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

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

In [3]:
```
#2
df.isnull().sum()
```

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]:
```
#3
df=df.dropna(how='any')
```

In [5]:
```
df.isnull().sum()
```

Out[5]:
```
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 [6]:
```
df
```

Out[6]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19M	10,000+	Free	0	Everyone
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14M	500,000+	Free	0	Everyone
2	U Launcher Lite – FREE Live Cool Themes, Hide ...	ART_AND_DESIGN	4.7	87510	8.7M	5,000,000+	Free	0	Everyone
3	Sketch - Draw & Paint	ART_AND_DESIGN	4.5	215644	25M	50,000,000+	Free	0	Teen
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	967	2.8M	100,000+	Free	0	Everyone
...	...	...	...	...	...	...	...	...	...
10834	FR Calculator	FAMILY	4.0	7	2.6M	500+	Free	0	Everyone
10836	Sya9a Maroc - FR	FAMILY	4.5	38	53M	5,000+	Free	0	Everyone
10837	Fr. Mike Schmitz Audio Teachings	FAMILY	5.0	4	3.6M	100+	Free	0	Everyone
10839	The SCP Foundation DB fr nn5n	BOOKS_AND_REFERENCE	4.5	114	Varies with device	1,000+	Free	0	Mature 17+
10840	iHoroscope - 2018 Daily Horoscope & Astrology	LIFESTYLE	4.5	398307	19M	10,000,000+	Free	0	Everyone

9360 rows × 13 columns

In [7]:

```
#4.1
def change(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 [8]:
```
df['Size'] = df['Size'].map(change)
```

In [9]:
```
df
```

Out[9]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Conter Ratin
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19000.0	10,000+	Free	0	Everyon
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14000.0	500,000+	Free	0	Everyon
2	U Launcher Lite – FREE Live Cool Themes, Hide ...	ART_AND_DESIGN	4.7	87510	8700.0	5,000,000+	Free	0	Everyon
3	Sketch - Draw & Paint	ART_AND_DESIGN	4.5	215644	25000.0	50,000,000+	Free	0	Tee
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	967	2800.0	100,000+	Free	0	Everyon
...	...	...	...	...	...	...	...	...	
10834	FR Calculator	FAMILY	4.0	7	2600.0	500+	Free	0	Everyon
10836	Sya9a Maroc - FR	FAMILY	4.5	38	53000.0	5,000+	Free	0	Everyon
10837	Fr. Mike Schmitz Audio Teachings	FAMILY	5.0	4	3600.0	100+	Free	0	Everyon
10839	The SCP Foundation DB fr nn5n	BOOKS_AND_REFERENCE	4.5	114	NaN	1,000+	Free	0	Matur 17

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Conter Ratin
10840	iHoroscope - 2018 Daily Horoscope & Astrology	LIFESTYLE	4.5	398307	19000.0	10,000,000+	Free	0	Everyon

9360 rows × 13 columns

In [10]:
```
#4.2
df[['Reviews']] = df[['Reviews']].astype('float64')
```

In [11]:
```
df.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   float64
 4   Size            7723 non-null   float64
 5   Installs        9360 non-null   object 
 6   Type            9360 non-null   object 
 7   Price           9360 non-null   object 
 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), object(10)
memory usage: 1023.8+ KB
```

In [12]:
```
#4.3
def change(installs):
    if '+' and ',' in installs:
        installs = installs.replace('+','')
        installs = installs.replace(',','')
        return(installs)
```

In [13]:
```
df['Installs'] = df['Installs'].map(change)
```

In [14]:
```
df[['Installs']] = df[['Installs']].fillna(0)
```

In [15]:
```
df[['Installs']] = df[['Installs']].astype('int64')
```

In [16]:

```
df[['Reviews']] = df[['Reviews']].astype('int64')
```

In [17]:
```
df.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            7723 non-null   float64
 5   Installs        9360 non-null   int64  
 6   Type            9360 non-null   object 
 7   Price           9360 non-null   object 
 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(2), int64(2), object(9)
memory usage: 1023.8+ KB
```

In [18]:
```
df.Installs
```

Out[18]:
```
0           10000
1          500000
2         5000000
3        50000000
4          100000
           ...   
10834           0
10836        5000
10837           0
10839        1000
10840    10000000
Name: Installs, Length: 9360, dtype: int64
```

In [19]:
```
#4.4
def change(price):
        if '$' in price:
            price = price.replace('$','')
            price = pd.to_numeric(price)
            return(price)
        else:
            return(0)
```

In [20]:
```
df['Price']=df['Price'].map(change)
```

In [21]:
```
df.Price[220:230]
```

Out[21]:
```
232    0.00
233    0.00
234    4.99
```

```
235    4.99
236    0.00
237    0.00
238    0.00
239    0.00
240    0.00
241    0.00
Name: Price, dtype: float64
```

In [22]:
```
#4.5.1
def change(rating):
    if rating < 1 or rating >5 :
        rating = rating.drop()
        return(rating)
    else:
        return(rating)
```

In [23]:
```
df['Rating'] = df['Rating'].map(change)
```

In [24]:
```
df[(df.Rating>5)]
```

Out[24]:

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

In [25]:
```
df.Reviews
```

Out[25]:
```
0           159
1           967
2         87510
3        215644
4           967
          ...
10834         7
10836        38
10837         4
10839       114
10840    398307
Name: Reviews, Length: 9360, dtype: int64
```

In [26]:
```
df.Installs
```

Out[26]:
```
0           10000
1          500000
2         5000000
3        50000000
4          100000
           ...
10834           0
10836        5000
10837           0
10839        1000
10840    10000000
Name: Installs, Length: 9360, dtype: int64
```

In [27]:

```
'''
5.sanity checks:
2.
Reviews and installs are of same length.

'''
```

Out[27]: '\n5.sanity checks:\n2.\nReviews and installs are of same length.\n\n'

In [28]:
```
#4.5.3
df['Type']=df['Type'].replace('Paid','Free')
```

In [29]:
```
df.Type[4250:4300]
```

Out[29]:
```
4423    Free
4424    Free
4425    Free
4426    Free
4427    Free
4428    Free
4429    Free
4430    Free
4432    Free
4433    Free
4434    Free
4435    Free
4436    Free
4437    Free
4438    Free
4439    Free
4440    Free
4441    Free
4442    Free
4443    Free
4444    Free
4445    Free
4446    Free
4447    Free
4448    Free
4449    Free
4450    Free
4452    Free
4454    Free
4455    Free
4456    Free
4458    Free
4460    Free
4461    Free
4463    Free
4464    Free
4466    Free
4467    Free
4468    Free
4469    Free
4470    Free
4472    Free
4473    Free
4474    Free
4476    Free
4477    Free
```

```
4478    Free
4479    Free
4480    Free
4481    Free
Name: Type, dtype: object
```

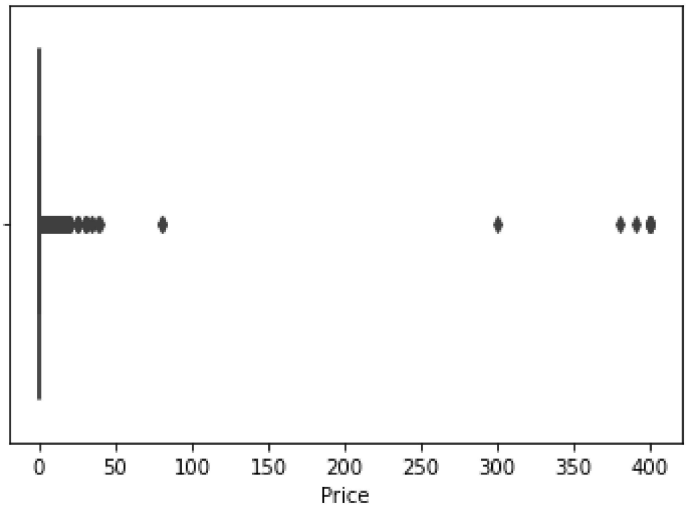
In [30]:

```
#5
sns.boxplot(df['Price'])
df[['Price']].describe()
#Most of the apps price falls under 150 and there are only 4 apps which goes beyond the
#There are 3 apps with maximum price range between 350-400.
# The price of maximum apps falls under 0-50.
# Q1,Q2,Q3 are 0 for the price.so,there is no IQR.
```

```
C:\Users\sbhav\Dropbox\anaconda\lib\site-packages\seaborn\_decorators.py:36: FutureWarni
ng: Pass the following variable as a keyword arg: x. From version 0.12, the only valid p
ositional argument will be `data`, and passing other arguments without an explicit keywo
rd will result in an error or misinterpretation.
  warnings.warn(
```

Out[30]:

	Price
count	9360.000000
mean	0.961279
std	15.821640
min	0.000000
25%	0.000000
50%	0.000000
75%	0.000000
max	400.000000



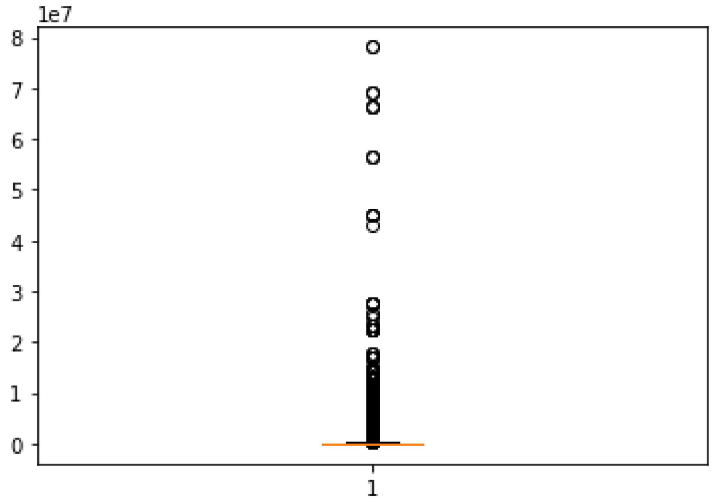
In [31]:

```
plt.boxplot(df['Reviews'])
df['Reviews'].describe()
# yes, there are few apps which have very high number of reviews and the maximum rewvie
# The value seems  correct for Facebook social,Instagram social and Whatsapp social app
```

Out[31]:

```
count    9.360000e+03
```

```
mean     5.143767e+05
std      3.145023e+06
min      1.000000e+00
25%      1.867500e+02
50%      5.955000e+03
75%      8.162750e+04
max      7.815831e+07
Name: Reviews, dtype: float64
```



In [32]:

```
plt.hist(df['Rating'])
# Ratings are distributed more towards the high.
#less than 500 people are given ratings between 1 to 3.5.
#Around 4000 people are given 4.5 rating which is the maximum rating.
df['Rating'].describe()
#outliers
Q1 = 4
Q3 = 4.5
IQR = Q3 - Q1

outliers = df[(df['Rating'] > Q3 +1.5*IQR) | (df['Rating'] < Q1 - 1.5*IQR)]
print(outliers)
```

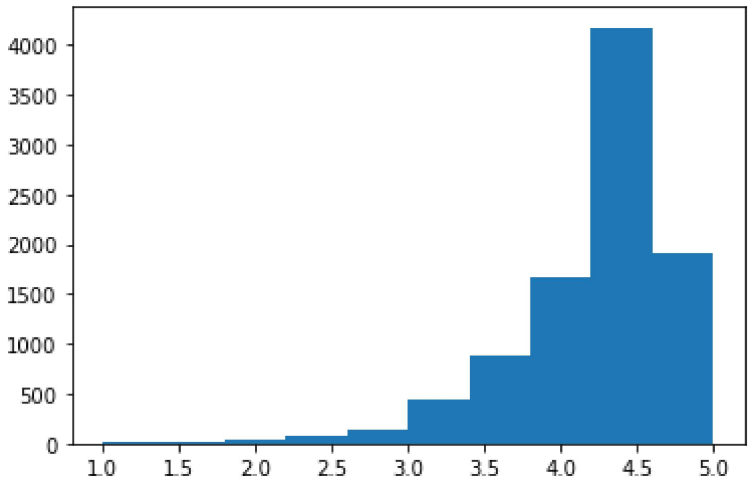
```
                                                  App             Category  \
87                          RST - Sale of cars on the PCT    AUTO_AND_VEHICLES
209                                        Plugin:AOT v5.0             BUSINESS
311    comico Popular Original Cartoon Updated Everyd...               COMICS
312                          Daily Manga - Comic & Webtoon               COMICS
477                                           Calculator               DATING
...                                                  ...                  ...
10665                          SB · FN 1870 Mobile Banking             FINANCE
10677                                      Pint - FN Theme      PERSONALIZATION
10715                                FarmersOnly Dating               DATING
10757                       Fisher-Price® Smart Connect™                TOOLS
10766                             FreedomPop Diagnostics                TOOLS

       Rating  Reviews     Size  Installs  Type  Price Content Rating  \
87        3.2      250   1100.0    100000  Free   0.00       Everyone
209       3.1     4034     23.0    100000  Free   0.00       Everyone
311       3.2    93965  15000.0   5000000  Free   0.00           Teen
312       3.2     1446   7100.0    100000  Free   0.00     Mature 17+
477       2.6       57   6200.0      1000  Free   6.99       Everyone
...       ...      ...      ...       ...   ...    ...            ...
10665     2.9      139   3300.0     10000  Free   0.00       Everyone
10677     2.5        6    234.0         0  Free   0.00       Everyone
10715     3.0     1145   1400.0    100000  Free   0.00     Mature 17+
10757     2.7      422  72000.0     50000  Free   0.00       Everyone
```

```
10766     2.9      452   7000.0    100000  Free   0.00       Everyone

               Genres        Last Updated           Current Ver   Android Ver
87     Auto & Vehicles      April 27, 2018                   1.4  4.0.3 and up
209           Business  September 11, 2015  3.0.1.11 (Build 311)    2.2 and up
311             Comics        July 3, 2018                 6.3.0  4.0.3 and up
312             Comics         May 18, 2018                   1.0  4.0.3 and up
477             Dating     October 25, 2017                 1.1.6    4.0 and up
...                ...                  ...                   ...          ...
10665          Finance       June 19, 2017                 3.0.5    4.0 and up
10677  Personalization     August 10, 2013                   1.0    2.2 and up
10715           Dating   February 25, 2016                   2.2    4.0 and up
10757            Tools   February 23, 2018                 2.4.1    4.4 and up
10766            Tools       July 17, 2017         1.03.123.0713  4.0.3 and up

[502 rows x 13 columns]
```



In [33]:

```
plt.hist(df['Size'])
df['Size'].describe()
#outliers
Q1 = 5300
Q3 = 33000
IQR = Q3 - Q1

outliers = df[(df['Size'] > Q3 +1.5*IQR) | (df['Size'] < Q1 - 1.5*IQR)]
print(outliers)
```

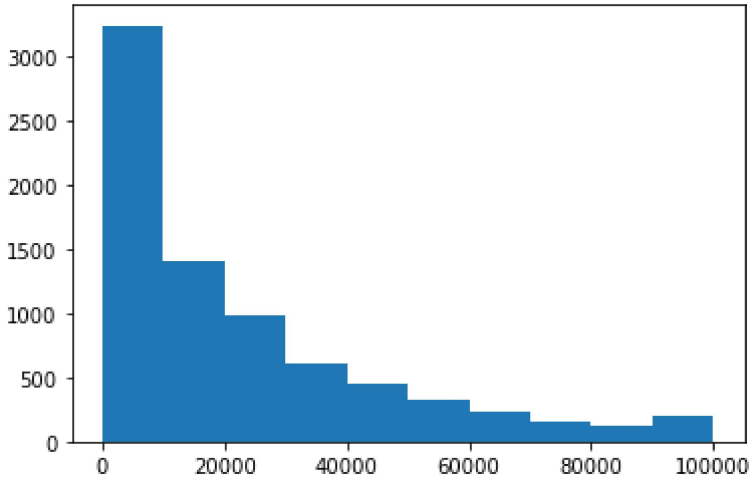
```
                                                  App          Category  \
600                           iPair-Meet, Chat, Dating            DATING
695                           iPair-Meet, Chat, Dating            DATING
728       Free intellectual training game application |         EDUCATION
748                             Memorado - Brain Games         EDUCATION
790    Rosetta Stone: Learn to Speak & Read New Langu...         EDUCATION
...                                                 ...               ...
10588                          Florida Travel Guide  TRAVEL_AND_LOCAL
10779                        Fortune Quest: Savior             FAMILY
10784                                    Big Hunter               GAME
10793                                     Sid Story               GAME
10803                  Fatal Raid - No.1 Mobile FPS               GAME

       Rating  Reviews     Size  Installs  Type  Price Content Rating  \
600       4.5   182986  77000.0   5000000  Free    0.0     Mature 17+
695       4.5   182986  77000.0   5000000  Free    0.0     Mature 17+
728       4.2     5741  84000.0   1000000  Free    0.0       Everyone
748       4.4    56897  97000.0   1000000  Free    0.0       Everyone
790       4.5   172505  76000.0   5000000  Free    0.0       Everyone
...       ...      ...      ...       ...   ...    ...            ...
```

```
10588     3.8       11  86000.0      1000  Free    0.0       Everyone
10779     3.6      135  75000.0     10000  Free    0.0   Everyone 10+
10784     4.3   245455  84000.0  10000000  Free    0.0   Everyone 10+
10793     4.4    28510  78000.0    500000  Free    0.0           Teen
10803     4.3    56496  81000.0   1000000  Free    0.0           Teen

                      Genres      Last Updated Current Ver   Android Ver
600                   Dating    August 2, 2018       5.0.8    4.1 and up
695                   Dating    August 2, 2018       5.0.8    4.1 and up
728    Education;Pretend Play     July 25, 2018       3.7.0    4.4 and up
748     Education;Brain Games  January 16, 2017      1.10.0    4.1 and up
790       Education;Education     June 27, 2018       5.2.1    5.0 and up
...                       ...               ...         ...         ...
10588          Travel & Local     July 11, 2018       6.8.2    4.1 and up
10779            Role Playing      June 1, 2018       1.022    4.4 and up
10784                  Action      May 31, 2018       2.8.6    4.0 and up
10793                    Card    August 1, 2018       2.6.6  4.0.3 and up
10803                  Action    August 7, 2018     1.5.447    4.0 and up

[412 rows x 13 columns]
```



In [34]:
```
df['Price'].max()
```

Out[34]:
```
400.0
```

In [35]:
```
#6.1
df= df.drop(df[df['Price'] >= 200].index)
```

In [36]:
```
df['Price'].max() #removed the values above 200
```

Out[36]:
```
79.99
```

In [37]:
```
df['Reviews'].max()
```

Out[37]:
```
78158306
```

In [38]:
```
df= df.drop(df[df['Reviews'] >= 2000000].index)
```

In [39]:
```
#6.2
```

```
df['Reviews'].max() #Removed the reviews more than 2millions
```

Out[39]: 1986068

In [40]:
```
#6.3
print(df[['Installs']].quantile(.10))
print(df[['Installs']].quantile(.25))
print(df[['Installs']].quantile(.50))
print(df[['Installs']].quantile(.70))
print(df[['Installs']].quantile(.90))
print(df[['Installs']].quantile(.95))
print(df[['Installs']].quantile(.99))
```

```
Installs    1000.0
Name: 0.1, dtype: float64
Installs    10000.0
Name: 0.25, dtype: float64
Installs    500000.0
Name: 0.5, dtype: float64
Installs    1000000.0
Name: 0.7, dtype: float64
Installs    10000000.0
Name: 0.9, dtype: float64
Installs    10000000.0
Name: 0.95, dtype: float64
Installs    100000000.0
Name: 0.99, dtype: float64
```

In [41]:
```
df['Installs'].max()
```

Out[41]: 1000000000

In [42]:
```
df= df.drop(df[df['Installs'] >= 10000000].index) # i'm deciding the cutoff as 10000000
```

In [43]:
```
df['Installs'].max()
```

Out[43]: 5000000

7

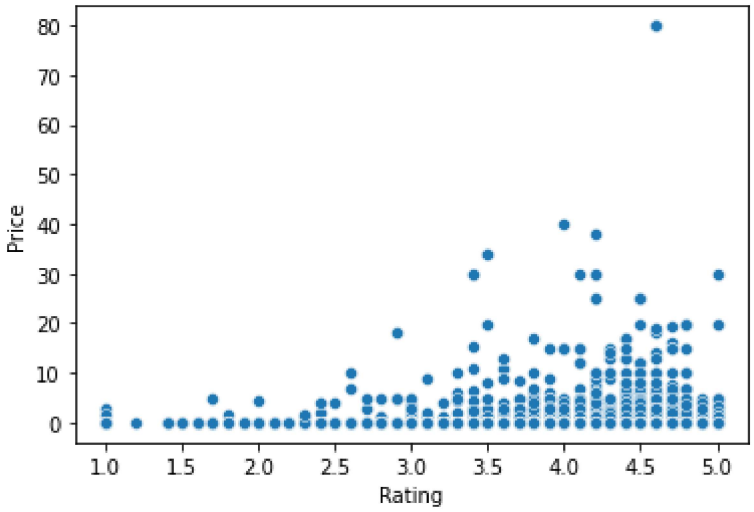
In [44]:
```
sns.scatterplot(df['Rating'],df['Price'])
'''
Ratings increases with the price.
Mostly the rating were between 3.5-5 for the price below 40.
The price below 10 has given more ratings and ratings got between 1-5 for free apps.
The minimum rating is 1 given by the below 10 price range users.

'''
```

```
C:\Users\sbhav\Dropbox\anaconda\lib\site-packages\seaborn\_decorators.py:36: FutureWarni
ng: Pass the following variables as keyword args: x, y. From version 0.12, the only vali
d positional argument will be `data`, and passing other arguments without an explicit ke
```

```
yword will result in an error or misinterpretation.
  warnings.warn(
```

Out[44]:
```
'\nRatings increases with the price.\nMostly the rating were between 3.5-5 for the price below 40.\nThe price below 10 has given more ratings and ratings got between 1-5 for free apps.\nThe minimum rating is 1 given by the below 10 price range users.\n\n'
```

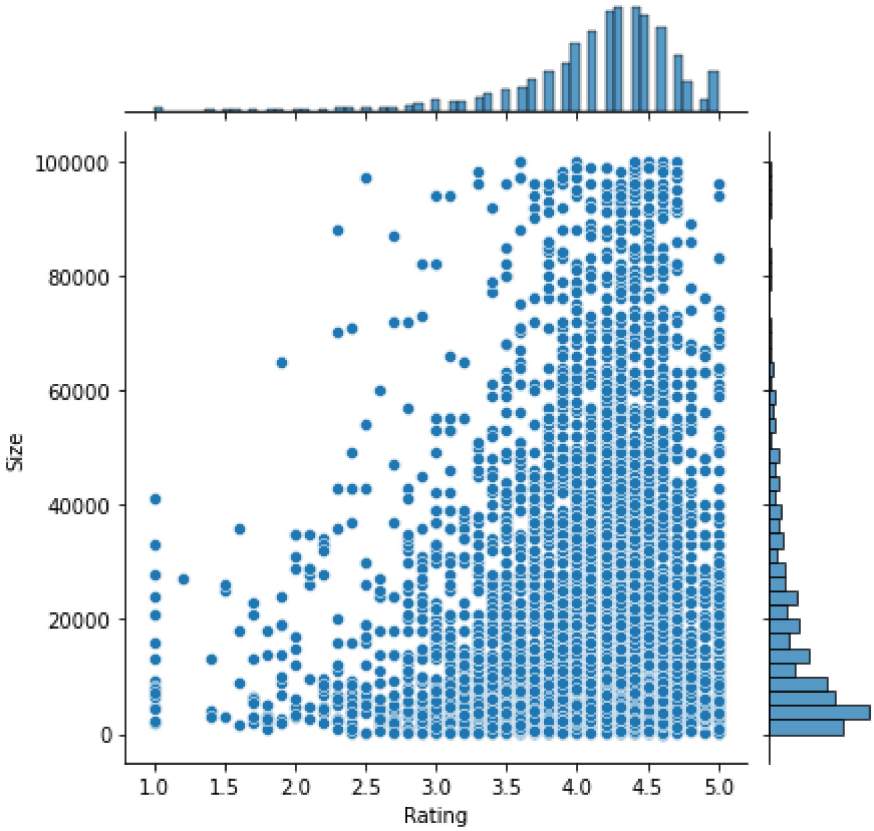


In [45]:
```
sns.jointplot(df['Rating'],df['Size'])
sns.xlabel=df['Rating']
sns.ylabel=df['Size']
'''
Almost each and every size has given the ratings.
Mostly size with 0-100000 has given the highest ratings between 3.5 to 4.5.
Very high range of ratings are given by the size below 60000.
As the size increases the ratings also increases.
4.5 is maximum rating given by the users.
1.2 is the minimum rating given by the size 30000


'''
```

```
C:\Users\sbhav\Dropbox\anaconda\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

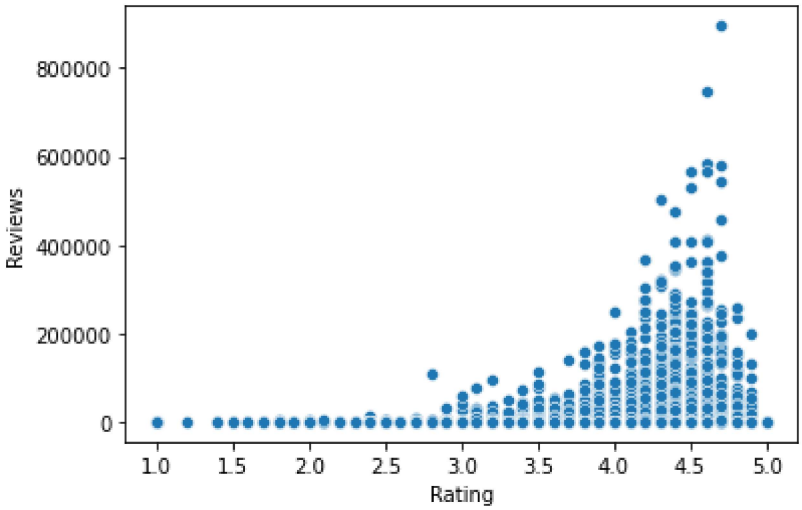
Out[45]:
```
'\nAlmost each and every size has given the ratings.\nMostly size with 0-100000 has given the highest ratings between 3.5 to 4.5.\nVery high range of ratings are given by the size below 60000.\nAs the size increases the ratings also increases.\n4.5 is maximum rating given by the users.\n1.2 is the minimum rating given by the size 30000\n\n\n'
```



In [46]:
```
sns.scatterplot(df['Rating'],df['Reviews'])
'''
0-200000 reviewers had given most of the ratings.
most of the reviewers has given 4.5 rating.
Reviews are very less above 600000.
'''
```

```
C:\Users\sbhav\Dropbox\anaconda\lib\site-packages\seaborn\_decorators.py:36: FutureWarni
ng: Pass the following variables as keyword args: x, y. From version 0.12, the only vali
d positional argument will be `data`, and passing other arguments without an explicit ke
yword will result in an error or misinterpretation.
  warnings.warn(
```

Out[46]:
```
'\n0-200000 reviewers had given most of the ratings.\nmost of the reviewers has given 4.
5 rating.\nReviews are very less above 600000.\n'
```

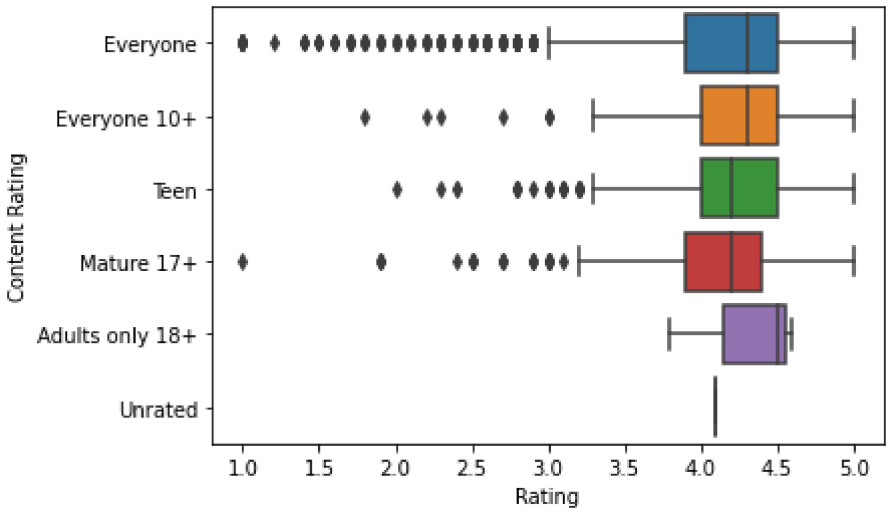


In [47]:
```
sns.boxplot(df['Rating'],df['Content Rating'])
'''
The mean of the content rating and rating varies between 4 to 4.5.
Minimum of the both starts between 3 to 3.5.
maximum ratings given by the above mature 17+ is 5.
everyone has given almost all the ratings between 1-5.
unrated has given only 4.2 rating.
18+ has given ratings between 3.7 to 4.6.

'''
```

```
C:\Users\sbhav\Dropbox\anaconda\lib\site-packages\seaborn\_decorators.py:36: FutureWarni
ng: Pass the following variables as keyword args: x, y. From version 0.12, the only vali
d positional argument will be `data`, and passing other arguments without an explicit ke
yword will result in an error or misinterpretation.
  warnings.warn(
```

Out[47]:
```
'\nThe mean of the content rating and rating varies between 4 to 4.5.\nMinimum of the bo
th starts between 3 to 3.5.\nmaximum ratings given by the above mature 17+ is 5.\neveryo
ne has given almost all the ratings between 1-5.\nunrated has given only 4.2 rating.\n18
+ has given ratings between 3.7 to 4.6.\n\n'
```

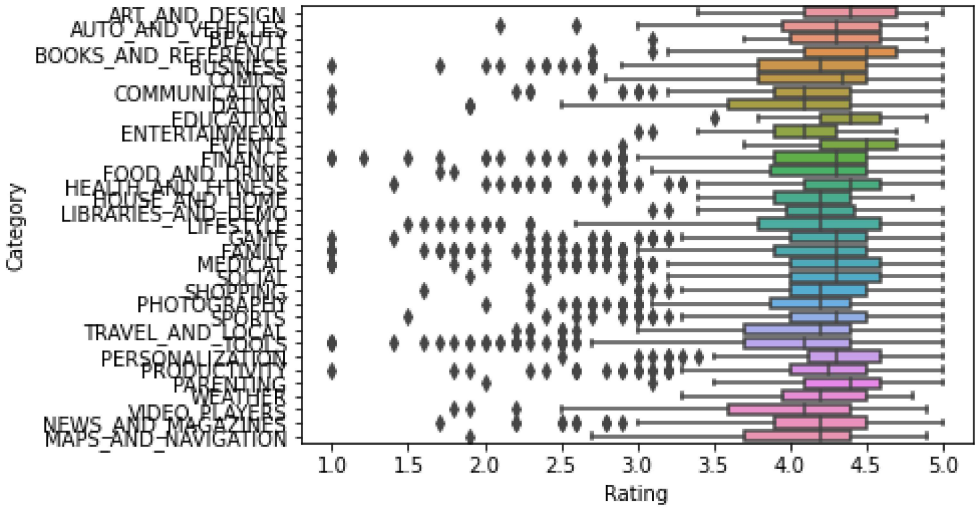


In [48]:
```
sns.boxplot(df['Rating'],df['Category'])
'''
Max,mean,median for every category have ratings between 3-5.
Art& Design and education category have less ratings compared to other categories.
Finance,Family,Tools category have high number of ratings.

'''
```

```
C:\Users\sbhav\Dropbox\anaconda\lib\site-packages\seaborn\_decorators.py:36: FutureWarni
ng: Pass the following variables as keyword args: x, y. From version 0.12, the only vali
d positional argument will be `data`, and passing other arguments without an explicit ke
yword will result in an error or misinterpretation.
  warnings.warn(
```

Out[48]:
```
'\nMax,mean,median for every category have ratings between 3-5.\nArt& Design and educati
on category have less ratings compared to other categories.\nFinance,Family,Tools catego
ry have high number of ratings.\n\n'
```



8

In [49]:
```
inp1=df.copy(deep=True)
```

In [50]:
```
inp1[['Reviews','Installs']]= np.log1p(inp1[['Reviews','Installs']])
```

In [51]:
```
inp1[['Reviews','Installs']]
```

Out[51]:

	Reviews	Installs
0	5.075174	9.210440
1	6.875232	13.122365
2	11.379520	15.424949
4	6.875232	11.512935
5	5.123964	10.819798
...	...	...
10833	3.806662	6.908755
10834	2.079442	0.000000
10836	3.663562	8.517393
10837	1.609438	0.000000
10839	4.744932	6.908755

7265 rows × 2 columns

In [52]:
```
inp1=inp1.drop(['Last Updated','Current Ver','Android Ver'],axis=1)
```

In [53]:
```
inp1=inp1.dropna(how='any')
```

In [54]:
```
inp1.isnull().sum()
```

Out[54]:
```
App               0
Category          0
Rating            0
Reviews           0
Size              0
Installs          0
Type              0
Price             0
Content Rating    0
Genres            0
dtype: int64
```

In [55]:
```
inp2=pd.get_dummies(['Category','Genres','Content Rating'])
```

In [56]:
```
inp2.Category.shape
```

Out[56]:
```
(3,)
```

In [57]:
```
inp1_num=inp1[['Rating','Reviews','Size','Installs','Price']]
```

In [58]:
```
df_combined=pd.concat([inp1_num,inp2],axis=1)
```

9

In [59]:
```
X=df_combined[['Price','Reviews','Size','Installs']]
y =df_combined['Rating']
```

10

In [60]:
```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X, y, train_size = 0.7)
```

In [61]:
```
print(X_train.shape)
print(y_train.shape)
print(X_test.shape)
print(y_test.shape)
```

```
(4546, 4)
(4546,)
(1949, 4)
(1949,)
```

In [62]:
```
y_test
```

Out[62]:
```
9311     4.1
1036     4.8
2083     4.5
9767     3.5
9287     3.0
        ...
2047     4.5
537      4.2
10125    4.4
3627     4.8
1625     4.4
Name: Rating, Length: 1949, dtype: float64
```

11

In [63]:
```
from sklearn.linear_model import LinearRegression
lm = LinearRegression()
```

In [64]:
```
lm.fit(X_train,y_train)
```

Out[64]:
```
LinearRegression()
```

In [65]:
```
print(lm.coef_)
```

```
[ 3.26199133e-03  1.14964918e-01 -1.78534527e-07 -7.64422268e-02]
```

In [66]:
```
print(lm.intercept_)
```

```
4.15691652565069
```

In [67]:
```
pred_test = lm.predict(X_test)
```

In [68]:
```
pred_train=lm.predict(X_train)
```

In [69]:
```
# Evaluation of Model
from sklearn.metrics import mean_squared_error, r2_score
from math import sqrt
```

12

In [70]:
```
mse_train = mean_squared_error(y_train, pred_train)
rmse_train = sqrt(mse_train)

print("Mean Square erorr for train data is :", mse_train)
print("Root Mean Square erorr for train data is :", rmse_train)


r2_train = r2_score(y_train, pred_train)
print('R2 value of model is :', r2_train)
```

```
Mean Square erorr for train data is : 0.3018965965206742
Root Mean Square erorr for train data is : 0.5494511775587292
R2 value of model is : 0.08619822546242006
```

In [71]:

```
mse_test = mean_squared_error(y_test, pred_test)
rmse_test = sqrt(mse_train)

print("Mean Square erorr for test data is :", mse_test)
print("Root Mean Square erorr for test data is :", rmse_test)


r2_test = r2_score(y_test, pred_test)
print('R2 value of model is :', r2_test)
```

```
Mean Square erorr for test data is : 0.30876438437684256
Root Mean Square erorr for test data is : 0.5494511775587292
R2 value of model is : 0.10201809592545863
```

In [ ]: