```
# The dataset gives us electronics sales data at Amazon.

# It contains user ratings for various electronics items sold, along
with category of each item and time of sell.

# The dataset is available at
https://www.kaggle.com/datasets/edusanketdk/electronics

# Importing the libraries

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# visualization

import seaborn as sns

# Importing the dataset

dataset = pd.read_csv('electronics.csv')

# list of first five rows

dataset.head()

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

  brand  year user_attr  split
0   NaN  1999       NaN      0
1   NaN  1999       NaN      0
2   NaN  1999       NaN      0
3   NaN  1999       NaN      0
4   NaN  1999       NaN      0

# list of last five rows

dataset.tail()
```

```
         item_id  user_id  rating   timestamp model_attr  \
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
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
```

```
# shape

dataset.shape
```

```
(1292954, 10)
```

```
# It is also a good practice to know the columns and their
corresponding data types
# along with finding whether they contain null values or not.

dataset.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
```

```
# We can see that the dataset contains 5 columns and 10000 rows.

# The columns are as follows:

# 1. User ID

# 2. Product ID
# 3. Rating

# 4. Timestamp

# 5. Category

# The data types of the columns are as follows:

# 1. User ID - int64

# 2. Product ID - object

# 3. Rating - int64

# 4. Timestamp - int64

# 5. Category - object

# We can see that the columns User ID and Rating are of int64 data
type, while the columns Product ID and Category are of object data
type.

# We can also see that there are no null values in the dataset.

# We can also see that the column Timestamp is of int64 data type, but
it is actually a timestamp.

# We can convert it to a timestamp using the following code:

from datetime import datetime

pd.to_datetime(dataset['timestamp'])

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]

# We can also see that the column Product ID is of object data type,
but it is actually a string.

# We can convert it to a string using the following code:
```

```
dataset['brand'] = dataset['brand'].astype(str)

# We can also see that the column Category is of object data type, but
it is actually a string.

# We can convert it to a string using the following code:

dataset['category'] = dataset['category'].astype(str)

# We can also see that the column Timestamp is of int64 data type, but
it is actually a timestamp.

# We can convert it to a timestamp using the following code:

dataset['timestamp'] = pd.to_datetime(dataset['timestamp'])

# We can also see that the column Rating is of int64 data type, but it
is actually a float.

# We can convert it to a float using the following code:

dataset['rating'] = dataset['rating'].astype(float)

# We can also see that the column User ID is of int64 data type, but
it is actually a string.

# We can convert it to a string using the following code:

dataset['user_id'] = dataset['user_id'].astype(str)

# We can also see that the column Product ID is of object data type,
but it is actually a string.

# We can convert it to a string using the following code:

dataset['item_id'] = dataset['item_id'].astype(str)

# to get a better understanding of the dataset,

# we can also see the statistical summary of the dataset.

dataset.describe()
```

```
             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
```

```
# the statistical summary of the dataset gives us the following information:

# 1. The mean rating is 4.

# 2. The minimum rating is 1.

# 3. The maximum rating is 5.

# 4. The standard deviation of the ratings is 1.4.

# 5. The 25th percentile of the ratings is 4.

# 6. The 50th percentile of the ratings is 5.

# 7. The 75th percentile of the ratings is 5.

# We can also see the number of unique users and items in the dataset.

dataset.nunique()
```

```
item_id          9560
user_id       1157633
rating              5
timestamp        6354
model_attr          3
category           10
brand              51
year               20
user_attr           2
split               3
dtype: int64
```

```
# check for duplicates

dataset.duplicated().sum()
```

```
0
```

```
# check for missing values

dataset.isnull().sum()
```

```
item_id      0
user_id      0
rating       0
timestamp    0
category     0
brand        0
```

```
year         0
user_attr    0
split        0
month        0
day          0
dtype: int64
```

```
# the distribution of ratings
```

```
dataset['rating'].value_counts()
```

```
5.0    107593
4.0     30104
3.0     14593
1.0     12652
2.0      9182
Name: rating, dtype: int64
```

```
# most of the ratings are 5
```

```
# what was the best year of sales
```

```
dataset['year'] = pd.DatetimeIndex(dataset['timestamp']).year
```

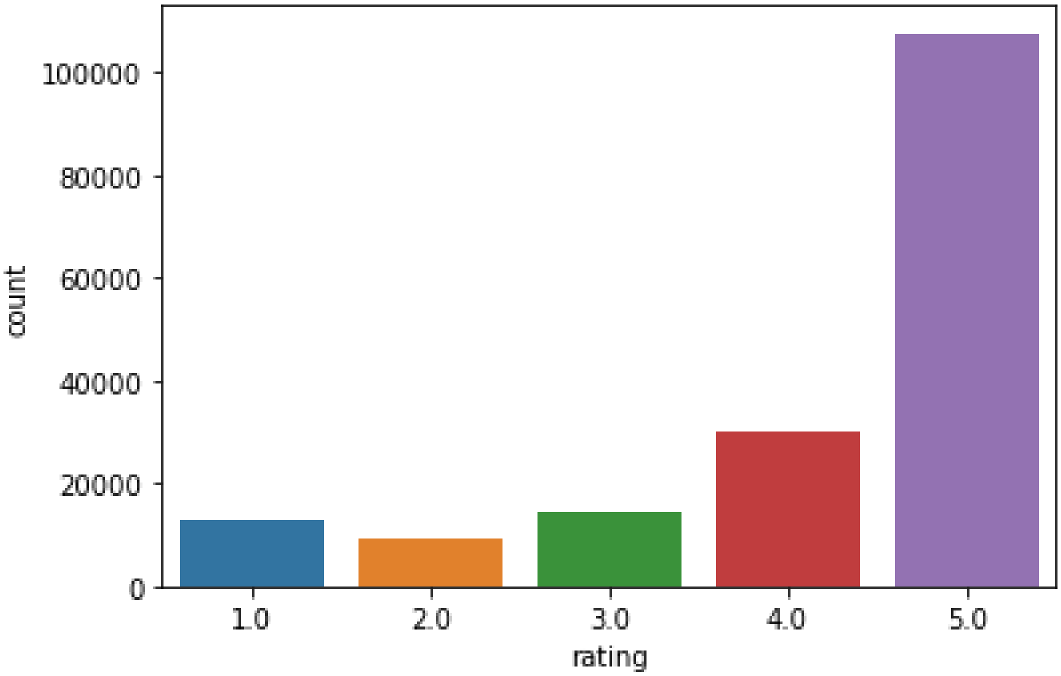
```
dataset['year'].value_counts()
```

```
2015    46891
2016    43907
2014    25475
2017    24753
2013    12355
2018     8874
2012     4357
2011     2679
2010     1717
2009     1220
2008      834
2007      525
2006      196
2005      149
2004       87
2003       55
2002       26
2001       18
2000        5
1999        1
Name: year, dtype: int64
```

```
# 2015 was the best year of sales
# what was the best month of sales

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

dataset['month'].value_counts()

1     18762
12    17134
2     15033
3     14853
8     14789
7     14439
11    13412
4     13359
5     13258
9     13155
6     12970
10    12960
Name: month, dtype: int64

# January was the best month of sales

# drop all null values

dataset.dropna(inplace=True)

# check for missing values

dataset.isnull().sum()

item_id       0
user_id       0
rating        0
timestamp     0
model_attr    0
category      0
brand         0
year          0
user_attr     0
split         0
month         0
day           0
dtype: int64
```

#FINDING ANSWERS WITH THE DATA WE HAVE WITH VISUALIZATIONS

```
# the distribution of ratings

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

<AxesSubplot:xlabel='rating', ylabel='count'>
```

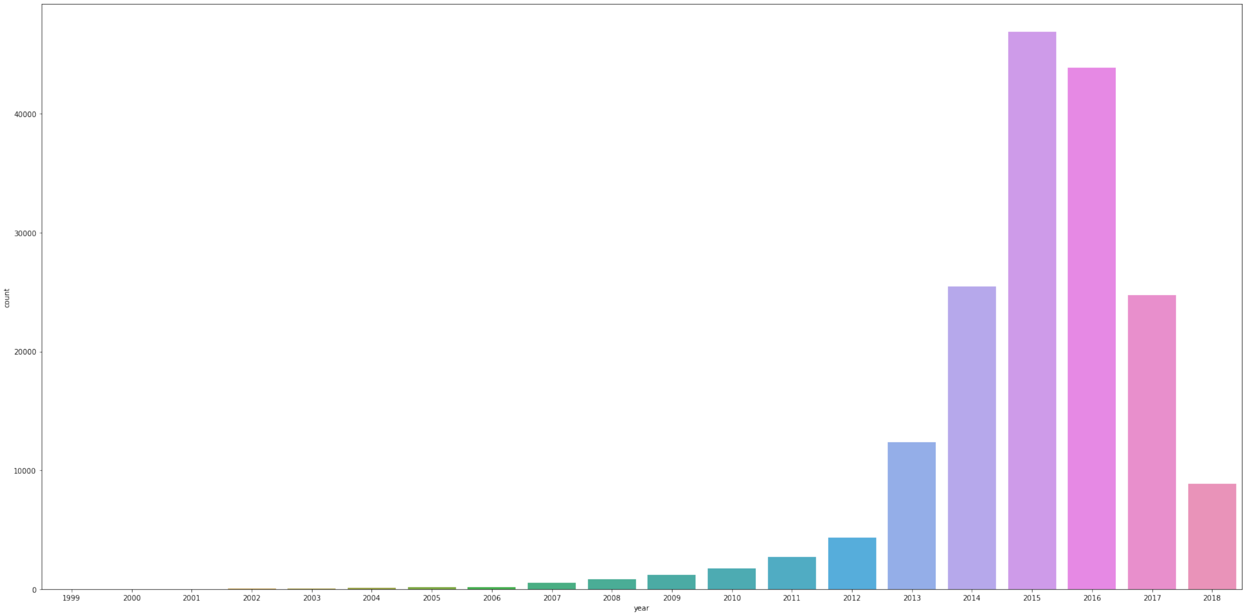


```
# the distribution of ratings

# The distribution of ratings is as follows:

# most of the ratings are 5

dataset['rating'].value_counts()
```

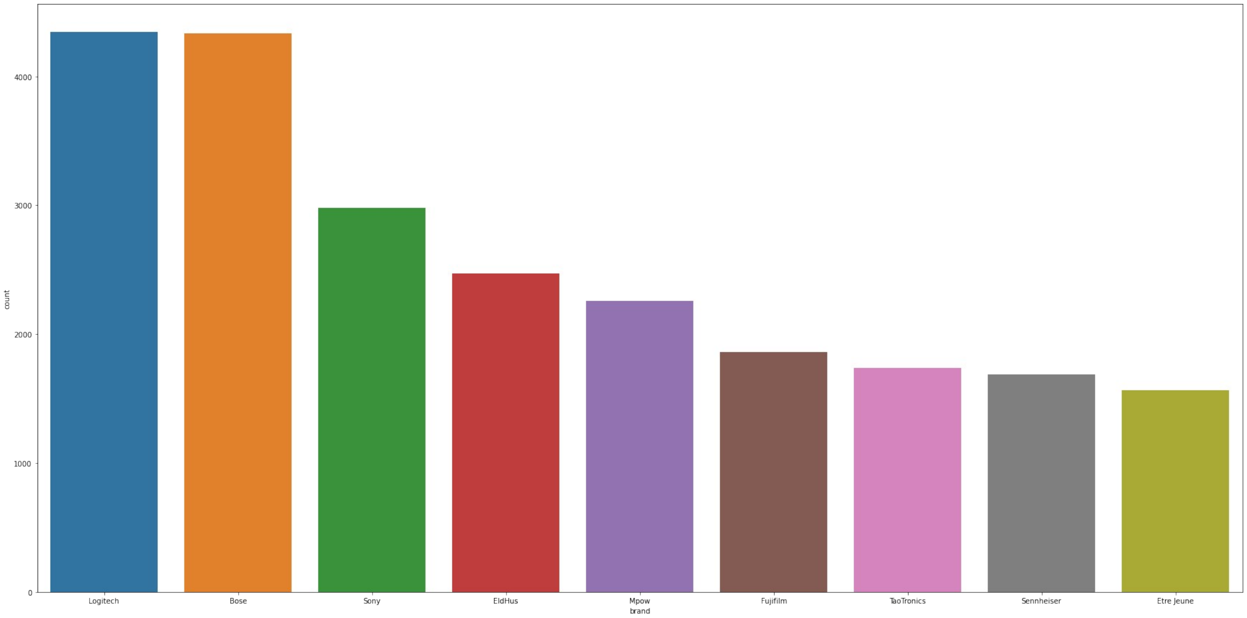
```
5.0    107593
4.0     30104
3.0     14593
1.0     12652
2.0      9182
Name: rating, dtype: int64
```

```
# the distribution of sales by year

sns.countplot(x='year', data=dataset)

# the distribution of sales by year

# The distribution of sales by year is as follows:

# 2015 was the best year of sales
```

```
<AxesSubplot:xlabel='year', ylabel='count'>
```

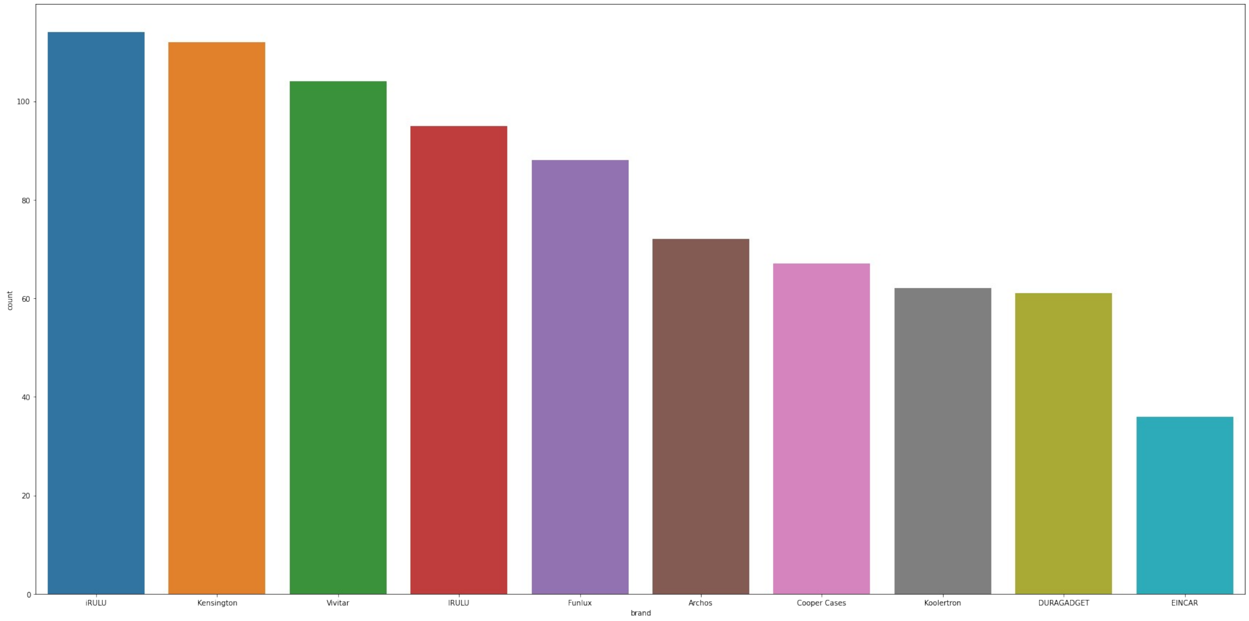


```
# brands with the most sales

sns.countplot(x='brand', data=dataset,
order=dataset['brand'].value_counts().iloc[1:10].index)

<AxesSubplot:xlabel='brand', ylabel='count'>
```

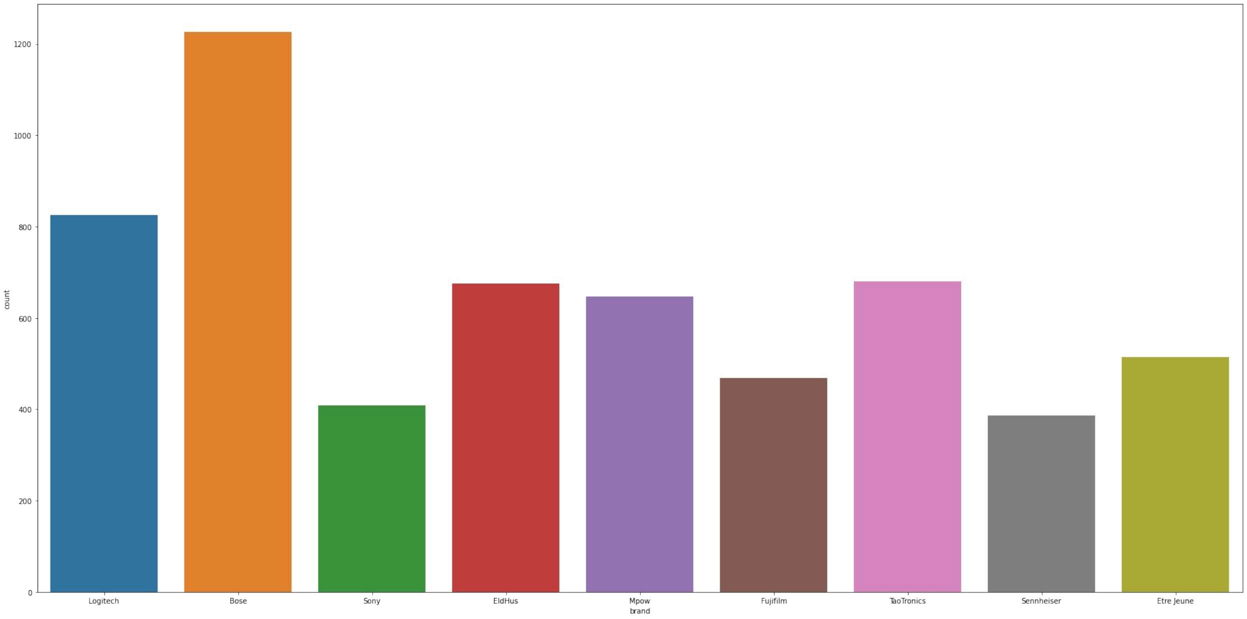


```
# What brand name sold the least?

sns.countplot(x='brand', data=dataset,
order=dataset['brand'].value_counts().iloc[-10:].index)
```

```
<AxesSubplot:xlabel='brand', ylabel='count'>
```

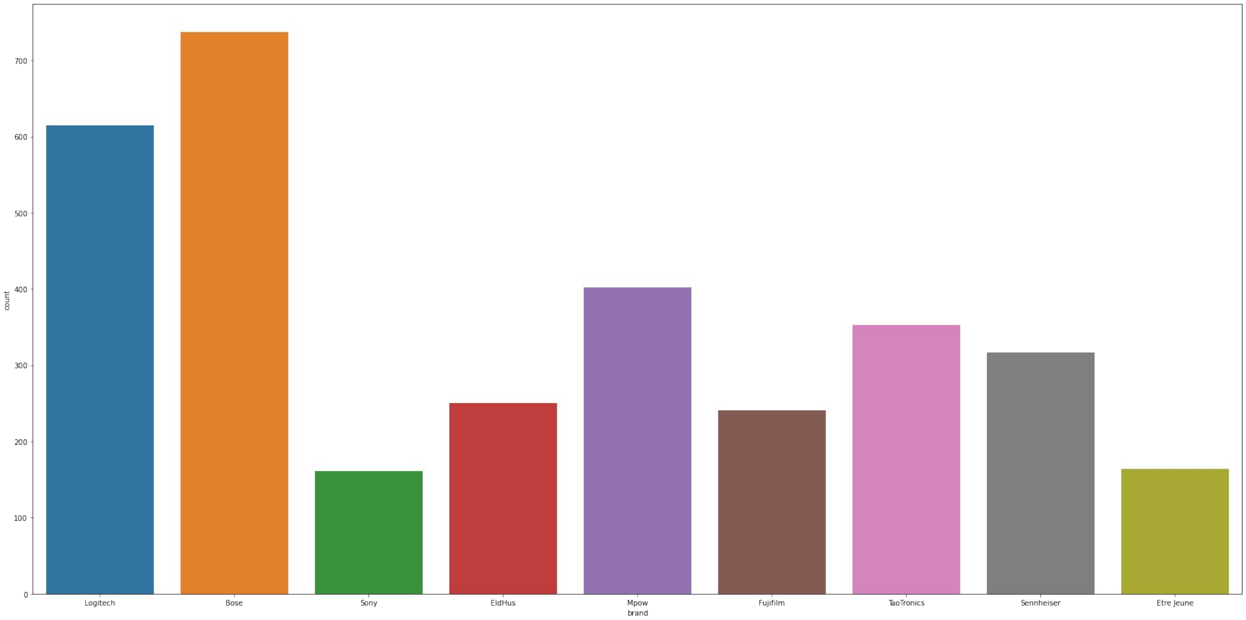


```
# We can see that the brand name of EINCAR sold the least followed
closely with DURAGADGET.

# Logitech & Bose had the most sales followed by Sony.

# brands with the most sales in 2016

sns.countplot(x='brand', data=dataset[dataset['year'] == 2016],
order=dataset['brand'].value_counts().iloc[1:10].index)

<AxesSubplot:xlabel='brand', ylabel='count'>
```

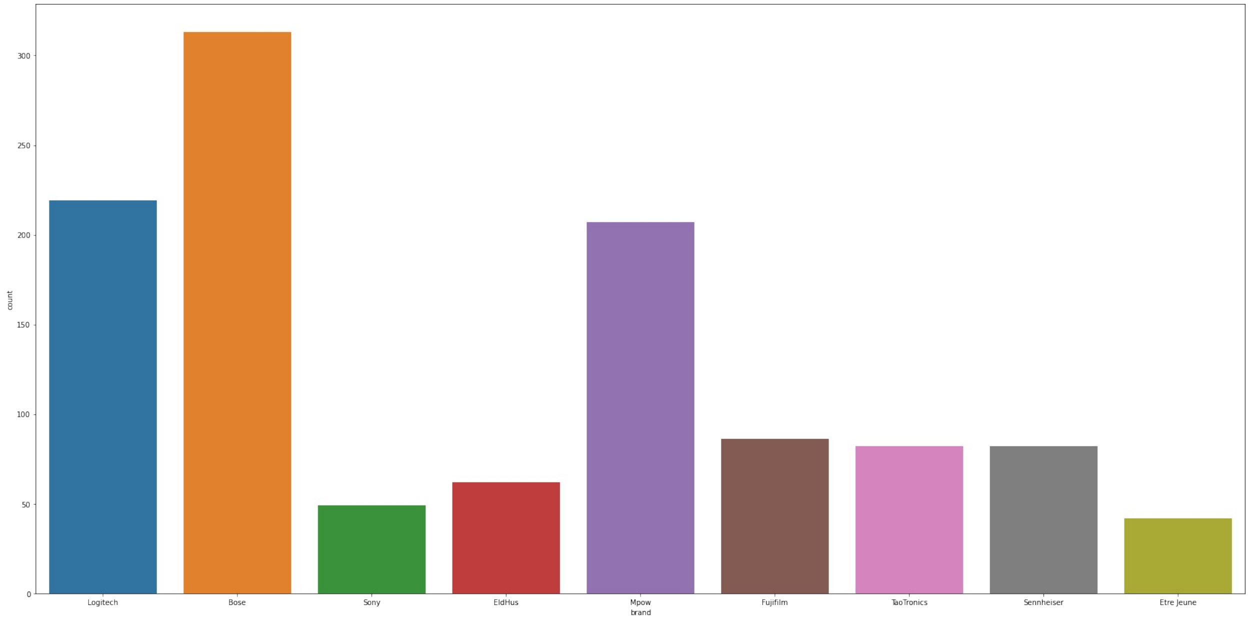


```
# in 2016 Bose overtook Logitech to have the most sales.

# TaoTronics had the third most sales that year

# brands with the most sales in 2017

sns.countplot(x='brand', data=dataset[dataset['year'] == 2017],
order=dataset['brand'].value_counts().iloc[1:10].index)
```

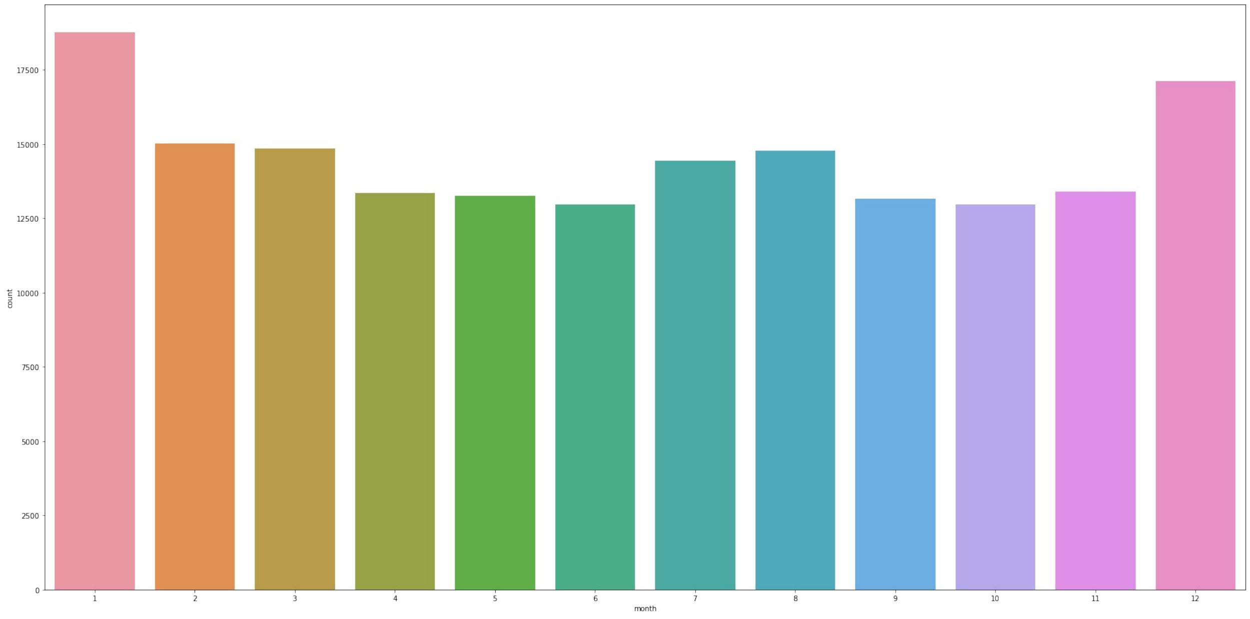
```
<AxesSubplot:xlabel='brand', ylabel='count'>
```



```
# the top 3 products sold in 2017 were Bose, Logitech and Mpow.
# brands with the most sales in 2018

sns.countplot(x='brand', data=dataset[dataset['year'] == 2018],
order=dataset['brand'].value_counts().iloc[1:10].index)

<AxesSubplot:xlabel='brand', ylabel='count'>
```

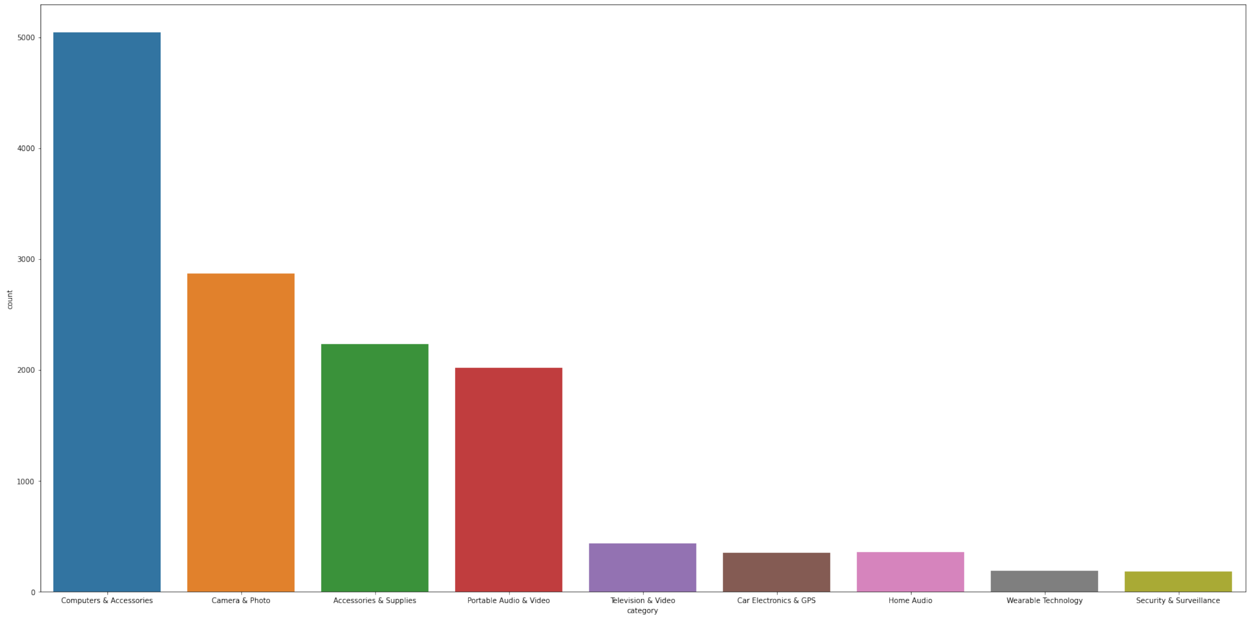


```
# For 2018, Bose was the most sold for a third year in a row followed
by Logitech while Mpow was the third most sold.

# month with most sales

sns.countplot(x='month', data=dataset)

<AxesSubplot:xlabel='month', ylabel='count'>
```



```
# January[#1] was the month with the most sales

# What products by category were sold the most in January

sns.countplot(x='category', data=dataset[dataset['month'] == 1],
order=dataset['category'].value_counts().iloc[1:10].index)
```

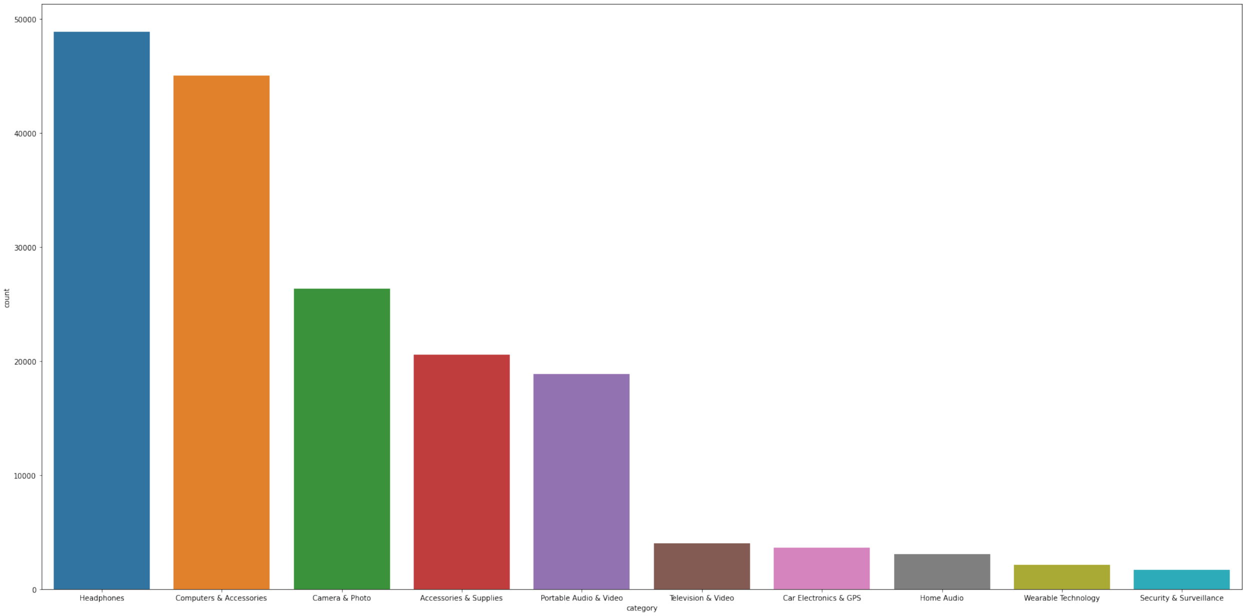
```
<AxesSubplot:xlabel='category', ylabel='count'>
```



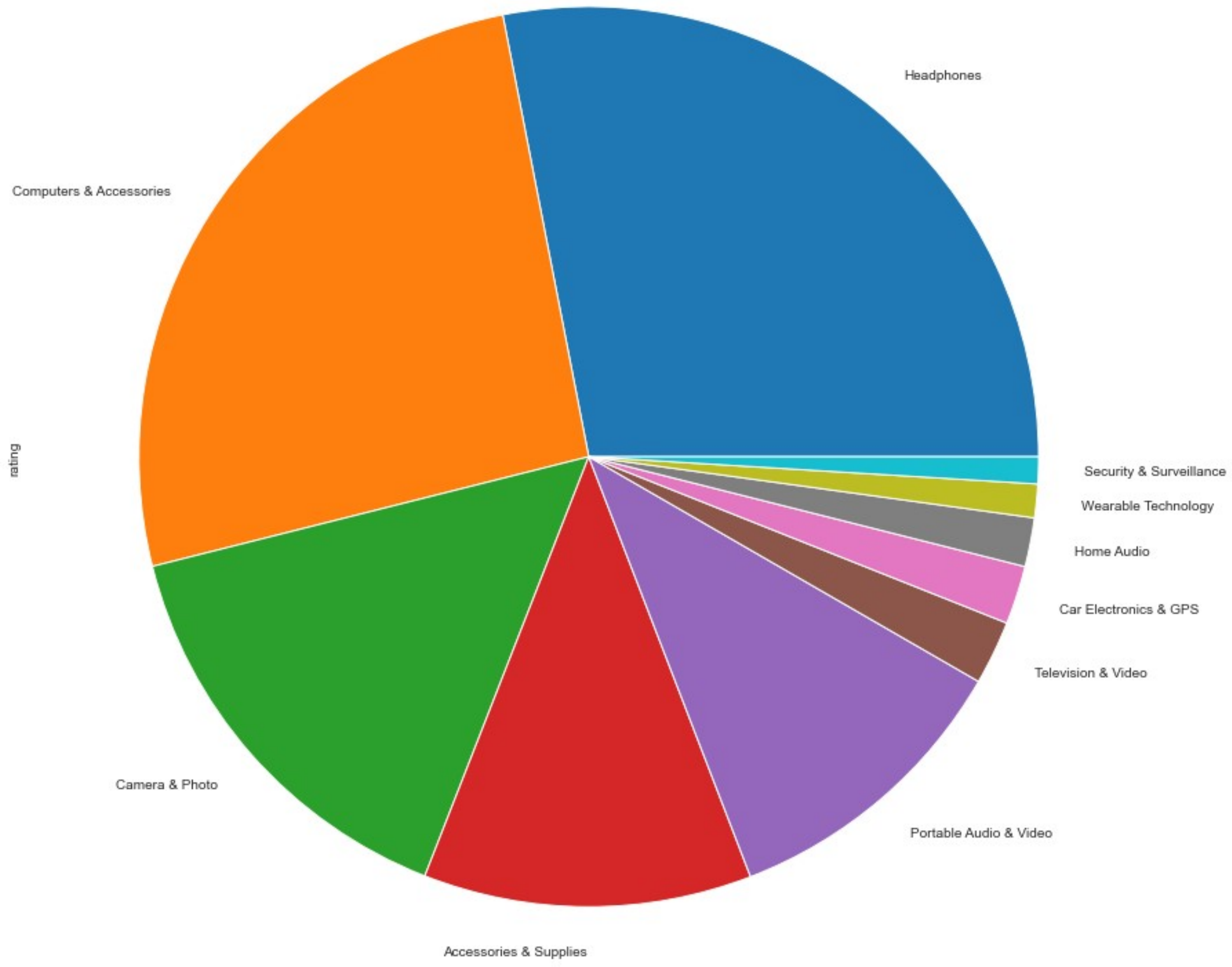
```
# The top 3 products sold in January were Computers & Accesories,
Camera & Photo and Accesories & Supplies.
```

```
# Category with the least sales

sns.countplot(x='category', data=dataset,
order=dataset['category'].value_counts().iloc[-10:].index)
```

```
<AxesSubplot:xlabel='category', ylabel='count'>
```



```
# The category with the least sales was Security & Surveillance while
the most sales were Headphones.

# distribution of sales presented in a pie chart

dataset['category'].value_counts(normalize=True)
dataset.groupby('category')
['rating'].count().sort_values(ascending=False).head(10).plot(kind='pi
e')

# white background

sns.set_style('white')
```



```
# conclusion of our analysis

# We can see that the year 2015 had the best sales.

# The month of January had the best sales.

# We can see that the brands Bose and Logitech sold the most

# We can see that the category of Headphones sold the most.

# We can see that the brand name of EINCAR sold the least followed closely with DURAGADGET.

# We can see that the category of Security and Surveillance sold the least.
```