Import Libraries

In [2]:
```
import pandas as pd
```

Read the Dataset

In [4]:
```
movies=pd.read_csv(r'C:\Users\HP\Downloads\archive\movie.csv')
```

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

Out[5]:
```
(27278, 3)
```

In [6]:
```
type(movies)
```

Out[6]:
```
pandas.core.frame.DataFrame
```

In [7]:
```
movies.head
```

Out[7]:
```
<bound method NDFrame.head of        movieId                                title
\
0            1                    Toy Story (1995)
1            2                      Jumanji (1995)
2            3             Grumpier Old Men (1995)
3            4            Waiting to Exhale (1995)
4            5  Father of the Bride Part II (1995)
...        ...                                  ...
27273   131254        Kein Bund für's Leben (2007)
27274   131256       Feuer, Eis & Dosenbier (2002)
27275   131258                  The Pirates (2014)
27276   131260                 Rentun Ruusu (2001)
27277   131262                    Innocence (2014)

                                       genres
0      Adventure|Animation|Children|Comedy|Fantasy
1                       Adventure|Children|Fantasy
2                                   Comedy|Romance
3                            Comedy|Drama|Romance
4                                          Comedy
...                                           ...
27273                                      Comedy
27274                                      Comedy
27275                                   Adventure
27276                         (no genres listed)
27277                      Adventure|Fantasy|Horror

[27278 rows x 3 columns]>
```

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

Out[8]:

	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 [9]:
```
movies.head(20)
```

Out[9]:

	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 [10]:
```
movies.tail
```

Out[10]:
```
<bound method NDFrame.tail of        movieId                               title
\
0            1                    Toy Story (1995)
1            2                      Jumanji (1995)
2            3             Grumpier Old Men (1995)
3            4            Waiting to Exhale (1995)
4            5  Father of the Bride Part II (1995)
...        ...                                 ...
27273   131254        Kein Bund für's Leben (2007)
27274   131256       Feuer, Eis & Dosenbier (2002)
27275   131258                  The Pirates (2014)
27276   131260                 Rentun Ruusu (2001)
27277   131262                    Innocence (2014)

                                          genres
0      Adventure|Animation|Children|Comedy|Fantasy
1                       Adventure|Children|Fantasy
2                                   Comedy|Romance
3                             Comedy|Drama|Romance
4                                           Comedy
...                                            ...
27273                                       Comedy
27274                                       Comedy
27275                                    Adventure
27276                          (no genres listed)
27277                       Adventure|Fantasy|Horror

[27278 rows x 3 columns]>
```

In [11]:
```
movies.tail()
```

Out[11]:

	movieId	title	genres
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

In [12]:
```
movies.tail(20)
```

Out[12]:

	movieId	title	genres
27258	131166	WWII IN HD (2009)	(no genres listed)
27259	131168	Phoenix (2014)	Drama
27260	131170	Parallels (2015)	Sci-Fi
27261	131172	Closed Curtain (2013)	(no genres listed)
27262	131174	Gentlemen (2014)	Drama|Romance|Thriller
27263	131176	A Second Chance (2014)	Drama
27264	131180	Dead Rising: Watchtower (2015)	Action|Horror|Thriller
27265	131231	Standby (2014)	Comedy|Romance
27266	131237	What Men Talk About (2010)	Comedy
27267	131239	Three Quarter Moon (2011)	Comedy|Drama
27268	131241	Ants in the Pants (2000)	Comedy|Romance
27269	131243	Werner - Gekotzt wird später (2003)	Animation|Comedy
27270	131248	Brother Bear 2 (2006)	Adventure|Animation|Children|Comedy|Fantasy
27271	131250	No More School (2000)	Comedy
27272	131252	Forklift Driver Klaus: The First Day on the Jo...	Comedy|Horror
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

In [13]:
```
tags=pd.read_csv(r'C:\Users\HP\Downloads\archive\tag.csv')
```

In [14]:
```
tags.shape
```

Out[14]:
```
(465564, 4)
```

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

Out[15]:
```
<bound method NDFrame.head of         userId  movieId           tag            ti
mestamp
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

[465564 rows x 4 columns]>
```

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

Out[16]:

	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 [17]:
```
tags.tail()
```

Out[17]:

	userId	movieId	tag	timestamp
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

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

In [19]:
```
ratings.shape
```

Out[19]:
```
(20000263, 4)
```

In [75]:
```
type(ratings)
```

Out[75]:
```
pandas.core.frame.DataFrame
```

In [20]:
```
ratings.head()
```

Out[20]:

	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 [21]:
```
ratings.tail()
```

Out[21]:

	userId	movieId	rating	timestamp
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 [77]:
```
del ratings['timestamp']
del tags['timestamp']
```

Data Structures:

- Series

In [24]:
```
row_0 = tags.iloc[0]
type(row_0)
```

Out[24]:
```
pandas.core.series.Series
```

In [25]:
```
print(row_0)
```

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

In [26]:
```
row_0.index
```

Out[26]:
```
Index(['userId', 'movieId', 'tag', 'timestamp'], dtype='object')
```

In [27]:
```
row_0['userId']
```

Out[27]:
```
18
```

In [28]:
```
'rating' in row_0
```

Out[28]:
```
False
```

In [29]:
```
row_0.name
```

Out[29]:
```
0
```

In [30]:
```
row_0 = row_0.rename('firstRow')
row_0.name
```

Out[30]:
```
'firstRow'
```

DataFrames

In [32]:
```
tags.head()
```

Out[32]:

	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 [33]:
```
tags.index
```

Out[33]:
```
RangeIndex(start=0, stop=465564, step=1)
```

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

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

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

Out[35]:

	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

Descriptive Statistics

- Let's look how the ratings are distributed!

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

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

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

In [79]:
```
ratings.min()
```

Out[79]:
```
userId     1.0
movieId    1.0
rating     0.5
dtype: float64
```

In [97]:
```
ratings.mean()
```

Out[97]:
```
userId     69045.872583
movieId     9041.567330
rating         3.525529
dtype: float64
```

In [81]:
```
ratings['rating'].min()
```

Out[81]:
```
0.5
```

In [83]:
```
ratings.max()
```

Out[83]:
```
userId     138493.0
movieId    131262.0
rating          5.0
dtype: float64
```

In [85]:
```
ratings['rating'].max()
```

Out[85]:
```
5.0
```

In [87]:
```
ratings.std()
```

Out[87]:
```
userId     40038.626653
movieId    19789.477445
rating         1.051989
dtype: float64
```

In [89]:
```
ratings['rating'].std()
```

Out[89]:
```
1.051988919275684
```

In [91]:
```
ratings.mode()
```

Out[91]:

	userId	movieId	rating
0	118205	296	4.0

In [93]:
```
ratings['rating'].mode()
```

Out[93]:
```
0    4.0
Name: rating, dtype: float64
```

In [95]:
```
ratings.corr() # correlation
```

Out[95]:

	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 [99]:
```
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[99]: False

In [101…
```
filter2 = ratings['rating'] > 0
filter2.all()
```

Out[101… True

Data Cleaning: Handling Missing Data

In [104…
```
movies.shape
```

Out[104… (27278, 3)

In [106…
```
movies.isnull().any().any()  # No NULL values !
```

Out[106… False

In [108…
```
ratings.shape
```

Out[108… (20000263, 3)

In [110…
```
ratings.isnull().any().any()      # No NULL values !
```

Out[110… False

In [112…
```
tags.shape
```

Out[112… (465564, 3)

In [114…
```
tags.isnull().any().any()    # We have some tags which are NULL.
```

Out[114… True

In [116…
```
tags=tags.dropna()
```

In [118…
```
tags.isnull().any().any()    # No NULL values ! Notice the number of lines have red
```

Out[118… False

In [120…
```
tags.shape
```

Out[120… (465548, 3)

Data Visualization

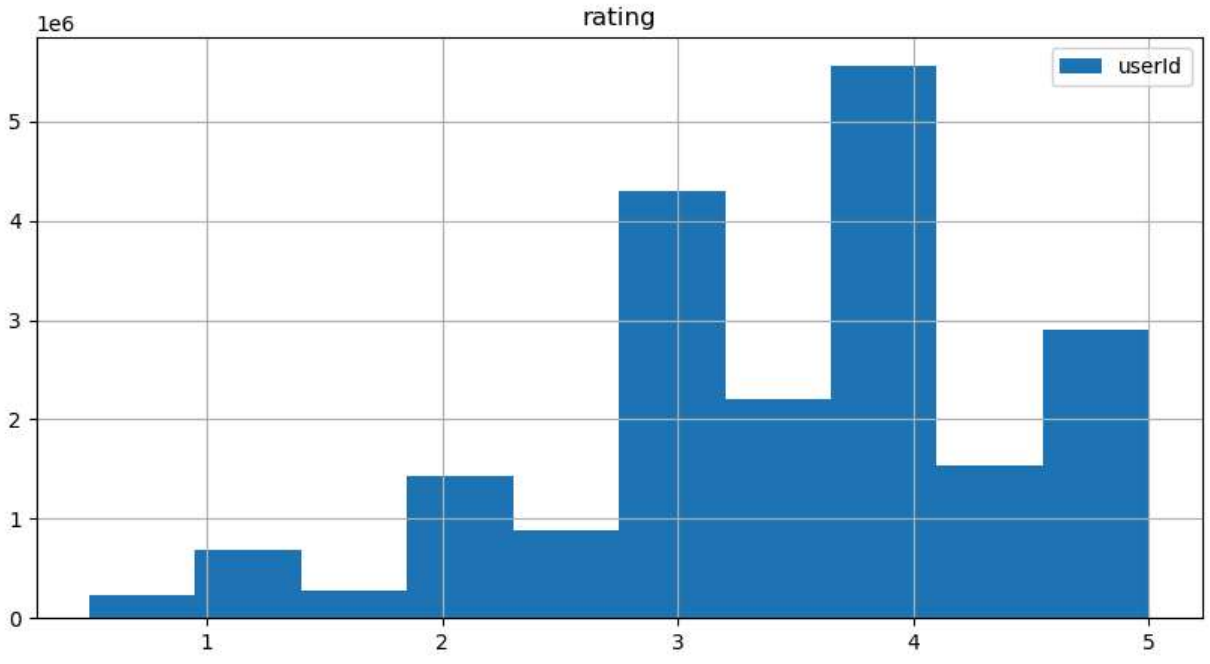
In [123…
```
import matplotlib.pyplot as plt
```

In [125…
```
import numpy as np
```

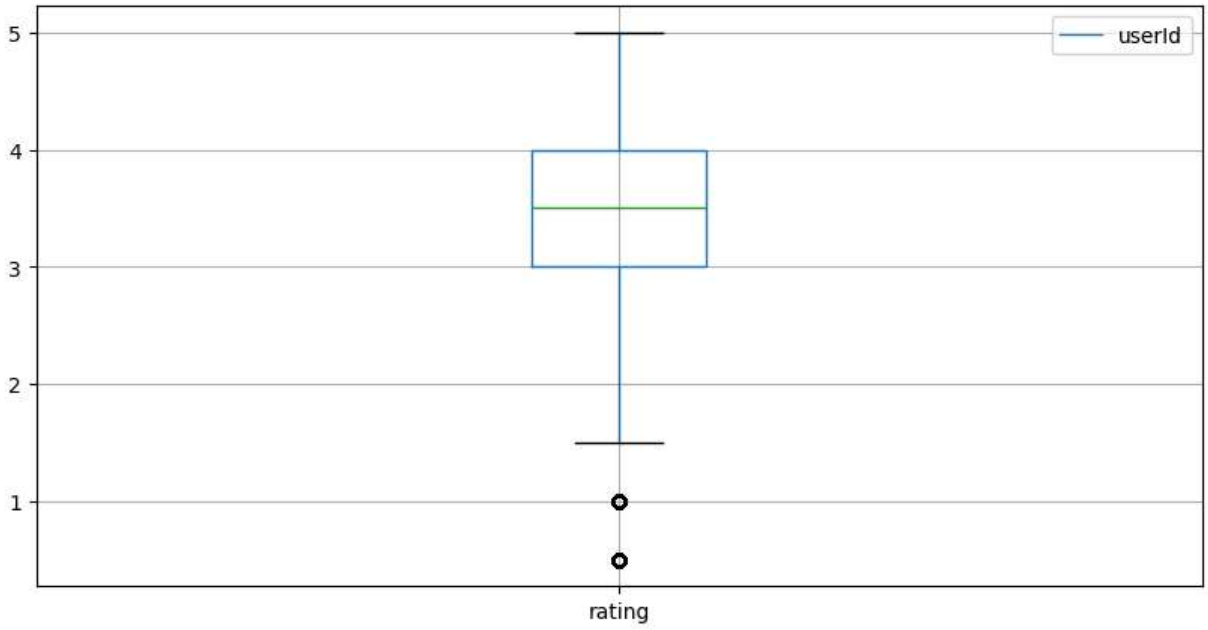
In [127…
```
import cycler
```

In [154…
```
import warnings
warnings.filterwarnings('ignore')
```

In [129…
```
%matplotlib inline
ratings.hist(column='rating', figsize=(10,5))
plt.legend(ratings)
plt.show()
```



In [131…
```
ratings.boxplot(column='rating', figsize=(10,5))
plt.legend(ratings)
plt.show()
```



Slicing Out Columns

In [136…
```
tags.head()
```

Out[136…

	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 [134…
```
tags['tag'].head()
```

Out[134…
```
0      Mark Waters
1        dark hero
2        dark hero
3    noir thriller
4        dark hero
Name: tag, dtype: object
```

In [138…
```
movies.head()
```

Out[138…

	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 [140…
```
movies[['title','genres']].head()
```

Out[140…

	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 [142…
```
ratings.head()
```

Out[142…

	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 [144…
```
ratings[['movieId','rating']].head()
```

Out[144…

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

In [146…
```
ratings[-10:]
```

Out[146…

	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 [148…

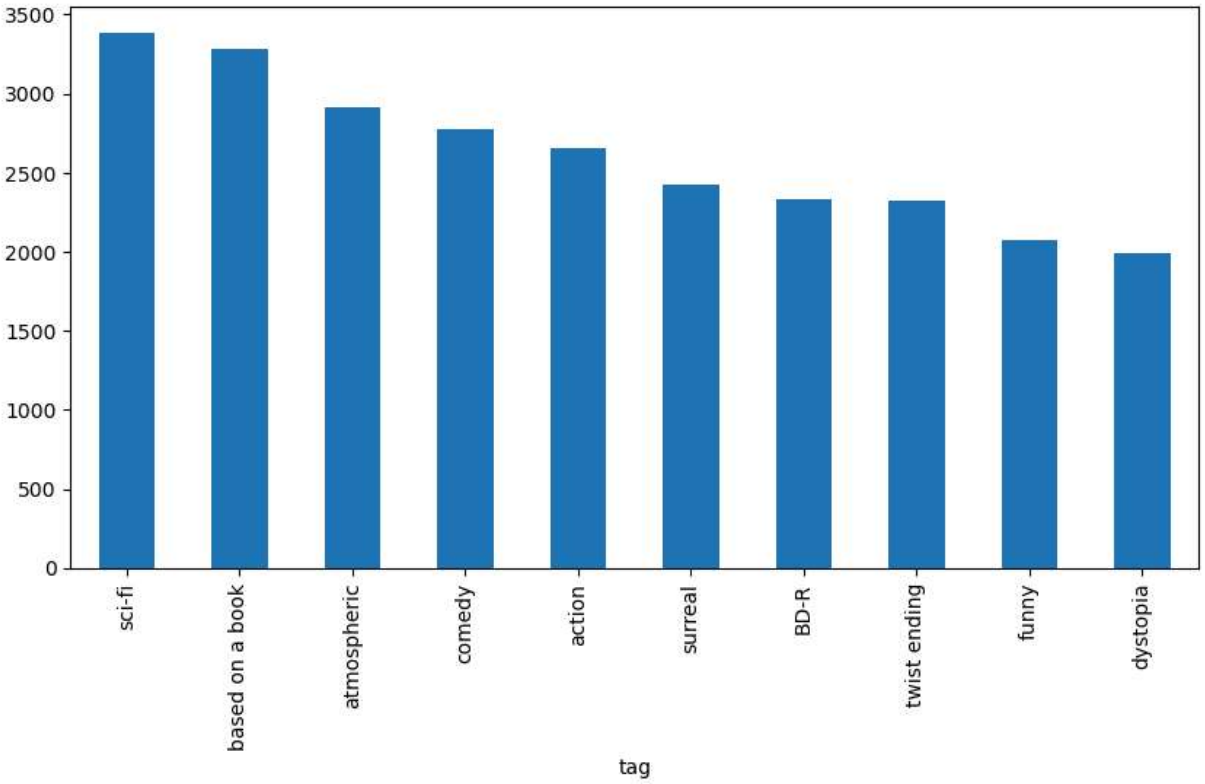
```
tag_counts = tags['tag'].value_counts()
tag_counts[-10:]
```

Out[148…

```
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 [150…

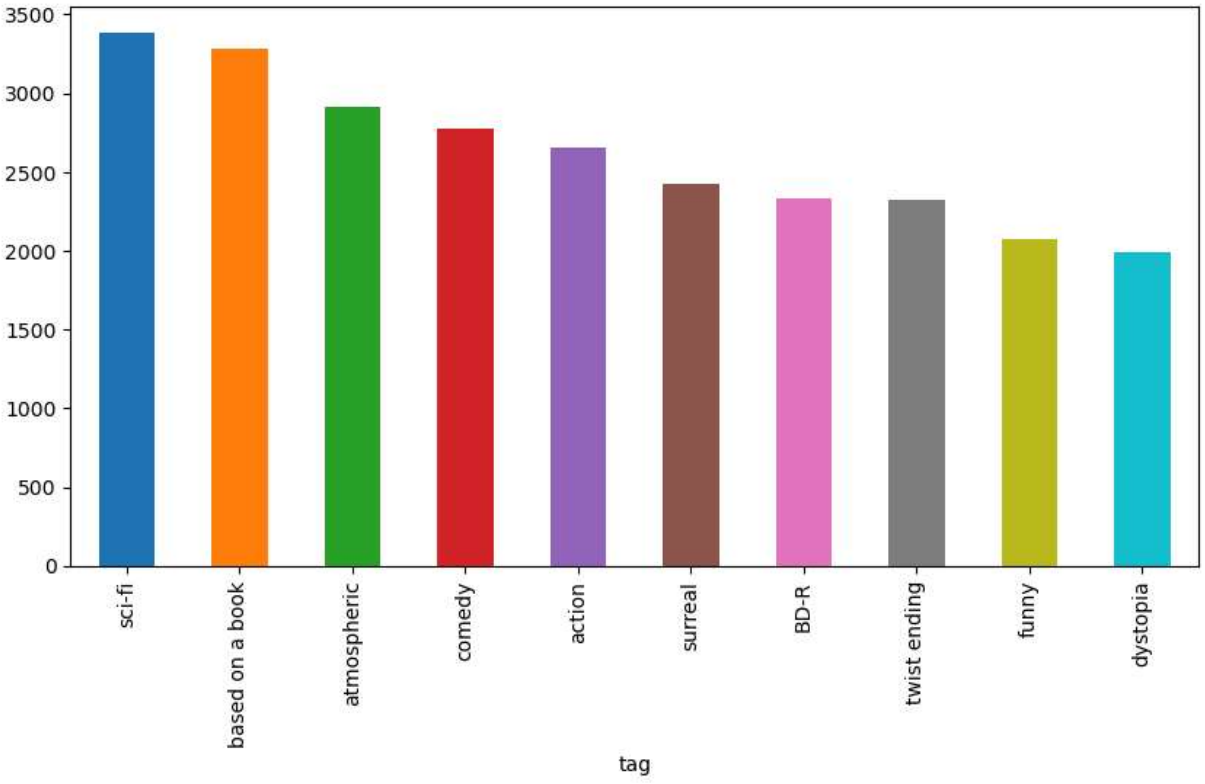
```
tag_counts[:10].plot(kind='bar', figsize=(10,5))
plt.show()
```



In [158…

```
# Generate a list of distinct colors
colors = plt.cm.get_cmap('tab10').colors
colors = colors[:10]

# Plot the bar chart with custom colors
tag_counts[:10].plot(kind='bar', figsize=(10, 5), color=colors)
plt.show()
```



Filters for Selecting Rows

In [164…
```
is_highly_rated = ratings['rating'] >= 5.0
ratings[is_highly_rated][30:50]
```

Out[164…

	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 [166…
```
is_action= movies['genres'].str.contains('Action')
movies[is_action][5:15]
```

Out[166…

	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 [168…

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

Out[168…

	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 [170…
```
ratings_count = ratings[['movieId','rating']].groupby('rating').count()
ratings_count
```

Out[170…

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 [172…
```
average_rating = ratings[['movieId','rating']].groupby('movieId').mean()
average_rating.head()
```

Out[172…

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

In [174…
```
movie_count = ratings[['movieId','rating']].groupby('movieId').count()
movie_count.head()
```

Out[174…

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

In [176…
```
movie_count = ratings[['movieId','rating']].groupby('movieId').count()
movie_count.head()
```

Out[176…

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

Merge DataFrames

In [178…
```
tags.head()
```

Out[178…

	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 [180…
```
movies.head()
```

Out[180…

	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 [182…

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

Out[182…

	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

In [184…

```
t=movies.merge(tags, on='movieId', how='outer') # it includes decimal values
t.head()
```

Out[184…

	movieId	title	genres	userId	tag
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy	1644.0	Watched
1	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy	1741.0	computer animation
2	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy	1741.0	Disney animated feature
3	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy	1741.0	Pixar animation
4	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy	1741.0	TÃ©a Leoni does not star in this movie

Combine Aggregation,Merging and Filters to Get Useful Analytics

In [186…

```
avg_ratings = ratings.groupby('movieId',as_index=False).mean()
del avg_ratings['userId']
avg_ratings.head()
```

Out[186…

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

In [188…

```
box_office = movies.merge(avg_ratings, on='movieId', how='inner')
box_office.tail()
```

Out[188…

	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 [190…

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

Out[190…

	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 [192…

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

Out[192…

	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 [194…

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

Out[194...

	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 [198...

```
movies.head()
```

Out[198...

	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 [201...

```
movie_genres = movies['genres'].str.split('|', expand=True)
movie_genres[:10]
```

Out[201…

	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 [203…

```
movie_genres['isComedy'] = movies['genres'].str.contains('Comedy')
movie_genres[:10]      # Add a New Column for comedy Genre Flag
```

Out[203…

	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	
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 [205…

```
movies['year'] = movies['title'].str.extract('.*\((.*)\).*', expand=True)
movies.tail()           # Extract year from title e.g.(2007)
```

Out[205…

	movieId	title	genres	year
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

Parsing Timestamps

In [208…
```
tags=pd.read_csv(r'C:\Users\HP\Downloads\archive\tag.csv')
```

In [210…
```
tags.dtypes
```

Out[210…
```
userId        int64
movieId       int64
tag          object
timestamp    object
dtype: object
```

In [212…
```
tags.head(5)
```

Out[212…

	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 [233…
```
tags['parsed_time'] = pd.to_datetime(tags['timestamp']) #Data Type datetime64[ns] m
```

In [235…
```
tags['parsed_time'].dtype
```

Out[235…
```
dtype('<M8[ns]')
```

In [219…
```
tags.head(2)
```

Out[219…

	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

In [237…
```
greater_than_t = tags['parsed_time'] > '01-02-2015'
selected_rows = tags[greater_than_t]
tags.shape, selected_rows.shape
```

Out[237…
```
((465564, 5), (19757, 5))
```

In [239…
```
tags.sort_values(by='parsed_time', ascending=True)
```

Out[239…

	userId	movieId	tag	timestamp	parsed_time
333932	100371	2788	monty python	2005-12-24 13:00:10	2005-12-24 13:00:10
333927	100371	1732	coen brothers	2005-12-24 13:00:36	2005-12-24 13:00:36
333924	100371	1206	stanley kubrick	2005-12-24 13:00:48	2005-12-24 13:00:48
333923	100371	1193	jack nicholson	2005-12-24 13:02:51	2005-12-24 13:02:51
333939	100371	5004	peter sellers	2005-12-24 13:03:19	2005-12-24 13:03:19
…	…	…	…	…	…
290535	88044	106782	profanity	2015-03-30 22:21:36	2015-03-30 22:21:36
288375	87797	215	Vienna	2015-03-30 22:50:01	2015-03-30 22:50:01
158763	46072	3409	premonition	2015-03-31 00:12:06	2015-03-31 00:12:06
158780	46072	6058	premonition	2015-03-31 00:12:44	2015-03-31 00:12:44
339178	102853	115149	russian mafia	2015-03-31 03:09:12	2015-03-31 03:09:12

465564 rows × 5 columns

Average Movie Ratings Over Time

In [225…
```
average_rating = ratings[['movieId','rating']].groupby('movieId', as_index=False).m
average_rating.tail()
```

Out[225…

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

In [ ]:
```
```

In [ ]: