In [92]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
data = pd.read_csv('googleplaystore.csv')
data.head()
```

Out[92]:

	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	De

In [93]:

```
data.isna().sum()
```

Out[93]:

```
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 [94]:

```
new_data = data.dropna()
```

In [95]:

```
new_data.isna().sum()
```

Out[95]:

```
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 [96]:

```
new_data["Size"].unique()
```

Out[96]:

```
array(['19M', '14M', '8.7M', '25M', '2.8M', '5.6M', '29M', '33M', '3.1M',
       '28M', '12M', '20M', '21M', '37M', '5.5M', '17M', '39M', '31M',
       '4.2M', '23M', '6.0M', '6.1M', '4.6M', '9.2M', '5.2M', '11M',
       '24M', 'Varies with device', '9.4M', '15M', '10M', '1.2M', '26M',
       '8.0M', '7.9M', '56M', '57M', '35M', '54M', '201k', '3.6M', '5.7M',
       '8.6M', '2.4M', '27M', '2.7M', '2.5M', '7.0M', '16M', '3.4M',
       '8.9M', '3.9M', '2.9M', '38M', '32M', '5.4M', '18M', '1.1M',
       '2.2M', '4.5M', '9.8M', '52M', '9.0M', '6.7M', '30M', '2.6M',
       '7.1M', '22M', '6.4M', '3.2M', '8.2M', '4.9M', '9.5M', '5.0M',
       '5.9M', '13M', '73M', '6.8M', '3.5M', '4.0M', '2.3M', '2.1M',
       '42M', '9.1M', '55M', '23k', '7.3M', '6.5M', '1.5M', '7.5M', '51M',
       '41M', '48M', '8.5M', '46M', '8.3M', '4.3M', '4.7M', '3.3M', '40M',
       '7.8M', '8.8M', '6.6M', '5.1M', '61M', '66M', '79k', '8.4M',
       '3.7M', '118k', '44M', '695k', '1.6M', '6.2M', '53M', '1.4M',
       '3.0M', '7.2M', '5.8M', '3.8M', '9.6M', '45M', '63M', '49M', '77M',
       '4.4M', '70M', '9.3M', '8.1M', '36M', '6.9M', '7.4M', '84M', '97M',
       '2.0M', '1.9M', '1.8M', '5.3M', '47M', '556k', '526k', '76M',
       '7.6M', '59M', '9.7M', '78M', '72M', '43M', '7.7M', '6.3M', '334k',
       '93M', '65M', '79M', '100M', '58M', '50M', '68M', '64M', '34M',
       '67M', '60M', '94M', '9.9M', '232k', '99M', '624k', '95M', '8.5k',
       '41k', '292k', '80M', '1.7M', '10.0M', '74M', '62M', '69M', '75M',
       '98M', '85M', '82M', '96M', '87M', '71M', '86M', '91M', '81M',
       '92M', '83M', '88M', '704k', '862k', '899k', '378k', '4.8M',
       '266k', '375k', '1.3M', '975k', '980k', '4.1M', '89M', '696k',
       '544k', '525k', '920k', '779k', '853k', '720k', '713k', '772k',
       '318k', '58k', '241k', '196k', '857k', '51k', '953k', '865k',
       '251k', '930k', '540k', '313k', '746k', '203k', '26k', '314k',
       '239k', '371k', '220k', '730k', '756k', '91k', '293k', '17k',
       '74k', '14k', '317k', '78k', '924k', '818k', '81k', '939k', '169k',
       '45k', '965k', '90M', '545k', '61k', '283k', '655k', '714k', '93k',
       '872k', '121k', '322k', '976k', '206k', '954k', '444k', '717k',
       '210k', '609k', '308k', '306k', '175k', '350k', '383k', '454k',
       '1.0M', '70k', '812k', '442k', '842k', '417k', '412k', '459k',
       '478k', '335k', '782k', '721k', '430k', '429k', '192k', '460k',
       '728k', '496k', '816k', '414k', '506k', '887k', '613k', '778k',
       '683k', '592k', '186k', '840k', '647k', '373k', '437k', '598k',
       '716k', '585k', '982k', '219k', '55k', '323k', '691k', '511k',
       '951k', '963k', '25k', '554k', '351k', '27k', '82k', '208k',
       '551k', '29k', '103k', '116k', '153k', '209k', '499k', '173k',
       '597k', '809k', '122k', '411k', '400k', '801k', '787k', '50k',
       '643k', '986k', '516k', '837k', '780k', '20k', '498k', '600k',
       '656k', '221k', '228k', '176k', '34k', '259k', '164k', '458k',
       '629k', '28k', '288k', '775k', '785k', '636k', '916k', '994k',
       '309k', '485k', '914k', '903k', '608k', '500k', '54k', '562k',
       '847k', '948k', '811k', '270k', '48k', '523k', '784k', '280k',
       '24k', '892k', '154k', '18k', '33k', '860k', '364k', '387k',
       '626k', '161k', '879k', '39k', '170k', '141k', '160k', '144k',
       '143k', '190k', '376k', '193k', '473k', '246k', '73k', '253k',
       '957k', '420k', '72k', '404k', '470k', '226k', '240k', '89k',
       '234k', '257k', '861k', '467k', '676k', '552k', '582k', '619k'],
      dtype=object)
```

In [97]:

```
new_data["Size"].dtype
```

Out[97]:

```
dtype('O')
```

In [98]:

```
new_data=new_data[-new_data['Size'].str.contains('Var')]
```

In [100]:

```
new_data
```

Out[100]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19M	10,000+	Free	0
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14M	500,000+	Free	0
2	U Launcher Lite – FREE Live Cool Themes, Hide ...	ART_AND_DESIGN	4.7	87510	8.7M	5,000,000+	Free	0
3	Sketch - Draw & Paint	ART_AND_DESIGN	4.5	215644	25M	50,000,000+	Free	0
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	967	2.8M	100,000+	Free	0
...	...	...	...	...	...	...	...	...
10833	Chemin (fr)	BOOKS_AND_REFERENCE	4.8	44	619k	1,000+	Free	0
10834	FR Calculator	FAMILY	4.0	7	2.6M	500+	Free	0
10836	Sya9a Maroc - FR	FAMILY	4.5	38	53M	5,000+	Free	0
10837	Fr. Mike Schmitz Audio Teachings	FAMILY	5.0	4	3.6M	100+	Free	0
10840	iHoroscope - 2018 Daily Horoscope & Astrology	LIFESTYLE	4.5	398307	19M	10,000,000+	Free	0

7723 rows × 13 columns

In [12]:

```
new_data.loc[:,"SizeNum"] = new_data.Size
new_data.SizeNum=pd.to_numeric(new_data['Size'])
new_data.Size.dtype
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
pandas/_libs/lib.pyx in pandas._libs.lib.maybe_convert_numeric()

ValueError: Unable to parse string "19M"

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
<ipython-input-12-92cc4f94dd08> in <module>
      1 new_data.loc[:,"SizeNum"] = new_data.Size
----> 2 new_data.SizeNum=pd.to_numeric(new_data['Size'])
      3 new_data.Size.dtype

/usr/local/lib/python3.7/site-packages/pandas/core/tools/numeric.py in to_num
eric(arg, errors, downcast)
    148         try:
    149             values = lib.maybe_convert_numeric(
--> 150                 values, set(), coerce_numeric=coerce_numeric
    151             )
    152         except (ValueError, TypeError):

pandas/_libs/lib.pyx in pandas._libs.lib.maybe_convert_numeric()

ValueError: Unable to parse string "19M" at position 0
```

In [101]:

```
new_data.loc[:,"SizeNum"] = new_data.Size.str.rstrip("Mk+")
new_data.SizeNum=pd.to_numeric(new_data['SizeNum'])
new_data.SizeNum.dtype
```

Out[101]:

```
dtype('float64')
```

In [102]:

```
new_data['SizeNum']=np.where(new_data.Size.str.contains('M'),new_data.SizeNum*1000, new_da
ta.SizeNum)
```

In [103]:

```
import warnings
warnings.filterwarnings('ignore')
```

In [104]:

```
new_data["SizeNum"]
```

Out[104]:

```
0        19000.0
1        14000.0
2         8700.0
3        25000.0
4         2800.0
          ...
10833      619.0
10834     2600.0
10836    53000.0
10837     3600.0
10840    19000.0
Name: SizeNum, Length: 7723, dtype: float64
```

In [105]:

```
# Size no more needed, replace it with SizeNum and drop SizeNum
new_data.Size=new_data.SizeNum
new_data.drop('SizeNum',axis=1,inplace=True)
```

In [106]:

```
new_data.Reviews = pd.to_numeric(new_data.Reviews)
```

In [107]:

```
new_data['Installs']=new_data.Installs.str.replace("+","")
```

In [108]:

```
new_data.Installs=new_data.Installs.str.replace(",","")
new_data.Installs=pd.to_numeric(new_data.Installs)
new_data.Installs.dtype
```

Out[108]:

```
dtype('int64')
```

In [109]:

```
new_data.Price=new_data.Price.str.replace("$","")
new_data.Price=pd.to_numeric(new_data.Price)
new_data.Price.dtype
```

Out[109]:

```
dtype('float64')
```

In [110]:

```
new_data=new_data[(new_data.Rating>=1) & (new_data.Rating<=5) ]
```

In [111]:

```
new_data.drop(new_data.index[new_data.Reviews>new_data.Installs],axis=0,inplace=True)
len(new_data.index)
```

Out[111]:

```
7717
```

In [112]:

```
index_free_and_price_gt_0 = new_data.index[((new_data.Type=='Free')&(new_data.Price>0))]
```

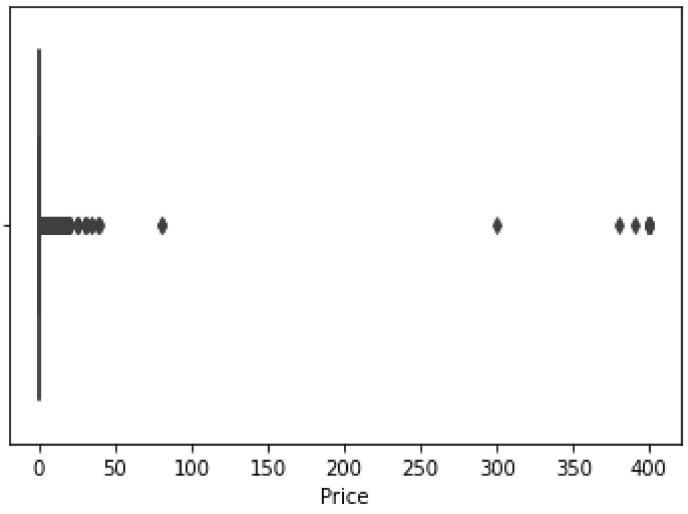
In [113]:

```
if len(index_free_and_price_gt_0)>0:
    print("Dropping following indices:",index_free_and_price_gt_0)
    new_data.drop(index_free_and_price_gt_0,axis=0,inplace=True)
else:
    print("There is no Free Apps with price >0")
```

```
There is no Free Apps with price >0
```

In [114]:

```
import seaborn as sns
ax = sns.boxplot(x='Price', data=new_data)
```



In [115]:

```
import statistics as stc
price_std=stc.stdev(new_data.Price)
price_std
```

Out[115]:

```
17.414783874309933
```

In [116]:

```
price_mean=stc.mean(new_data.Price)
price_mean
```

Out[116]:

```
1.128724893093171
```

In [117]:

```
price_outlier_uplimit=price_mean+3*price_std
price_outlier_uplimit
```

Out[117]:

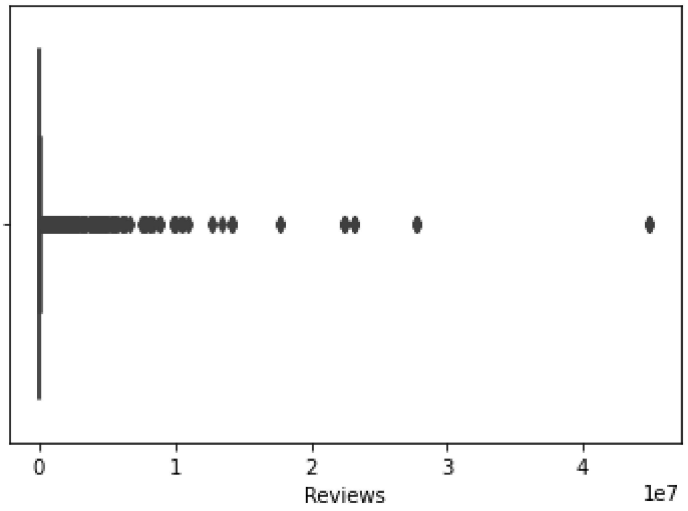
```
53.37307651602297
```

In [118]:

```
sns.boxplot(x='Reviews',data=new_data)
```

Out[118]:

```
<AxesSubplot:xlabel='Reviews'>
```



In [121]:

```
rev_std=stc.stdev(new_data.Reviews)
rev_std
```

Out[121]:

```
1864639.6094670836
```

In [122]:

```
rev_mean=stc.mean(new_data.Reviews)
rev_mean
```

Out[122]:

295127.5482700531

In [123]:

```
rev_outlier_uplimit=rev_mean+3*rev_std
rev_outlier_uplimit
```

Out[123]:

5889046.376671304

In [124]:

```
rev_outlier_downlimit=rev_mean-3*rev_std
rev_outlier_downlimit
```

Out[124]:

-5298791.280131198

In [125]:

```
new_data[new_data.Reviews>rev_outlier_uplimit]
print("# of upper outliers is ",len(new_data[(new_data.Reviews>rev_outlier_uplimit) ]))
```

```
# of upper outliers is  89
```

In [31]:

```
sns.histplot(x='Rating',data=new_data)
```

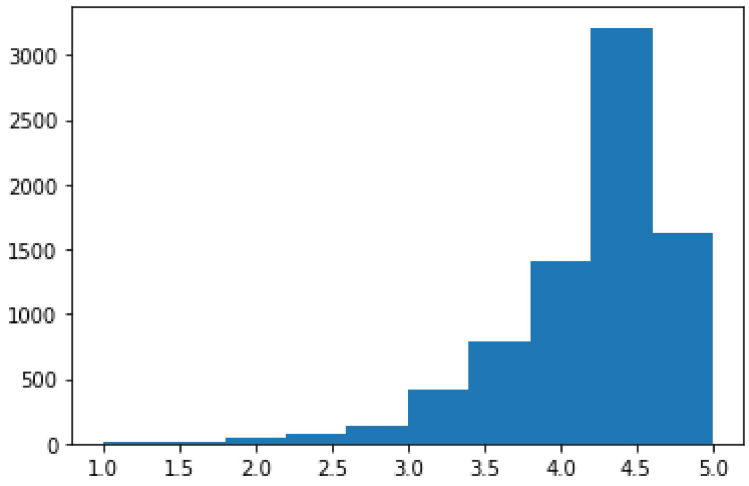
```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-31-33864a45ed39> in <module>
----> 1 sns.histplot(x='Rating',data=new_data)

AttributeError: module 'seaborn' has no attribute 'histplot'
```

In [126]:

```
import matplotlib.pyplot as plt
plt.hist(x='Rating',data=new_data)
```

Out[126]:

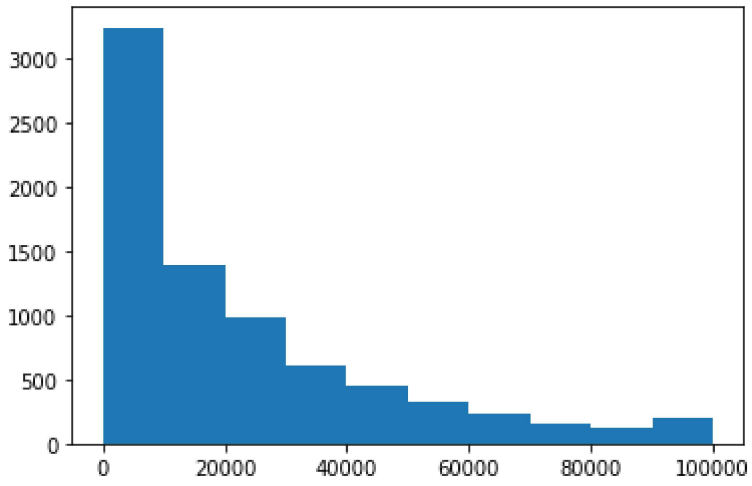
```
(array([  17.,   18.,   39.,   72.,  132.,  408.,  781., 1406., 3212.,
        1632.]),
 array([1. , 1.4, 1.8, 2.2, 2.6, 3. , 3.4, 3.8, 4.2, 4.6, 5. ]),
 <BarContainer object of 10 artists>)
```



In [127]:

```
plt.hist(x='Size',data=new_data,log_scale=True)
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-127-509863d052fb> in <module>
----> 1 plt.hist(x='Size',data=new_data,log_scale=True)

/usr/local/lib/python3.7/site-packages/matplotlib/pyplot.py in hist(x, bins,
 range, density, weights, cumulative, bottom, histtype, align, orientation, r
width, log, color, label, stacked, data, **kwargs)
   2671         align=align, orientation=orientation, rwidth=rwidth, log=log,
   2672         color=color, label=label, stacked=stacked,
-> 2673         **({"data": data} if data is not None else {}), **kwargs)
   2674
   2675

/usr/local/lib/python3.7/site-packages/matplotlib/__init__.py in inner(ax, da
ta, *args, **kwargs)
   1455                 args_and_kwargs.get(label_namer), auto_label)
   1456
-> 1457         return func(*new_args, **new_kwargs)
   1458
   1459     inner.__doc__ = _add_data_doc(inner.__doc__, replace_names)

/usr/local/lib/python3.7/site-packages/matplotlib/axes/_axes.py in hist(self,
x, bins, range, density, weights, cumulative, bottom, histtype, align, orient
ation, rwidth, log, color, label, stacked, **kwargs)
   6791             if patch:
   6792                 p = patch[0]
-> 6793                 p.update(kwargs)
   6794                 if lbl is not None:
   6795                     p.set_label(lbl)

/usr/local/lib/python3.7/site-packages/matplotlib/artist.py in update(self, p
rops)
    994                     func = getattr(self, f"set_{k}", None)
    995                     if not callable(func):
--> 996                         raise AttributeError(f"{type(self).__name__!
r} object "
    997                                              f"has no property {k!
r}")
    998                     ret.append(func(v))

AttributeError: 'Rectangle' object has no property 'log_scale'
```


3000
2500
2000
1500
1000
500
0
0
20000
40000
60000
80000
100000

In [128]:

```
new_data[new_data.Price>=200]
```

Out[128]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	
4197	most expensive app (H)	FAMILY	4.3	6	1500.0	100	Paid	399.99	Everyone	Ente
4362	💎 I'm rich	LIFESTYLE	3.8	718	26000.0	10000	Paid	399.99	Everyone	
4367	I'm Rich - Trump Edition	LIFESTYLE	3.6	275	7300.0	10000	Paid	400.00	Everyone	
5351	I am rich	LIFESTYLE	3.8	3547	1800.0	100000	Paid	399.99	Everyone	
5354	I am Rich Plus	FAMILY	4.0	856	8700.0	10000	Paid	399.99	Everyone	Ente
5355	I am rich VIP	LIFESTYLE	3.8	411	2600.0	10000	Paid	299.99	Everyone	
5356	I Am Rich Premium	FINANCE	4.1	1867	4700.0	50000	Paid	399.99	Everyone	
5357	I am extremely Rich	LIFESTYLE	2.9	41	2900.0	1000	Paid	379.99	Everyone	
5358	I am Rich!	FINANCE	3.8	93	22000.0	1000	Paid	399.99	Everyone	
5359	I am rich(premium)	FINANCE	3.5	472	965.0	5000	Paid	399.99	Everyone	
5362	I Am Rich Pro	FAMILY	4.4	201	2700.0	5000	Paid	399.99	Everyone	Ente
5364	I am rich (Most expensive app)	FINANCE	4.1	129	2700.0	1000	Paid	399.99	Teen	
5366	I Am Rich	FAMILY	3.6	217	4900.0	10000	Paid	389.99	Everyone	Ente
5369	I am Rich	FINANCE	4.3	180	3800.0	5000	Paid	399.99	Everyone	
5373	I AM RICH PRO PLUS	FINANCE	4.0	36	41000.0	1000	Paid	399.99	Everyone	

In [129]:

```
new_data.drop(new_data.index[(new_data.Price>=200)], inplace=True)
len(new_data.index)
```

Out[129]:

7702

In [130]:

```
new_data.drop(new_data.index[(new_data.Reviews>=2000000)], inplace=True)
len(new_data.index)
```

Out[130]:

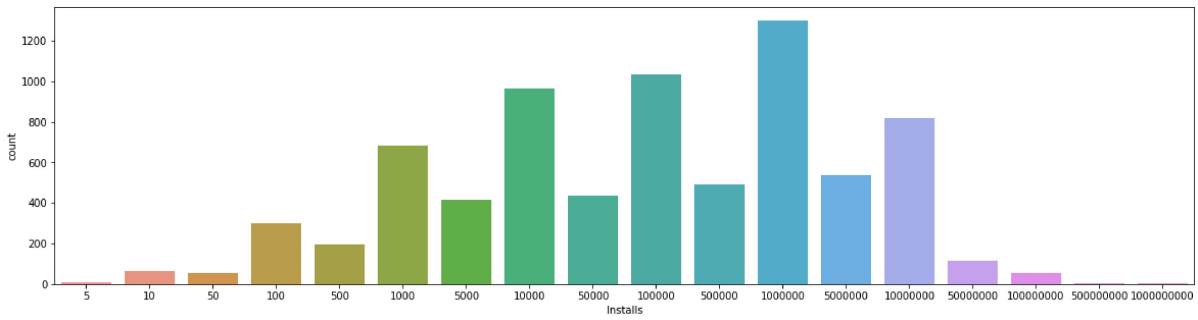
7483

In [131]:

```
plt.figure(figsize=(20,5))
sns.countplot(x='Installs',data=new_data)
```

Out[131]:

<AxesSubplot:xlabel='Installs', ylabel='count'>



In [132]:

```
new_data["Installs"]
```

Out[132]:

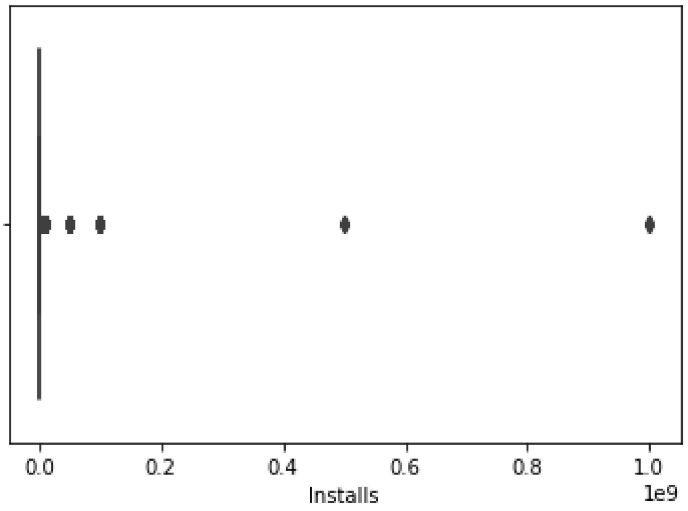
```
0           10000
1          500000
2         5000000
3        50000000
4          100000
           ...
10833        1000
10834         500
10836        5000
10837         100
10840    10000000
Name: Installs, Length: 7483, dtype: int64
```

In [133]:

```
sns.boxplot(x='Installs',data=new_data)
```

Out[133]:

```
<AxesSubplot:xlabel='Installs'>
```



In [134]:

```
install_10_perc=np.percentile(new_data.Installs, 10)
install_10_perc
```

Out[134]:

```
1000.0
```

In [135]:

```
install_25_perc=np.percentile(new_data.Installs, 25)
install_25_perc
```

Out[135]:

```
10000.0
```

In [138]:

```
install_50_perc=np.percentile(new_data.Installs, 50)
install_50_perc
```

Out[138]:

```
100000.0
```

In [137]:

```
install_70_perc=np.percentile(new_data.Installs, 70)
install_70_perc
```

Out[137]:

```
1000000.0
```

In [139]:

```
install_90_perc=np.percentile(new_data.Installs, 90)
install_90_perc
```

Out[139]:

```
10000000.0
```

In [140]:

```
install_99_perc=np.percentile(new_data.Installs, 99)
install_99_perc
```

Out[140]:

```
50000000.0
```

In [141]:

```
print("As result, ",len(new_data[new_data.Installs >= install_99_perc])," will be dropped"
)
```

```
As result,  176  will be dropped
```

In [142]:

```
new_data.drop(new_data.index[new_data.Installs >= install_99_perc],inplace=True)
len(new_data.index)
```

Out[142]:

```
7307
```

In [143]:

```
new_data["Reviews"].dtype
```

Out[143]:

```
dtype('int64')
```

In [144]:

```
inp1=new_data.copy()
inp1.Reviews=inp1.Reviews.apply(np.log1p)
```

In [145]:

```
inp1
```

Out[145]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	5.075174	19000.0	10000	Free	0.0
1	Coloring book moana	ART_AND_DESIGN	3.9	6.875232	14000.0	500000	Free	0.0
2	U Launcher Lite – FREE Live Cool Themes, Hide ...	ART_AND_DESIGN	4.7	11.379520	8700.0	5000000	Free	0.0
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	6.875232	2800.0	100000	Free	0.0
5	Paper flowers instructions	ART_AND_DESIGN	4.4	5.123964	5600.0	50000	Free	0.0
...	...	...	...	...	...	...	...	...
10833	Chemin (fr)	BOOKS_AND_REFERENCE	4.8	3.806662	619.0	1000	Free	0.0
10834	FR Calculator	FAMILY	4.0	2.079442	2600.0	500	Free	0.0
10836	Sya9a Maroc - FR	FAMILY	4.5	3.663562	53000.0	5000	Free	0.0
10837	Fr. Mike Schmitz Audio Teachings	FAMILY	5.0	1.609438	3600.0	100	Free	0.0
10840	iHoroscope - 2018 Daily Horoscope & Astrology	LIFESTYLE	4.5	12.894981	19000.0	10000000	Free	0.0

7307 rows × 13 columns

In [146]:

```
inp1.Installs=inp1.Installs.apply(np.log1p)
inp1.drop(columns=['App','Last Updated','Current Ver','Android Ver'],inplace=True,axis=1)
```

In [147]:

```
inp1.columns
```

Out[147]:

```
Index(['Category', 'Rating', 'Reviews', 'Size', 'Installs', 'Type', 'Price',
       'Content Rating', 'Genres'],
      dtype='object')
```

In [148]:

```
inp1.shape
```

Out[148]:

```
(7307, 9)
```

In [149]:

```
inp2= pd.get_dummies(inp1)
```

In [150]:

```
inp2.head()
```

Out[150]:

	Rating	Reviews	Size	Installs	Price	Category_ART_AND_DESIGN	Category_AUTO_AN
0	4.1	5.075174	19000.0	9.210440	0.0	1	
1	3.9	6.875232	14000.0	13.122365	0.0	1	
2	4.7	11.379520	8700.0	15.424949	0.0	1	
4	4.3	6.875232	2800.0	11.512935	0.0	1	
5	4.4	5.123964	5600.0	10.819798	0.0	1	

5 rows × 158 columns

In [151]:

```
inp2.shape
```

Out[151]:

```
(7307, 158)
```

In [152]:

```
set(inp2.columns)
```

Out[152]:

```
{'Category_ART_AND_DESIGN',
 'Category_AUTO_AND_VEHICLES',
 'Category_BEAUTY',
 'Category_BOOKS_AND_REFERENCE',
 'Category_BUSINESS',
 'Category_COMICS',
 'Category_COMMUNICATION',
 'Category_DATING',
 'Category_EDUCATION',
 'Category_ENTERTAINMENT',
 'Category_EVENTS',
 'Category_FAMILY',
 'Category_FINANCE',
 'Category_FOOD_AND_DRINK',
 'Category_GAME',
 'Category_HEALTH_AND_FITNESS',
 'Category_HOUSE_AND_HOME',
 'Category_LIBRARIES_AND_DEMO',
 'Category_LIFESTYLE',
 'Category_MAPS_AND_NAVIGATION',
 'Category_MEDICAL',
 'Category_NEWS_AND_MAGAZINES',
 'Category_PARENTING',
 'Category_PERSONALIZATION',
 'Category_PHOTOGRAPHY',
 'Category_PRODUCTIVITY',
 'Category_SHOPPING',
 'Category_SOCIAL',
 'Category_SPORTS',
 'Category_TOOLS',
 'Category_TRAVEL_AND_LOCAL',
 'Category_VIDEO_PLAYERS',
 'Category_WEATHER',
 'Content Rating_Adults only 18+',
 'Content Rating_Everyone',
 'Content Rating_Everyone 10+',
 'Content Rating_Mature 17+',
 'Content Rating_Teen',
 'Content Rating_Unrated',
 'Genres_Action',
 'Genres_Action;Action & Adventure',
 'Genres_Adventure',
 'Genres_Adventure;Action & Adventure',
 'Genres_Adventure;Brain Games',
 'Genres_Adventure;Education',
 'Genres_Arcade',
 'Genres_Arcade;Action & Adventure',
 'Genres_Arcade;Pretend Play',
 'Genres_Art & Design',
 'Genres_Art & Design;Creativity',
 'Genres_Art & Design;Pretend Play',
 'Genres_Auto & Vehicles',
 'Genres_Beauty',
 'Genres_Board',
 'Genres_Board;Action & Adventure',
```

```
    'Genres_Board;Brain Games',
    'Genres_Board;Pretend Play',
    'Genres_Books & Reference',
    'Genres_Books & Reference;Education',
    'Genres_Business',
    'Genres_Card',
    'Genres_Card;Action & Adventure',
    'Genres_Card;Brain Games',
    'Genres_Casino',
    'Genres_Casual',
    'Genres_Casual;Action & Adventure',
    'Genres_Casual;Brain Games',
    'Genres_Casual;Creativity',
    'Genres_Casual;Education',
    'Genres_Casual;Music & Video',
    'Genres_Casual;Pretend Play',
    'Genres_Comics',
    'Genres_Comics;Creativity',
    'Genres_Communication',
    'Genres_Dating',
    'Genres_Education',
    'Genres_Education;Action & Adventure',
    'Genres_Education;Brain Games',
    'Genres_Education;Creativity',
    'Genres_Education;Education',
    'Genres_Education;Music & Video',
    'Genres_Education;Pretend Play',
    'Genres_Educational',
    'Genres_Educational;Action & Adventure',
    'Genres_Educational;Brain Games',
    'Genres_Educational;Creativity',
    'Genres_Educational;Education',
    'Genres_Educational;Pretend Play',
    'Genres_Entertainment',
    'Genres_Entertainment;Action & Adventure',
    'Genres_Entertainment;Brain Games',
    'Genres_Entertainment;Creativity',
    'Genres_Entertainment;Education',
    'Genres_Entertainment;Music & Video',
    'Genres_Entertainment;Pretend Play',
    'Genres_Events',
    'Genres_Finance',
    'Genres_Food & Drink',
    'Genres_Health & Fitness',
    'Genres_Health & Fitness;Action & Adventure',
    'Genres_Health & Fitness;Education',
    'Genres_House & Home',
    'Genres_Libraries & Demo',
    'Genres_Lifestyle',
    'Genres_Lifestyle;Pretend Play',
    'Genres_Maps & Navigation',
    'Genres_Medical',
    'Genres_Music',
    'Genres_Music & Audio;Music & Video',
    'Genres_Music;Music & Video',
    'Genres_News & Magazines',
    'Genres_Parenting',
```

```
 'Genres_Parenting;Brain Games',
 'Genres_Parenting;Education',
 'Genres_Parenting;Music & Video',
 'Genres_Personalization',
 'Genres_Photography',
 'Genres_Productivity',
 'Genres_Puzzle',
 'Genres_Puzzle;Action & Adventure',
 'Genres_Puzzle;Brain Games',
 'Genres_Puzzle;Creativity',
 'Genres_Puzzle;Education',
 'Genres_Racing',
 'Genres_Racing;Action & Adventure',
 'Genres_Racing;Pretend Play',
 'Genres_Role Playing',
 'Genres_Role Playing;Action & Adventure',
 'Genres_Role Playing;Brain Games',
 'Genres_Role Playing;Pretend Play',
 'Genres_Shopping',
 'Genres_Simulation',
 'Genres_Simulation;Action & Adventure',
 'Genres_Simulation;Education',
 'Genres_Simulation;Pretend Play',
 'Genres_Social',
 'Genres_Sports',
 'Genres_Sports;Action & Adventure',
 'Genres_Strategy',
 'Genres_Strategy;Action & Adventure',
 'Genres_Strategy;Creativity',
 'Genres_Strategy;Education',
 'Genres_Tools',
 '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',
 'Installs',
 'Price',
 'Rating',
 'Reviews',
 'Size',
 'Type_Free',
 'Type_Paid'}
```

In [153]:

```
data = inp2.drop(columns='Rating')
data.shape
target = pd.DataFrame(inp2.Rating)
target.shape
```

Out[153]:

```
(7307, 1)
```

In [154]:

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(data, target, test_size=0.3, random_st
ate=3)
print("x_train shape is ", x_train.shape)
print("y_train shape is ", y_train.shape)
print("x_test shape is ", x_test.shape)
print("y_test shape is ", y_test.shape)
```

```
x_train shape is  (5114, 157)
y_train shape is  (5114, 1)
x_test shape is  (2193, 157)
y_test shape is  (2193, 1)
```

In [155]:

```
from sklearn.linear_model import LinearRegression
model=LinearRegression()
model.fit(x_train, y_train)
```

Out[155]:

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=Fals
e)
```

In [156]:

```
from sklearn.metrics import r2_score
train_pred=model.predict(x_train)
print("R2 value of the model(by train) is ", r2_score(y_train, train_pred))
```

```
R2 value of the model(by train) is  0.15264772134593896
```

In [157]:

```
#Make predictions on test set and report R2.
test_pred=model.predict(x_test)
print("R2 value of the model(by test) is ", r2_score(y_test, test_pred))
```

```
R2 value of the model(by test) is  0.1426226303095114
```

In [ ]:

```

```