In [1]:
```
from numpy import arange
from matplotlib import pyplot as plt
from scipy.stats import norm
import pandas as pd
```

In [2]:
```
data=pd.read_csv("/home/praveen/Desktop/SEM/ML/0301/0.txt",sep="\t",names
```

In [3]:
```
data.head()
```

Out[3]:

	Video ID	Uploader	Age	Category	length	views	rate	ratings	comment	relate
0	2rwktobtv9s	EA	742.0	Gadgets & Games	83.0	389536.0	2.65	2294.0	268.0	SQI9xPI
1	h6Ghupxbj9g	KB42PAH	742.0	Sports	28.0	276207.0	4.57	297.0	424.0	O1dXfik
2	mfeZibn3vmU	Gromek66	742.0	Comedy	278.0	151693.0	4.68	228.0	96.0	FJwVwvl
3	86Fe6LICKKk	lonelygirl15	742.0	People & Blogs	148.0	125061.0	2.77	1343.0	1419.0	86Fe6LI
4	XbRkmBcVWlc	Htiwan	742.0	Film & Animation	79.0	108868.0	4.33	282.0	245.0	vDtUZ0_I

5 rows × 29 columns

In [4]:
```
data.dtypes
```

Out[4]:
```
Video ID          object
Uploader          object
Age              float64
Category          object
length           float64
views            float64
rate             float64
ratings          float64
comment          float64
related id1       object
related id2       object
related id3       object
related id4       object
related id5       object
related id6       object
related id7       object
related id8       object
related id9       object
related id10      object
related id11      object
related id12      object
related id13      object
related id14      object
related id15      object
related id16      object
related id17      object
related id18      object
related id19      object
related id20      object
dtype: object
```

In [5]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 359 entries, 0 to 358
Data columns (total 29 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   Video ID       359 non-null    object 
 1   Uploader       353 non-null    object 
 2   Age            353 non-null    float64
 3   Category       353 non-null    object 
 4   length         353 non-null    float64
 5   views          353 non-null    float64
 6   rate           353 non-null    float64
 7   ratings        353 non-null    float64
 8   comment        353 non-null    float64
 9   related id1    349 non-null    object 
 10  related id2    349 non-null    object 
 11  related id3    348 non-null    object 
 12  related id4    348 non-null    object 
 13  related id5    348 non-null    object 
 14  related id6    348 non-null    object 
 15  related id7    347 non-null    object 
 16  related id8    347 non-null    object 
 17  related id9    347 non-null    object 
 18  related id10   347 non-null    object 
 19  related id11   346 non-null    object 
 20  related id12   346 non-null    object 
 21  related id13   346 non-null    object 
 22  related id14   346 non-null    object 
 23  related id15   346 non-null    object 
 24  related id16   346 non-null    object 
 25  related id17   346 non-null    object 
 26  related id18   346 non-null    object 
 27  related id19   345 non-null    object 
 28  related id20   343 non-null    object 
dtypes: float64(6), object(23)
memory usage: 81.5+ KB
```

In [6]:
```
data.memory_usage()
```

Out[6]:
```
Index            128
Video ID        2872
Uploader        2872
Age             2872
Category        2872
length          2872
views           2872
rate            2872
ratings         2872
comment         2872
related id1     2872
related id2     2872
related id3     2872
related id4     2872
related id5     2872
related id6     2872
related id7     2872
related id8     2872
related id9     2872
related id10    2872
related id11    2872
related id12    2872
related id13    2872
related id14    2872
related id15    2872
related id16    2872
related id17    2872
related id18    2872
related id19    2872
related id20    2872
dtype: int64
```

In [7]:
```
data.memory_usage().sum()
```

Out[7]:
```
83416
```

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

Out[8]:

	Age	length	views	rate	ratings	comment
count	353.000000	353.000000	3.530000e+02	353.000000	353.000000	353.000000
mean	738.671388	288.920680	6.545571e+04	4.248725	251.113314	185.736544
std	39.477910	260.720776	1.388875e+05	0.804386	366.200460	250.564184
min	0.000000	5.000000	1.930000e+02	0.000000	0.000000	0.000000
25%	739.000000	94.000000	4.274000e+03	4.050000	55.000000	35.000000
50%	742.000000	218.000000	1.166600e+04	4.520000	119.000000	90.000000
75%	742.000000	436.000000	8.375500e+04	4.780000	306.000000	252.000000
max	743.000000	1732.000000	1.726429e+06	5.000000	3019.000000	1951.000000

In [9]:
```
data.mean()
```

```
<ipython-input-9-abc01cf6c622>:1: FutureWarning: Dropping of nuisance co
lumns in DataFrame reductions (with 'numeric_only=None') is deprecated;
in a future version this will raise TypeError.  Select only valid column
s before calling the reduction.
  data.mean()
```

Out[9]:
```
Age          738.671388
length       288.920680
views      65455.708215
rate           4.248725
ratings      251.113314
comment      185.736544
dtype: float64
```

In [10]:
```
data['views'].mean()
```

Out[10]:
```
65455.70821529745
```

In [11]:
```
data.corr()
```

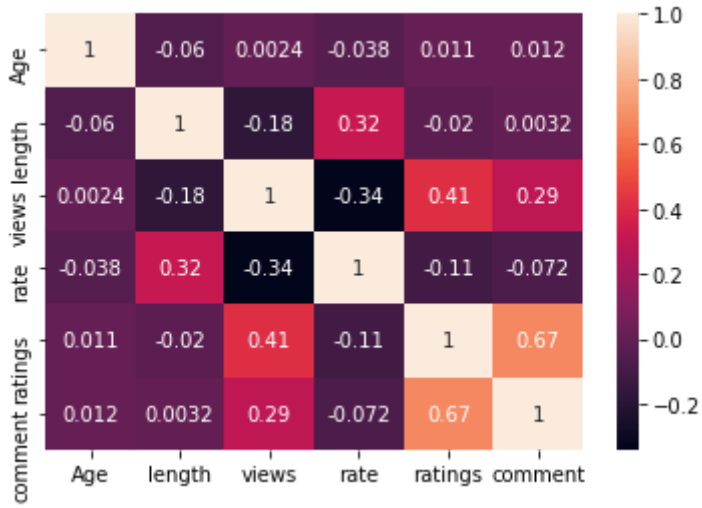
Out[11]:

	Age	length	views	rate	ratings	comment
Age	1.000000	-0.059594	0.002375	-0.037906	0.010645	0.011828
length	-0.059594	1.000000	-0.177573	0.319274	-0.020210	0.003234
views	0.002375	-0.177573	1.000000	-0.340352	0.412502	0.290656
rate	-0.037906	0.319274	-0.340352	1.000000	-0.105435	-0.072211
ratings	0.010645	-0.020210	0.412502	-0.105435	1.000000	0.669227
comment	0.011828	0.003234	0.290656	-0.072211	0.669227	1.000000

In [12]:
```
import seaborn as s
```

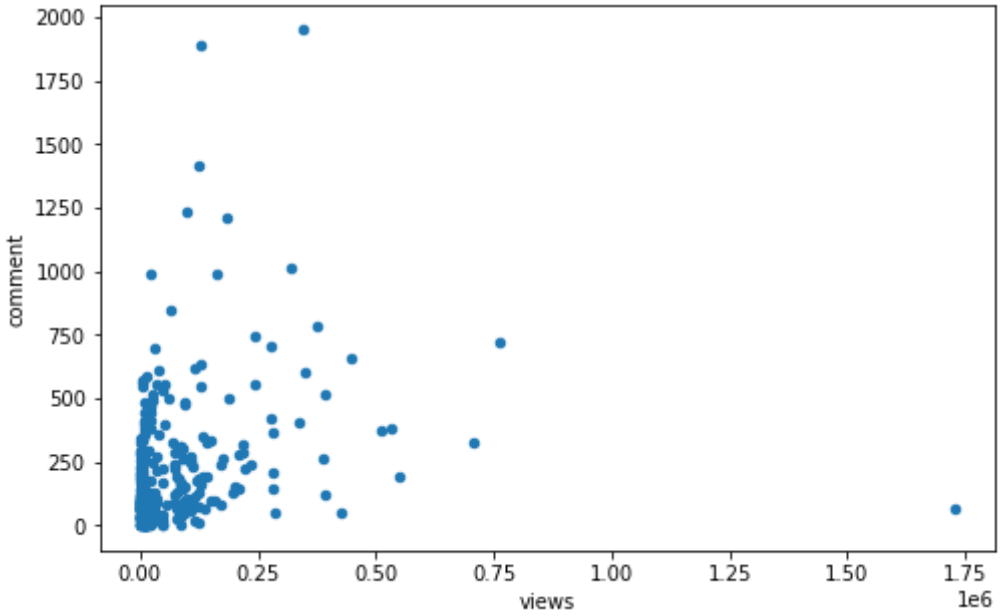
In [13]:
```
s.heatmap(data.corr(),annot=True)
```

Out[13]:
```
<AxesSubplot:>
```

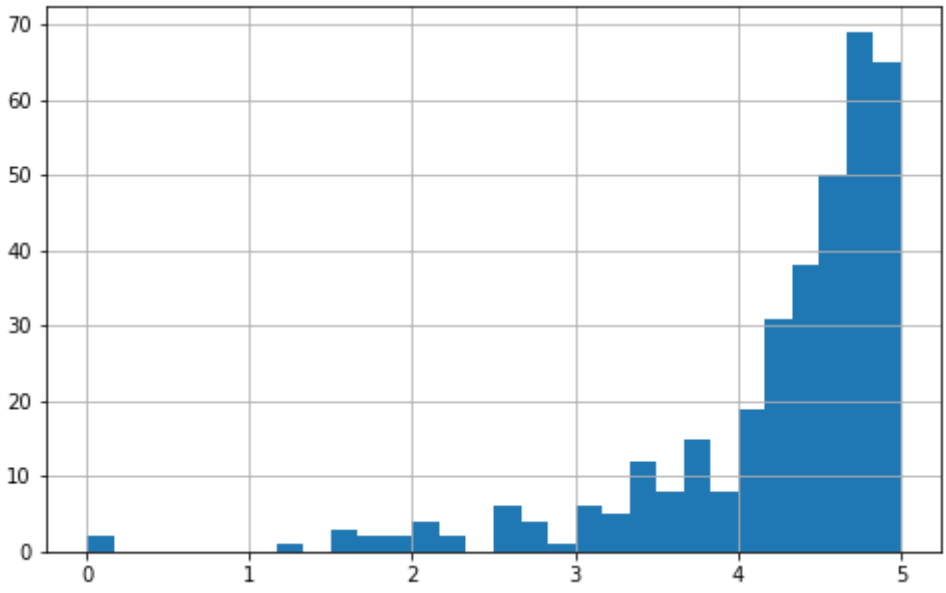


In [14]:
```
data.plot.scatter('views', 'comment', figsize=(8, 5))
```

Out[14]:
```
<AxesSubplot:xlabel='views', ylabel='comment'>
```

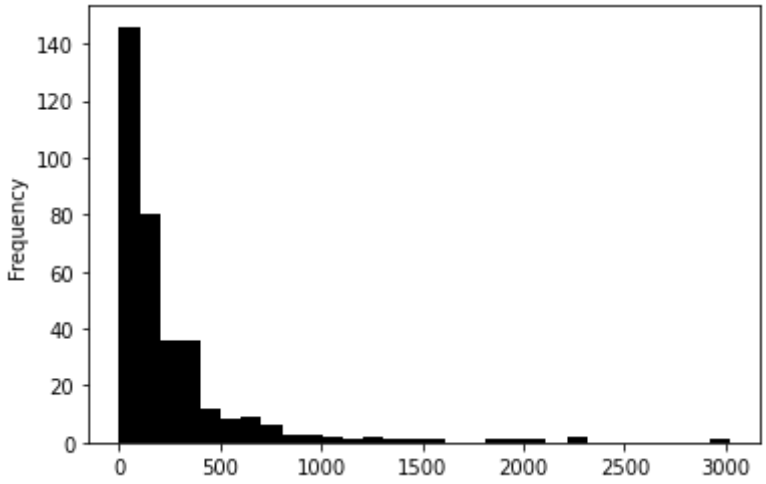


In [15]:
```
data['rate'].hist(bins=30, figsize=(8, 5));
```

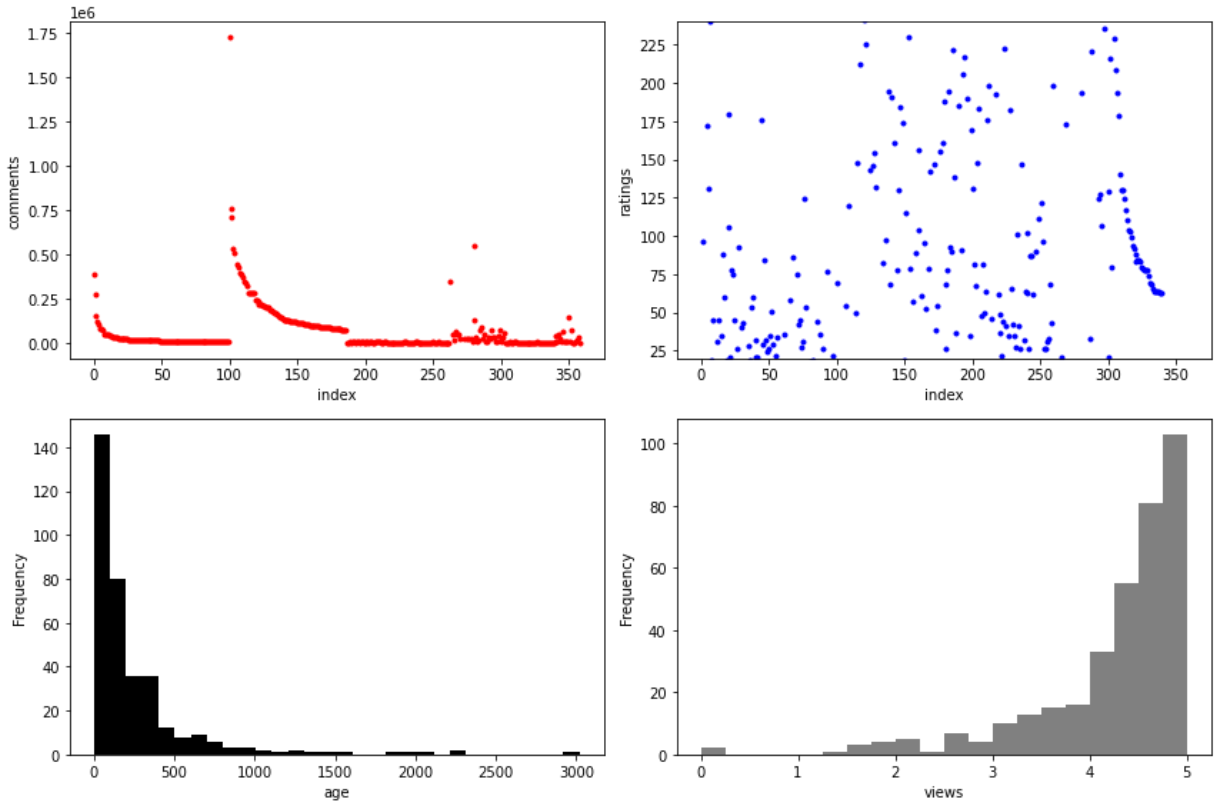


In [16]:
```
data['ratings'].plot.hist(bins=30, color='black')
```

Out[16]: <AxesSubplot:ylabel='Frequency'>

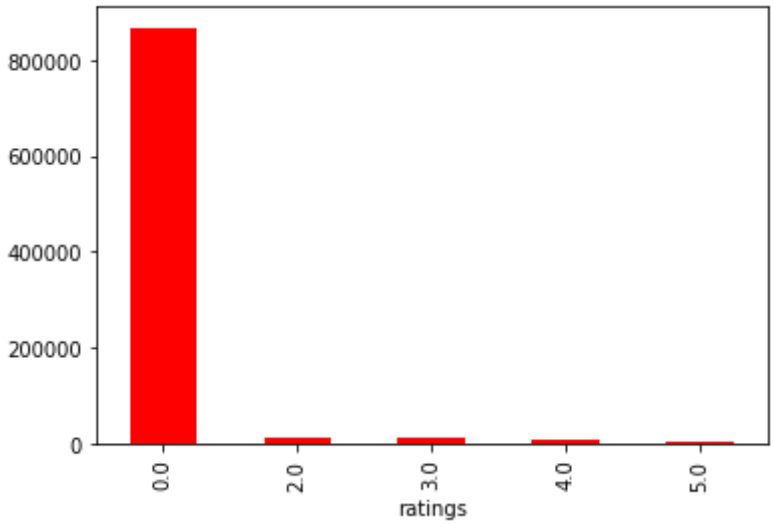


In [17]:
```
fig, axes = plt.subplots(2, 2, figsize=(12, 8))
# or fig, (ax1, ax2, ax3, ax4) = plt.subplots(2, 2, figsize=(12, 8))

# axes is the axes object(s). It can be a single object or an array of ob
# In this case, it is an array of dimension 2-by-2

data['views'].plot(ax = axes[0][0], style='.', color='red') # top left
data['comment'].plot(ax = axes[0][1], style='.', color='blue') # top righ

data['ratings'].plot.hist(bins=30, ax = axes[1][0], color='black') # bott
data['rate'].plot.hist(bins=20, ax = axes[1][1], color='gray') # bottom r

axes[0][0].set_xlabel('index')
axes[0][1].set_xlabel('index')
axes[1][0].set_xlabel('age')
axes[1][1].set_xlabel('views')

axes[0][0].set_ylabel('comments')
axes[0][1].set_ylabel('ratings')
# axes[1][0].set_ylabel('')

# axes[0][0].set_ylim(20, 120)
axes[0][1].set_ylim(20, 240)

# axes[1][0].set_xlim(0, 60)
# axes[1][1].set_xlim(20, 80)

fig.tight_layout()
```



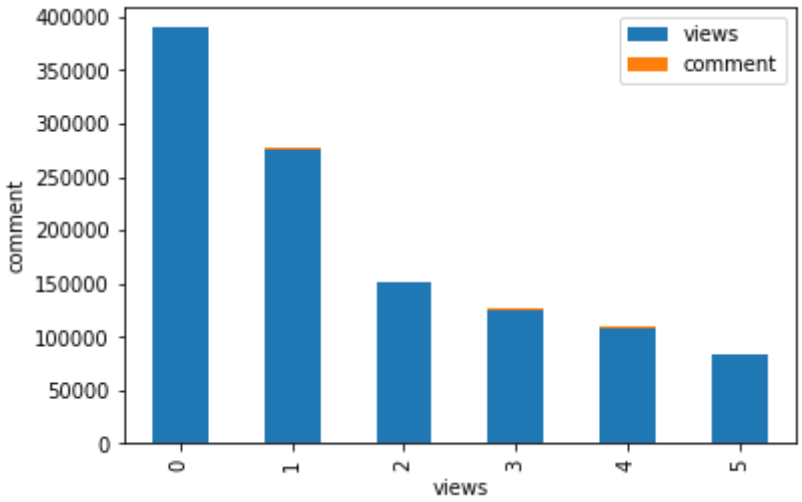
In [18]:

```
data_avg_rate = data.groupby('ratings')['views'].mean()
data_avg_rate[:5].plot.bar(color='red');
```



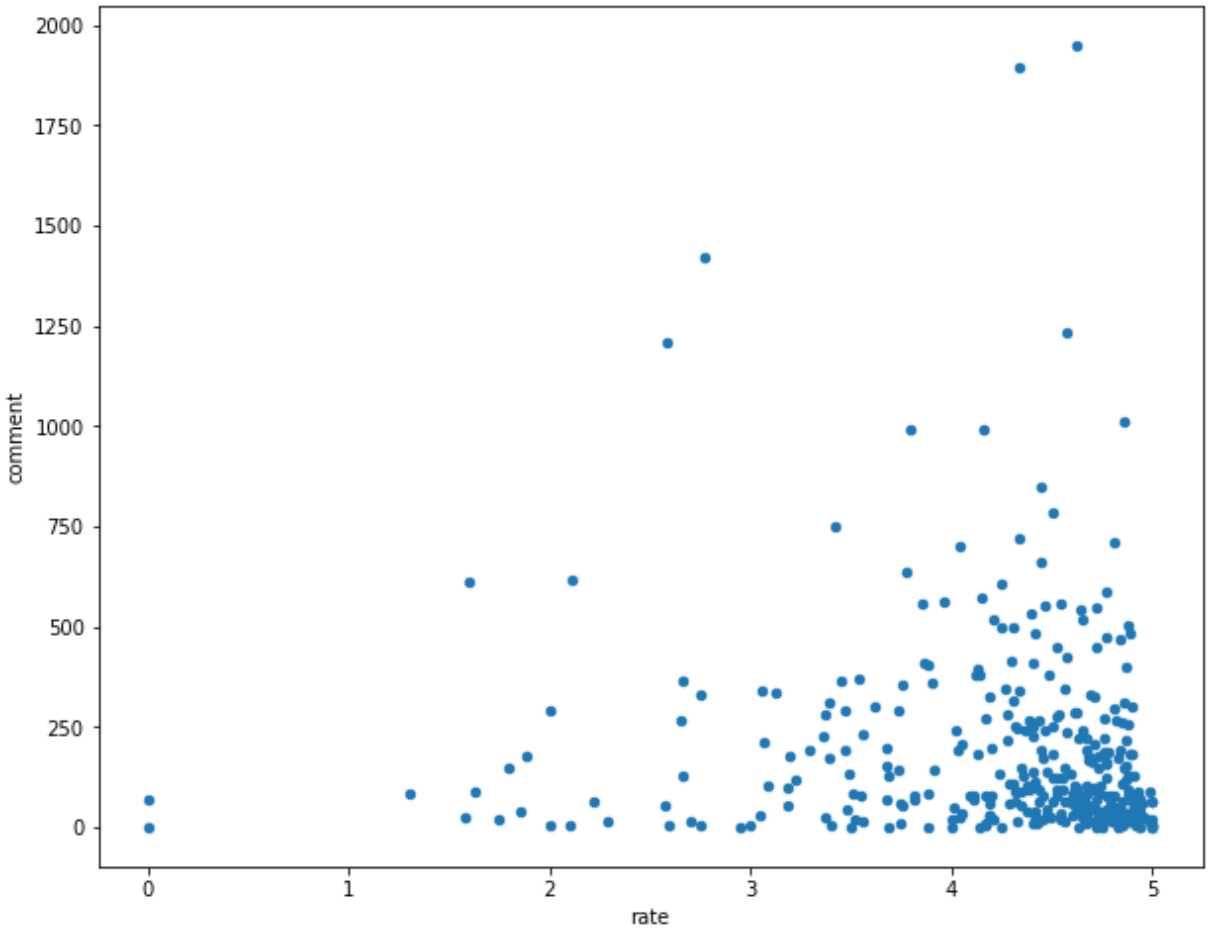
In [19]:

```
ax = data[['views', 'comment']][:6].plot.bar(stacked=True)

ax.set_xlabel("views")
ax.set_ylabel("comment");
```



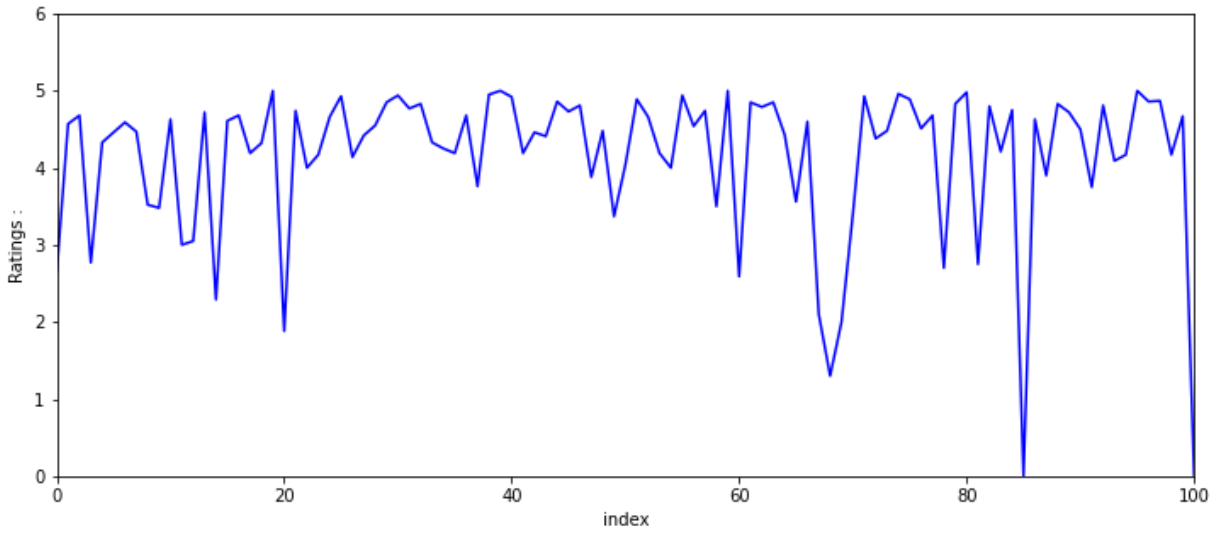
In [20]:
```
data.plot.scatter('rate', 'comment', figsize=(10, 8))
```

Out[20]:
```
<AxesSubplot:xlabel='rate', ylabel='comment'>
```



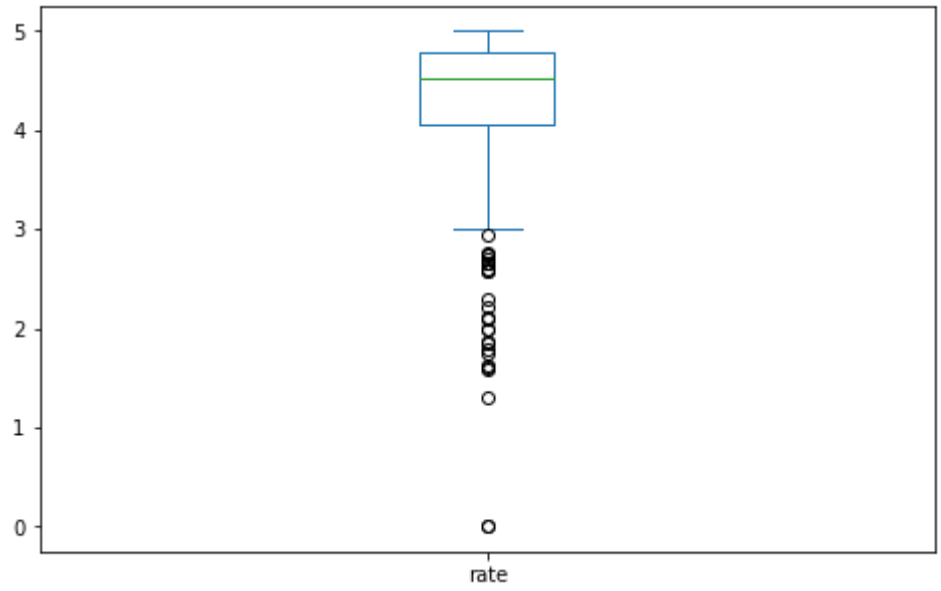
In [21]:

```
data['rate'].plot(figsize=(12, 5), color='blue')

plt.xlim(0, 100)
plt.ylim(0, 6)
plt.xlabel('index')
plt.ylabel('Ratings :');
```



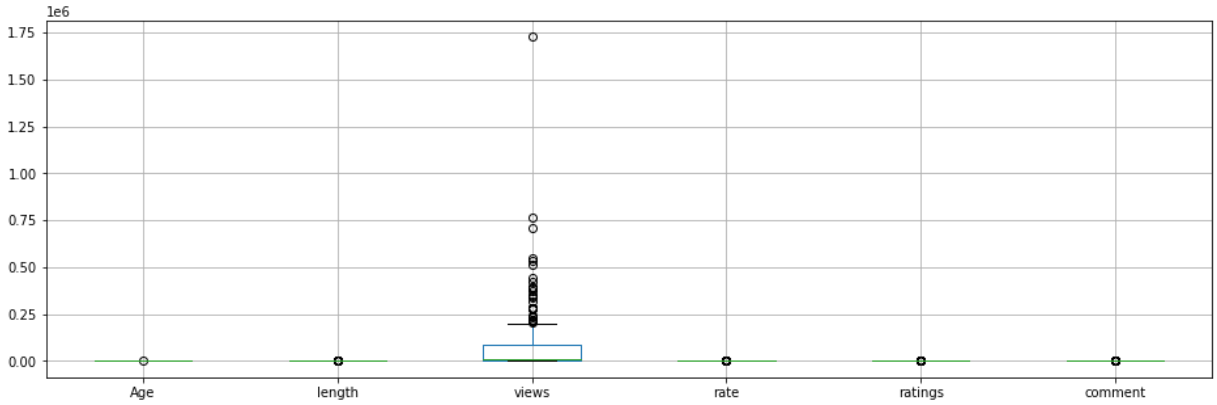
In [22]:

```
data['rate'].plot.box(figsize=(8, 5));
```



In [23]:
```
data.boxplot(figsize=(16, 5))
```

Out[23]: <AxesSubplot:>



In [ ]: