genres listed)	3.0
26743	131262	Innocence (2014)	Adventure|Fantasy|Horror	4.0

In [85]:
```
is_highly_rated=box_office['rating'] >= 4.0
box_office[is_highly_rated][-5:]
```

Out[85]:

	movieId	title	genres	rating
26737	131250	No More School (2000)	Comedy	4.0
26738	131252	Forklift Driver Klaus: The First Day on the Jo...	Comedy|Horror	4.0
26739	131254	Kein Bund für's Leben (2007)	Comedy	4.0
26740	131256	Feuer, Eis & Dosenbier (2002)	Comedy	4.0
26743	131262	Innocence (2014)	Adventure|Fantasy|Horror	4.0

In [86]:
```
is_Adventure=box_office['genres'].str.contains('Adventure')
box_office[is_Adventure]
```

Out[86]:

	movieId	title	genres	rating
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy	3.921240
1	2	Jumanji (1995)	Adventure|Children|Fantasy	3.211977
7	8	Tom and Huck (1995)	Adventure|Children	3.142049
9	10	GoldenEye (1995)	Action|Adventure|Thriller	3.430029
12	13	Balto (1995)	Adventure|Animation|Children	3.272416
...	...	...	...	...
26683	131084	Hui Buh: The Castle Ghost (2006)	Adventure|Comedy|Fantasy	2.500000
26687	131092	Mickey, Donald, Goofy: The Three Musketeers (2...	Adventure|Animation|Children|Comedy	3.000000
26736	131248	Brother Bear 2 (2006)	Adventure|Animation|Children|Comedy|Fantasy	4.000000
26741	131258	The Pirates (2014)	Adventure	2.500000
26743	131262	Innocence (2014)	Adventure|Fantasy|Horror	4.000000

2287 rows × 4 columns

In [87]:
```
box_office[is_Adventure & is_highly_rated]
```

Out[87]:

	movieId	title	genres	rating
257	260	Star Wars: Episode IV - A New Hope (1977)	Action|Adventure|Sci-Fi	4.190672
593	599	Wild Bunch, The (1969)	Adventure|Western	4.004726
708	720	Wallace & Gromit: The Best of Aardman Animatio...	Adventure|Animation|Comedy	4.109473
891	908	North by Northwest (1959)	Action|Adventure|Mystery|Romance|Thriller	4.233538
952	969	African Queen, The (1951)	Adventure|Comedy|Romance|War	4.101558
...	...	...	...	...
26611	130586	Itinerary of a Spoiled Child (1988)	Adventure|Drama	4.500000
26655	130996	The Beautiful Story (1992)	Adventure|Drama|Fantasy	5.000000
26667	131050	Stargate SG-1 Children of the Gods - Final Cut...	Adventure|Sci-Fi|Thriller	5.000000
26736	131248	Brother Bear 2 (2006)	Adventure|Animation|Children|Comedy|Fantasy	4.000000
26743	131262	Innocence (2014)	Adventure|Fantasy|Horror	4.000000

113 rows × 4 columns