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

In [2]:
```
rating=pd.read_csv(r"F:\Data science\kAGGLE pROJECT\archive\rating.csv")
```

In [3]:
```
rating.shape
```

Out[3]: (20000263, 4)

In [4]:
```
movie=pd.read_csv(r"F:\Data science\kAGGLE pROJECT\archive\movie.csv")
```

In [5]:
```
movie.shape
```

Out[5]: (27278, 3)

In [6]:
```
tag=pd.read_csv(r"F:\Data science\kAGGLE pROJECT\archive\tag.csv")
```

In [7]:
```
tag.shape
```

Out[7]: (465564, 4)

In [8]:
```
tag.head()
```

Out[8]:

	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 [9]:
```
# del rating["timestamp"]
# del tag["timestamp"]
```

In [10]:
```
rating
```

Out[10]:

	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 [11]:
```
row_0=tag.iloc[0]
type(row_0)
```

Out[11]:
```
pandas.core.series.Series
```

In [12]:
```
print(row_0)
```

```
userId                      18
movieId                   4141
tag                Mark Waters
timestamp    2009-04-24 18:19:40
Name: 0, dtype: object
```

In [13]:
```
row_0.index
```

Out[13]:
```
Index(['userId', 'movieId', 'tag', 'timestamp'], dtype='object')
```

In [14]:
```
tag["tag"].head()
```

Out[14]:
```
0      Mark Waters
1        dark hero
2        dark hero
3    noir thriller
4        dark hero
Name: tag, dtype: object
```

In [15]:
```
movie[["title","genres"]].head()
```

Out[15]:

	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 [16]:
```
row_0["userId"]
```

Out[16]:
```
18
```

In [17]:
```
"rating" in row_0
```

Out[17]:
```
False
```

In [18]:
```
"tag" in row_0
```

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

In [19]:
```
row_0.name
```

Out[19]:
```
0
```

In [20]:
```
row_0=row_0.rename("First row")
row_0.name
```

Out[20]:
```
'First row'
```

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

Out[21]:

	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 [22]:
```
tag.index
```

Out[22]:
```
RangeIndex(start=0, stop=465564, step=1)
```

In [23]:
```
tag.columns
```

Out[23]:
```
Index(['userId', 'movieId', 'tag', 'timestamp'], dtype='object')
```

Descriptive Statistics

In [25]:
```
tag.iloc[[0,11,500]]  #Display 3 record wich index are 0,11,500
```

Out[25]:

	userId	movieId	tag	timestamp
0	18	4141	Mark Waters	2009-04-24 18:19:40
11	65	1783	noir thriller	2013-05-10 01:39:43
500	342	55908	entirely dialogue	2012-01-31 18:41:16

In [26]:
```
rating["rating"].describe()
```

Out[26]:
```
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 [27]:
```
rating.describe()
```

Out[27]:

	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 [28]:
```
rating["rating"].mean()
```

Out[28]:
```
3.5255285642993797
```

In [29]:
```
numeric_data=rating[["userId","movieId","rating"]]
```

In [30]:
```
# rating.mean()
numeric_data.mean()
```

Out[30]:
```
userId     69045.872583
movieId     9041.567330
rating         3.525529
dtype: float64
```

In [31]:
```
rating["rating"].min()
```

Out[31]:
```
0.5
```

In [32]:
```
rating["rating"].max()
```

Out[32]:
```
5.0
```

In [33]:
```
rating["rating"].std()
```

Out[33]:
```
1.051988919275684
```

In [34]:
```
rating["rating"].mode()
```

Out[34]:
```
0    4.0
Name: rating, dtype: float64
```

In [35]:
```
numeric_data.corr()  # Find out Correlation
```

Out[35]:

	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 [36]:
```
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[36]:
```
False
```

In [37]:
```
filter2=rating["rating"]>0
filter2.all()
```

Out[37]:
```
True
```

Data cleaning

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

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

In [40]:
```
movie.isnull().any()  #Series of false value coumn wise
```

Out[40]:
```
movieId    False
title      False
genres     False
dtype: bool
```

In [41]:
```
movie.isnull().any().any() #if data is not missing then it returns false
```

Out[41]:
```
False
```

-> Thats nice no null value

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

Out[43]:
```
(20000263, 4)
```

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

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

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

Out[45]:
```
True
```

In tag table we have so null fiels

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

Out[47]:
```
(465564, 4)
```

In [48]:
```
tag=tag.dropna()  # It removes null value records
```

In [49]:
```
tag.shape
```

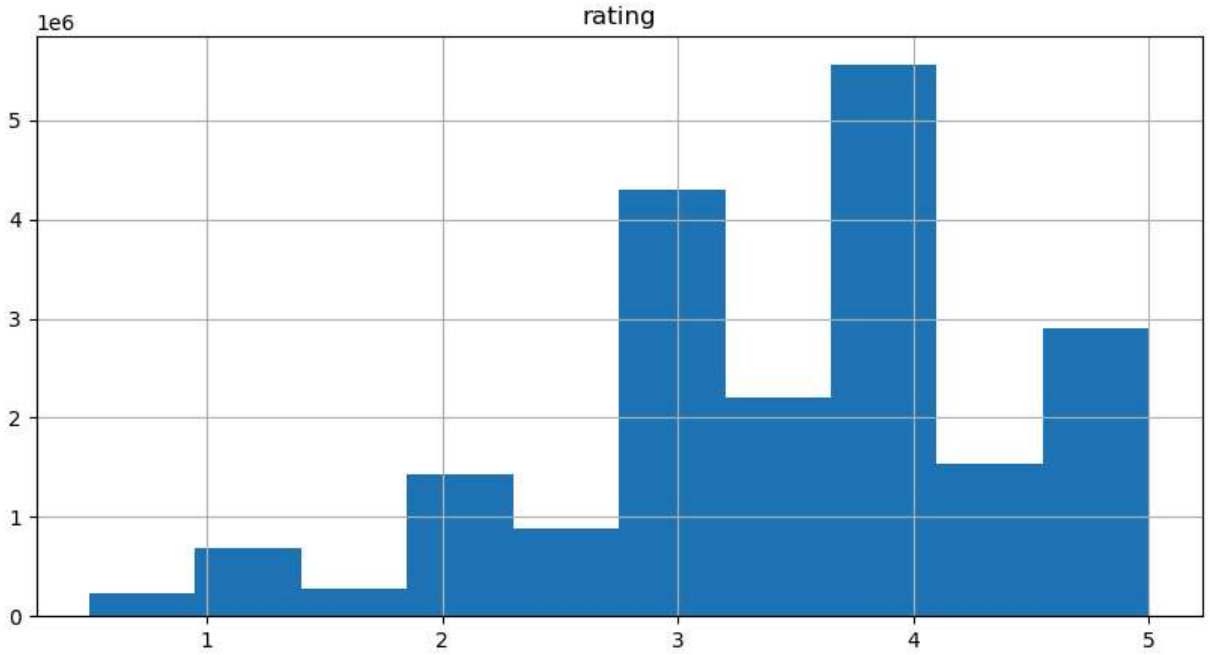
Out[49]:
```
(465548, 4)
```

In [50]:
```
tag.isnull().any().any()
```

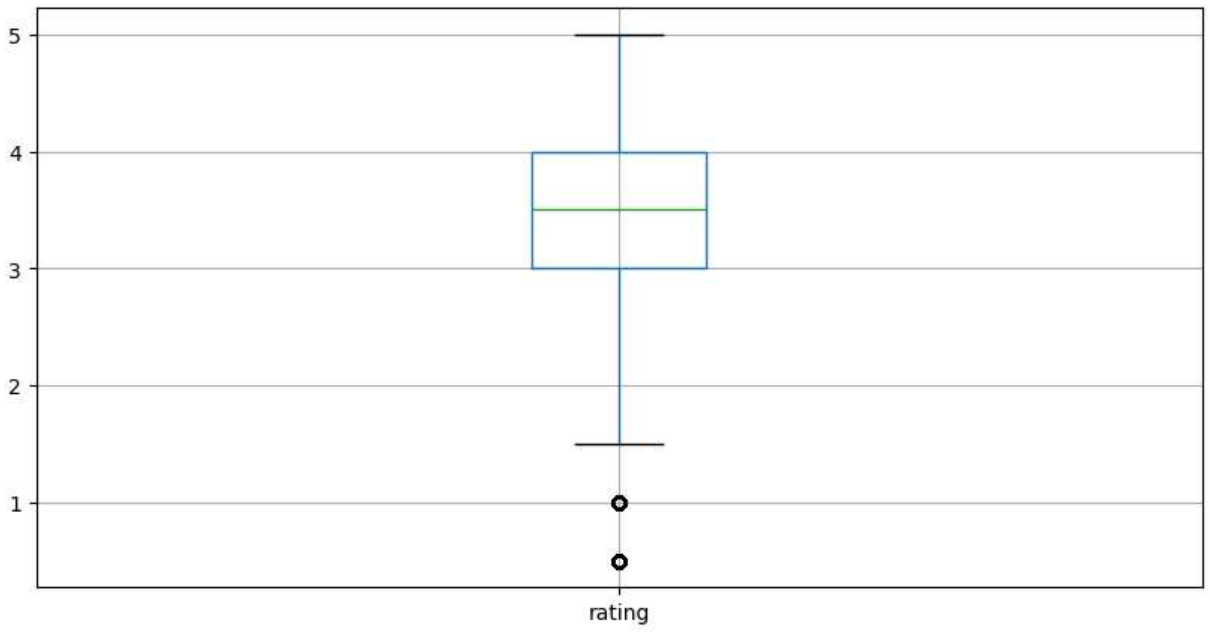
Out[50]:
```
False
```

Data Visualization

In [52]:
```
import matplotlib.pyplot as plt
%matplotlib inline
rating.hist(column="rating",figsize=(10,5))
plt.show()
```



In [53]:
```
%matplotlib inline
rating.boxplot(column="rating",figsize=(10,5))
plt.show()
```



Slicing out columns

In [55]:
```
tag["tag"].head()
```

Out[55]:
```
0      Mark Waters
1        dark hero
2        dark hero
3    noir thriller
4        dark hero
Name: tag, dtype: object
```

In [56]:
```
movie.columns
```

Out[56]:
```
Index(['movieId', 'title', 'genres'], dtype='object')
```

In [57]:
```
movie[["title","genres"]].head()
```

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

In [58]:
```
rating[-10:]  #show last 10 record from bottum
```

Out[58]:

	userId	movieId	rating	timestamp
20000253	138493	60816	4.5	2009-12-03 18:32:43
20000254	138493	61160	4.0	2009-11-16 16:55:37
20000255	138493	65682	4.5	2009-10-17 21:52:53
20000256	138493	66762	4.5	2009-10-17 18:50:08
20000257	138493	68319	4.5	2009-12-07 18:15:20
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

In [59]:
```
tag
```

Out[59]:

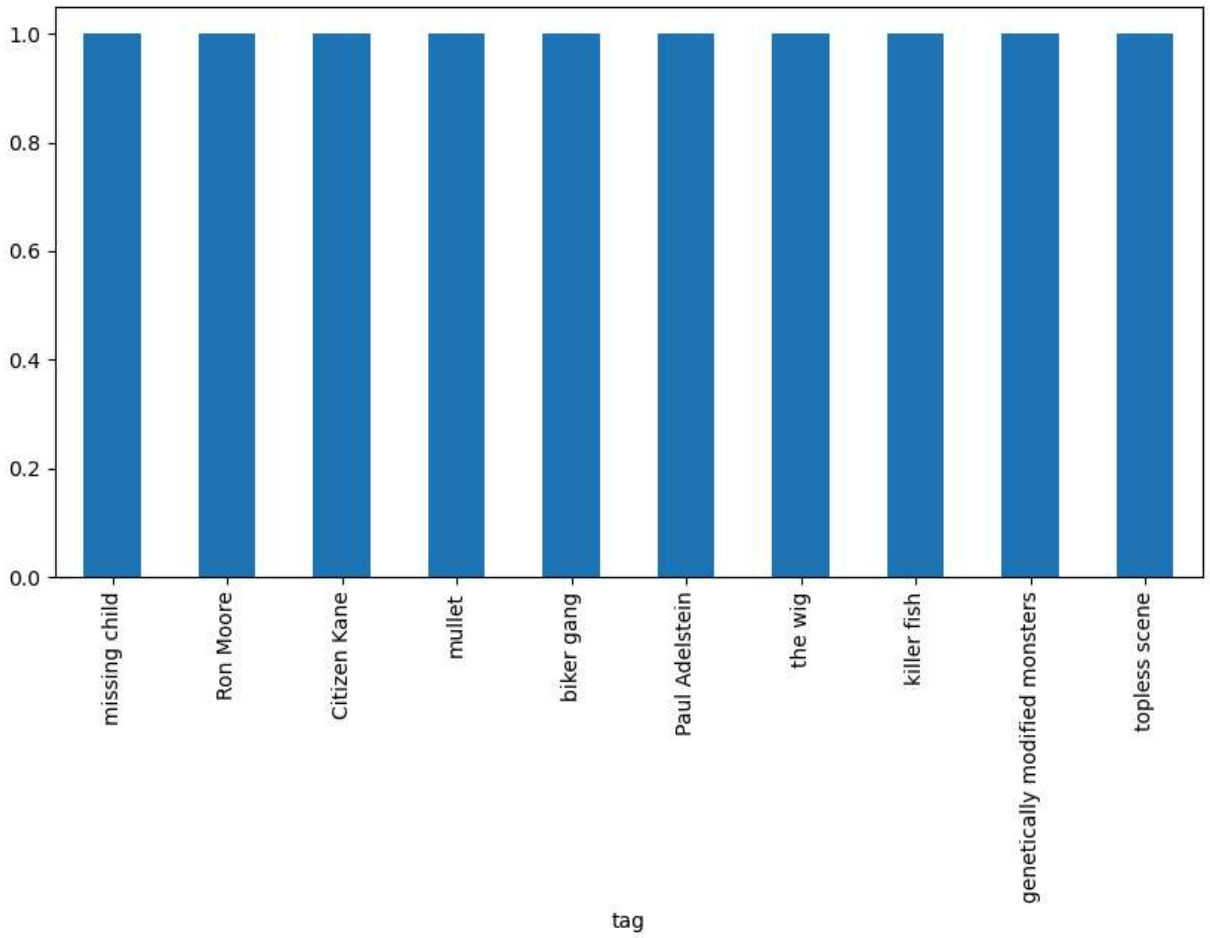
	userId	movieId	tag	timestamp
0	18	4141	Mark Waters	2009-04-24 18:19:40
1	65	208	dark hero	2013-05-10 01:41:18
2	65	353	dark hero	2013-05-10 01:41:19
3	65	521	noir thriller	2013-05-10 01:39:43
4	65	592	dark hero	2013-05-10 01:41:18
...	...	...	...	...
465559	138446	55999	dragged	2013-01-23 23:29:32
465560	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

465548 rows × 4 columns

In [60]:
```
tag_count=tag["tag"].value_counts()#It display number of occurance value
tag_count[-10:]  #Display last 10 occurance value of least frequent
```

Out[60]:
```
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 [61]:
```
tag_count[-10:].plot(kind="bar",figsize=(10,5))
plt.show()
```



Filters for Selecting Rows

In [63]:
```
is_highly_rated = rating['rating']>=5 # 5.0 #return those rating >=5.0
# is_highly_rated[30:50]
rating[is_highly_rated][30:50]  #it display record from 30-50 records
```

Out[63]:

	userId	movieId	rating	timestamp
239	3	50	5.0	1999-12-11 13:13:38
242	3	175	5.0	1999-12-11 13:32:13
244	3	223	5.0	1999-12-11 13:20:44
245	3	260	5.0	1999-12-11 13:09:02
246	3	316	5.0	1999-12-14 12:51:10
247	3	318	5.0	1999-12-11 13:09:26
248	3	329	5.0	1999-12-14 12:53:41
252	3	457	5.0	1999-12-11 13:16:55
253	3	480	5.0	1999-12-14 12:50:20
254	3	490	5.0	1999-12-11 13:30:41
256	3	541	5.0	1999-12-11 13:14:07
258	3	593	5.0	1999-12-11 13:16:55
263	3	858	5.0	1999-12-11 13:01:07
264	3	904	5.0	1999-12-11 13:04:10
267	3	924	5.0	1999-12-11 13:10:12
268	3	953	5.0	1999-12-11 13:11:52
271	3	1060	5.0	1999-12-11 13:18:30
272	3	1073	5.0	1999-12-11 13:20:44
275	3	1084	5.0	1999-12-11 13:15:03
276	3	1089	5.0	1999-12-11 13:05:58

In [64]:
```
movie[5:15]
```

Out[64]:

	movieId	title	genres
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

In [65]:
```
is_action=movie["genres"].str.contains("Action")
movie[is_action][5:15]
```

Out[65]:

	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 [66]:
```
movie[is_action].head(15)
```

Out[66]:

	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 [68]:
```
rating_count=rating[["movieId","rating"]].groupby("rating").count()
rating_count
```

Out[68]:

rating	movieId
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 [69]:
```
average_rating=rating[["movieId","rating"]].groupby("movieId").mean()
average_rating.head()
```

Out[69]:

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

In [70]:
```
movie_count = rating[['movieId','rating']].groupby('movieId').count()
movie_count.head()
```

Out[70]:

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

In [71]:
```
movie_count = rating[['movieId','rating']].groupby('movieId').count()
movie_count.tail()
```

Out[71]:

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

Merge Dataframes

In [73]:
```
tag.head()
```

Out[73]:

	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 [74]:
```
movie.head()
```

Out[74]:

	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 [75]:
```
movie.columns
```

Out[75]:
```
Index(['movieId', 'title', 'genres'], dtype='object')
```

In [76]:
```
tag.columns
```

Out[76]:
```
Index(['userId', 'movieId', 'tag', 'timestamp'], dtype='object')
```

In [77]:
```
t = movie.merge(tag, on='movieId', how='inner')
t.head()
```

Out[77]:

	movieId	title	genres	userId	tag	timestam
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy	1644	Watched	2014-1 04 23:44:4
1	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy	1741	computer animation	2007-0 08 13:59:1
2	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy	1741	Disney animated feature	2007-0 08 22:21:4
3	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy	1741	Pixar animation	2007-0 08 22:46:1
4	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy	1741	TÃ©a Leoni does not star in this movie	2009-0 15 19:19:3

📚 Combine aggreagation, merging, and filters to get useful analytics

In [79]:
```
rating
```

Out[79]:

	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 [80]:
```
avg_rating=rating.groupby("movieId",as_index=False)["rating"].mean()
avg_rating.head()
```

Out[80]:

	movieId	rating
0	1	3.921240
1	2	3.211977
2	3	3.151040
3	4	2.861393
4	5	3.064592

In [81]:
```
box_office = movie.merge(avg_rating, on='movieId', how='inner')
box_office.tail()
```

Out[81]:

	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 [82]:
```
is_highly_rated = box_office['rating'] >= 4.0
box_office[is_highly_rated][-5:]
```

Out[82]:

	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 [83]:
```
is_Adventure = box_office['genres'].str.contains('Adventure')
box_office[is_Adventure][:5]
```

Out[83]:

	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

In [84]:
```
box_office[is_Adventure & is_highly_rated][-5:]
```

Out[84]:

	movieId	title	genres	rating
26611	130586	Itinerary of a Spoiled Child (1988)	Adventure|Drama	4.5
26655	130996	The Beautiful Story (1992)	Adventure|Drama|Fantasy	5.0
26667	131050	Stargate SG-1 Children of the Gods - Final Cut...	Adventure|Sci-Fi|Thriller	5.0
26736	131248	Brother Bear 2 (2006)	Adventure|Animation|Children|Comedy|Fantasy	4.0
26743	131262	Innocence (2014)	Adventure|Fantasy|Horror	4.0

📝 Vectorized String Operations

In [86]:
```
movie.head()
```

Out[86]:

	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

Split 'genres' into multiple columns

In [88]:
```
movie_genres=movie["genres"].str.split("|", expand=True)
movie_genres.head(10)
```

Out[88]:

	0	1	2	3	4	5	6	7	8	9
0	Adventure	Animation	Children	Comedy	Fantasy	None	None	None	None	None
1	Adventure	Children	Fantasy	None	None	None	None	None	None	None
2	Comedy	Romance	None	None	None	None	None	None	None	None
3	Comedy	Drama	Romance	None	None	None	None	None	None	None
4	Comedy	None	None	None	None	None	None	None	None	None
5	Action	Crime	Thriller	None	None	None	None	None	None	None
6	Comedy	Romance	None	None	None	None	None	None	None	None
7	Adventure	Children	None	None	None	None	None	None	None	None
8	Action	None	None	None	None	None	None	None	None	None
9	Action	Adventure	Thriller	None	None	None	None	None	None	None

In [89]:
```
movie_genres["isComedy"]=movie["genres"].str.contains("Comedy")
movie_genres.head()
```

Out[89]:

	0	1	2	3	4	5	6	7	8	9	isCo
0	Adventure	Animation	Children	Comedy	Fantasy	None	None	None	None	None	
1	Adventure	Children	Fantasy	None	None	None	None	None	None	None	
2	Comedy	Romance	None	None	None	None	None	None	None	None	
3	Comedy	Drama	Romance	None	None	None	None	None	None	None	
4	Comedy	None	None	None	None	None	None	None	None	None	

In [90]:
```
movie_genres["isComedy"]=movie_genres["isComedy"].replace({True:"Comedy",False:"No
movie_genres.head()
```

Out[90]:

	0	1	2	3	4	5	6	7	8	9	isCo
0	Adventure	Animation	Children	Comedy	Fantasy	None	None	None	None	None	C
1	Adventure	Children	Fantasy	None	None	None	None	None	None	None	C
2	Comedy	Romance	None	None	None	None	None	None	None	None	C
3	Comedy	Drama	Romance	None	None	None	None	None	None	None	C
4	Comedy	None	None	None	None	None	None	None	None	None	C

In [91]:
```
movie["year"]=movie["title"].str.extract(r'.*\((.*)\).*', expand=True)
movie
```

Out[91]:

	movieId	title	genres	year
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy	1995
1	2	Jumanji (1995)	Adventure|Children|Fantasy	1995
2	3	Grumpier Old Men (1995)	Comedy|Romance	1995
3	4	Waiting to Exhale (1995)	Comedy|Drama|Romance	1995
4	5	Father of the Bride Part II (1995)	Comedy	1995
...	...	...	...	...
27273	131254	Kein Bund für's Leben (2007)	Comedy	2007
27274	131256	Feuer, Eis & Dosenbier (2002)	Comedy	2002
27275	131258	The Pirates (2014)	Adventure	2014
27276	131260	Rentun Ruusu (2001)	(no genres listed)	2001
27277	131262	Innocence (2014)	Adventure|Fantasy|Horror	2014

27278 rows × 4 columns

In [92]:
```
tag.dtypes
```

Out[92]:
```
userId        int64
movieId       int64
tag          object
timestamp    object
dtype: object
```

In [93]:
```
tag.head()
```

Out[93]:

	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 [175…
```
tag['timestamp']=pd.to_datetime(tag['timestamp'])
```

In [179…
```
tag['sparse_time']=(tag['timestamp']-pd.Timestamp("1990-01-01"))//pd.Timedelta("1s"
```

In [181…
```
tag.head()
```

Out[181…

	userId	movieId	tag	timestamp	sparse_time
0	18	4141	Mark Waters	2009-04-24 18:19:40	609445180
1	65	208	dark hero	2013-05-10 01:41:18	736998078
2	65	353	dark hero	2013-05-10 01:41:19	736998079
3	65	521	noir thriller	2013-05-10 01:39:43	736997983
4	65	592	dark hero	2013-05-10 01:41:18	736998078

In [189…
```
tag.dtypes
```

Out[189…
```
userId                int64
movieId               int64
tag                  object
timestamp      datetime64[ns]
sparse_time           int64
dtype: object
```

In [193…
```
tag.sort_values(by='timestamp',ascending=True).head()
```

Out[193…

	userId	movieId	tag	timestamp	sparse_time
333932	100371	2788	monty python	2005-12-24 13:00:10	504277210
333927	100371	1732	coen brothers	2005-12-24 13:00:36	504277236
333924	100371	1206	stanley kubrick	2005-12-24 13:00:48	504277248
333923	100371	1193	jack nicholson	2005-12-24 13:02:51	504277371
333939	100371	5004	peter sellers	2005-12-24 13:03:19	504277399

In [195…

```
# Calculate average rating per movie
average_rating = rating[['movieId', 'rating']].groupby('movieId', as_index=False).m

print(average_rating.tail())  # Last 5 rows
```

```
       movieId  rating
26739   131254     4.0
26740   131256     4.0
26741   131258     2.5
26742   131260     3.0
26743   131262     4.0
```

In [199…

```
movie.head()
```

Out[199…

	movieId	title	genres	year
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy	1995
1	2	Jumanji (1995)	Adventure|Children|Fantasy	1995
2	3	Grumpier Old Men (1995)	Comedy|Romance	1995
3	4	Waiting to Exhale (1995)	Comedy|Drama|Romance	1995
4	5	Father of the Bride Part II (1995)	Comedy	1995

In [201…

```
joined=movie.merge(average_rating, on="movieId",how="inner")
joined
```

Out[201…

	movieId	title	genres	year	rating
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy	1995	3.921240
1	2	Jumanji (1995)	Adventure|Children|Fantasy	1995	3.211977
2	3	Grumpier Old Men (1995)	Comedy|Romance	1995	3.151040
3	4	Waiting to Exhale (1995)	Comedy|Drama|Romance	1995	2.861393
4	5	Father of the Bride Part II (1995)	Comedy	1995	3.064592
...	...	...	...	...	...
26739	131254	Kein Bund für's Leben (2007)	Comedy	2007	4.000000
26740	131256	Feuer, Eis & Dosenbier (2002)	Comedy	2002	4.000000
26741	131258	The Pirates (2014)	Adventure	2014	2.500000
26742	131260	Rentun Ruusu (2001)	(no genres listed)	2001	3.000000
26743	131262	Innocence (2014)	Adventure|Fantasy|Horror	2014	4.000000

26744 rows × 5 columns

In [ ]: