In [2]:
```
import pandas as pd#import pandas as pd sub name
```

In [3]:
```
data=pd.read_csv("/home/placement/Downloads/movies.csv")#reading a file from downloas using read command
```

In [4]:
```
data.head(10)#show top 10 rows
```

Out[4]:

	year	srno	movie	rating	time
0	1913	3	3	3	3
1	1914	20	20	5	18
2	1915	1	1	1	1
3	1916	1	1	1	1
4	1918	1	1	1	1
5	1919	3	3	3	3
6	1920	6	6	6	6
7	1921	2	2	2	2
8	1922	2	2	2	2
9	1923	4	4	4	4

In [5]:
```
data.describe()#describe count,mean,max
```

Out[5]:

	year	srno	movie	rating	time
count	101.000000	101.000000	101.000000	101.000000	101.000000
mean	1963.960396	490.990099	490.990099	107.069307	453.821782
std	29.366961	1068.727397	1068.727397	237.255864	977.783682
min	1913.000000	1.000000	1.000000	1.000000	1.000000
25%	1939.000000	7.000000	7.000000	5.000000	7.000000
50%	1964.000000	107.000000	107.000000	21.000000	101.000000
75%	1989.000000	334.000000	334.000000	79.000000	317.000000
max	2014.000000	5511.000000	5511.000000	1346.000000	4992.000000

In [6]:
```
data.isna().sum()#how many rows can empty can be seen using this command
```

Out[6]:
```
year      0
srno      0
movie     0
rating    0
time      0
dtype: int64
```

In [7]:
```
data.shape#display no of rows and columns
```

Out[7]:
```
(101, 5)
```

In [8]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 101 entries, 0 to 100
Data columns (total 5 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   year    101 non-null    int64
 1   srno    101 non-null    int64
 2   movie   101 non-null    int64
 3   rating  101 non-null    int64
 4   time    101 non-null    int64
dtypes: int64(5)
memory usage: 4.1 KB
```

In [9]:
```
data1=data.groupby(['year']).count()#how many colums can be count
```

In [10]:
```
data1
```

Out[10]:

year	srno	movie	rating	time
1913	1	1	1	1
1914	1	1	1	1
1915	1	1	1	1
1916	1	1	1	1
1918	1	1	1	1
...	...	...	...	...
2010	1	1	1	1
2011	1	1	1	1
2012	1	1	1	1
2013	1	1	1	1
2014	1	1	1	1

101 rows × 4 columns

In [11]:
```
data1.to_csv("/home/placement/Downloads/movies1.csv")
```

In [12]:
```
data1
```

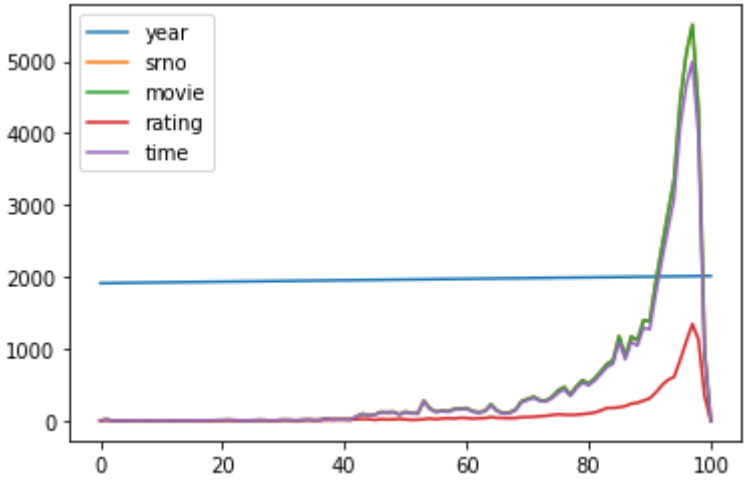
Out[12]:

year	srno	movie	rating	time
1913	1	1	1	1
1914	1	1	1	1
1915	1	1	1	1
1916	1	1	1	1
1918	1	1	1	1
...	...	...	...	...
2010	1	1	1	1
2011	1	1	1	1
2012	1	1	1	1
2013	1	1	1	1
2014	1	1	1	1

101 rows × 4 columns

In [13]:
```
data.plot()
#plotting the graph for data
```

Out[13]: <AxesSubplot:>



In [14]:
```
import warnings#remove warnings
```

In [15]:
```
warnings.filterwarnings("ignore")
```

In [16]:
```
data.tail()#display from bottom
```

Out[16]:

	year	srno	movie	rating	time
96	2010	5107	5107	1102	4671
97	2011	5511	5511	1346	4992
98	2012	4339	4339	1130	3978
99	2013	981	981	345	901
100	2014	1	1	1	1

In [17]:
```
data.info()#data type can be seen using this command
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 101 entries, 0 to 100
Data columns (total 5 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   year    101 non-null    int64
 1   srno    101 non-null    int64
 2   movie   101 non-null    int64
 3   rating  101 non-null    int64
 4   time    101 non-null    int64
dtypes: int64(5)
memory usage: 4.1 KB
```

In [19]:
```
data['year'].unique()#unique year can be printed in array
```

Out[19]:
```
array([1913, 1914, 1915, 1916, 1918, 1919, 1920, 1921, 1922, 1923, 1924,
       1925, 1926, 1927, 1928, 1929, 1930, 1931, 1932, 1933, 1934, 1935,
       1936, 1937, 1938, 1939, 1940, 1941, 1942, 1943, 1944, 1945, 1946,
       1947, 1948, 1949, 1950, 1951, 1952, 1953, 1954, 1955, 1956, 1957,
       1958, 1959, 1960, 1961, 1962, 1963, 1964, 1965, 1966, 1967, 1968,
       1969, 1970, 1971, 1972, 1973, 1974, 1975, 1976, 1977, 1978, 1979,
       1980, 1981, 1982, 1983, 1984, 1985, 1986, 1987, 1988, 1989, 1990,
       1991, 1992, 1993, 1994, 1995, 1996, 1997, 1998, 1999, 2000, 2001,
       2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012,
       2013, 2014])
```

In [20]:
```
data1=data.drop(['year'],axis=1)#to remove year column
```

In [21]:
```
data1
```

Out[21]:

	srno	movie	rating	time
0	3	3	3	3
1	20	20	5	18
2	1	1	1	1
3	1	1	1	1
4	1	1	1	1
...	...	...	...	...
96	5107	5107	1102	4671
97	5511	5511	1346	4992
98	4339	4339	1130	3978
99	981	981	345	901
100	1	1	1	1

101 rows × 4 columns

In [23]:
```
list(data1.columns)#identifying the columns
```

Out[23]:
```
['srno', 'movie', 'rating', 'time']
```

In [24]:
```
data['rating'].sum()#sum of rating column
```

Out[24]:
```
10814
```

In [28]:
```
data1=data.loc[(data.time>=5000)]#greater than 5000 can be printed
```

In [30]:
```
data1
```

Out[30]:

	srno	movie	rating	time
0	3	3	3	3
1	20	20	5	18
2	1	1	1	1
3	1	1	1	1
4	1	1	1	1
...	...	...	...	...
96	5107	5107	1102	4671
97	5511	5511	1346	4992
98	4339	4339	1130	3978
99	981	981	345	901
100	1	1	1	1

101 rows × 4 columns

In [31]:
```
data=data.loc[(data.year==2000)&(data.time>5000)]#testing both conditions
```

In [33]:
```
data
```

Out[33]:

	year	srno	movie	rating	time
0	1913	3	3	3	3
1	1914	20	20	5	18
2	1915	1	1	1	1
3	1916	1	1	1	1
4	1918	1	1	1	1
...	...	...	...	...	...
96	2010	5107	5107	1102	4671
97	2011	5511	5511	1346	4992
98	2012	4339	4339	1130	3978
99	2013	981	981	345	901
100	2014	1	1	1	1

101 rows × 5 columns

In [34]:
```
datar=data.sort_values('rating')#ascending order rating column
datar
```

Out[34]:

	year	srno	movie	rating	time
100	2014	1	1	1	1
2	1915	1	1	1	1
3	1916	1	1	1	1
4	1918	1	1	1	1
22	1936	7	7	2	7
...	...	...	...	...	...
94	2008	3358	3358	609	3088
95	2009	4451	4451	844	4092
96	2010	5107	5107	1102	4671
98	2012	4339	4339	1130	3978
97	2011	5511	5511	1346	4992

101 rows × 5 columns

In [35]:
```
datat=data.sort_values('time')#ascending order time column
datat
```

Out[35]:

	year	srno	movie	rating	time
100	2014	1	1	1	1
2	1915	1	1	1	1
3	1916	1	1	1	1
4	1918	1	1	1	1
14	1928	2	2	2	2
...	...	...	...	...	...
94	2008	3358	3358	609	3088
98	2012	4339	4339	1130	3978
95	2009	4451	4451	844	4092
96	2010	5107	5107	1102	4671
97	2011	5511	5511	1346	4992

101 rows × 5 columns

In [36]:
```
data.iloc[5]#5th column can be displayed
```

Out[36]:
```
year      1919
srno         3
movie        3
rating       3
time         3
Name: 5, dtype: int64
```

In [37]:
```
data2=data.loc[(data.rating>=4)&(data.year>=2000)&(data.year<=2010)]#test multiple conditions using loc func
data2
```

Out[37]:

	year	srno	movie	rating	time
86	2000	902	902	203	854
87	2001	1173	1173	237	1093
88	2002	1117	1117	250	1044
89	2003	1399	1399	279	1294
90	2004	1381	1381	311	1269
91	2005	1937	1937	392	1765
92	2006	2416	2416	492	2229
93	2007	2892	2892	569	2665
94	2008	3358	3358	609	3088
95	2009	4451	4451	844	4092
96	2010	5107	5107	1102	4671

In [ ]:
```

```