In [1]:
```
# Import the libraries
import pandas as pd
import numpy as np

#visualization

import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
# Import the data set
# first from the Jupiter home tab we have to upload the data set.
data=pd.read_csv('electronics.csv')
```

In [3]:
```
data.shape
# we have 10 columns and 1292954 rows
```

Out[3]:
```
(1292954, 10)
```

In [4]:
```
#information of data set
data.info
```

Out[4]:
```
<bound method DataFrame.info of          item_id  user_id  rating   timestamp model_a
ttr  \
0              0        0     5.0  1999-06-13     Female
1              0        1     5.0  1999-06-14     Female
2              0        2     3.0  1999-06-17     Female
3              0        3     1.0  1999-07-01     Female
4              0        4     2.0  1999-07-06     Female
...          ...      ...     ...         ...        ...
1292949     9478  1157628     1.0  2018-09-26     Female
1292950     9435  1157629     5.0  2018-09-26     Female
1292951     9305  1157630     3.0  2018-09-26     Female
1292952     9303  1157631     5.0  2018-09-29       Male
1292953     9478  1157632     1.0  2018-10-01     Female

                        category       brand  year user_attr  split
0         Portable Audio & Video         NaN  1999       NaN      0
1         Portable Audio & Video         NaN  1999       NaN      0
2         Portable Audio & Video         NaN  1999       NaN      0
3         Portable Audio & Video         NaN  1999       NaN      0
4         Portable Audio & Video         NaN  1999       NaN      0
...                          ...         ...   ...       ...    ...
1292949              Headphones  Etre Jeune  2017       NaN      0
1292950  Computers & Accessories         NaN  2017       NaN      0
1292951  Computers & Accessories         NaN  2016       NaN      0
1292952              Headphones         NaN  2018       NaN      0
1292953              Headphones  Etre Jeune  2017    Female      0

[1292954 rows x 10 columns]>
```

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

Out[5]:

	item_id	user_id	rating	timestamp	model_attr	category	brand	year	user_attr	split
0	0	0	5.0	1999-06-13	Female	Portable Audio & Video	NaN	1999	NaN	0
1	0	1	5.0	1999-06-14	Female	Portable Audio & Video	NaN	1999	NaN	0
2	0	2	3.0	1999-06-17	Female	Portable Audio & Video	NaN	1999	NaN	0
3	0	3	1.0	1999-07-01	Female	Portable Audio & Video	NaN	1999	NaN	0
4	0	4	2.0	1999-07-06	Female	Portable Audio & Video	NaN	1999	NaN	0

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

Out[11]:

	item_id	user_id	rating	timestamp	model_attr	category	brand	year	user_attr	split
1292949	9478	1157628	1.0	2018-09-26	Female	Headphones	Etre Jeune	2017	NaN	0
1292950	9435	1157629	5.0	2018-09-26	Female	Computers & Accessories	NaN	2017	NaN	0
1292951	9305	1157630	3.0	2018-09-26	Female	Computers & Accessories	NaN	2016	NaN	0
1292952	9303	1157631	5.0	2018-09-29	Male	Headphones	NaN	2018	NaN	0
1292953	9478	1157632	1.0	2018-10-01	Female	Headphones	Etre Jeune	2017	Female	0

In [8]:
```
#to know the data type
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1292954 entries, 0 to 1292953
Data columns (total 10 columns):
 #   Column      Non-Null Count    Dtype  
---  ------      --------------    -----  
 0   item_id     1292954 non-null  int64  
 1   user_id     1292954 non-null  int64  
 2   rating      1292954 non-null  float64
 3   timestamp   1292954 non-null  object 
 4   model_attr  1292954 non-null  object 
 5   category    1292954 non-null  object 
 6   brand       331120 non-null   object 
 7   year        1292954 non-null  int64  
 8   user_attr   174124 non-null   object 
 9   split       1292954 non-null  int64  
dtypes: float64(1), int64(4), object(5)
memory usage: 98.6+ MB
```

In [10]:
```
# We can also see that the column Timestamp is of object data type, but it is actually

# We can convert it to a timestamp using the following code:
import datetime as datetime
pd.to_datetime(data['timestamp'])
```

Out[10]:
```
0         1999-06-13
1         1999-06-14
2         1999-06-17
3         1999-07-01
4         1999-07-06
             ...
1292949   2018-09-26
1292950   2018-09-26
1292951   2018-09-26
1292952   2018-09-29
1292953   2018-10-01
Name: timestamp, Length: 1292954, dtype: datetime64[ns]
```

In [18]:
```
#convert category to string data type
data['category'] = data['category'].astype(str)

#convert user id to string data type
data['user_id'] = data['user_id'].astype(str)

#convert item_id to string data type
data['item_id'] = data['item_id'].astype(str)
```

In [19]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1292954 entries, 0 to 1292953
Data columns (total 10 columns):
 #   Column      Non-Null Count    Dtype
---  ------      --------------    -----
 0   item_id     1292954 non-null  object
 1   user_id     1292954 non-null  object
 2   rating      1292954 non-null  float64
 3   timestamp   1292954 non-null  object
 4   model_attr  1292954 non-null  object
 5   category    1292954 non-null  object
 6   brand       331120 non-null   object
 7   year        1292954 non-null  int64
 8   user_attr   174124 non-null   object
 9   split       1292954 non-null  int64
dtypes: float64(1), int64(2), object(7)
memory usage: 98.6+ MB
```

In [20]:
```
data.describe()
```

Out[20]:

	rating	year	split
count	1.292954e+06	1.292954e+06	1.292954e+06
mean	4.051482e+00	2.012938e+03	1.747587e-01
std	1.379732e+00	2.643513e+00	5.506810e-01
min	1.000000e+00	1.999000e+03	0.000000e+00
25%	4.000000e+00	2.012000e+03	0.000000e+00
50%	5.000000e+00	2.014000e+03	0.000000e+00
75%	5.000000e+00	2.015000e+03	0.000000e+00
max	5.000000e+00	2.018000e+03	2.000000e+00

In [21]:
```
# to find the unique users and items in the data.

data.nunique()
```

Out[21]:
```
item_id          9560
user_id       1157633
rating              5
timestamp        6354
model_attr          3
category           10
brand              50
year               20
user_attr           2
split               3
dtype: int64
```

In [22]:
```
#to check null values in the data
data.isnull().sum()
```

Out[22]:
```
item_id             0
user_id             0
rating              0
timestamp           0
model_attr          0
category            0
brand          961834
year                0
user_attr     1118830
split               0
dtype: int64
```

In [24]:
```
#Drop the null values
data.dropna(inplace=True)
```

In [25]:
```
data.isnull().sum()
```

Out[25]:
```
item_id       0
user_id       0
rating        0
timestamp     0
model_attr    0
category      0
brand         0
year          0
user_attr     0
split         0
dtype: int64
```

In [26]:
```
data.nunique()
```

Out[26]:
```
item_id        1892
user_id       40401
rating            5
timestamp      4179
model_attr        3
category         10
brand            50
year             19
user_attr         2
split             3
dtype: int64
```

Sanity Check

In [27]:
```
#Rating should be between 1 to 5
#Drop the rows that has values outside this range
```

In [28]:
```
# We need to check if there is any data in "rating" greater than 5
data[data['rating']>5]
```

Out[28]:

item_id	user_id	rating	timestamp	model_attr	category	brand	year	user_attr	split

In [29]:
```
# By doing above check we know now that there is no rating above 5
```

In [30]:
```
# We need to check if there is any data in "rating" less than 1
data[data['rating']<1]
```

Out[30]:

item_id	user_id	rating	timestamp	model_attr	category	brand	year	user_attr	split

In [31]:
```
# By doing above check we know now that there is no rating under 1
```

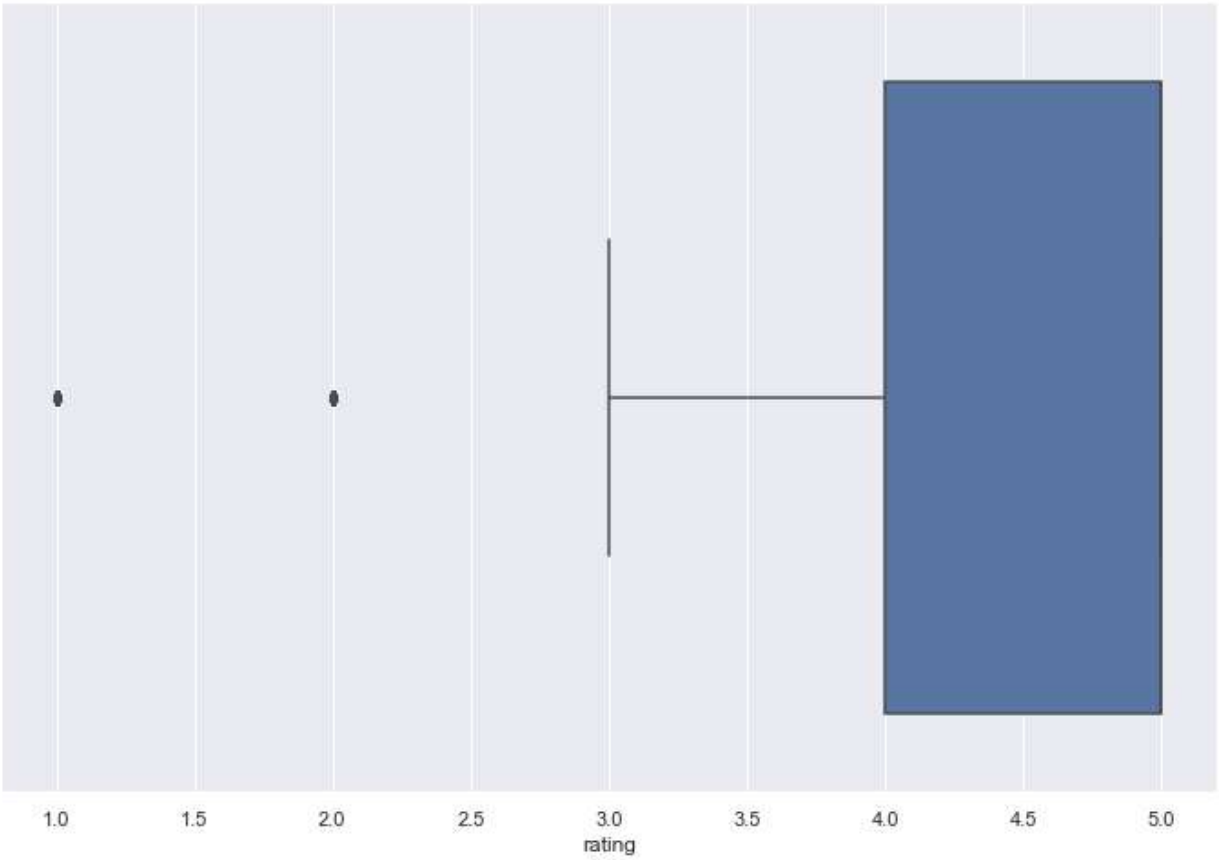
Analysing the data set

In [32]:
```
# Box plot
#Box plot of rating and brand
```

In [35]:
```
sns.boxplot(data['rating'])
```

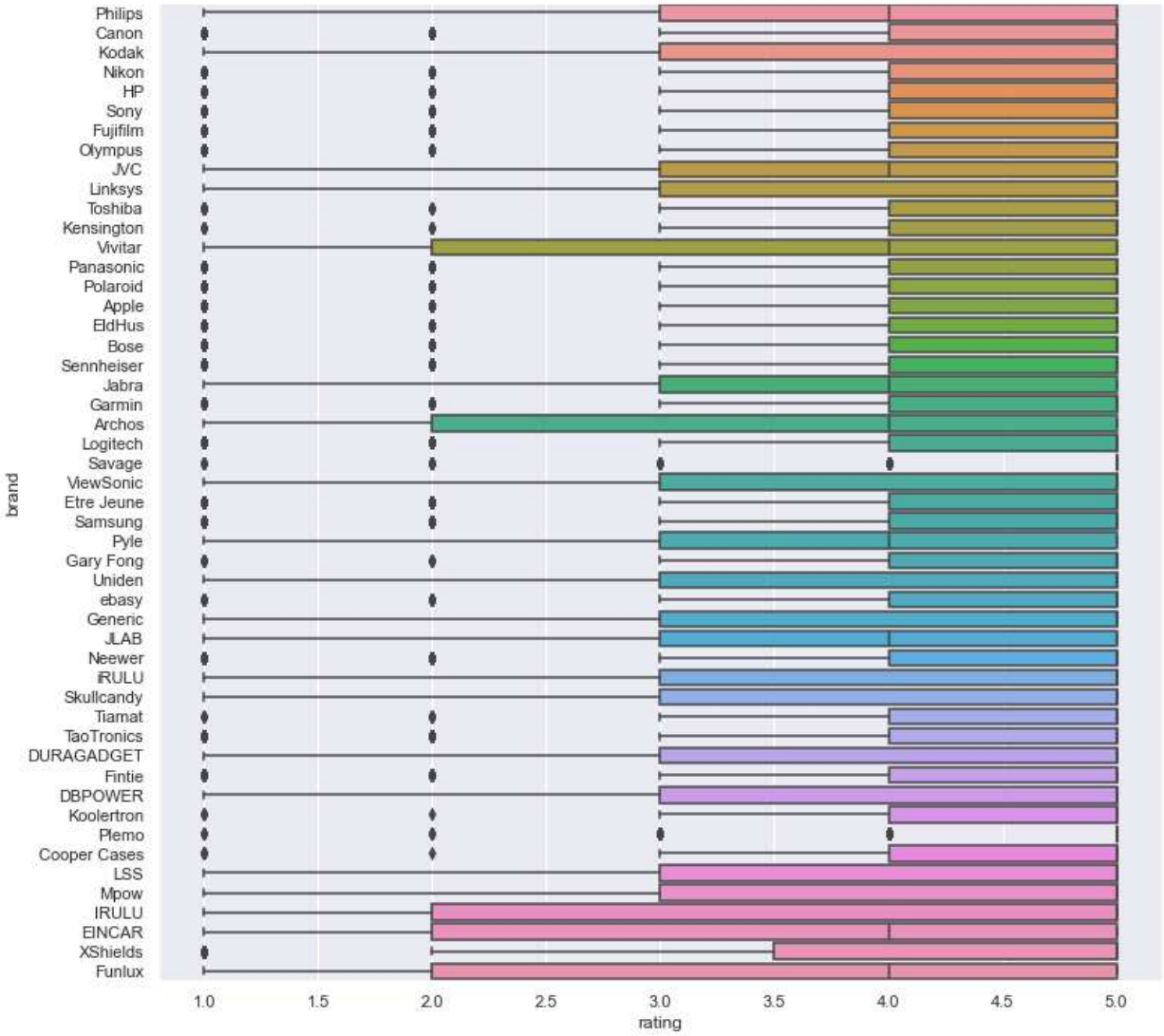
```
C:\Users\Anjan\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning:
Pass the following variable as a keyword arg: x. From version 0.12, the only valid po
sitional argument will be `data`, and passing other arguments without an explicit key
word will result in an error or misinterpretation.
  warnings.warn(
```

Out[35]: <AxesSubplot:xlabel='rating'>



In [39]:
```
sns.set(rc={'figure.figsize':(12,12)}) # for graph size
sns.boxplot(x="rating", y="brand", data=data)
```

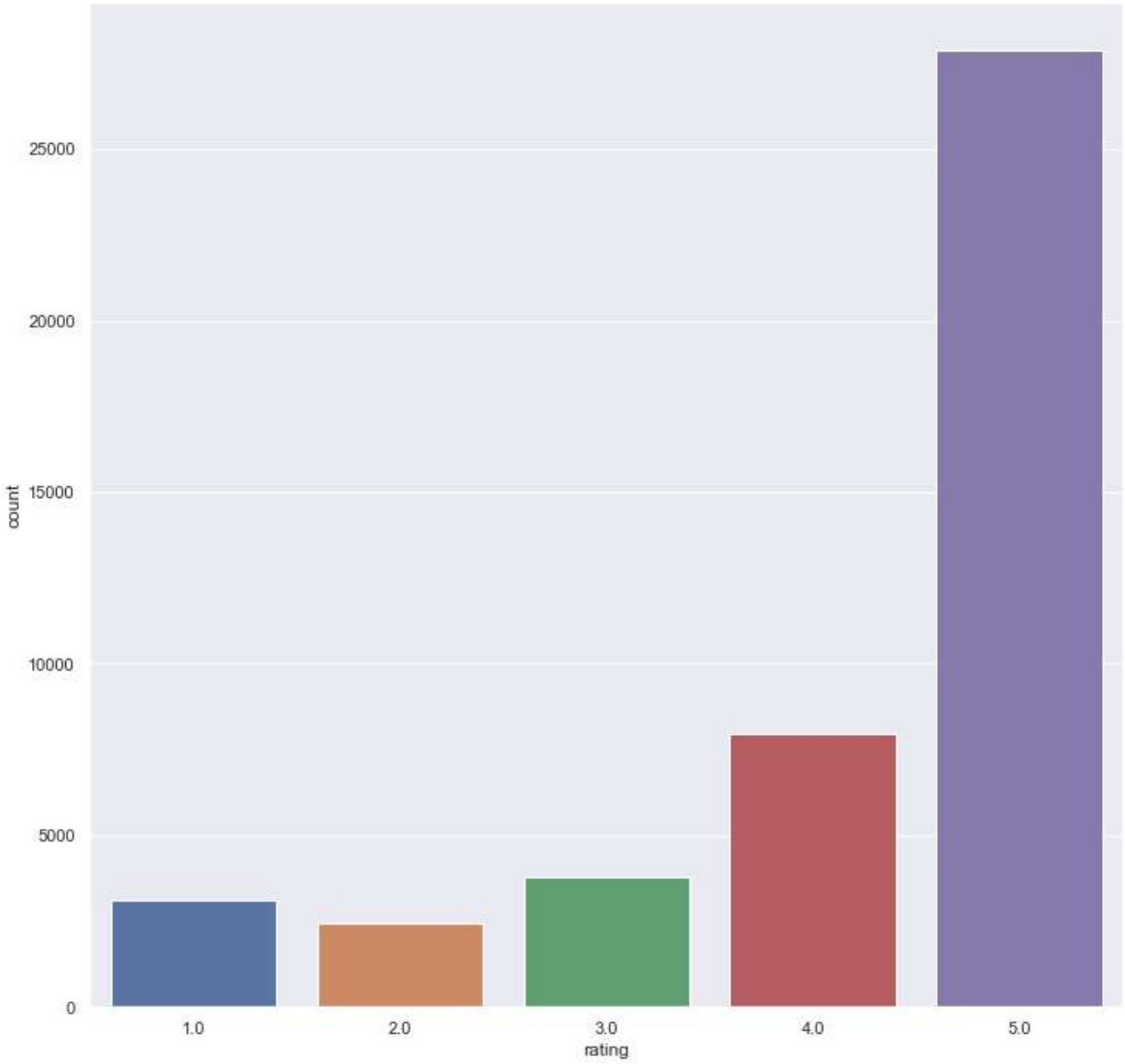
Out[39]: <AxesSubplot:xlabel='rating', ylabel='brand'>



In [42]:
```
# the distribution of ratings

sns.countplot(x='rating', data=data)
```

Out[42]:
```
<AxesSubplot:xlabel='rating', ylabel='count'>
```

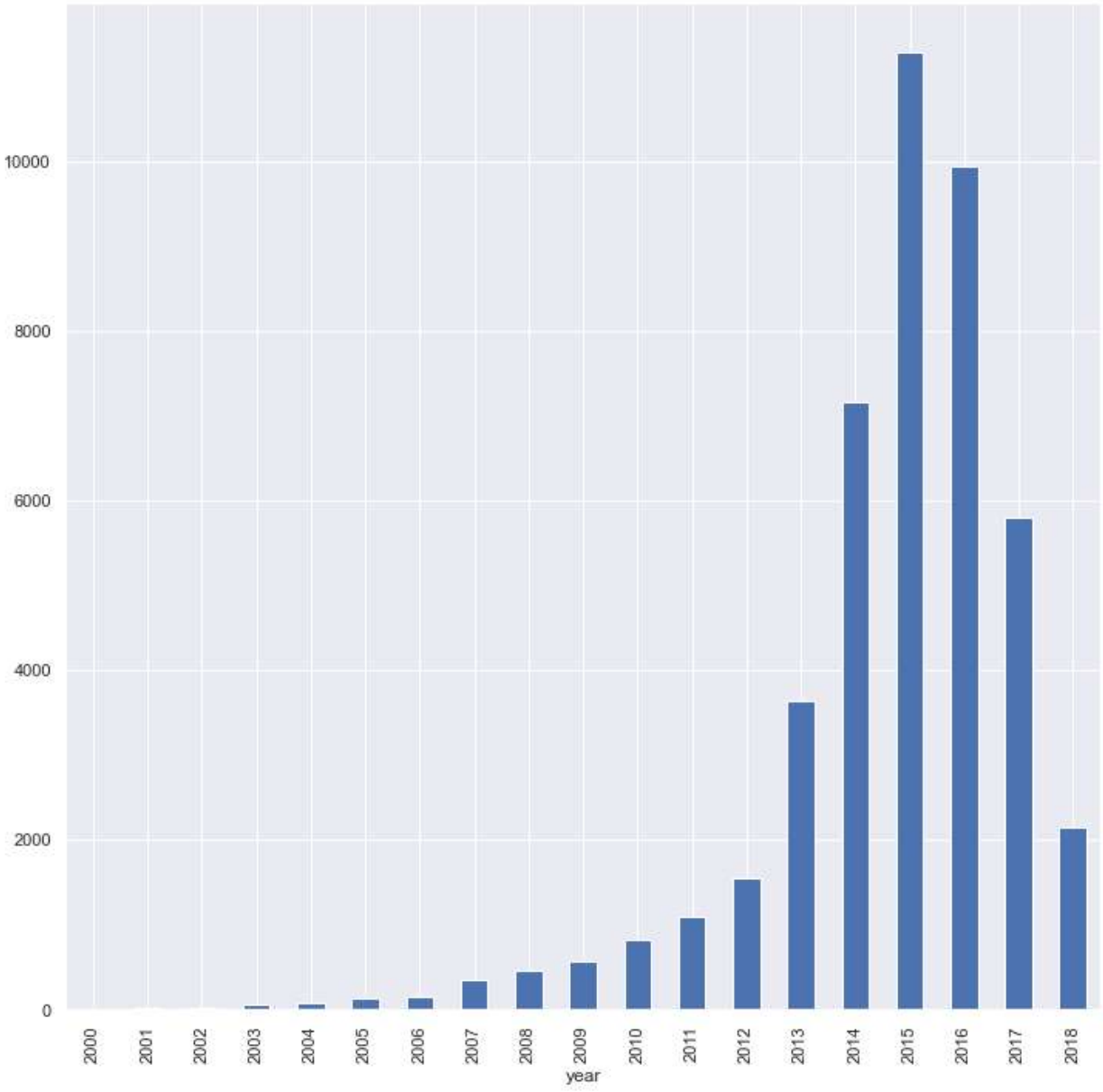


In [43]:
```
# the above plot we understand the maximum rtaing given is 5 and the least being 1.
```

Analysing sales per brand in a given year

In [45]:
```
# best year of sales
data['year'] = pd.DatetimeIndex(data['timestamp']).year
data.groupby('year')['rating'].count().plot(kind='bar')
```

Out[45]:
```
<AxesSubplot:xlabel='year'>
```

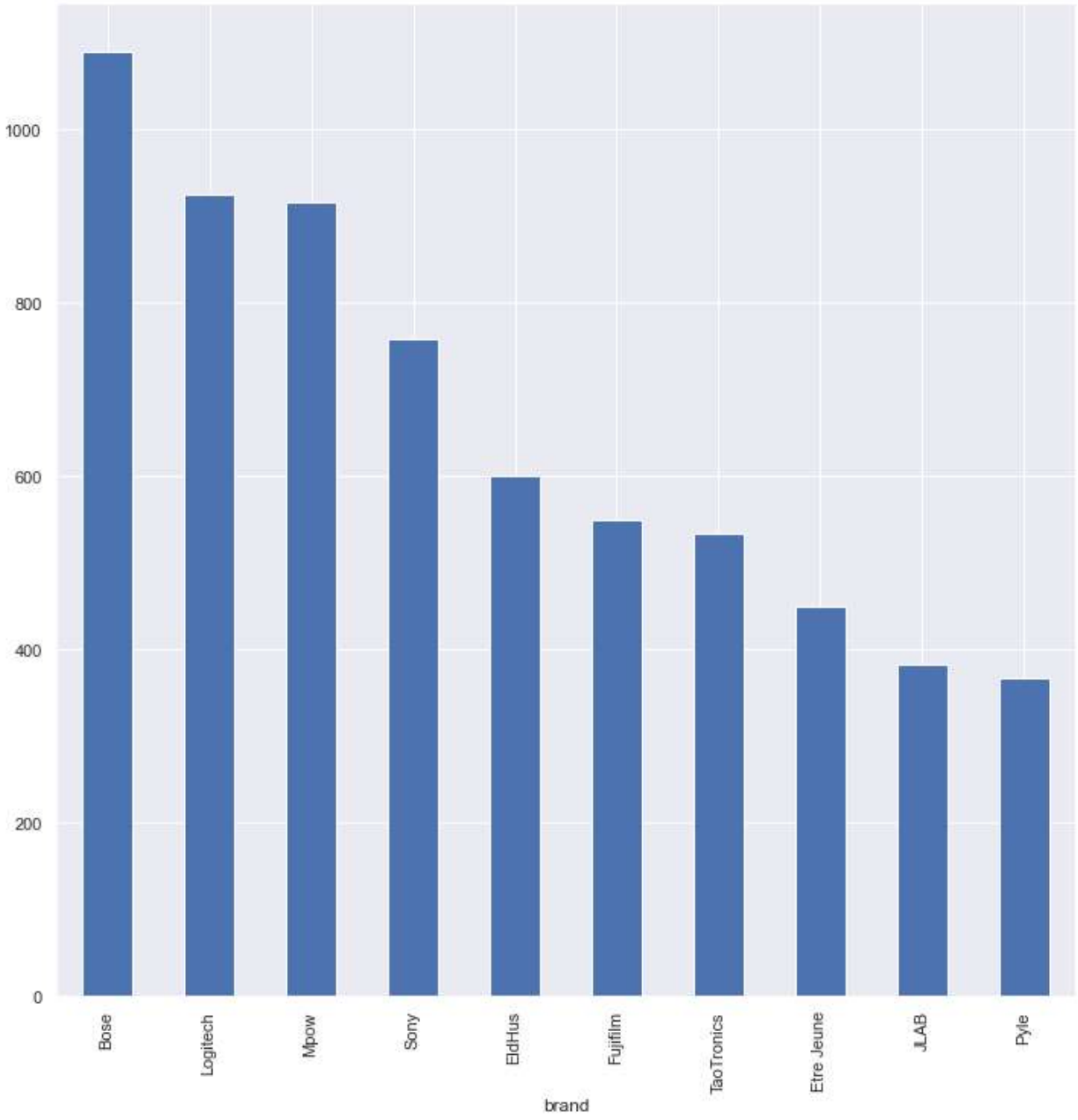


In [46]:
```
# By above gragh we know the to sales year are 2015, 2016 and 2014
# Lets analyse the top brands sold in the above top three years
```

In [47]:
```
#2015
data_2015 = data[data['year'] == 2015]

data_2015.groupby('brand')['rating'].count().sort_values(ascending=False).head(10).plo
```

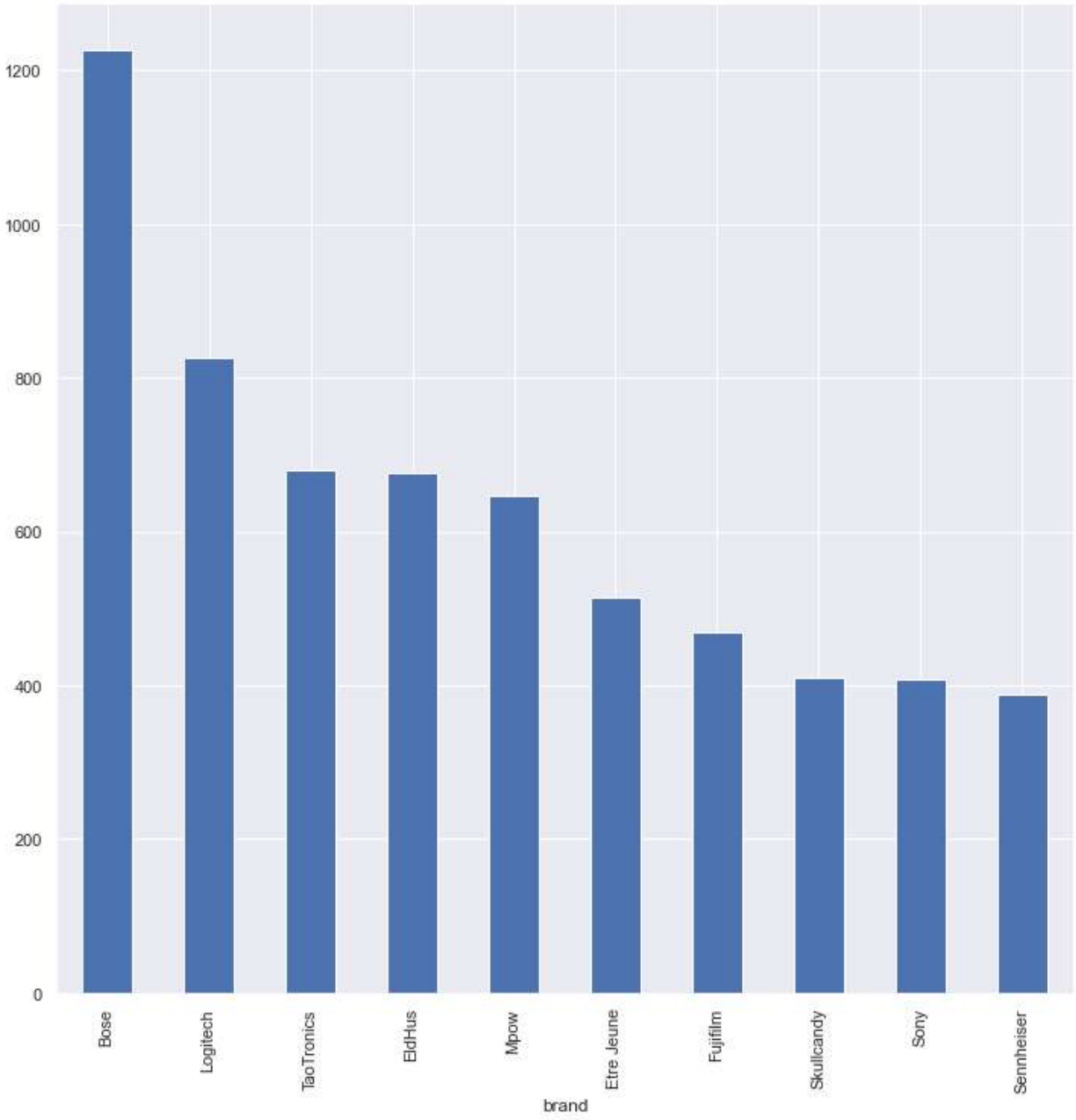
Out[47]:
```
<AxesSubplot:xlabel='brand'>
```



In [48]:
```
#2016
data_2016 = data[data['year'] == 2016]

data_2016.groupby('brand')['rating'].count().sort_values(ascending=False).head(10).plo
```

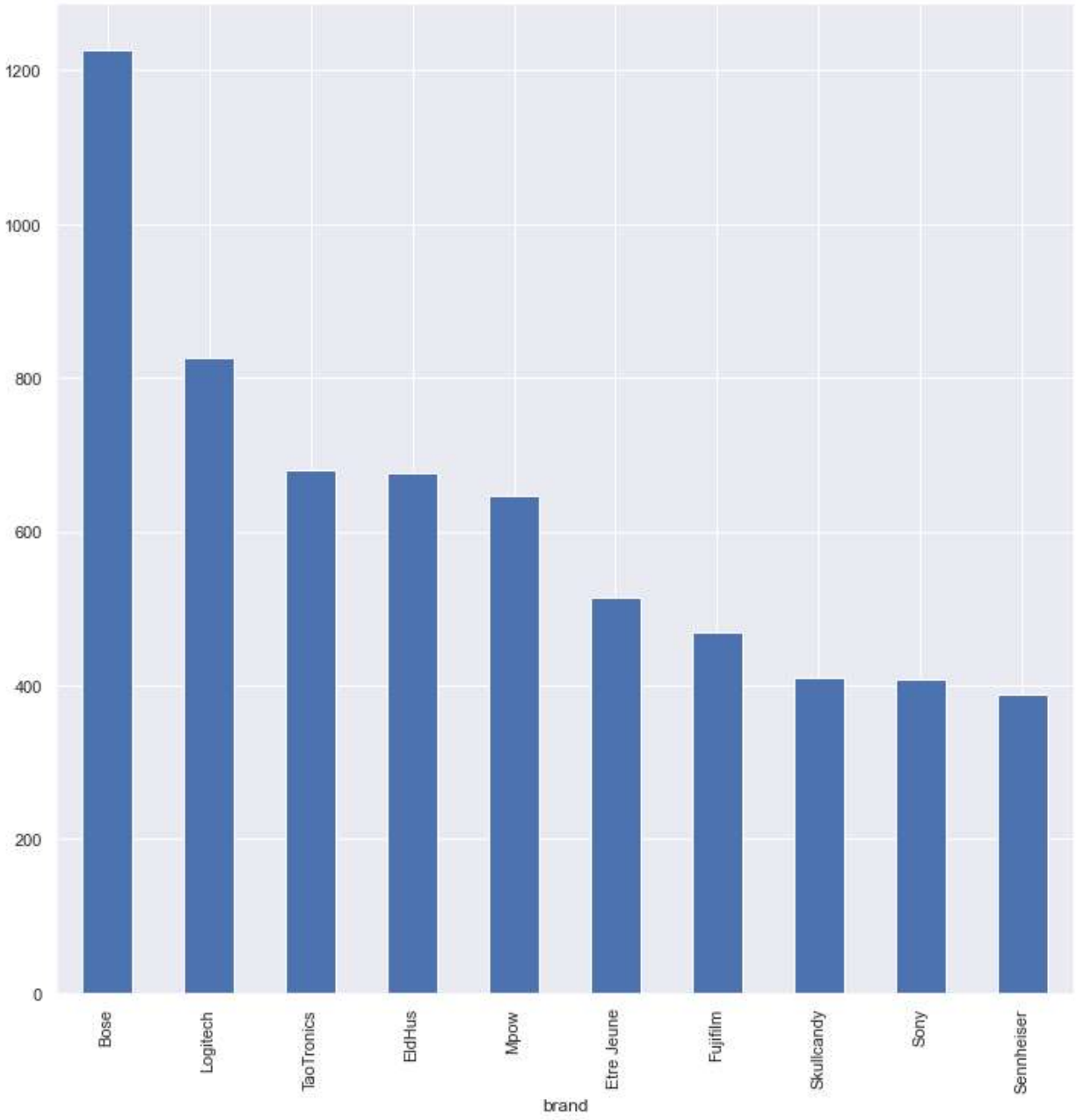
Out[48]:
```
<AxesSubplot:xlabel='brand'>
```



In [49]:
```
#2014
data_2014 = data[data['year'] == 2016]

data_2014.groupby('brand')['rating'].count().sort_values(ascending=False).head(10).plo
```

Out[49]:
```
<AxesSubplot:xlabel='brand'>
```



In [50]:
```
# we can see that top saleing brand is Bose followed by Logitech in above three year
```

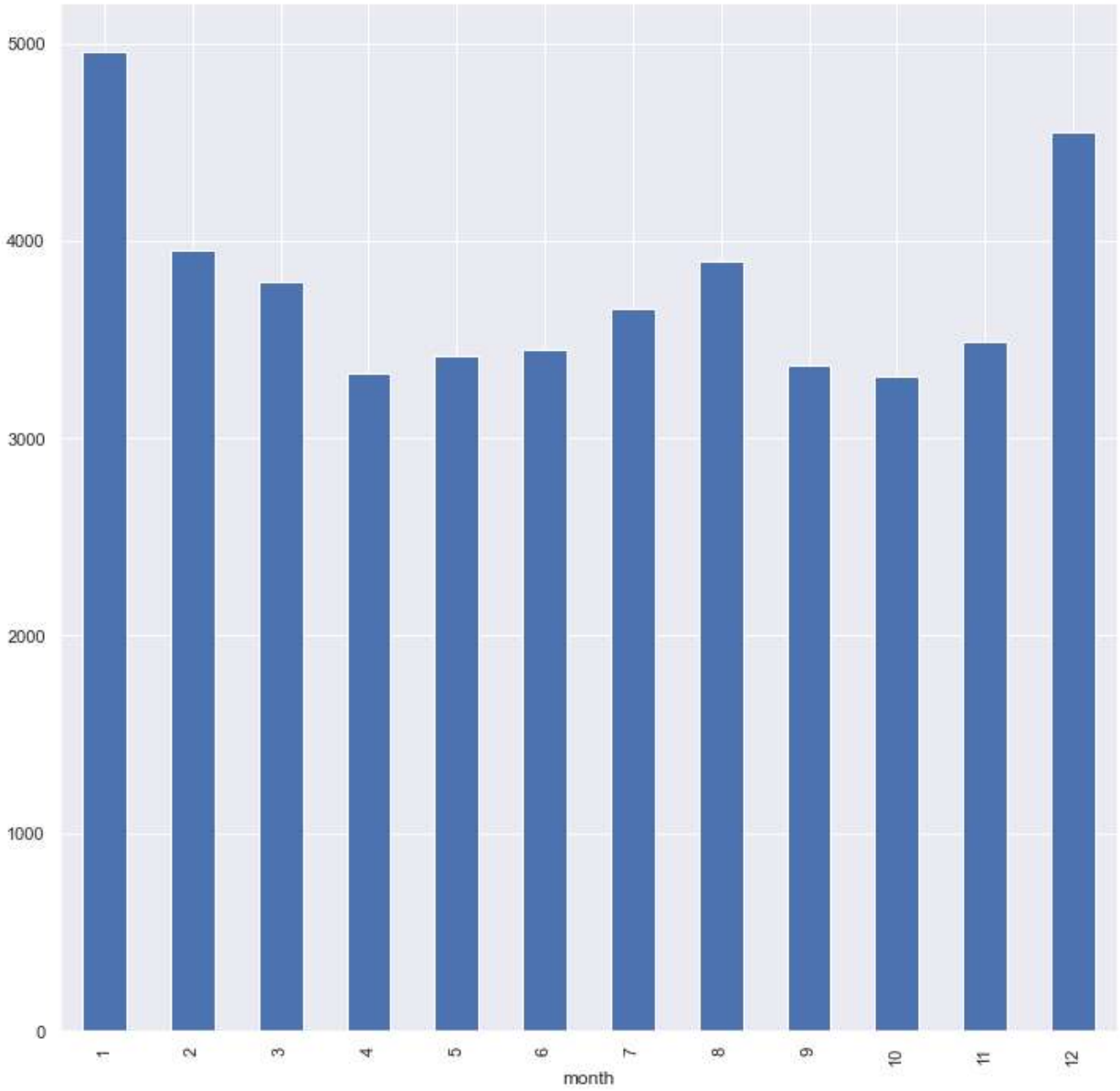
In [75]:
```
# Year 2015 had the best sales.

# Best month of sales

data['month'] = pd.DatetimeIndex(data['timestamp']).month

data.groupby('month')['rating'].count().plot(kind='bar')
```

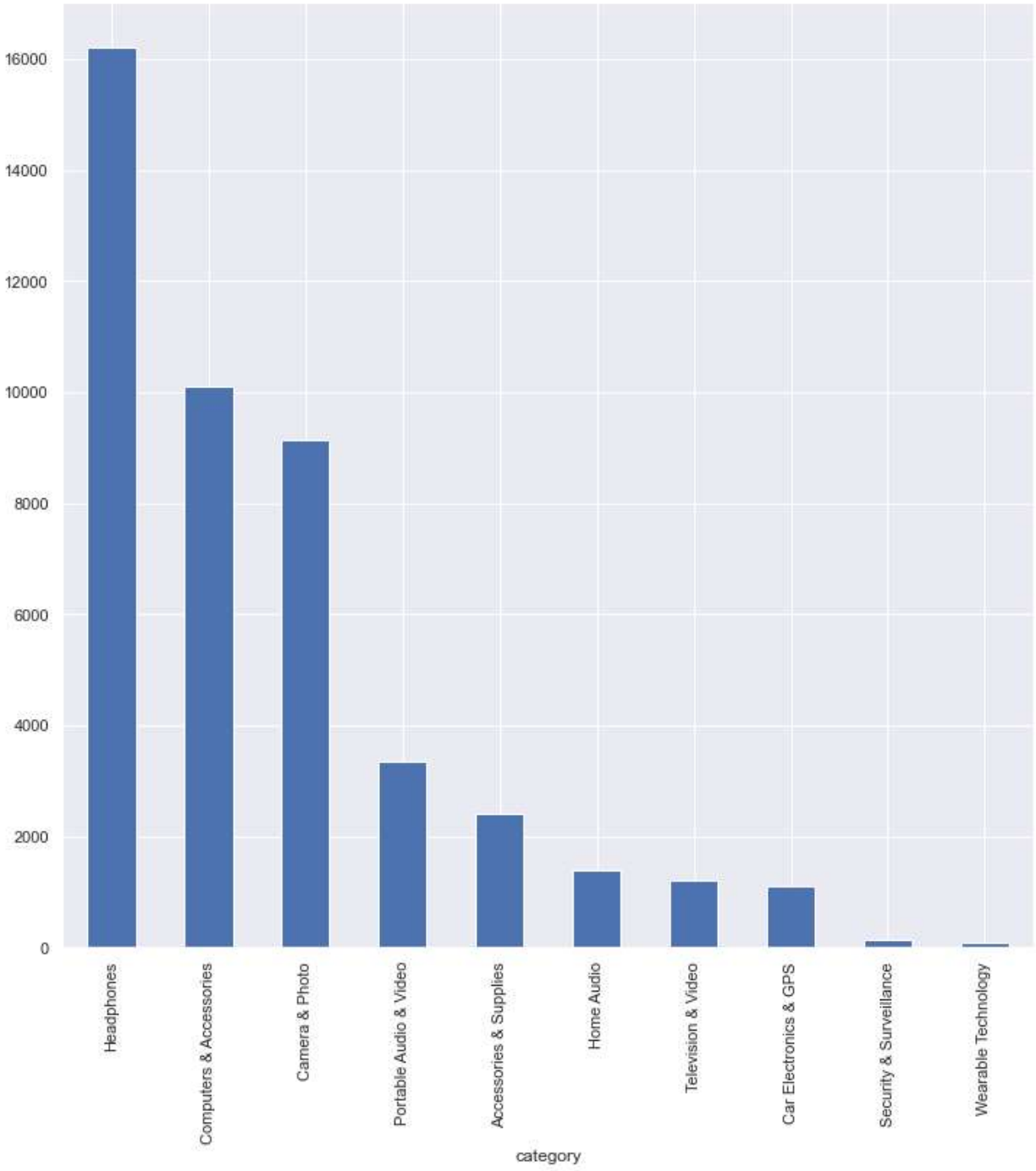
Out[75]:
```
<AxesSubplot:xlabel='month'>
```



In [64]:
```
# Product by category sold the most?

data.groupby('category')['rating'].count().sort_values(ascending=False).head(10).plot(
```

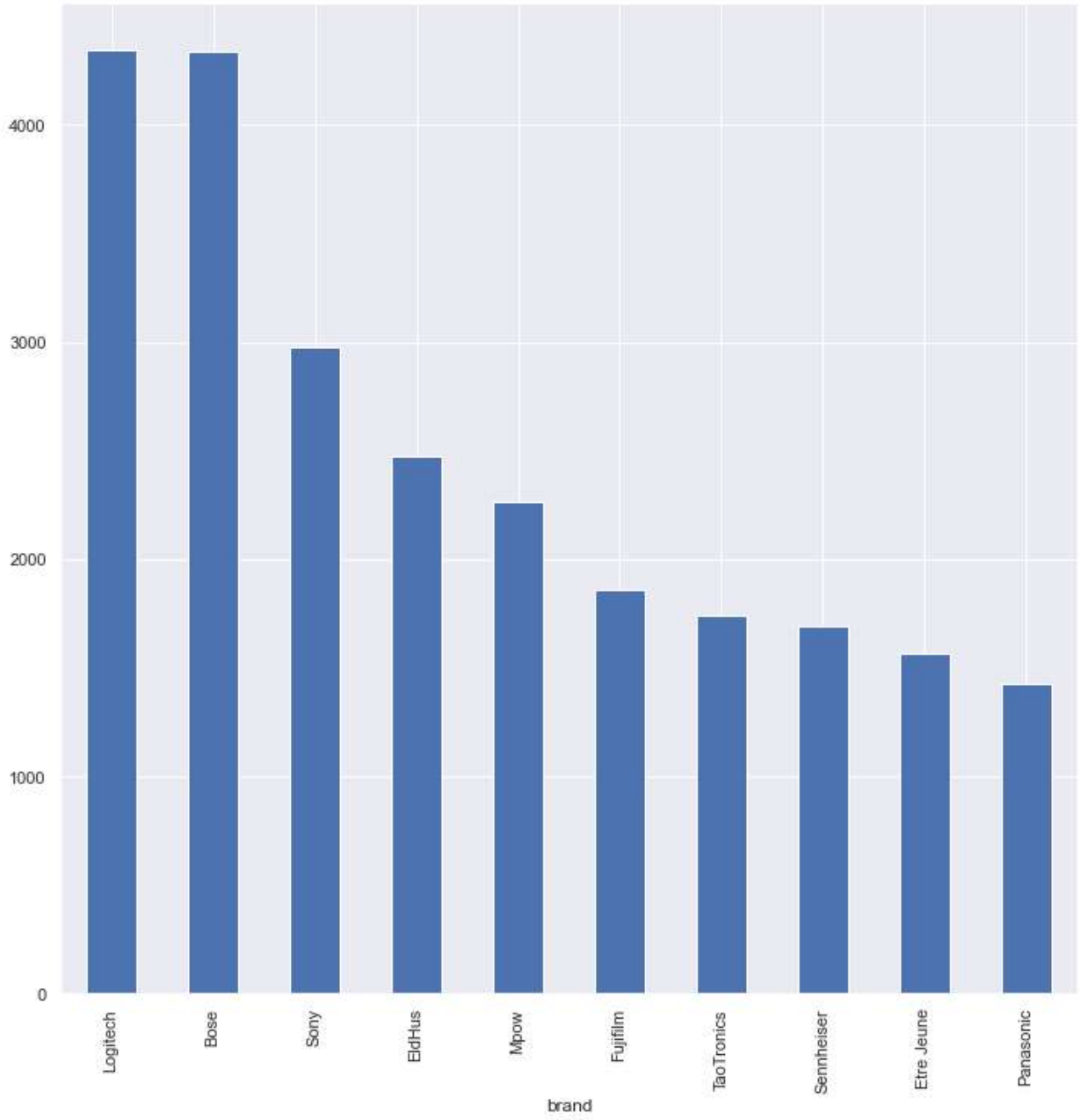
Out[64]:
```
<AxesSubplot:xlabel='category'>
```



In [71]:
```
#  From above we know the top product type sold is Headphones.
```

In [73]:
```
data.groupby('brand')['rating'].count().sort_values(ascending=False).head(10).plot(kir
```

Out[73]:
```
<AxesSubplot:xlabel='brand'>
```



In [74]:
```
#  From above we know the top Brand sold is Logitech and Bose.
```

Final Anlysis

In [76]:
```
# Year 2015 had the best sales.
# we know the top Brand sold is Logitech and Bose.
# We know the top product type sold is Headphones.
# Jan followed by Dec are the top month of sales.
# Dec sales is high due to christmas.
# Customer spend more in new year.
# The sales could be bosted if more offers are in the above two months.
```

In [ ]:
```

```