In [31]:
```
import pandas as r
d=r.read_csv("/home/placement/Downloads/movies.csv") #reading the file into the jupyter
```

In [32]:
```
#This command is to describe the data present in the DataFrame in statistically
d.describe()
```

Out[32]:

	srno	year	rating	time
count	49590.000000	49590.000000	10814.000000	45838.000000
mean	24795.500000	2002.303428	3.451248	2628.344605
std	14315.544261	12.534555	0.495601	1604.684505
min	1.000000	1913.000000	1.400000	52.000000
25%	12398.250000	1999.000000	3.100000	1356.000000
50%	24795.500000	2007.000000	3.500000	2563.000000
75%	37192.750000	2010.000000	3.800000	2877.000000
max	49590.000000	2014.000000	4.500000	28813.000000

In [33]:
```
#The head() method returns a specified number of rows, string from the top
d.head(100)
```

Out[33]:

	srno	movie	year	rating	time
0	1	The Nightmare Before	1993	3.9	4568.0
1	2	The Mummy	1932	3.5	4388.0
2	3	Orphans of the Storm	1921	3.2	9062.0
3	4	The Object of Beauty	1991	2.8	6150.0
4	5	Night Tide	1963	2.8	5126.0
...	...	...	...	...	...
95	96	The Hunted	1995	3.4	6605.0
96	97	The Great Waldo Pepper	1975	3.5	6467.0
97	98	Godzilla: King of the Monsters	1956	3.5	4828.0
98	99	Highlander 2: Renegade Version	1991	3.1	6585.0
99	100	High Noon	1952	3.9	5087.0

100 rows × 5 columns

In [34]:
```
#The tail() method returns a specified number of rows, string from the bottom
d.tail(5)
```

Out[34]:

	srno	movie	year	rating	time
49585	49586	Winter Wonderland	2013	2.8	1812.0
49586	49587	Top Gear: Series 19: Africa Special	2013	NaN	6822.0
49587	49588	Fireplace For Your Home: Crackling Fireplace w...	2010	NaN	3610.0
49588	49589	Kate Plus Ei8ht	2010	2.7	100.0
49589	49590	Kate Plus Ei8ht: Season 1	2010	2.7	535.0

In [35]:
```
#loc[] is used to retrieve the group of rows and columns by labels
data2=d.loc[(d.time>5000) & (d.year==2000) & (d.rating>3.7)]
data2
```

Out[35]:

	srno	movie	year	rating	time
437	438	Requiem for a Dream	2000	3.9	6087.0
484	485	Coming to Light: Edward S. Curtis and the Nort...	2000	4.0	5027.0
578	579	Traffic	2000	3.8	8849.0
611	612	Mohabbatein	2000	3.8	12966.0
765	766	Memento	2000	3.8	6806.0
1837	1838	Battle Royale	2000	3.9	6832.0

In [36]:
```
#To arrange the rows in ascending order
d2=d.sort_values('year')
d2
```

Out[36]:

	srno	movie	year	rating	time
42670	42671	Fantômas II: Juve vs. Fantômas	1913	2.7	3718.0
42664	42665	Fantômas I: In the Shadow of the Guillotine	1913	2.9	3268.0
14327	14328	Fantômas III: The Murderous Corpse	1913	2.6	5432.0
42672	42673	Fantômas V: The False Magistrate	1914	2.4	4247.0
609	610	Cabiria	1914	2.9	7684.0
...	...	...	...	...	...
47906	47907	Twisted: Season 1: Three for the Road	2013	NaN	2626.0
47907	47908	Twisted: Season 1: Sleeping with the Frenemy	2013	NaN	2624.0
47901	47902	Twisted: Season 1: The Truth Will Out	2013	NaN	2627.0
47359	47360	Eve of Destruction	2013	3.3	NaN
49560	49561	The Square (Trailer)	2014	3.6	154.0

49590 rows × 5 columns

In [37]:
```
#We can count the NaN values in Pandas DataFrame using the isna() function and with the sum() function
d.isna().sum()
```

Out[37]:
```
srno          0
movie         0
year          0
rating    38776
time       3752
dtype: int64
```

In [38]:
```
#It is to count the columns and rows in the dataframe
d.shape
```

Out[38]:
```
(49590, 5)
```

In [39]:
```
#method is used to prints information about the DataFrame
d.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 49590 entries, 0 to 49589
Data columns (total 5 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   srno    49590 non-null  int64  
 1   movie   49590 non-null  object 
 2   year    49590 non-null  int64  
 3   rating  10814 non-null  float64
 4   time    45838 non-null  float64
dtypes: float64(2), int64(2), object(1)
memory usage: 1.9+ MB
```

In [40]:
```
'''groupby count method is used to count the values in each group by
ignoring the missing values or NaN values in the data frame.'''

d1=d.groupby(['year']).count()
d1
```

Out[40]:

year	srno	movie	rating	time
1913	3	3	3	3
1914	20	20	5	18
1915	1	1	1	1
1916	1	1	1	1
1918	1	1	1	1
...	...	...	...	...
2010	5107	5107	1102	4673
2011	5511	5511	1346	4992
2012	4339	4339	1130	3978
2013	981	981	345	901
2014	1	1	1	1

101 rows × 4 columns

In [41]:
```
#This is to create a duplicate file
d1.to_csv('/home/placement/duplicatefile.csv')
```

In [42]:
```
#This create a graph
d1.plot()
```

Out[42]: <Axes: xlabel='year'>