

Import Libraries

```
In [1]: import pandas as pd
import matplotlib.pyplot as plt
```

Read the Dataset

```
In [2]: movies = pd.read_csv(r'C:\Users\Ramya\Downloads\archive\movie.csv')
```

```
print(type(movies))
```

```
In [3]: movies.head(20) #print 1st 20 rows
```

```
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
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 [4]: tags = pd.read_csv(r"C:\Users\Ramya\Downloads\archive\tag.csv")
tags.head() #print 1st 5 rows and col
```

```
Out[4]:
```

	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 [5]: ratings = pd.read_csv(r"C:\Users\Ramya\Downloads\archive\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]: del ratings['timestamp']
del tags['timestamp']
```

Data Structures:

Series

```
In [7]: row_0=tags.iloc[0] #prints index location
type(row_0)
```

```
Out[7]: pandas.core.series.Series
```

```
In [8]: print(row_0)
```

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

```
In [9]: row_0.index
```

```
Out[9]: Index(['userId', 'movieId', 'tag'], dtype='object')
```

```
In [10]: row_0['userId']
```

```
Out[10]: 18
```

```
In [11]: 'rating' in row_0
```

```
Out[11]: False
```

```
In [12]: row_0.name
```

```
Out[12]: 0
```

```
In [13]: row_0=row_0.rename('firstRow')
row_0.name
```

```
Out[13]: 'firstRow'
```

DataFrames

```
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]: tags.index
```

```
Out[15]: RangeIndex(start=0, stop=465564, step=1)
```

```
In [16]: tags.columns
```

```
Out[16]: Index(['userId', 'movieId', 'tag'], dtype='object')
```

```
In [17]: tags.iloc[ [0,11,500] ] #printing location 0,11,500 only
```

```
Out[17]:
```

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

Descriptive Statistics

how the ratings are distributed!

```
In [18]: ratings['rating'].describe()
```

```
Out[18]: 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 [19]: ratings.describe()
```

```
Out[19]:
```

	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 [20]: ratings.mean()
```

```
Out[20]: userId      69045.872583
         movieId     9041.567330
         rating       3.525529
         dtype: float64
```

```
In [21]: ratings['rating'].min()
```

```
Out[21]: 0.5
```

```
In [22]: ratings['rating'].max()
```

```
Out[22]: 5.0
```

```
In [23]: ratings['rating'].std() #calculates the standard deviation
```

```
Out[23]: 1.051988919275684
```

```
In [24]: ratings['rating'].mode()#rating col,the most frequently occurring value(s).
```

```
Out[24]: 0    4.0
         Name: rating, dtype: float64
```

```
In [25]: ratings.corr()#two numerical columns are linearly related
```

Out[25]:

	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 [26]: 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[26]: False

```
In [27]: filter2 = ratings['rating'] > 0
filter2.all()
```

Out[27]: True

Data Cleaning: Handling Missing Data

```
In [28]: movies.shape #how many rows & columns are there
```

Out[28]: (27278, 3)

```
In [29]: movies.isnull().any().any()
```

Out[29]: False

```
In [30]: ratings.shape
```

Out[30]: (20000263, 3)

```
In [31]: ratings.isnull().any().any()
```

Out[31]: False

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

Out[32]: (465564, 3)

```
In [33]: tags.isnull().any().any()
```

Out[33]: True

```
In [34]: tags=tags.dropna()
```

```
In [35]: tags.isnull().any().any()
```

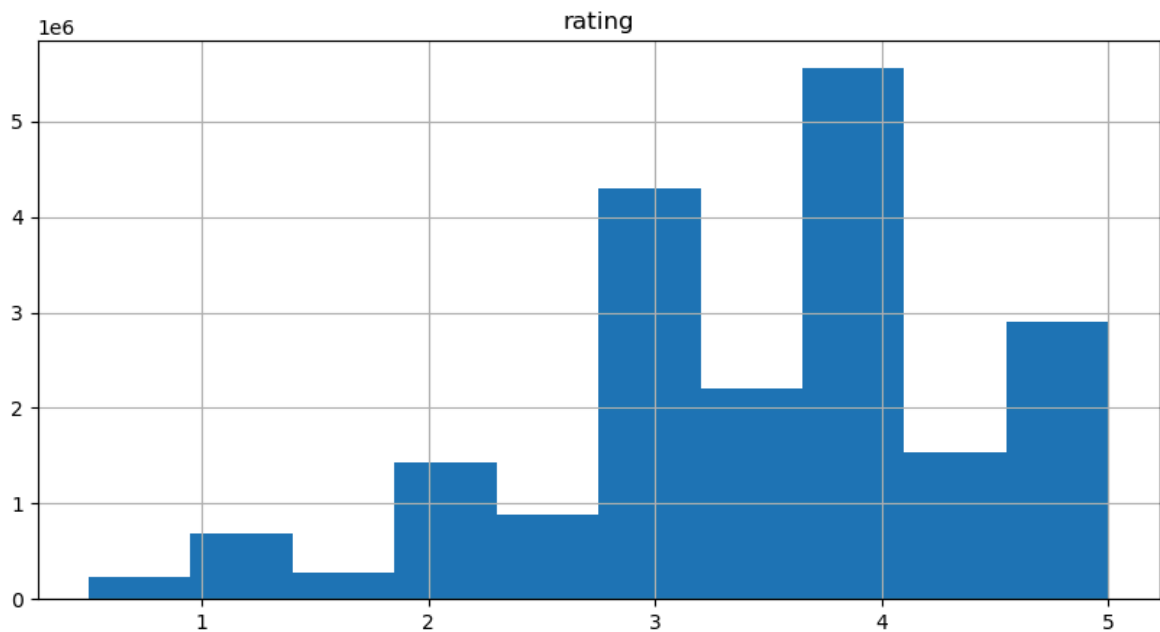
Out[35]: False

```
In [36]: tags.shape
```

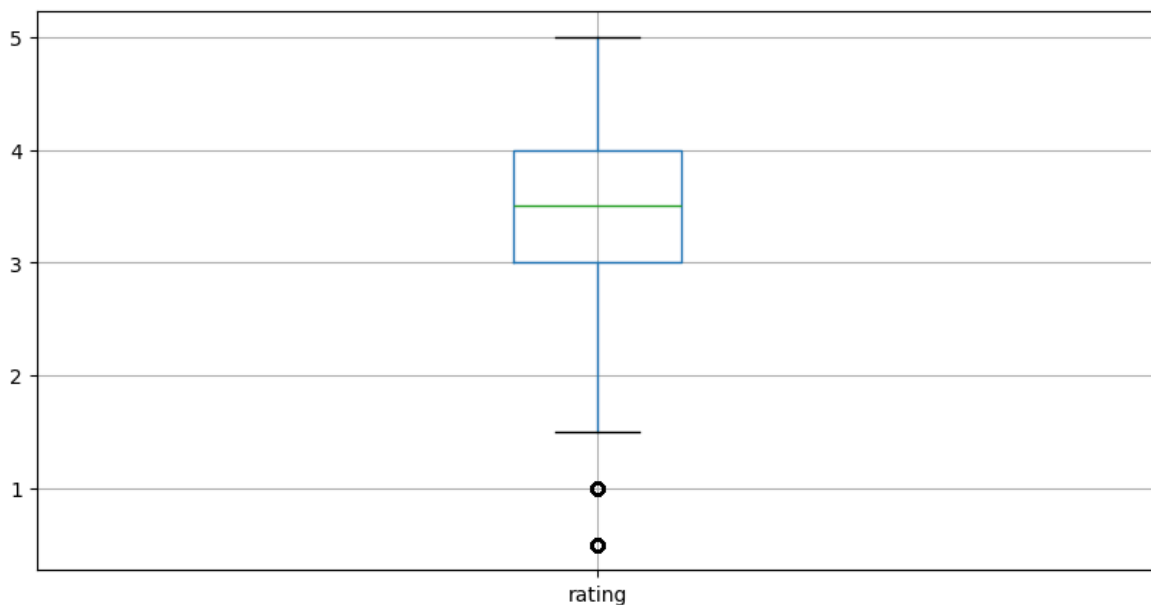
Out[36]: (465548, 3)

Data Visualization

```
In [37]: %matplotlib inline
ratings.hist(column='rating', figsize=(10,5))
plt.show()
```



```
In [38]: ratings.boxplot(column='rating', figsize=(10,5))
plt.show()
```



Slicing Out Columns

```
In [39]: tags['tag'].head()
```

```
Out[39]: 0    Mark Waters
1    dark hero
2    dark hero
3    noir thriller
4    dark hero
Name: tag, dtype: object
```

```
In [40]: movies[['title', 'genres']].head()
```

```
Out[40]:
```

	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 [41]: ratings[-10:]
```

Out[41]:

	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 [48]: `tag_counts = tags['tag'].value_counts()`In [49]: `tag_counts[-10:]`

Out[49]:

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

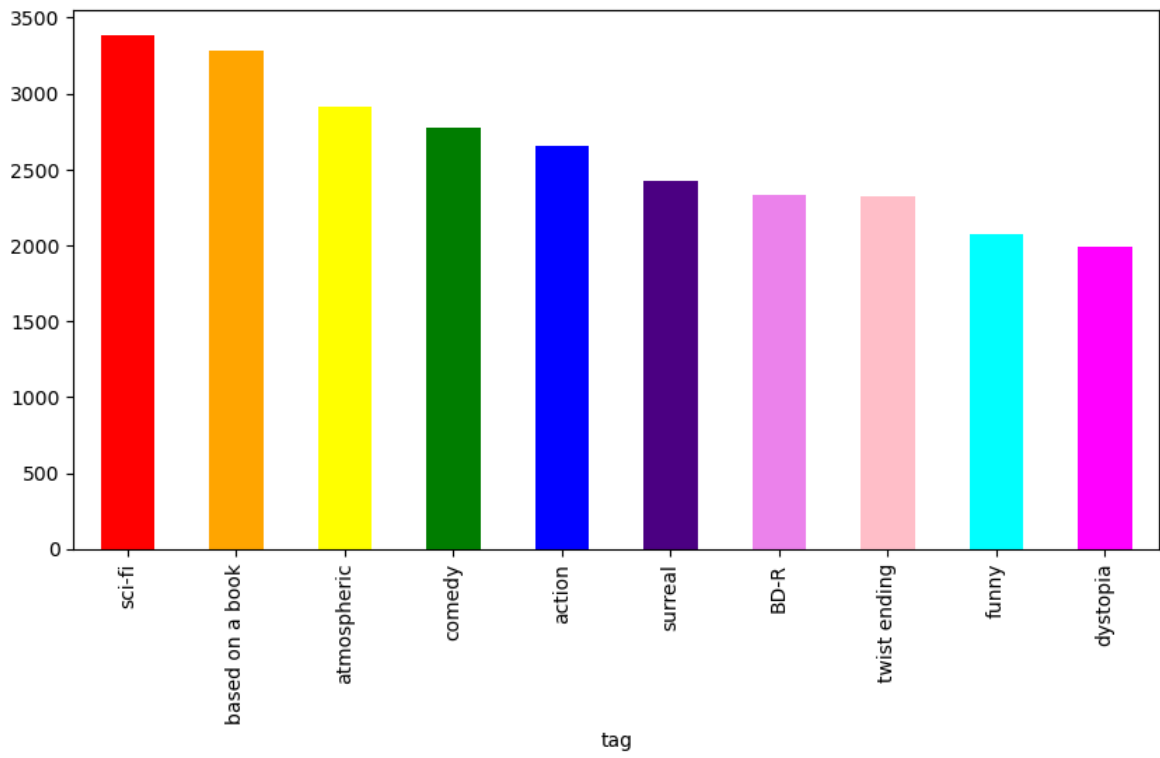
Name: count, dtype: int64

In [59]:

```

tag_counts[:10].plot(kind='bar',figsize=(10,5))
colors = ['red', 'orange', 'yellow', 'green', 'blue', 'indigo', 'violet', 'pink']
tag_counts[:10].plot(kind='bar', figsize=(10,5), color=colors)
plt.show()

```

In []:

In []: