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

In [2]:
```
ratings = pd.read_csv(r'C:\AI Course Naresh\2-15-2025\New folder (4)\rating.csv'
```

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

Out[3]:
```
(20000263, 4)
```

In [4]:
```
ratings.columns
```

Out[4]:
```
Index(['userId', 'movieId', 'rating', 'timestamp'], dtype='object')
```

In [5]:
```
ratings.head(1)
```

Out[5]:

	userId	movieId	rating	timestamp
0	1	2	3.5	2005-04-02 23:53:47

In [6]:
```
tags = pd.read_csv(r'C:\AI Course Naresh\2-15-2025\New folder (4)\tag.csv')
```

In [7]:
```
tags.head(1)
```

Out[7]:

	userId	movieId	tag	timestamp
0	18	4141	Mark Waters	2009-04-24 18:19:40

In [8]:
```
movies = pd.read_csv(r'C:\AI Course Naresh\2-15-2025\New folder (4)\movie.csv')
```

In [9]:
```
movies.head(1)
```

Out[9]:

	movieId	title	genres
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy

In [10]:
```
del ratings['timestamp']
del tags['timestamp']
```

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

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

Series

In [13]:
```
tags.iloc[0]
```

Out[13]:
```
userId              18
movieId           4141
tag        Mark Waters
Name: 0, dtype: object
```

In [14]:
```
tags.iloc[1]
```

Out[14]:

```
userId           65
movieId         208
tag       dark hero
Name: 1, dtype: object
```

In [15]:

```
tags_0 = tags.iloc[0]
```

In [16]:

```
print(tags_0)
```

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

Pandas With Data Science.AI

In [18]:

```
row_0 = tags.iloc[0]
```

In [19]:

```
print(row_0)
```

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

In [20]:

```
row_0.index
```

Out[20]:

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

In [21]:

```
row_0['userId']
```

Out[21]:

```
18
```

In [22]:

```
'rating' in row_0
```

Out[22]:

```
False
```

In [23]:

```
row_0.name
```

Out[23]:

```
0
```

In [24]:

```
row_0 = row_0.rename('firstRow')
row_0
```

Out[24]:

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

DataFrames

In [26]:

```
tags.head()
```

Out[26]:

	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 [27]:

```
tags.index
```

Out[27]:

```
RangeIndex(start=0, stop=465564, step=1)
```

In [28]:

```
tags.columns
```

Out[28]:

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

In [29]:

```
tags.iloc[[11,50,100]]
```

Out[29]:

	userId	movieId	tag
11	65	1783	noir thriller
50	96	106696	writing
100	121	52973	drugs

In [30]:

```
ratings['rating'].describe()
```

Out[30]:

```
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 [31]:

```
ratings.describe()
# ratings.show()
```

Out[31]:

	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 [32]:

```
ratings['rating'].mean()
```

Out[32]:

```
3.5255285642993797
```

In [33]:

```
ratings.mean()
```

Out[33]:

```
userId     69045.872583
movieId     9041.567330
rating         3.525529
dtype: float64
```

In [34]:

```
ratings['rating'].min()
```

Out[34]:

```
0.5
```

In [35]:

```
ratings['rating'].max()
```

Out[35]:

```
5.0
```

In [36]:

```
ratings['rating'].std()
```

Out[36]:

```
1.051988919275684
```

In [37]:

```
ratings['rating'].mode()
```

Out[37]:

```
0    4.0
Name: rating, dtype: float64
```

In [38]:

```
ratings.corr()
```

Out[38]:

	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 [39]:

```
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[39]:
```
False
```

In [40]:
```
filter2 = ratings['rating'] > 0
filter2.all()
```

Out[40]:
```
True
```

In [41]:
```
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
```

Data Cleaning: Handling Missing Data

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

Out[43]:
```
(27278, 3)
```

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

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

In [45]:
```
ratings.shape
```

Out[45]:
```
(20000263, 3)
```

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

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

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

Out[47]:
```
(465564, 3)
```

In [48]:
```python
tags.isnull().any().any()
```

Out[48]:
```
True
```

In [49]:
```python
tags = tags.dropna()
```

In [50]:
```python
tags.isnull().any().any()
```

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

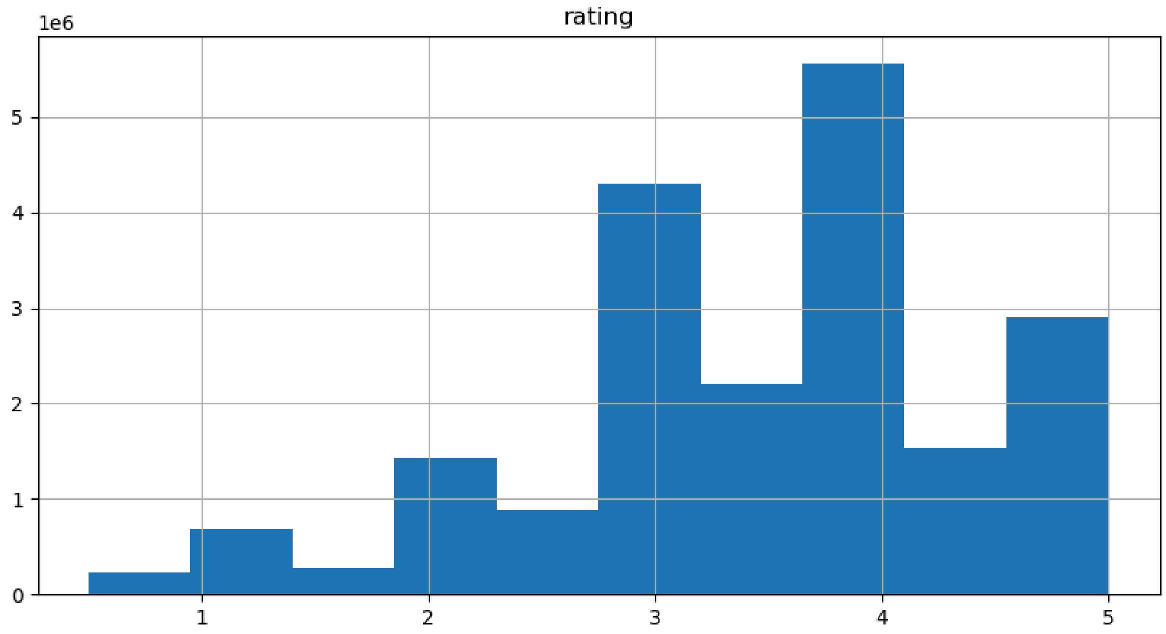
In [51]:
```python
tags.shape
```

Out[51]:
```
(465548, 3)
```

Data Visualization

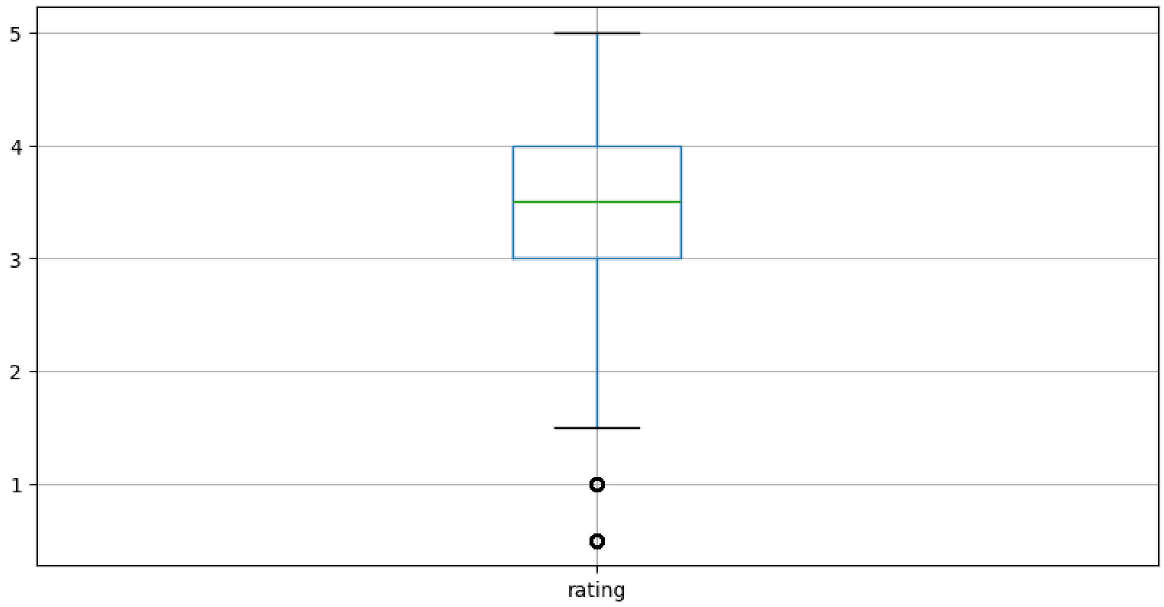
In [115…
```python
import matplotlib.pyplot as plt
```

In [119…
```python
%matplotlib inline

ratings.hist(column='rating', figsize = (10,5))

plt.show()
```



In [ ]:
```python

```

In [124…
```python
ratings.boxplot(column='rating', figsize=(10,5))
plt.show()
```



In [128…

```
tags.columns
```

Out[128…

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

In [127…

```
tags
```

Out[127…

	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 [131…

```
tags['tag'].head()
```

Out[131…

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

In [133…

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

Out[133…

	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 [ ]:

```
ratings
```

In [141…

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

Out[141…

```
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 [147…

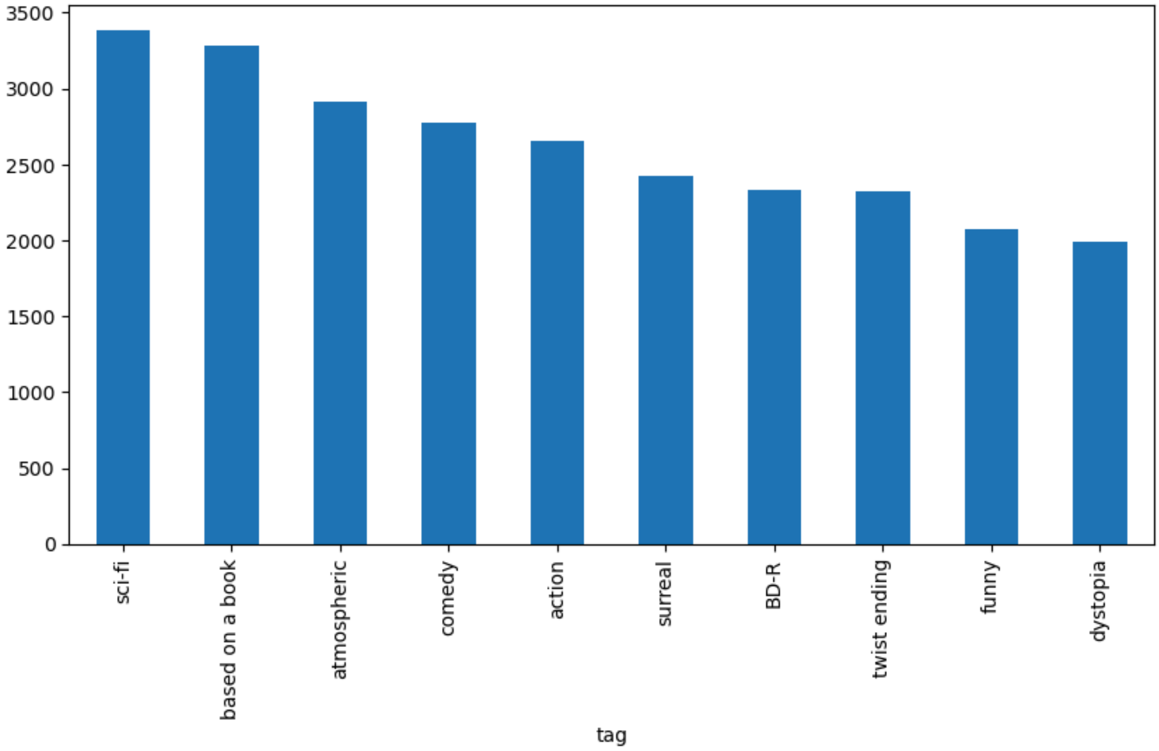
```
tags['tag'].tail(10)
```

Out[147…

```
465554                   visually appealing
465555                      family friendly
465556    Scary Movies To See on Halloween
465557                            Peter Pan
465558                   visually appealing
465559                              dragged
465560                        Jason Bateman
465561                               quirky
465562                                  sad
465563                        rise to power
Name: tag, dtype: object
```

In [155…

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


3500
3000
2500
2000
1500
1000
500
0
sci-fi
based on a book
atmospheric
comedy
action
surreal
BD-R
twist ending
funny
dystopia
tag