In [372]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [373]:
```
infor=pd.read_csv('googleplaystore.csv')
```

In [374]:
```
infor.head()
```

Out[374]:

	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	D
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	De

In [375]:
```
infor.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10841 entries, 0 to 10840
Data columns (total 13 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   App             10841 non-null  object 
 1   Category        10841 non-null  object 
 2   Rating          9367 non-null   float64
 3   Reviews         10841 non-null  object 
 4   Size            10841 non-null  object 
 5   Installs        10841 non-null  object 
 6   Type            10840 non-null  object 
 7   Price           10841 non-null  object 
 8   Content Rating  10840 non-null  object 
 9   Genres          10841 non-null  object 
 10  Last Updated    10841 non-null  object 
 11  Current Ver     10833 non-null  object 
 12  Android Ver     10838 non-null  object 
dtypes: float64(1), object(12)
memory usage: 1.1+ MB
```

In [376]:
```
infor.isnull().sum(axis=0)
```

Out[376]:
```
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 [377]:
```
infor.dropna(how ='any', inplace = True)
```

In [378]:
```
infor.isnull().sum(axis=0)
```

Out[378]:
```
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 [379]:
```
infor.dtypes
```

Out[379]:
```
App                object
Category           object
Rating            float64
Reviews            object
Size               object
Installs           object
Type               object
Price              object
Content Rating     object
Genres             object
Last Updated       object
Current Ver        object
Android Ver        object
dtype: object
```

In [380]:
```
infor.head()
```

Out[380]:

	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	D
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	De

In [381]:
```
infor.dtypes
```

Out[381]:
```
App               object
Category          object
Rating           float64
Reviews           object
Size              object
Installs          object
Type              object
Price             object
Content Rating    object
Genres            object
Last Updated      object
Current Ver       object
Android Ver       object
dtype: object
```

In [382]:
```
infor.head()
```

Out[382]:

	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	D
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	De

In [383]:
```
infor.Price.value_counts()[:5]
```

Out[383]:
```
0        8715
$2.99     114
$0.99     106
$4.99      70
$1.99      59
Name: Price, dtype: int64
```

In [384]:
```
infor['Price'] = infor.Price.map(lambda x: 0 if x=='0' else float(x[1:]))
```

In [385]:
```
infor.Reviews = infor.Reviews.astype("int32")
```

In [386]:
```
infor.Reviews.describe()
```

Out[386]:
```
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 [387]:
```
infor.Installs.value_counts()
```

Out[387]:
```
1,000,000+        1576
10,000,000+       1252
100,000+          1150
10,000+           1009
5,000,000+         752
1,000+             712
500,000+           537
50,000+            466
5,000+             431
100,000,000+       409
100+               309
50,000,000+        289
500+               201
500,000,000+        72
10+                 69
1,000,000,000+      58
50+                 56
5+                   9
1+                   3
Name: Installs, dtype: int64
```

In [388]:
```
def clean_installs(val):
    return int(val.replace(",","").replace("+",""))
```

In [389]:
```
infor.Installs = infor.Installs.map(clean_installs)
```

In [390]:
```
infor.Installs.describe()
```

Out[390]:
```
count    9.360000e+03
mean     1.790875e+07
std      9.126637e+07
min      1.000000e+00
25%      1.000000e+04
50%      5.000000e+05
75%      5.000000e+06
max      1.000000e+09
Name: Installs, dtype: float64
```

In [391]:
```
def change_size(size):
    if 'M' in size:
        x = size[:-1]
        x = float(x)
        return(x)
    elif 'k' == size[-1:]:
        x = size[:-1]
        x = float(x)
        return(x)
    else:
        return None
```

In [392]:
```
infor["Size"] = infor["Size"].map(change_size)
```

In [393]:
```
infor.Size.describe()
```

Out[393]:
```
count    7723.00000
mean       37.30707
std        93.54223
min         1.00000
25%         6.10000
50%        16.00000
75%        37.00000
max       994.00000
Name: Size, dtype: float64
```

In [394]:
```
infor.Size.fillna(method = 'ffill', inplace=True)
```

In [395]:
```
infor.dtypes
```

Out[395]:
```
App               object
Category          object
Rating           float64
Reviews            int32
Size             float64
Installs           int64
Type              object
Price            float64
Content Rating    object
Genres            object
Last Updated      object
Current Ver       object
Android Ver       object
dtype: object
```

In [396]:
```
infor.Rating.describe()
```

Out[396]:
```
count    9360.000000
mean        4.191838
std         0.515263
min         1.000000
25%         4.000000
50%         4.300000
75%         4.500000
max         5.000000
Name: Rating, dtype: float64
```

In [397]:
```
len(infor[infor.Reviews > infor.Installs])
```

Out[397]:
```
7
```

In [398]:
```
infor[infor.Reviews > infor.Installs]
```

Out[398]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Upd
2454	KBA-EZ Health Guide	MEDICAL	5.0	4	25.0	1	Free	0.00	Everyone	Medical	Au 2, 2
4663	Alarmy (Sleep If U Can) - Pro	LIFESTYLE	4.8	10249	30.0	10000	Paid	2.49	Everyone	Lifestyle	Jul 2
5917	Ra Ga Ba	GAME	5.0	2	20.0	1	Paid	1.49	Everyone	Arcade	Febr 8, 2
6700	Brick Breaker BR	GAME	5.0	7	19.0	5	Free	0.00	Everyone	Arcade	Jul 2
7402	Trovami se ci riesci	GAME	5.0	11	6.1	10	Free	0.00	Everyone	Arcade	M 11, 2
8591	DN Blog	SOCIAL	5.0	20	4.2	10	Free	0.00	Teen	Social	Jul 2
10697	Mu.F.O.	GAME	5.0	2	16.0	1	Paid	0.99	Everyone	Arcade	Mar 2

In [399]:
```
infor = infor[infor.Reviews <= infor.Installs].copy()
```

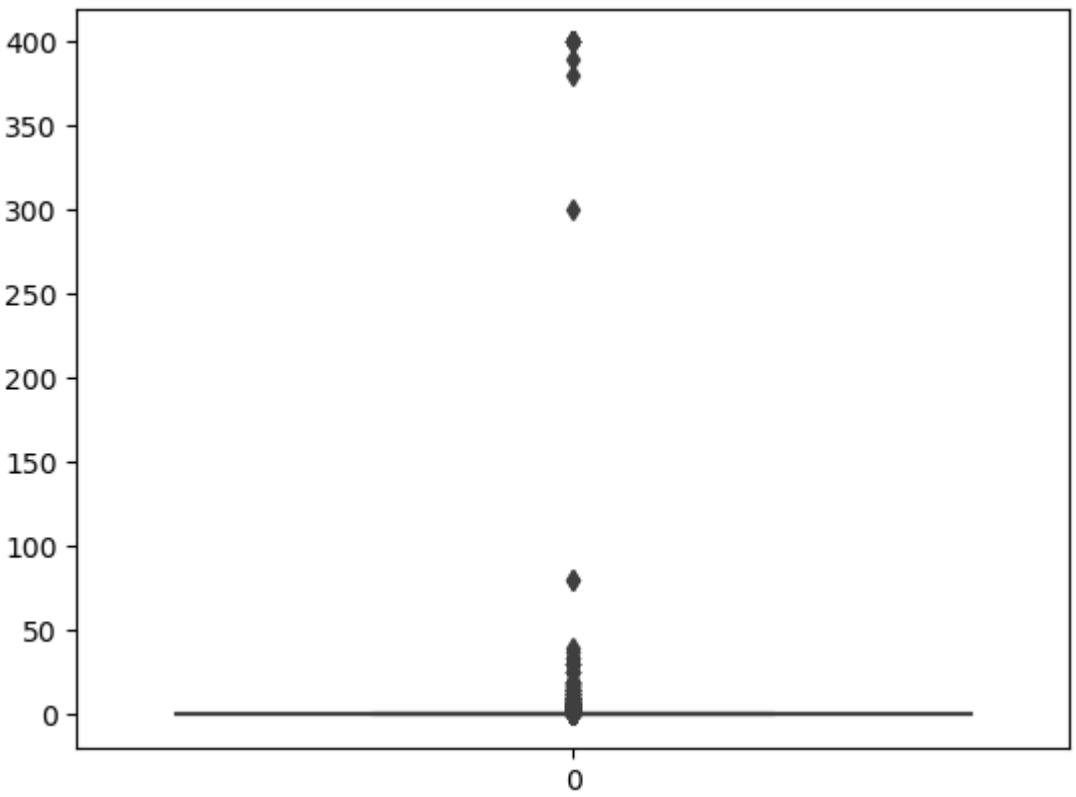
In [400]:
```
infor.shape
```

Out[400]:
```
(9353, 13)
```

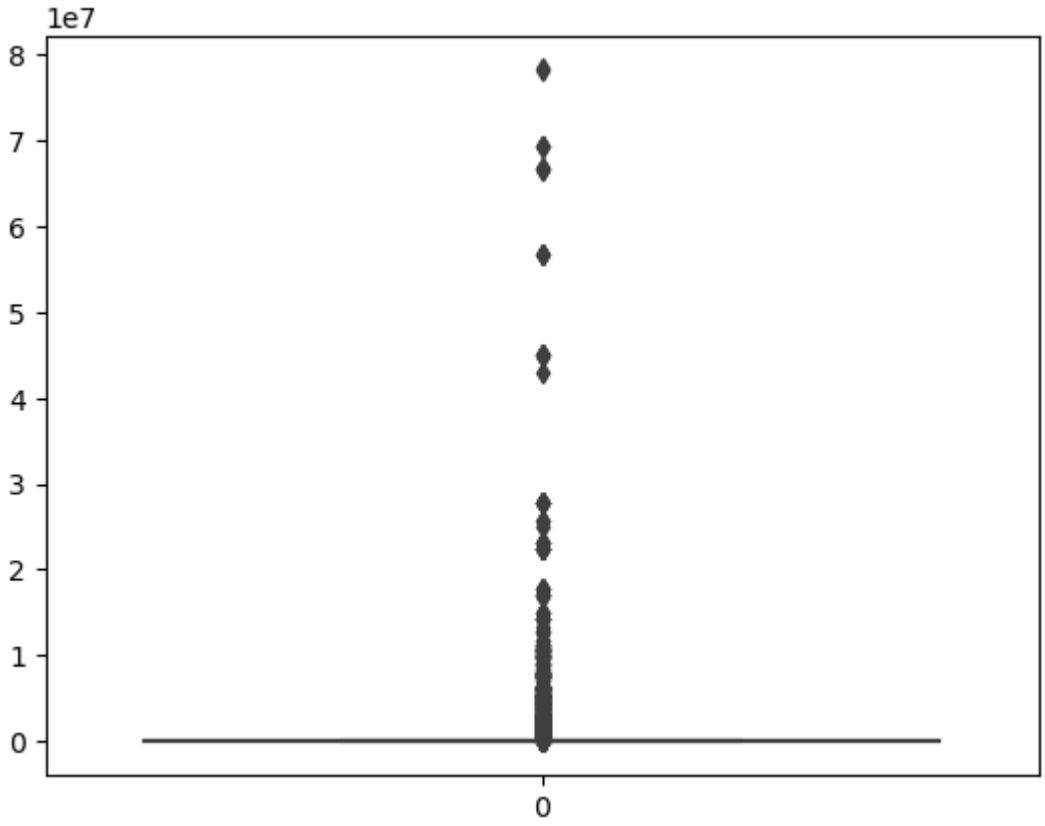
In [401]:
```
len(infor[(infor.Type == "Free") & (infor.Price>0)])
```

Out[401]:
```
0
```

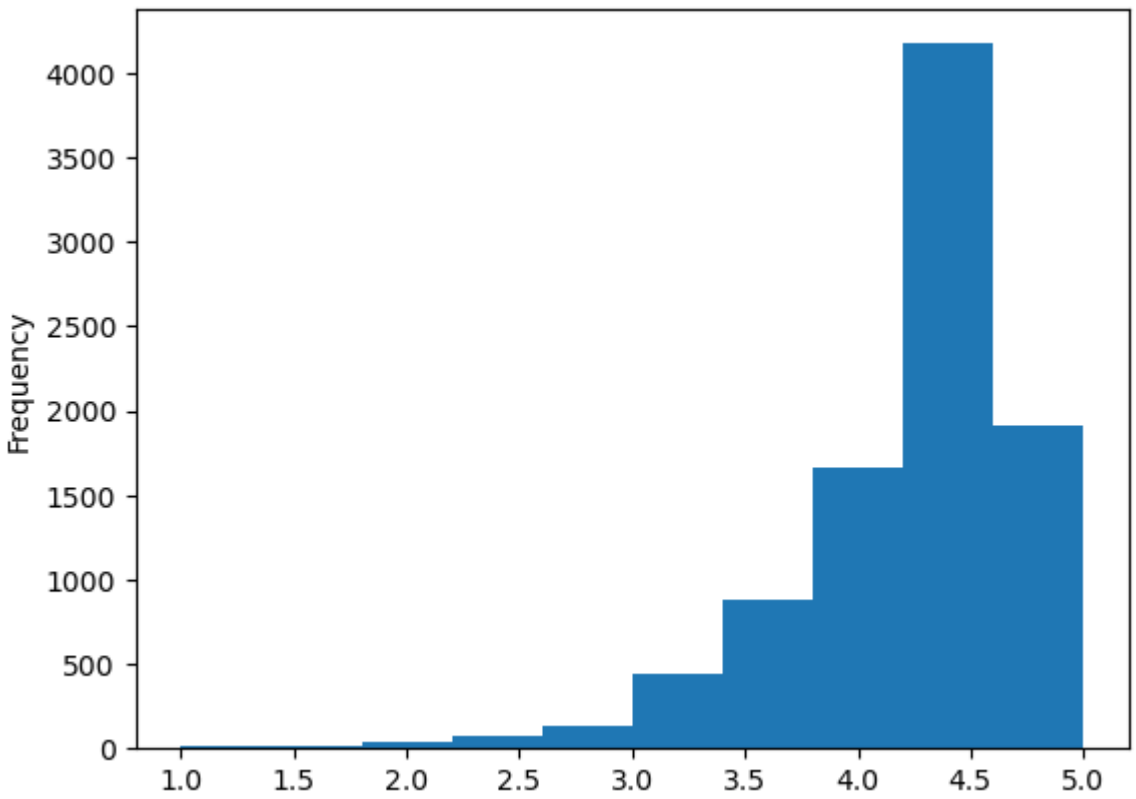
In [402]:
```
sns.boxplot(infor.Price)
plt.show()
```



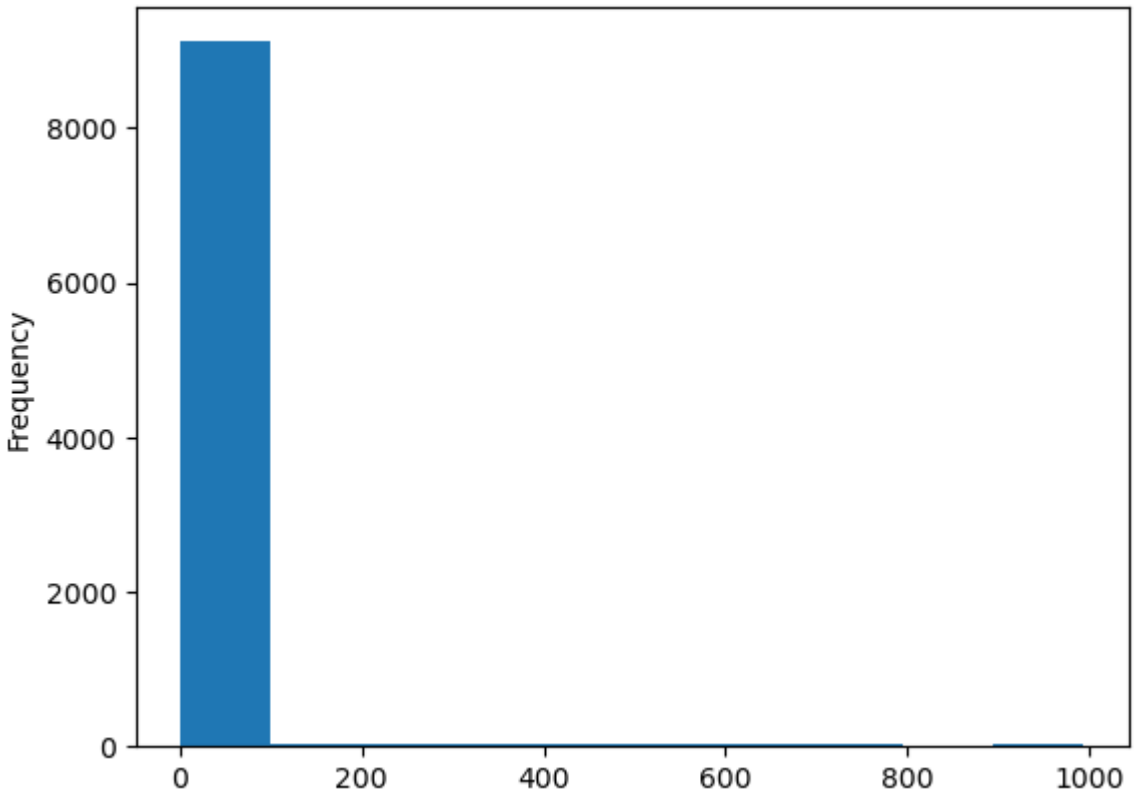
In [403]:
```
sns.boxplot(infor.Reviews)
plt.show()
```



In [404]:
```
infor.Rating.plot.hist()
plt.show()    #Distribution of Rating
```



In [405]:
```
infor['Size'].plot.hist()
plt.show()
```



In [406]:
```
len(infor[infor.Price > 200])
```

Out[406]: 15

In [407]:
```
infor[infor.Price > 200]
```

Out[407]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	G
4197	most expensive app (H)	FAMILY	4.3	6	1.5	100	Paid	399.99	Everyone	Entertai
4362	💎 I'm rich	LIFESTYLE	3.8	718	26.0	10000	Paid	399.99	Everyone	Lif
4367	I'm Rich - Trump Edition	LIFESTYLE	3.6	275	7.3	10000	Paid	400.00	Everyone	Lif
5351	I am rich	LIFESTYLE	3.8	3547	1.8	100000	Paid	399.99	Everyone	Lif
5354	I am Rich Plus	FAMILY	4.0	856	8.7	10000	Paid	399.99	Everyone	Entertai
5355	I am rich VIP	LIFESTYLE	3.8	411	2.6	10000	Paid	299.99	Everyone	Lif
5356	I Am Rich Premium	FINANCE	4.1	1867	4.7	50000	Paid	399.99	Everyone	Fi
5357	I am extremely Rich	LIFESTYLE	2.9	41	2.9	1000	Paid	379.99	Everyone	Lif
5358	I am Rich!	FINANCE	3.8	93	22.0	1000	Paid	399.99	Everyone	Fi
5359	I am rich(premium)	FINANCE	3.5	472	965.0	5000	Paid	399.99	Everyone	Fi
5362	I Am Rich Pro	FAMILY	4.4	201	2.7	5000	Paid	399.99	Everyone	Entertai
5364	I am rich (Most expensive app)	FINANCE	4.1	129	2.7	1000	Paid	399.99	Teen	Fi
5366	I Am Rich	FAMILY	3.6	217	4.9	10000	Paid	389.99	Everyone	Entertai
5369	I am Rich	FINANCE	4.3	180	3.8	5000	Paid	399.99	Everyone	Fi
5373	I AM RICH PRO PLUS	FINANCE	4.0	36	41.0	1000	Paid	399.99	Everyone	Fi

In [408]:
```
infor = infor[infor.Price <= 200].copy()
infor.shape
```

Out[408]: (9338, 13)

In [409]:
```
infor = infor[infor.Reviews <= 2000000]
infor.shape
```

Out[409]:
```
(8885, 13)
```

In [410]:
```
infor.Installs.quantile([0.1, 0.25, 0.50, 0.70, 0.9, 0.95, 0.99])
```

Out[410]:
```
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 [411]:
```
len(infor[infor.Installs >= 1000000000])
```

Out[411]:
```
6
```

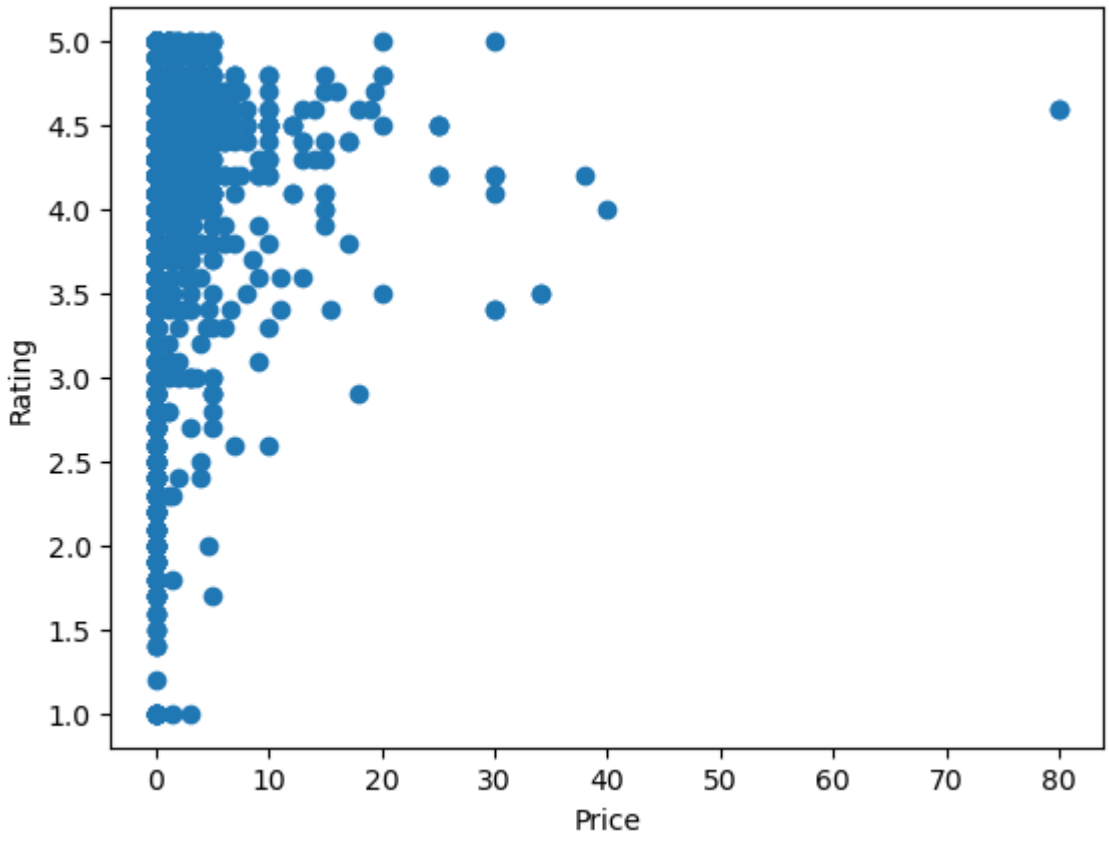
In [412]:
```
infor = infor[infor.Installs < 1000000000].copy()
infor.shape
```

Out[412]:
```
(8879, 13)
```

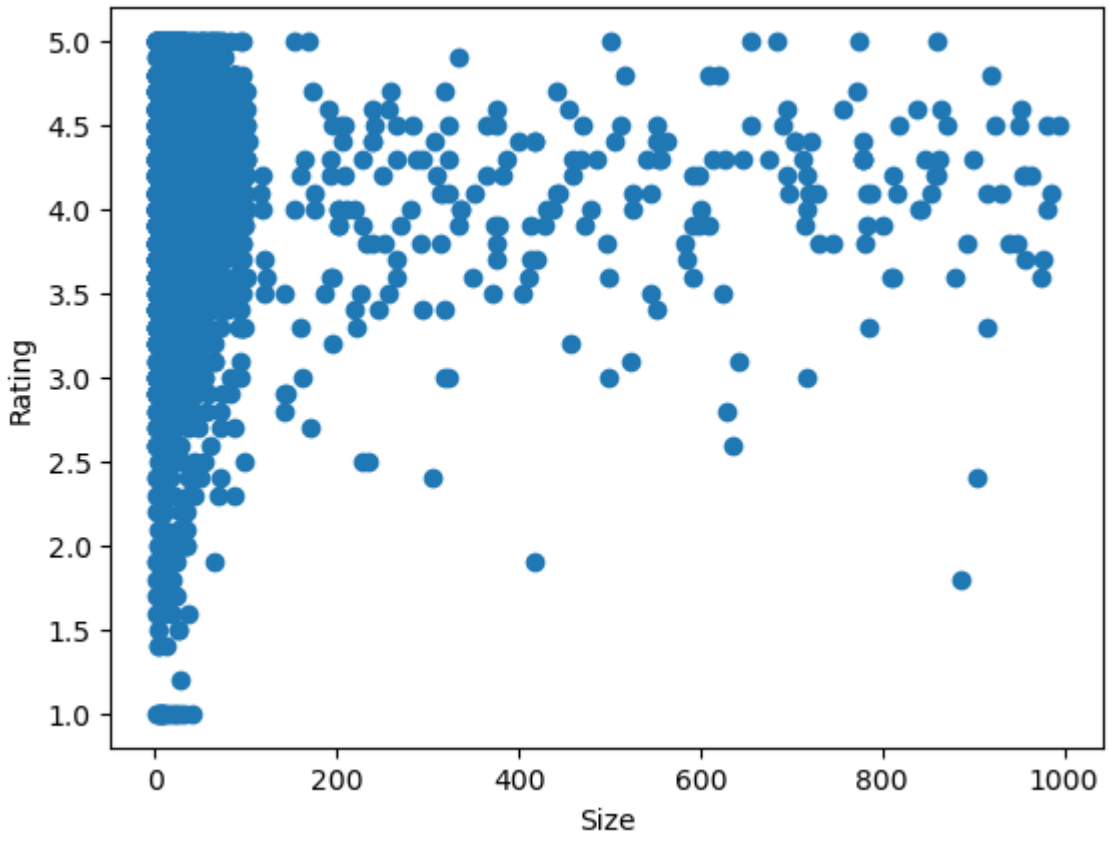
In [413]:
```
import warnings
warnings.filterwarnings("ignore")
```

In [414]:
```
import matplotlib.pyplot as plt
```

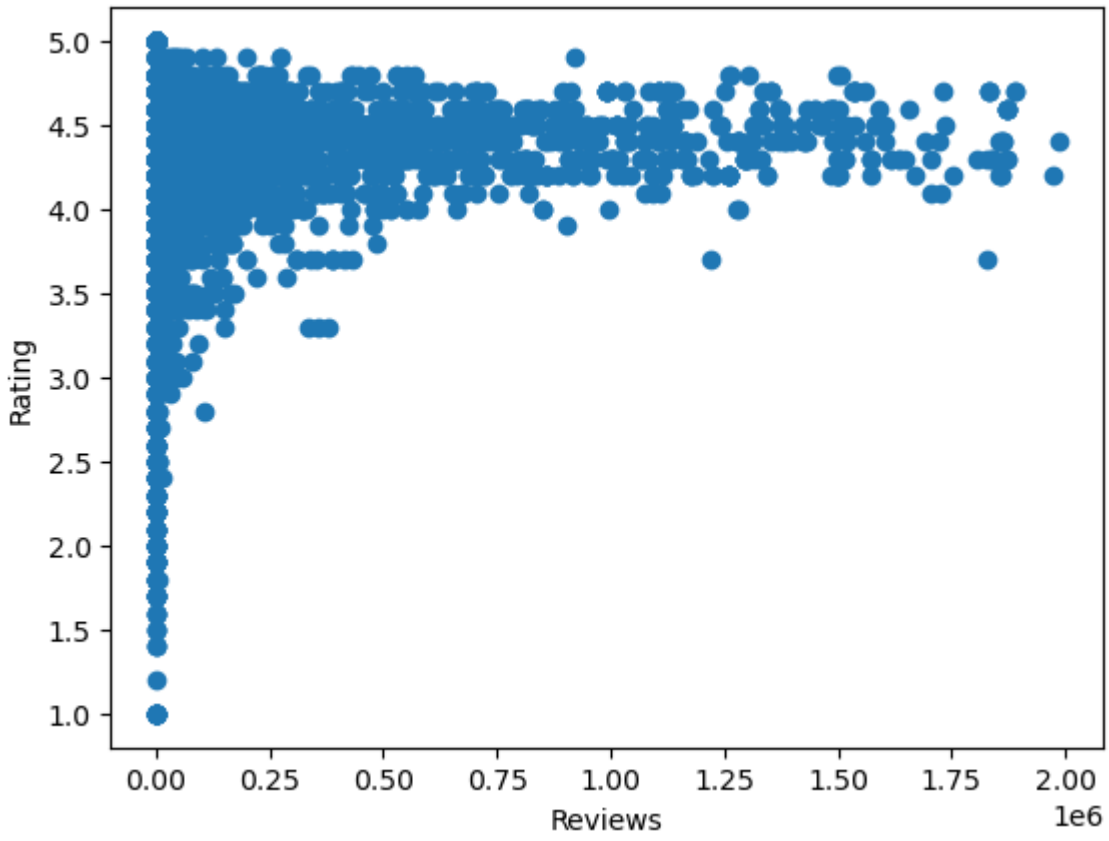
In [415]:
```
x=infor['Price']
y=infor['Rating']
plt.scatter(x,y)
plt.xlabel('Price')
plt.ylabel('Rating')
plt.show()
```



In [416]:
```
x=infor['Size']
y=infor['Rating']
plt.scatter(x,y)
plt.xlabel('Size')
plt.ylabel('Rating')
plt.show()
```

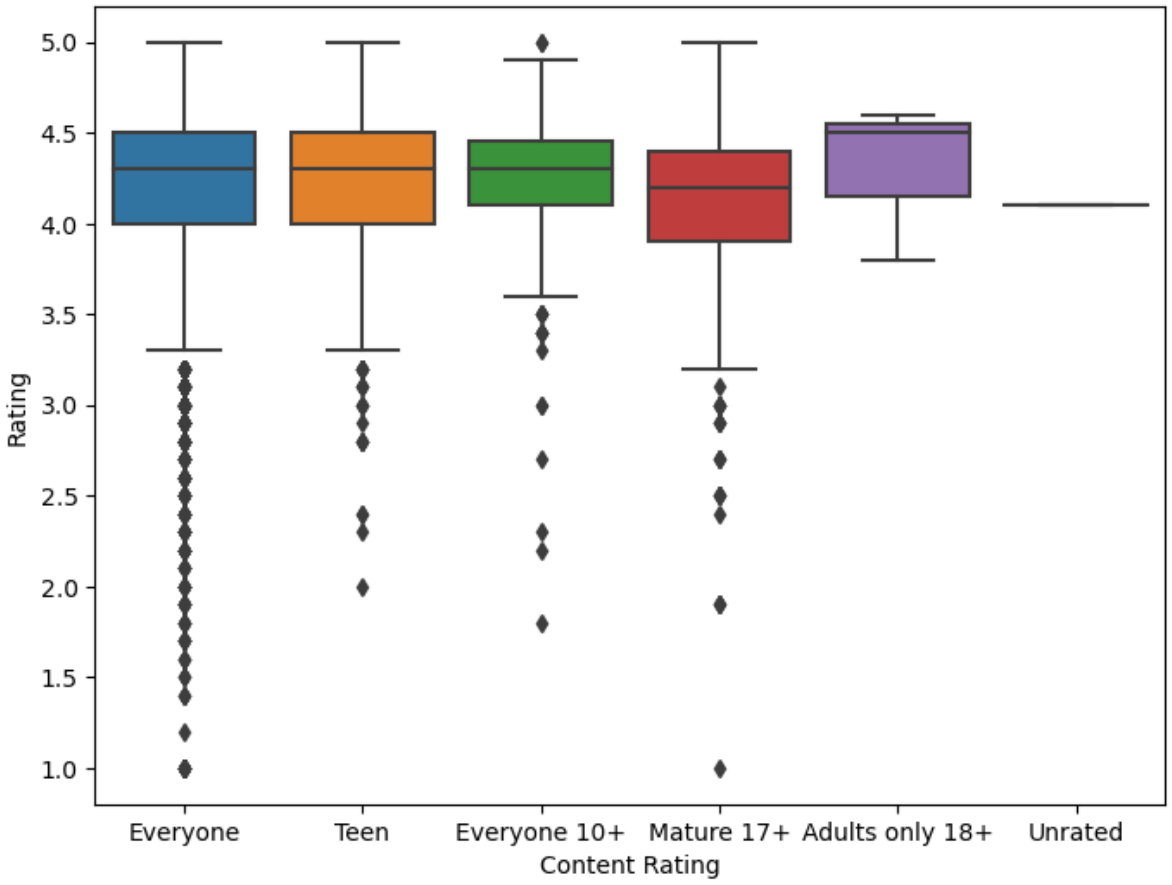


In [417]:
```
x=infor['Reviews']
y=infor['Rating']
plt.scatter(x,y)
plt.xlabel('Reviews')
plt.ylabel('Rating')
plt.show()
```



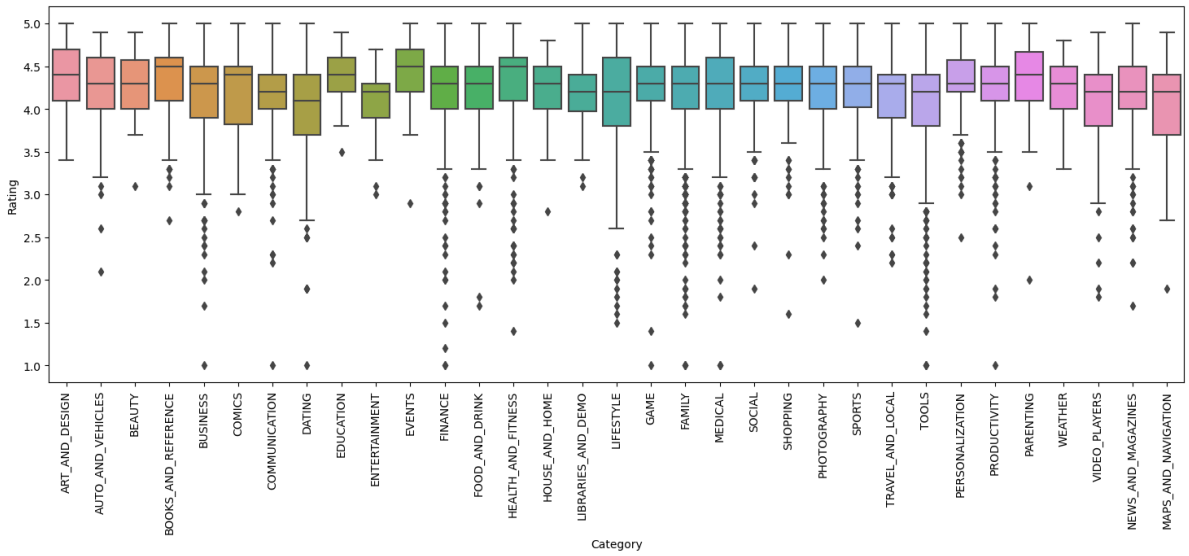
In [418]:
```
plt.figure(figsize=[8,6])
sns.boxplot(x='Content Rating', y='Rating', data=infor)
```

Out[418]:
```
<Axes: xlabel='Content Rating', ylabel='Rating'>
```



In [419]:
```
plt.figure(figsize=[18,6])
g = sns.boxplot(x='Category', y='Rating', data=infor)
plt.xticks(rotation=90)
```

Out[419]:
```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
        17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32]),
 [Text(0, 0, 'ART_AND_DESIGN'),
  Text(1, 0, 'AUTO_AND_VEHICLES'),
  Text(2, 0, 'BEAUTY'),
  Text(3, 0, 'BOOKS_AND_REFERENCE'),
  Text(4, 0, 'BUSINESS'),
  Text(5, 0, 'COMICS'),
  Text(6, 0, 'COMMUNICATION'),
  Text(7, 0, 'DATING'),
  Text(8, 0, 'EDUCATION'),
  Text(9, 0, 'ENTERTAINMENT'),
  Text(10, 0, 'EVENTS'),
  Text(11, 0, 'FINANCE'),
  Text(12, 0, 'FOOD_AND_DRINK'),
  Text(13, 0, 'HEALTH_AND_FITNESS'),
  Text(14, 0, 'HOUSE_AND_HOME'),
  Text(15, 0, 'LIBRARIES_AND_DEMO'),
  Text(16, 0, 'LIFESTYLE'),
  Text(17, 0, 'GAME'),
  Text(18, 0, 'FAMILY'),
  Text(19, 0, 'MEDICAL'),
  Text(20, 0, 'SOCIAL'),
  Text(21, 0, 'SHOPPING'),
  Text(22, 0, 'PHOTOGRAPHY'),
  Text(23, 0, 'SPORTS'),
  Text(24, 0, 'TRAVEL_AND_LOCAL'),
  Text(25, 0, 'TOOLS'),
  Text(26, 0, 'PERSONALIZATION'),
  Text(27, 0, 'PRODUCTIVITY'),
  Text(28, 0, 'PARENTING'),
  Text(29, 0, 'WEATHER'),
  Text(30, 0, 'VIDEO_PLAYERS'),
  Text(31, 0, 'NEWS_AND_MAGAZINES'),
  Text(32, 0, 'MAPS_AND_NAVIGATION')])
```



In [420]:
```
infor1 = infor.copy()
```

In [421]:
```
infor.Installs.describe()
```

Out[421]:
```
count    8.879000e+03
mean     5.595862e+06
std      2.421042e+07
min      5.000000e+00
25%      1.000000e+04
50%      5.000000e+05
75%      5.000000e+06
max      5.000000e+08
Name: Installs, dtype: float64
```

In [422]:
```
infor1.Installs = infor1.Installs.apply(np.log1p)
```

In [423]:
```
infor1.Reviews = infor1.Reviews.apply(np.log1p)
```

In [424]:
```
infor1.dtypes
```

Out[424]:
```
App               object
Category          object
Rating           float64
Reviews          float64
Size             float64
Installs         float64
Type              object
Price            float64
Content Rating    object
Genres            object
Last Updated      object
Current Ver       object
Android Ver       object
dtype: object
```

In [425]:
```
infor1.drop(["App","Last Updated","Current Ver","Android Ver"],axis=1, inplace
```

In [426]:
```
infor1.shape
```

Out[426]:
```
(8879, 9)
```

In [427]:
```
infor2 = pd.get_dummies(infor1,drop_first=True)
```

In [428]:
```
infor2.columns
```

Out[428]:
```
Index(['Rating', 'Reviews', 'Size', 'Installs', 'Price',
       'Category_AUTO_AND_VEHICLES', 'Category_BEAUTY',
       'Category_BOOKS_AND_REFERENCE', 'Category_BUSINESS', 'Category_COMIC
S',
       ...
       'Genres_Tools', 'Genres_Tools;Education', 'Genres_Travel & Local',
       'Genres_Travel & Local;Action & Adventure', 'Genres_Trivia',
       'Genres_Video Players & Editors',
       'Genres_Video Players & Editors;Creativity',
       'Genres_Video Players & Editors;Music & Video', 'Genres_Weather',
       'Genres_Word'],
      dtype='object', length=157)
```

In [429]:
```
from sklearn.model_selection import train_test_split
```

In [430]:
```
?train_test_split
```

In [431]:
```
df_train, df_test = train_test_split(infor2, train_size = 0.7,random_state=100
```

In [432]:
```
df_train.shape, df_test.shape
```

Out[432]:
```
((6215, 157), (2664, 157))
```

In [433]:
```
y_train = df_train.pop("Rating")
x_train = df_train
```

In [434]:
```
y_test = df_test.pop("Rating")
x_test = df_test
```

In [435]:
```
from sklearn.linear_model import LinearRegression
```

In [436]:
```
lr = LinearRegression()
```

In [437]:
```
lr.fit(x_train,y_train)
```

Out[437]:
```
▾ LinearRegression

LinearRegression()
```

In [438]:
```
from sklearn.metrics import r2_score
```

In [439]:
```
y_train_pred= lr.predict(x_train)
r2_score(y_train,y_train_pred)
```

Out[439]:
```
0.1655093129622186
```

In [440]:
```
y_test_pred= lr.predict(x_test)
r2_score(y_test,y_test_pred)
```

Out[440]:
```
0.13265347439179043
```

In [ ]:
```

```