In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [2]:
```
data=pd.read_csv('googleplaystore.csv')
data.head(2)
```

Out[2]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Las Update
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19M	10,000+	Free	0	Everyone	Art & Design	Januar 7, 201
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14M	500,000+	Free	0	Everyone	Art & Design;Pretend Play	Januar 15, 201

In [3]:
```
data.shape
```

Out[3]:
```
(10841, 13)
```

In [4]:
```
data.isnull().sum().sum()#Total number of null values present in the data.
```

Out[4]:
```
1487
```

In [3]:
```
data.isnull().sum()#Null values in each column.
```

Out[3]:
```
App                0
Category           0
Rating          1474
Reviews            0
Size               0
Installs           0
Type               1
Price              0
Content Rating     1
Genres             0
Last Updated       0
Current Ver        8
Android Ver        3
dtype: int64
```

In [4]:
```
data.dropna(inplace=True)#Dropping the null values from each columns.
```

In [7]:
```
data.isnull().sum()#Now,no null value is present in the dataset.
```

Out[7]:
```
App             0
Category        0
Rating          0
Reviews         0
Size            0
Installs        0
Type            0
Price           0
Content Rating  0
Genres          0
Last Updated    0
Current Ver     0
```

```
Android Ver       0
dtype: int64
```

In [10]:
```
data[data.duplicated()]
```

Out[10]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Up
229	Quick PDF Scanner + OCR FREE	BUSINESS	4.2	80805	Varies with device	5,000,000+	Free	0	Everyone	Business	Fe 26
236	Box	BUSINESS	4.2	159872	Varies with device	10,000,000+	Free	0	Everyone	Business	J
239	Google My Business	BUSINESS	4.4	70991	Varies with device	5,000,000+	Free	0	Everyone	Business	J
256	ZOOM Cloud Meetings	BUSINESS	4.4	31614	37M	10,000,000+	Free	0	Everyone	Business	J
261	join.me - Simple Meetings	BUSINESS	4.0	6989	Varies with device	1,000,000+	Free	0	Everyone	Business	J
...	...	...	...	...	...	...	...	...	...	...	
8643	Wunderlist: To-Do List & Tasks	PRODUCTIVITY	4.6	404610	Varies with device	10,000,000+	Free	0	Everyone	Productivity	A
8654	TickTick: To Do List with Reminder, Day Planner	PRODUCTIVITY	4.6	25370	Varies with device	1,000,000+	Free	0	Everyone	Productivity	A 6
8658	ColorNote Notepad Notes	PRODUCTIVITY	4.6	2401017	Varies with device	100,000,000+	Free	0	Everyone	Productivity	Ju
10049	Airway Ex - Intubate. Anesthetize. Train.	MEDICAL	4.3	123	86M	10,000+	Free	0	Everyone	Medical	J
10768	AAFP	MEDICAL	3.8	63	24M	10,000+	Free	0	Everyone	Medical	Ju

474 rows × 13 columns

In [9]:
```
data[data.duplicated(keep='last')]
```

Out[9]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Gen
164	Ebook Reader	BOOKS_AND_REFERENCE	4.1	85842	37M	5,000,000+	Free	0	Everyone	Book Refere
192	Docs To Go™ Free Office Suite	BUSINESS	4.1	217730	Varies with device	50,000,000+	Free	0	Everyone	Busin
193	Google My Business	BUSINESS	4.4	70991	Varies with device	5,000,000+	Free	0	Everyone	Busin
204	Box	BUSINESS	4.2	159872	Varies with device	10,000,000+	Free	0	Everyone	Busin

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Gen
213	ZOOM Cloud Meetings	BUSINESS	4.4	31614	37M	10,000,000+	Free	0	Everyone	Busin
...	...	...	...	...	...	...	...	...	...	
3787	ABC News - US & World News	NEWS_AND_MAGAZINES	4.0	18976	35M	1,000,000+	Free	0	Everyone 10+	New Magazi
3788	NBC News	NEWS_AND_MAGAZINES	4.1	63020	Varies with device	5,000,000+	Free	0	Everyone 10+	New Magazi
3790	USA TODAY	NEWS_AND_MAGAZINES	4.1	49259	Varies with device	5,000,000+	Free	0	Everyone 10+	New Magazi
3792	CNN Breaking US & World News	NEWS_AND_MAGAZINES	4.0	293080	25M	10,000,000+	Free	0	Everyone 10+	New Magazi
3795	Newsroom: News Worth Sharing	NEWS_AND_MAGAZINES	4.2	201737	Varies with device	10,000,000+	Free	0	Everyone 10+	New Magazi

474 rows × 13 columns

In [8]:
```
data.shape#This is the shape after remove the missing values.
```

Out[8]:
```
(9360, 13)
```

In [22]:
```
#s_mul_thou(size multiply by thousand) is a function name.
#We have to multiply those values by 1000 which are in Mb.
#Values which are in Kb are as it is.
#Now, convert size column dtype in numeric form which is present in object.
#Rest of the values in size column are empty.(i.e NaN)
def s_mul_thou(Size):
    if 'M' in Size:
        x=Size[:-1]
        x=float(x)*1000
        return x

    elif 'k' in Size:
        x=Size[:-1]
        x=float(x)
        return x

    else:
        return None
```

In [23]:
```
data['Size']=data['Size'].apply(s_mul_thou)#Now apply this function in size column.
```

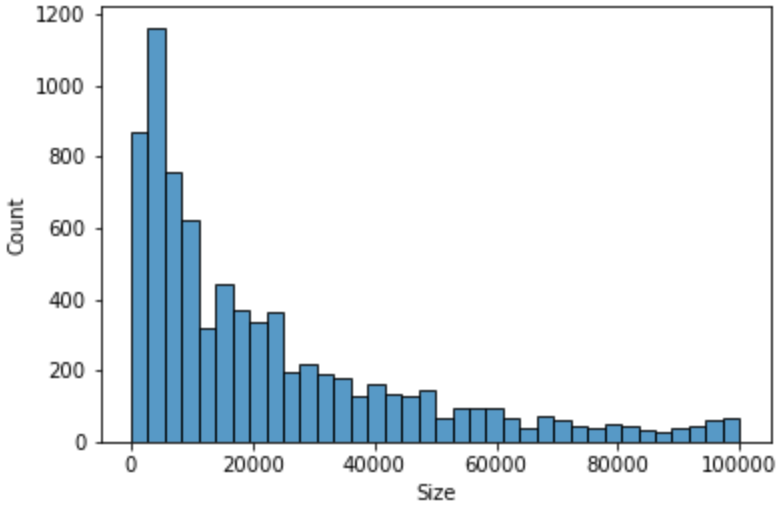
In [11]:
```
data.head(2)#Now Size is a numeric column as float64 dtype.
```

Out[11]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Upd

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Upd
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19000.0	10,000+	Free	0	Everyone	Art & Design	Jan 7, 2
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14000.0	500,000+	Free	0	Everyone	Art & Design;Pretend Play	Jan 15, 2

In [12]:
```
sns.histplot(data['Size'])
plt.show()
```



In [24]:
```
#But now some records are null in size column, and we have to fill it by fillna method.
#We use median method because the size column is a positive skewed data.(As shown in histo
data['Size'].fillna(data['Size'].median(),inplace=True)
```

In [14]:
```
data.tail(2)
```

Out[14]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	G
10839	The SCP Foundation DB fr nn5n	BOOKS_AND_REFERENCE	4.5	114	14000.0	1,000+	Free	0	Mature 17+	Bo Refe
10840	iHoroscope - 2018 Daily Horoscope & Astrology	LIFESTYLE	4.5	398307	19000.0	10,000,000+	Free	0	Everyone	Lif

In [26]:
```
data['Installs']
```

Out[26]:
```
0            10,000+
1           500,000+
2         5,000,000+
3        50,000,000+
4           100,000+
             ...
10834           500+
10836         5,000+
10837           100+
10839         1,000+
```

```
10840    10,000,000+
Name: Installs, Length: 9360, dtype: object
```

In [35]:
```
#Convert the string field in numeric field. Replace is the method to resolve this query.
#Regular expression must be true.
data['Reviews']=pd.to_numeric(data['Reviews'].replace('[^0-9]','',regex=True))
```

In [30]:
```
data['Price'].tail(50)#We have to remove the '$' sign from the values whenever it appears.
```

Out[30]:
```
10768         0
10770         0
10771         0
10776         0
10777         0
10778         0
10779         0
10780         0
10781         0
10782    $16.99
10783         0
10784         0
10785     $1.20
10786         0
10787         0
10789         0
10790         0
10791         0
10792         0
10793         0
10795         0
10796         0
10797         0
10799         0
10800         0
10801         0
10802         0
10803         0
10804         0
10805         0
10809         0
10810         0
10812         0
10814         0
10815         0
10817         0
10819         0
10820         0
10826         0
10827         0
10828         0
10829         0
10830         0
10832         0
10833         0
10834         0
10836         0
10837         0
10839         0
10840         0
Name: Price, dtype: object
```

In [31]:
```
#Convert the string field in numeric field. Replace is the method to resolve this query.
#Regular expression must be true.
data['Price']=pd.to_numeric(data['Price'].replace('[^0-9.]','',regex=True))
```

In [33]:
```
data['Price'].tail(50)
```

```
10770     0.00
10771     0.00
10776     0.00
10777     0.00
10778     0.00
10779     0.00
10780     0.00
10781     0.00
10782    16.99
10783     0.00
10784     0.00
10785     1.20
10786     0.00
10787     0.00
10789     0.00
10790     0.00
10791     0.00
10792     0.00
10793     0.00
10795     0.00
10796     0.00
10797     0.00
10799     0.00
10800     0.00
10801     0.00
10802     0.00
10803     0.00
10804     0.00
10805     0.00
10809     0.00
10810     0.00
10812     0.00
10814     0.00
10815     0.00
10817     0.00
10819     0.00
10820     0.00
10826     0.00
10827     0.00
10828     0.00
10829     0.00
10830     0.00
10832     0.00
10833     0.00
10834     0.00
10836     0.00
10837     0.00
10839     0.00
10840     0.00
Name: Price, dtype: float64
```

In [28]:
```
data['Installs']
```

Out[28]:
```
0          10000
1         500000
2        5000000
3       50000000
4         100000
          ...
10834        500
10836       5000
10837        100
10839       1000
10840    10000000
Name: Installs, Length: 9360, dtype: int64
```

In [27]:
```
#Convert the string field in numeric field(i.e integer). Replace is the method to resolve
#We have to remove the '+' sign from the values when it appears.
```

```
#Regular expression must be true.
data['Installs']=pd.to_numeric(data['Installs'].replace('[^0-9]','',regex=True))
```

In [36]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 9360 entries, 0 to 10840
Data columns (total 13 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   App             9360 non-null   object 
 1   Category        9360 non-null   object 
 2   Rating          9360 non-null   float64
 3   Reviews         9360 non-null   int64  
 4   Size            9360 non-null   float64
 5   Installs        9360 non-null   int64  
 6   Type            9360 non-null   object 
 7   Price           9360 non-null   float64
 8   Content Rating  9360 non-null   object 
 9   Genres          9360 non-null   object 
 10  Last Updated    9360 non-null   object 
 11  Current Ver     9360 non-null   object 
 12  Android Ver     9360 non-null   object 
dtypes: float64(3), int64(2), object(8)
memory usage: 1023.8+ KB
```

In [37]:
```
data['Rating'].unique()
```

Out[37]:
```
array([4.1, 3.9, 4.7, 4.5, 4.3, 4.4, 3.8, 4.2, 4.6, 4. , 4.8, 4.9, 3.6,
       3.7, 3.2, 3.3, 3.4, 3.5, 3.1, 5. , 2.6, 3. , 1.9, 2.5, 2.8, 2.7,
       1. , 2.9, 2.3, 2.2, 1.7, 2. , 1.8, 2.4, 1.6, 2.1, 1.4, 1.5, 1.2])
```

In [38]:
```
data[data['Reviews']>data['Installs']]#Reviews column values are more than Installs columr
```

Out[38]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver
2454	KBA-EZ Health Guide	MEDICAL	5.0	4	25000.0	1	Free	0.00	Everyone	Medical	August 2, 2018	1.0.72
4663	Alarmy (Sleep If U Can) - Pro	LIFESTYLE	4.8	10249	14000.0	10000	Paid	2.49	Everyone	Lifestyle	July 30, 2018	Varies with device
5917	Ra Ga Ba	GAME	5.0	2	20000.0	1	Paid	1.49	Everyone	Arcade	February 8, 2017	1.0.4
6700	Brick Breaker BR	GAME	5.0	7	19000.0	5	Free	0.00	Everyone	Arcade	July 23, 2018	1.0
7402	Trovami se ci riesci	GAME	5.0	11	6100.0	10	Free	0.00	Everyone	Arcade	March 11, 2017	0.1
8591	DN Blog	SOCIAL	5.0	20	4200.0	10	Free	0.00	Teen	Social	July 23, 2018	1.0
10697	Mu.F.O.	GAME	5.0	2	16000.0	1	Paid	0.99	Everyone	Arcade	March 3, 2017	1.0

In [39]:
```
data[data['Reviews']>data['Installs']].shape[0]
```

Out[39]:
```
7
```

In [40]:
```
#Drop records when reviews greater than installs.
#This '~' sign refer to the dropping the records permanently.
data=data[~(data['Reviews']>data['Installs'])]
```

In [41]:
```
data.shape[0]#Now, there are 9353 records in the dataset.
```

Out[41]:
```
9353
```

In [42]:
```
data[(data['Type']=='Free')&(data['Price']>0)]
```

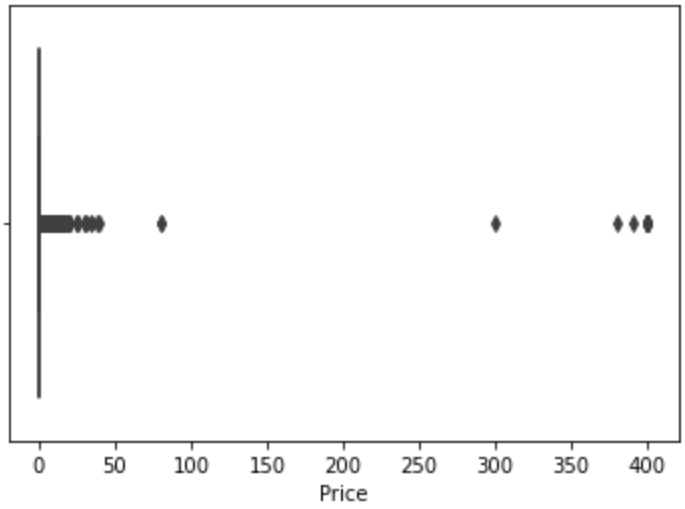
Out[42]:

App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver

Boxplot for Price

In [43]:
```
sns.boxplot(data['Price'])
plt.show()
```

```
C:\ProgramData\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 positional a
rgument will be `data`, and passing other arguments without an explicit keyword will resul
t in an error or misinterpretation.
  warnings.warn(
```



In [45]:
```
data['Price'].value_counts()
```

Out[45]:
```
0.00      8711
2.99       114
0.99       105
4.99        70
1.99        59
          ...
299.99       1
1.59         1
1.61         1
3.90         1
2.90         1
Name: Price, Length: 73, dtype: int64
```

In [52]:
```
data.loc[data['Price']>200,['App','Price']]#Some apps prices are very high.
```

Out[52]:

	App	Price
4197	most expensive app (H)	399.99
4362	I'm rich	399.99

	App	Price
4367	I'm Rich - Trump Edition	400.00
5351	I am rich	399.99
5354	I am Rich Plus	399.99
5355	I am rich VIP	299.99
5356	I Am Rich Premium	399.99
5357	I am extremely Rich	379.99
5358	I am Rich!	399.99
5359	I am rich(premium)	399.99
5362	I Am Rich Pro	399.99
5364	I am rich (Most expensive app)	399.99
5366	I Am Rich	389.99
5369	I am Rich	399.99
5373	I AM RICH PRO PLUS	399.99

In [53]:
```
data.loc[data['Price']>200,['App','Price']].shape[0]#There are 15 records when prices are
```

Out[53]:
```
15
```

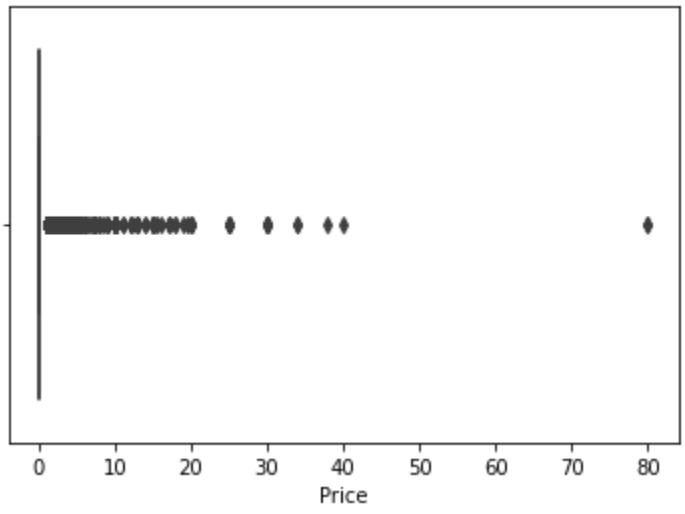
In [54]:
```
data=data[~(data['Price']>200)]#Drop records where price is greater than 200(Price).
```

In [55]:
```
data.shape
```

Out[55]:
```
(9338, 13)
```

In [56]:
```
#After removing records where price greater than 200, the boxplot look like this.
sns.boxplot(data.Price)
plt.show()
```

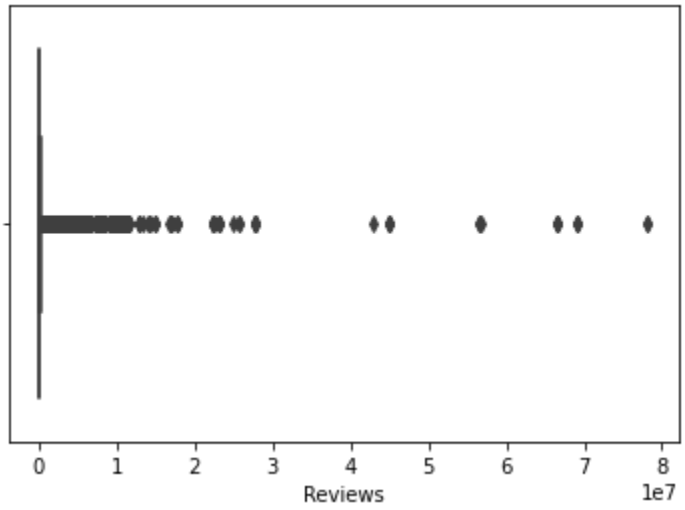
```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass
the following variable as a keyword arg: x. From version 0.12, the only valid positional a
rgument will be `data`, and passing other arguments without an explicit keyword will resul
t in an error or misinterpretation.
  warnings.warn(
```



Boxplot for Reviews

In [46]:
```
sns.boxplot(data.Reviews)
plt.show()
```

```
C:\ProgramData\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 positional a
rgument will be `data`, and passing other arguments without an explicit keyword will resul
t in an error or misinterpretation.
  warnings.warn(
```



Yes, there are some apps with very high price. There are values which are very high from the lowest value.

There are lot of values present under the 10 millions reviews.

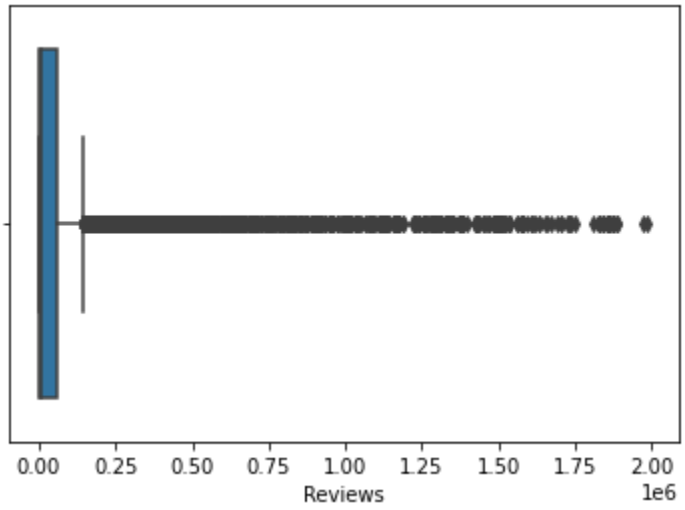
In [57]:
```
data=data[~(data['Reviews']>2000000)]#Dropping records when reviews are greater than 2 Mi
```

In [58]:
```
data.shape
```

Out[58]:
```
(8885, 13)
```

In [59]:
```
#After removing records where Reviews greater than 2 Millions, the boxplot look like this.
sns.boxplot(data.Reviews)
plt.show()
```

```
C:\ProgramData\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 positional a
rgument will be `data`, and passing other arguments without an explicit keyword will resul
t in an error or misinterpretation.
  warnings.warn(
```



In [60]:
```
data.shape
```

Out[60]:
```
(8885, 13)
```

In [114…
```
IQR=data['Reviews'].quantile(0.75)-data['Reviews'].quantile(0.25)
IQR
```

Out[114…
```
71363.0
```

In [115…
```
upper=data['Reviews'].quantile(0.75)+(1.5*IQR)
lower=data['Reviews'].quantile(0.25)-(1.5*IQR)
print(upper)
print(lower)
```

```
178573.5
-106878.5
```

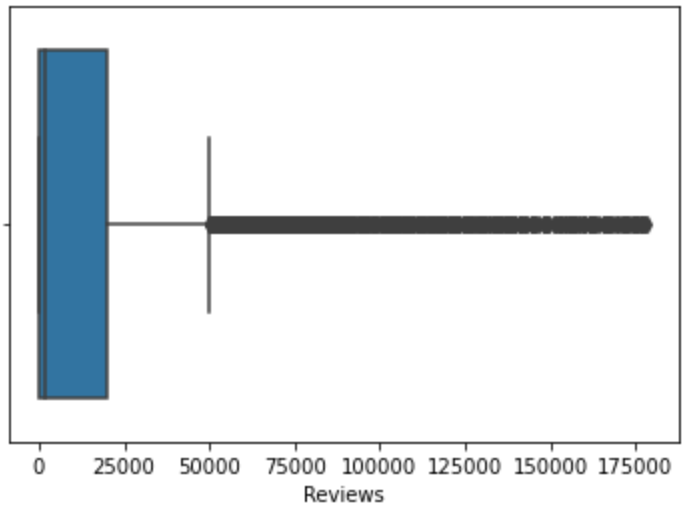
In [116…
```
data.drop(data[data['Reviews']>178573.5].index,inplace=True)
```

In [57]:
```
data.shape
```

Out[57]:
```
(7047, 13)
```

In [117…
```
sns.boxplot(data.Reviews)
plt.show()
```

```
C:\ProgramData\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 positional a
rgument will be `data`, and passing other arguments without an explicit keyword will resul
t in an error or misinterpretation.
  warnings.warn(
```

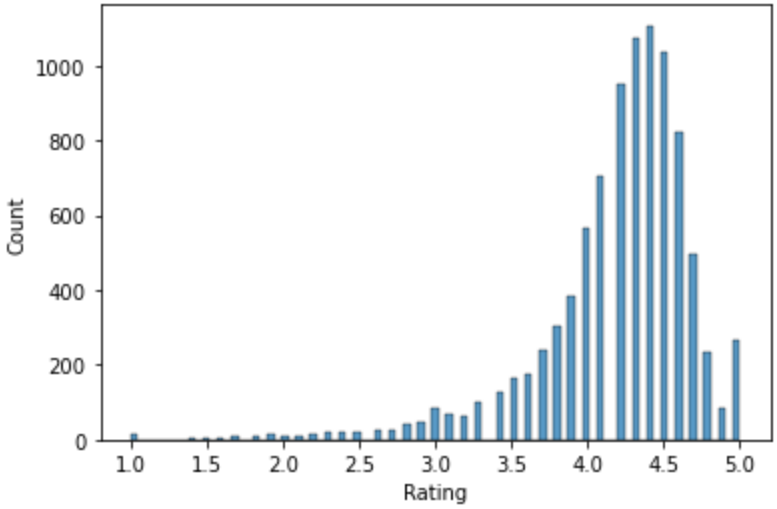


In [119…
```
data[data['Reviews']>=50000].shape[0]
```

Out[119…
```
971
```

Histogram for rating

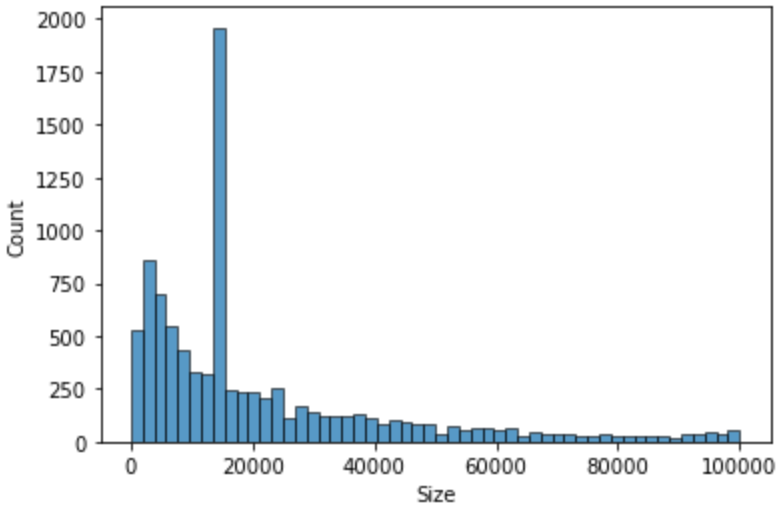
In [47]:
```
sns.histplot(data.Rating)
plt.show()
```



Between 4.0 and 4.5, there are maximum numbers of rating of an app. There are apps which has a low number of rating between 1.0 to 3.8, which is under 400 apps. Yes, this is a negative skewed data, which is more towards high rating.

Histogram for size

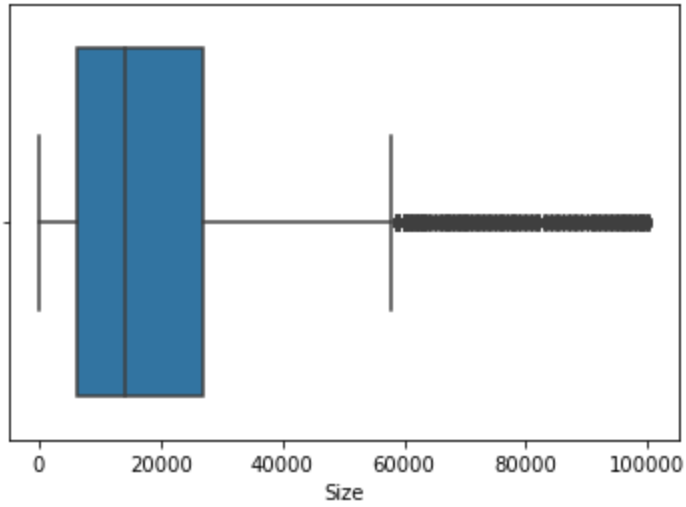
In [48]:
```
sns.histplot(data.Size)
plt.show()
```



This is a positive skewed data. 0 to 20,000 have the high values in size.

In [92]:
```
sns.boxplot(data.Size)
plt.show()
```

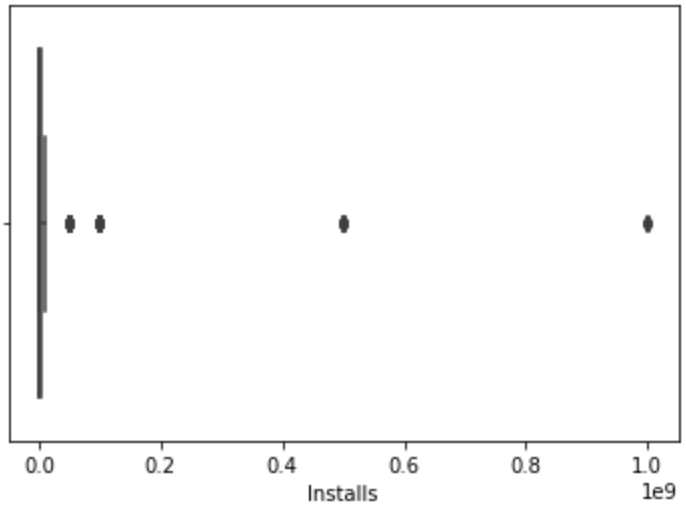
```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass
the following variable as a keyword arg: x. From version 0.12, the only valid positional a
rgument will be `data`, and passing other arguments without an explicit keyword will resul
t in an error or misinterpretation.
  warnings.warn(
```



Boxplot for Installs

In [62]:
```
#There are some outliers present in Installs columns.
sns.boxplot(data.Installs)
plt.show()
```

```
C:\ProgramData\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 positional a
rgument will be `data`, and passing other arguments without an explicit keyword will resul
t in an error or misinterpretation.
  warnings.warn(
```



In [63]:
```
data['Installs'].quantile([0.1,0.25,0.5,0.7,0.9,0.95,0.99])
```

Out[63]:
```
0.10         1000.0
0.25        10000.0
0.50       500000.0
0.70      1000000.0
0.90     10000000.0
0.95     10000000.0
0.99    100000000.0
Name: Installs, dtype: float64
```

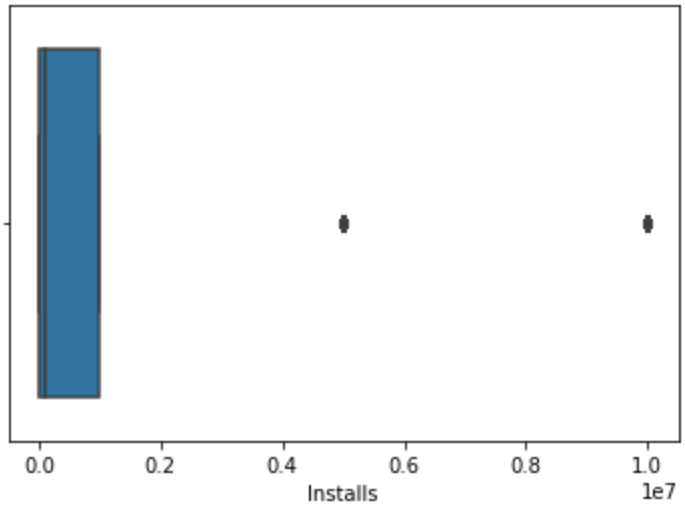
In [64]:
```
data=data[~(data['Installs']>10000000.0)]#95% is the cutoff threshold for outliers and rem
```

In [65]:
```
data.shape
```

Out[65]:
```
(8496, 13)
```

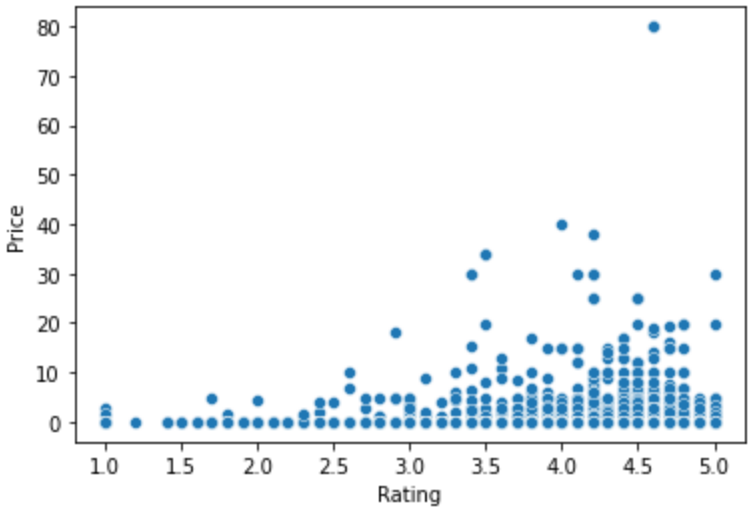
In [66]:
```
sns.boxplot(data.Installs)
plt.show()
```

```
C:\ProgramData\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 positional a
rgument will be `data`, and passing other arguments without an explicit keyword will resul
t in an error or misinterpretation.
  warnings.warn(
```



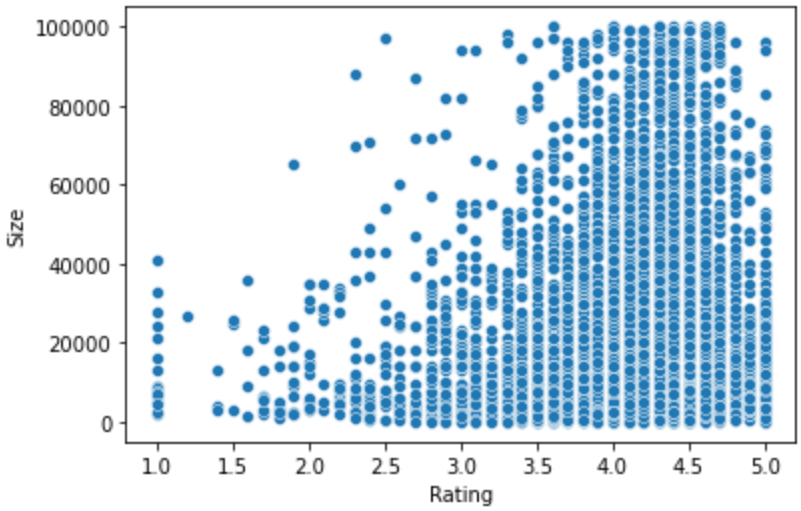
Bivariate analysis

In [67]:
```
sns.scatterplot(x=data['Rating'],y=data['Price'])
plt.show()
```



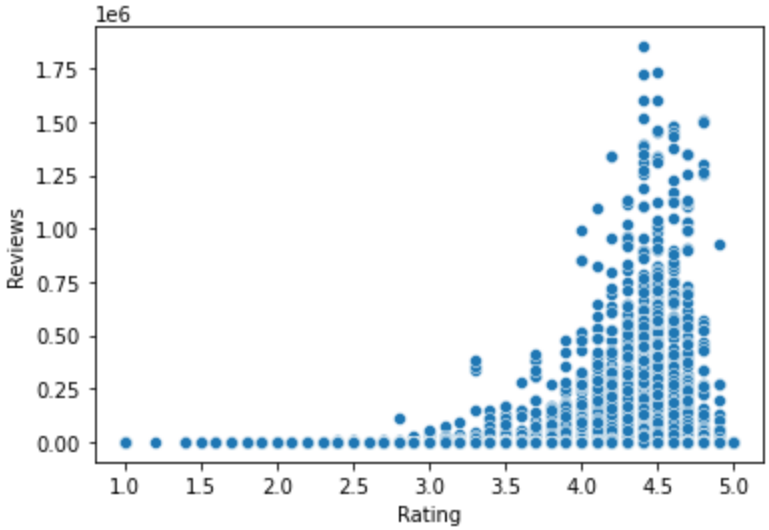
Yes, when rating increases price is also increase, but in most cases when rating increase the price is 0.

In [68]:
```
sns.scatterplot(x=data['Rating'],y=data['Size'])
plt.show()
```



Yes, heavier apps rated better, but also smaller size apps rated better too.

In [69]:
```
sns.scatterplot(x=data['Rating'],y=data['Reviews'])
plt.show()
```

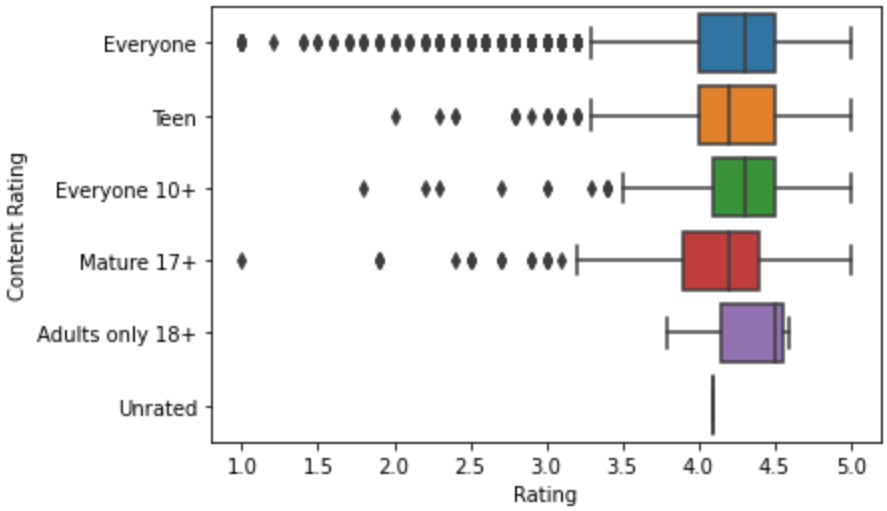


No, it is always not possible. As we can see lower reviews are also getting the high rating.

In [59]:
```
data['Content Rating'].value_counts()
```

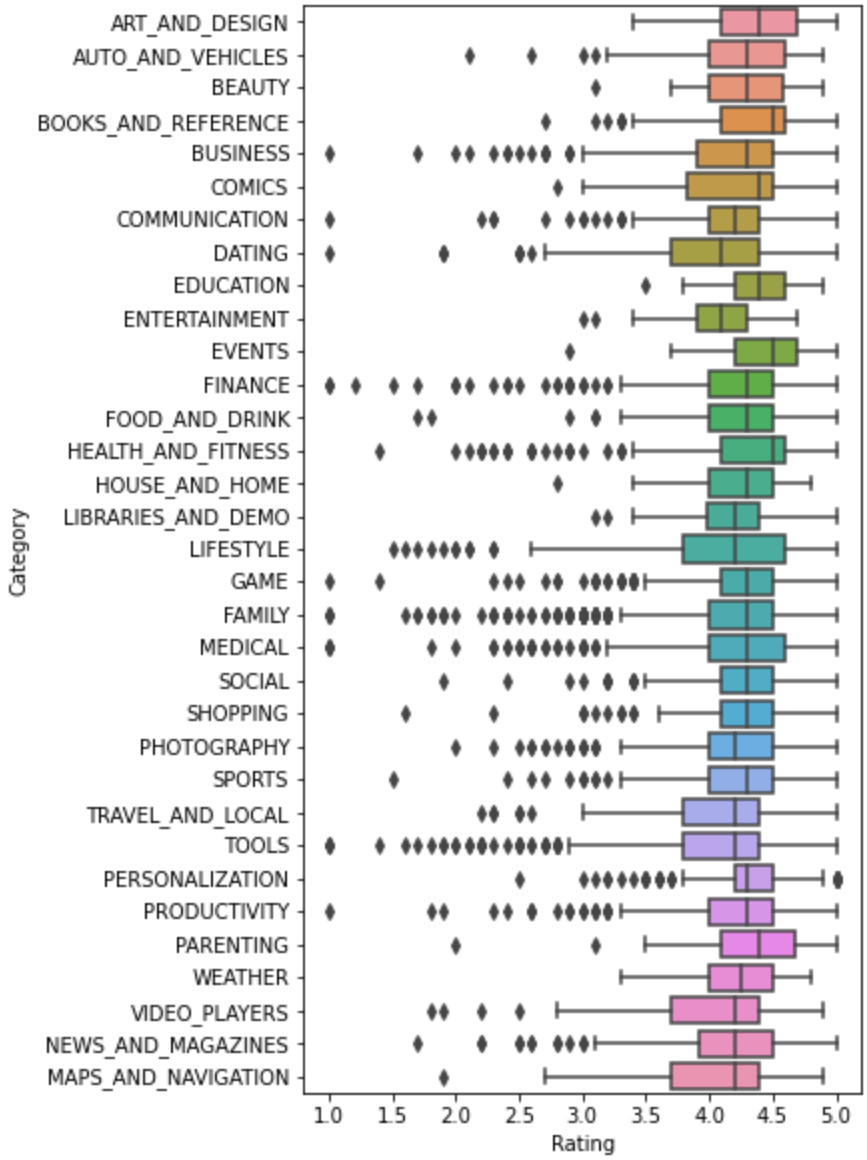
Out[59]:
```
Everyone           5197
Teen                603
Mature 17+          286
Everyone 10+        208
Adults only 18+       3
Unrated               1
Name: Content Rating, dtype: int64
```

In [70]:
```
sns.boxplot(x=data['Rating'],y=data['Content Rating'])
plt.show()
```



Yes, there are difference in the ratings. The value 'Everyone' has maximum number of outliers as compared to others. Some values such as ''Everyone','Everyone 10+','Teen','Mature 17+' have same maximum rating (i.e 5.0). Yes, some types are better like 'Adults only 18+','Unrated'. These values have less rating as compared to others.

In [71]:
```
plt.figure(figsize=(5,10))
sns.boxplot(x=data['Rating'],y=data['Category'])
plt.show()
```



Category 'Art and design' has the best ratings among all, because this category has maximum rating and lowest minimum rating. There are two categories like 'art and design' and 'weather' which have no outliers. There are several categories which have 5.0 rating as compared to others.

In [126…
```
data
```

Out[126…

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19000.0	10000	Free	0.0	Everyone	Art
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14000.0	500000	Free	0.0	Everyone	Desig
2	U Launcher Lite – FREE Live Cool Themes, Hide ...	ART_AND_DESIGN	4.7	87510	8700.0	5000000	Free	0.0	Everyone	Art
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	967	2800.0	100000	Free	0.0	Everyone	Design
5	Paper flowers instructions	ART_AND_DESIGN	4.4	167	5600.0	50000	Free	0.0	Everyone	Art
...	...	...	...	...	...	...	...	...	...	
10834	FR Calculator	FAMILY	4.0	7	2600.0	500	Free	0.0	Everyone	
10836	Sya9a Maroc - FR	FAMILY	4.5	38	53000.0	5000	Free	0.0	Everyone	
10837	Fr. Mike Schmitz Audio Teachings	FAMILY	5.0	4	3600.0	100	Free	0.0	Everyone	
10839	The SCP Foundation DB fr nn5n	BOOKS_AND_REFERENCE	4.5	114	14000.0	1000	Free	0.0	Mature 17+	
10840	iHoroscope - 2018 Daily Horoscope & Astrology	LIFESTYLE	4.5	398307	19000.0	10000000	Free	0.0	Everyone	

8496 rows × 13 columns

In [99]:
```
inp1=data.copy()
```

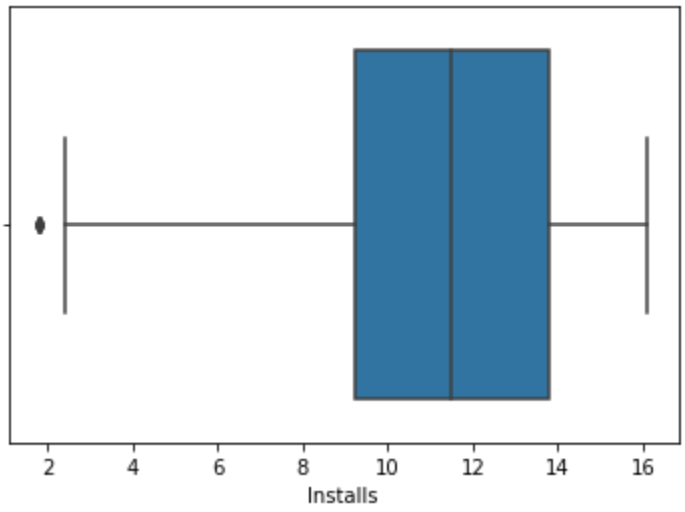
In [100…
```
inp1['Reviews']=np.log1p(inp1.Reviews)
inp1['Installs']=np.log1p(inp1.Installs)
```

Out[101…

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Up
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	5.075174	19000.0	9.210440	Free	0.0	Everyone	Art & Design	Ja 7
1	Coloring book moana	ART_AND_DESIGN	3.9	6.875232	14000.0	13.122365	Free	0.0	Everyone	Art & Design;Pretend Play	Ja 15

In [102…
```
sns.boxplot(inp1['Installs'])#Skewness of reviews column is reduced
plt.show()
```

```
C:\ProgramData\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 positional a
rgument will be `data`, and passing other arguments without an explicit keyword will resul
t in an error or misinterpretation.
  warnings.warn(
```



In [103…
```
inp1.drop(columns=['App','Last Updated','Current Ver','Android Ver','Type'],inplace=True)
```

In [104…
```
inp1.head(2)
```

Out[104…

	Category	Rating	Reviews	Size	Installs	Price	Content Rating	Genres
0	ART_AND_DESIGN	4.1	5.075174	19000.0	9.210440	0.0	Everyone	Art & Design
1	ART_AND_DESIGN	3.9	6.875232	14000.0	13.122365	0.0	Everyone	Art & Design;Pretend Play

In [105…
```
inp1.shape
```

Out[105… (8496, 8)

In [106…
```
inp2=pd.get_dummies(inp1,columns=['Category','Genres','Content Rating'])
```

In [107…
```
inp2.head(2)
```

Out[107…

	Rating	Reviews	Size	Installs	Price	Category_ART_AND_DESIGN	Category_AUTO_AND_VEHICLES	Cat
0	4.1	5.075174	19000.0	9.210440	0.0	1	0	

	Rating	Reviews	Size	Installs	Price	Category_ART_AND_DESIGN	Category_AUTO_AND_VEHICLES	Cat
1	3.9	6.875232	14000.0	13.122365	0.0	1	0	

2 rows × 159 columns

In [108…
```
inp2.shape
```

Out[108…
```
(8496, 159)
```

In [109…
```
from sklearn.model_selection import train_test_split
df_train,df_test=train_test_split(inp2,test_size=0.3,random_state=0)
```

In [110…
```
df_train
```

Out[110…

	Rating	Reviews	Size	Installs	Price	Category_ART_AND_DESIGN	Category_AUTO_AND_VEHICLES
6169	4.8	4.499810	14000.0	6.908755	0.0	0	0
5431	4.6	8.555644	64000.0	11.512935	0.0	0	0
9036	4.2	6.456770	4000.0	11.512935	0.0	0	0
6406	4.2	8.125631	62000.0	11.512935	0.0	0	0
10113	4.4	10.898275	21000.0	16.118096	0.0	0	0
...	...	...	...	...	...	...	...
5437	4.7	12.886243	75000.0	16.118096	0.0	0	0
10014	4.5	1.945910	22000.0	6.216606	0.0	0	0
6027	4.6	2.564949	2400.0	8.517393	0.0	0	0
4011	4.7	6.562444	26000.0	8.517393	0.0	0	0
3320	4.2	11.948461	14000.0	16.118096	0.0	0	0

5947 rows × 159 columns

In [111…
```
df_train.shape
```

Out[111…
```
(5947, 159)
```

In [112…
```
df_test
```

Out[112…

	Rating	Reviews	Size	Installs	Price	Category_ART_AND_DESIGN	Category_AUTO_AND_VEHICLES
5587	3.5	3.367296	121.0	6.908755	0.00	0	0
4732	4.5	10.534972	14000.0	13.815512	0.00	0	0
807	4.7	12.658106	3300.0	16.118096	0.00	0	0
6414	3.5	8.344267	10000.0	11.512935	0.00	0	0
4522	1.8	2.890372	3100.0	9.210440	0.00	0	0

	Rating	Reviews	Size	Installs	Price	Category_ART_AND_DESIGN	Category_AUTO_AND_VEHICLES
...	...	...	...	...	...	...	...
3420	4.1	11.967155	6400.0	16.118096	0.00	0	0
6142	4.5	11.304917	2600.0	16.118096	0.00	0	0
4946	4.8	8.637639	14000.0	9.210440	1.49	0	0
7375	4.2	6.118097	36000.0	9.210440	2.99	0	0
4865	4.3	7.395722	18000.0	11.512935	0.00	0	0

2549 rows × 159 columns

In [113…
```
df_test.shape
```

Out[113…
```
(2549, 159)
```

In [114…
```
y_train = df_train.pop('Rating')
X_train = df_train
```

In [115…
```
X_train
```

Out[115…

	Reviews	Size	Installs	Price	Category_ART_AND_DESIGN	Category_AUTO_AND_VEHICLES	Categ
6169	4.499810	14000.0	6.908755	0.0	0	0	
5431	8.555644	64000.0	11.512935	0.0	0	0	
9036	6.456770	4000.0	11.512935	0.0	0	0	
6406	8.125631	62000.0	11.512935	0.0	0	0	
10113	10.898275	21000.0	16.118096	0.0	0	0	
...	...	...	...	...	...	...	
5437	12.886243	75000.0	16.118096	0.0	0	0	
10014	1.945910	22000.0	6.216606	0.0	0	0	
6027	2.564949	2400.0	8.517393	0.0	0	0	
4011	6.562444	26000.0	8.517393	0.0	0	0	
3320	11.948461	14000.0	16.118096	0.0	0	0	

5947 rows × 158 columns

In [116…
```
y_train
```

Out[116…
```
6169     4.8
5431     4.6
9036     4.2
6406     4.2
10113    4.4
        ...
5437     4.7
10014    4.5
6027     4.6
```

```
3320     4.2
Name: Rating, Length: 5947, dtype: float64
```

In [117…
```
y_test = df_test.pop('Rating')
X_test = df_test
```

In [118…
```
X_test
```

Out[118…

	Reviews	Size	Installs	Price	Category_ART_AND_DESIGN	Category_AUTO_AND_VEHICLES	Categor
5587	3.367296	121.0	6.908755	0.00	0	0	
4732	10.534972	14000.0	13.815512	0.00	0	0	
807	12.658106	3300.0	16.118096	0.00	0	0	
6414	8.344267	10000.0	11.512935	0.00	0	0	
4522	2.890372	3100.0	9.210440	0.00	0	0	
...	...	...	...	...	...	...	
3420	11.967155	6400.0	16.118096	0.00	0	0	
6142	11.304917	2600.0	16.118096	0.00	0	0	
4946	8.637639	14000.0	9.210440	1.49	0	0	
7375	6.118097	36000.0	9.210440	2.99	0	0	
4865	7.395722	18000.0	11.512935	0.00	0	0	

2549 rows × 158 columns

In [119…
```
y_test
```

Out[119…
```
5587    3.5
4732    4.5
807     4.7
6414    3.5
4522    1.8
       ...
3420    4.1
6142    4.5
4946    4.8
7375    4.2
4865    4.3
Name: Rating, Length: 2549, dtype: float64
```

In [120…
```
#LinearRegression is our class name.
#lin_reg is our object name.
from sklearn.linear_model import LinearRegression
lin_reg=LinearRegression()
```

In [121…
```
#fit is the method which is use only in train data.
lin_reg.fit(X_train,y_train)
```

Out[121…
```
LinearRegression()
```

In [122…
```
#Prediction on test set.
y_pred=lin_reg.predict(X_test)
```

In [123…
```
y_pred
```

Out[123… array([4.10251906, 4.24059274, 4.50357966, ..., 4.51414891, 4.16190069,
3.98695217])

In [124…
```
#Report R2_score on test set.
from sklearn.metrics import r2_score
r_score=r2_score(y_test,y_pred)
print(r_score)
```

```
0.1554486771884962
```

In [125…
```
#Report R2_score on train set.
from sklearn.metrics import r2_score
r_score=r2_score(lin_reg.predict(X_train),y_train)
print(r_score)
```

```
-4.513193755139202
```

In [ ]: