Untitled4

July 25, 2023

[6]:
```
import pandas as pd
```

[7]:
```
import numpy as np
```

[8]:
```
import matplotlib.pyplot as plt
```

[9]:
```
import seaborn as sns
```

[10]:
```
df= pd.read_csv('googleplaystore.csv')
```

[11]:
```
df.shape
```

[11]:
```
(10841, 13)
```

[12]:
```
df.head (5)
```

[12]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19M	10,000+	Free	0	Everyone	Art & Design	January 7, 2018	1.0.0
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14M	500,000+	Free	0	Everyone	Art & Design;Pretend Play	January 15, 2018	2.0.0
2	U Launcher Lite – FREE Live Cool Themes, Hide …	ART_AND_DESIGN	4.7	87510	8.7M	5,000,000+	Free	0	Everyone	Art & Design	August 1, 2018	1.2.4
3	Sketch - Draw & Paint	ART_AND_DESIGN	4.5	215644	25M	50,000,000+	Free	0	Teen	Art & Design	June 8, 2018	Varies with device
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	967	2.8M	100,000+	Free	0	Everyone	Art & Design;Creativity	June 20, 2018	1.1

```
    Android Ver
0  4.0.3 and up
1  4.0.3 and up
2  4.0.3 and up
3    4.2 and up
4    4.4 and up
```

[13]:
```
df.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
```

[14]:
```
df.isnull().sum ()
```

[14]:
```
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
```

[15]:
```
df = df.dropna()
```

[16]:
```
df.shape
```

[16]:
```
(9360, 13)
```

[17]:
```
df.isnull().sum ()
```

[17]:
```
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
```

[18]:
```
df = df[df.Size != 'Varies with device']
```

[19]:
```
def MtoK(b):
    if b[len(b) -1: ] == 'M':
        return(float(b[0: len(b) -1 ])*1000)
    elif b[len(b) -1: ] == 'K' or  b[len(b) -1: ] == 'k':
        return(float(b[0: len(b) -1 ]))
    else:
        return b
```

[20]:
```
df.Size = df.Size.apply(MtoK)
```

[21]:
```
df.info ()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 7723 entries, 0 to 10840
Data columns (total 13 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   App             7723 non-null   object
 1   Category        7723 non-null   object
 2   Rating          7723 non-null   float64
 3   Reviews         7723 non-null   object
 4   Size            7723 non-null   float64
```

```
 5   Installs        7723 non-null   object
 6   Type            7723 non-null   object
 7   Price           7723 non-null   object
 8   Content Rating  7723 non-null   object
 9   Genres          7723 non-null   object
 10  Last Updated    7723 non-null   object
 11  Current Ver     7723 non-null   object
 12  Android Ver     7723 non-null   object
dtypes: float64(2), object(11)
memory usage: 844.7+ KB
```

[22]:
```
df.Size
```

[22]:
```
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: Size, Length: 7723, dtype: float64
```

[23]:
```
df = df[df.Size != 'Varies with device']
```

[24]:
```
df.shape
```

[24]:
```
(7723, 13)
```

[25]:
```
df ["Reviews"] = df ['Reviews'].astype ("int64")
```

[26]:
```
df ["Reviews"].dtype
```

[26]:
```
dtype('int64')
```

[27]:
```
def remove_char(val):
    return(int(val.replace(',','').replace('+','')))
```

[28]:
```
df.Installs = df.Installs.map (remove_char)
```

[29]:
```
df.Installs
```

[29]:
```
0          10000
1         500000
```

```
2          5000000
3         50000000
4           100000
             ...
10833         1000
10834          500
10836         5000
10837          100
10840    10000000
Name: Installs, Length: 7723, dtype: int64
```

[30]:
```
def remove_symbol(val):
    return(float(val.replace("$", "")))
```

[31]:
```
df.Price = df.Price.apply(remove_symbol)
```

[32]:
```
df ["Price"].dtype
```

[32]:
```
dtype('float64')
```

[33]:
```
df.Price
```

[33]:
```
0        0.0
1        0.0
2        0.0
3        0.0
4        0.0
        ...
10833    0.0
10834    0.0
10836    0.0
10837    0.0
10840    0.0
Name: Price, Length: 7723, dtype: float64
```

[34]:
```
df.shape
```

[34]:
```
(7723, 13)
```

[35]:
```
df[(df.Rating <1) | (df.Rating>5)]
```

[35]:
```
Empty DataFrame
Columns: [App, Category, Rating, Reviews, Size, Installs, Type, Price, Content
Rating, Genres, Last Updated, Current Ver, Android Ver]
Index: []
```

[36]:
```
df.loc[df.Rating < 1] & df.loc[df.Rating > 5]
```

[36]:
```
Empty DataFrame
Columns: [App, Category, Rating, Reviews, Size, Installs, Type, Price, Content
Rating, Genres, Last Updated, Current Ver, Android Ver]
Index: []
```

[37]:
```
df.loc[df.Reviews > df.Installs]
```

[37]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver
2454	KBA-EZ Health Guide	MEDICAL	5.0	4	25000.0	1	Free	0.00	Everyone	Medical	August 2, 2018	1.0.72	4.0.3 and up
5917	Ra Ga Ba	GAME	5.0	2	20000.0	1	Paid	1.49	Everyone	Arcade	February 8, 2017	1.0.4	2.3 and up
6700	Brick Breaker BR	GAME	5.0	7	19000.0	5	Free	0.00	Everyone	Arcade	July 23, 2018	1	4.1 and up
7402	Trovami se ci riesci	GAME	5.0	11	6100.0	10	Free	0.00	Everyone	Arcade	March 11, 2017	0.1	2.3 and up
8591	DN Blog	SOCIAL	5.0	20	4200.0	10	Free	0.00	Teen	Social	July 23, 2018	1	4.0 and up
10697	Mu.F.O.	GAME	5.0	2	16000.0	1	Paid	0.99	Everyone	Arcade	March 3, 2017	1	2.3 and up

[38]:
```
df.loc[df.Reviews > df.Installs].describe ()
```

[38]:

	Rating	Reviews	Size	Installs	Price
count	6.0	6.000000	6.000000	6.000000	6.000000
mean	5.0	7.666667	15050.000000	4.666667	0.413333
std	0.0	6.947422	8219.914841	4.412105	0.659566
min	5.0	2.000000	4200.000000	1.000000	0.000000
25%	5.0	2.500000	8575.000000	1.000000	0.000000
50%	5.0	5.500000	17500.000000	3.000000	0.000000
75%	5.0	10.000000	19750.000000	8.750000	0.742500
max	5.0	20.000000	25000.000000	10.000000	1.490000

[39]:
```
df.columns
```

[39]:
```
Index(['App', 'Category', 'Rating', 'Reviews', 'Size', 'Installs', 'Type',
       'Price', 'Content Rating', 'Genres', 'Last Updated', 'Current Ver',
```

```
       'Android Ver'],
      dtype='object')
```

[40]:
```
df[['Reviews','Installs']]
```

[40]:
```
       Reviews  Installs
0          159     10000
1          967    500000
2        87510   5000000
3       215644  50000000
4          967    100000
…          …        …
10833       44      1000
10834        7       500
10836       38      5000
10837        4       100
10840   398307  10000000

[7723 rows x 2 columns]
```

[41]:
```
df['RAI'] = np.where((df['Reviews'] <= df['Installs']), df['Installs'], np.nan)
```

[42]:
```
df['RAI'].shape
```

[42]:
```
(7723,)
```

[43]:
```
df['RAI'].describe()
```

[43]:
```
count    7.717000e+03
mean     8.430620e+06
std      5.017636e+07
min      5.000000e+00
25%      1.000000e+04
50%      1.000000e+05
75%      1.000000e+06
max      1.000000e+09
Name: RAI, dtype: float64
```

[52]:
```
df = df.dropna()
```

[53]:
```
df.columns
```

[53]:
```
Index(['App', 'Category', 'Rating', 'Reviews', 'Size', 'Installs', 'Type',
       'Price', 'Content Rating', 'Genres', 'Last Updated', 'Current Ver',
       'Android Ver'],
      dtype='object')
```

[ ]:
```
df = df.drop(['RAI'], axis = 1)
```

[49]:
```
df.shape
```

[49]:
```
(7717, 13)
```

[56]:
```
df.columns
```

[56]:
```
Index(['App', 'Category', 'Rating', 'Reviews', 'Size', 'Installs', 'Type',
       'Price', 'Content Rating', 'Genres', 'Last Updated', 'Current Ver',
       'Android Ver'],
      dtype='object')
```

[58]:
```
df.loc[df.Price > 0]
```

[58]:
```
                                            App           Category  \
234    TurboScan: scan documents and receipts in PDF           BUSINESS
235                  Tiny Scanner Pro: PDF Doc Scan           BUSINESS
290    TurboScan: scan documents and receipts in PDF           BUSINESS
291                  Tiny Scanner Pro: PDF Doc Scan           BUSINESS
477                                      Calculator             DATING
…                                                 …                  …
10682                           Fruit Ninja Classic               GAME
10690                                     FO Bixby      PERSONALIZATION
10760                                Fast Tract Diet  HEALTH_AND_FITNESS
10782                         Trine 2: Complete Story               GAME
10785                                   sugar, sugar             FAMILY

       Rating  Reviews     Size  Installs  Type  Price Content Rating  \
234       4.7    11442   6800.0    100000  Paid   4.99       Everyone
235       4.8    10295  39000.0    100000  Paid   4.99       Everyone
290       4.7    11442   6800.0    100000  Paid   4.99       Everyone
291       4.8    10295  39000.0    100000  Paid   4.99       Everyone
477       2.6       57   6200.0      1000  Paid   6.99       Everyone
…           …        …        …         …     …      …
10682     4.3    85468  36000.0   1000000  Paid   0.99       Everyone
10690     5.0        5    861.0       100  Paid   0.99       Everyone
10760     4.4       35   2400.0      1000  Paid   7.99       Everyone
10782     3.8      252  11000.0     10000  Paid  16.99           Teen
10785     4.2     1405   9500.0     10000  Paid   1.20       Everyone

          Genres      Last Updated  Current Ver  Android Ver
234     Business    March 25, 2018        1.5.2   4.0 and up
235     Business    April 11, 2017        3.4.6   3.0 and up
290     Business    March 25, 2018        1.5.2   4.0 and up
291     Business    April 11, 2017        3.4.6   3.0 and up
477       Dating  October 25, 2017        1.1.6   4.0 and up
```

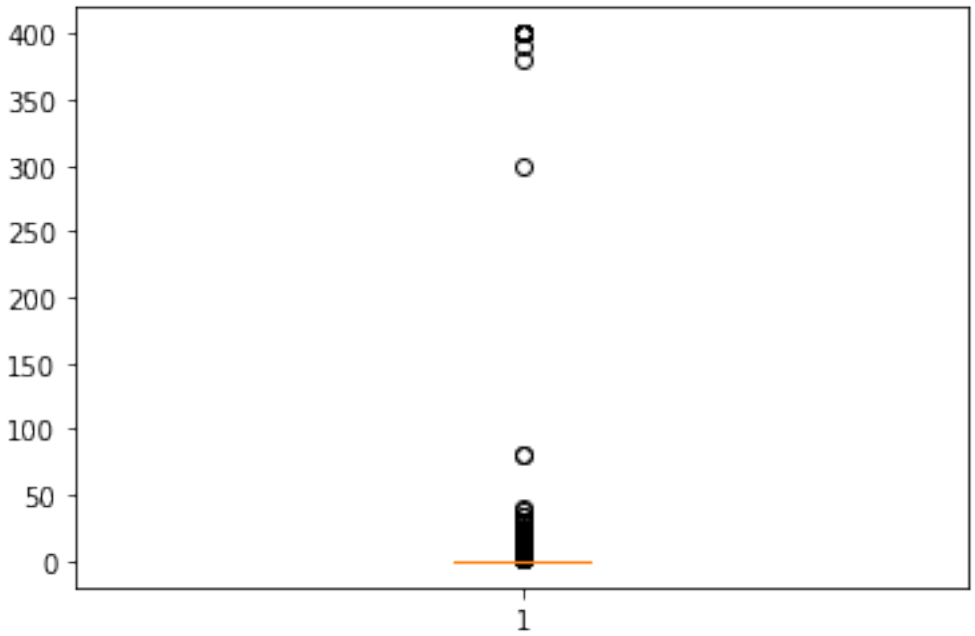
```
...                 ...                 ...           ...           ...
10682           Arcade       June 8, 2018  2.4.1.485300  4.0.3 and up
10690    Personalization     April 25, 2018           0.2    7.0 and up
10760  Health & Fitness     August 8, 2018         1.9.3    4.2 and up
10782           Action  February 27, 2015          2.22    5.0 and up
10785           Puzzle       June 5, 2018           2.7    2.3 and up

[575 rows x 13 columns]
```

[59]:
```
df[np.logical_and(df['Type'] == 'Free', df['Price'] > 0)]
```

[59]:
```
Empty DataFrame
Columns: [App, Category, Rating, Reviews, Size, Installs, Type, Price, Content
Rating, Genres, Last Updated, Current Ver, Android Ver]
Index: []
```

[60]:
```
plt.boxplot(df['Price'])
```

[60]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x7f6beb4f6310>,
  <matplotlib.lines.Line2D at 0x7f6beb4f6610>],
 'caps': [<matplotlib.lines.Line2D at 0x7f6beb4f6950>,
  <matplotlib.lines.Line2D at 0x7f6beb4f6c90>],
 'boxes': [<matplotlib.lines.Line2D at 0x7f6beb4f6050>],
 'medians': [<matplotlib.lines.Line2D at 0x7f6be9487050>],
 'fliers': [<matplotlib.lines.Line2D at 0x7f6be9487390>],
 'means': []}
```

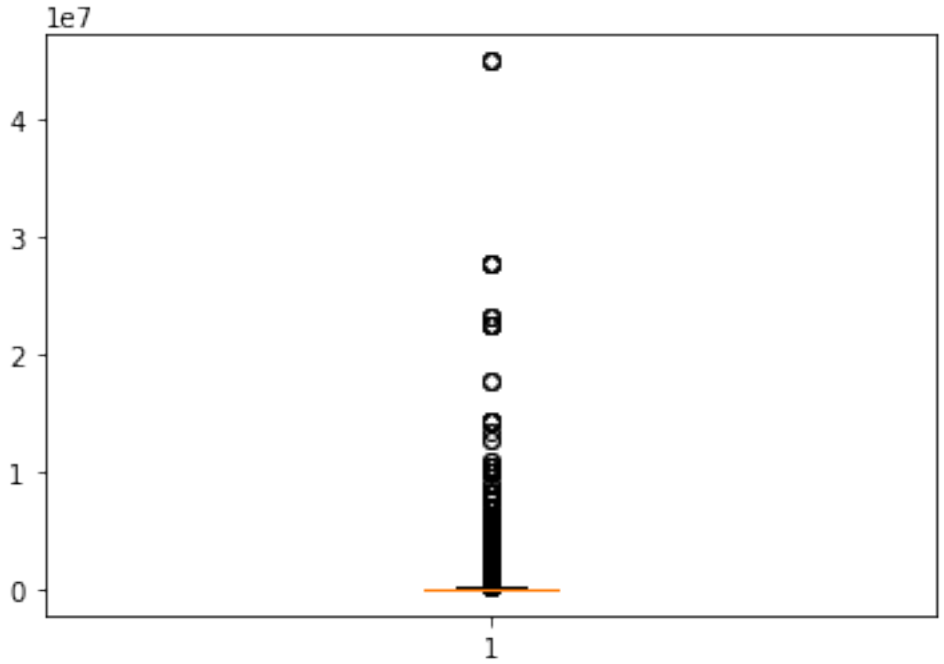


[62]:
```
df['Price'].describe()
```

[62]:
```
count    7717.000000
mean        1.128725
std        17.414784
min         0.000000
25%         0.000000
50%         0.000000
75%         0.000000
max       400.000000
Name: Price, dtype: float64
```

[63]:
```
plt.boxplot(df['Reviews'])
```

[63]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x7f6be93fbd90>,
  <matplotlib.lines.Line2D at 0x7f6be9384150>],
 'caps': [<matplotlib.lines.Line2D at 0x7f6be9384490>,
  <matplotlib.lines.Line2D at 0x7f6be93847d0>],
 'boxes': [<matplotlib.lines.Line2D at 0x7f6be93fbbd0>],
 'medians': [<matplotlib.lines.Line2D at 0x7f6be9384b50>],
 'fliers': [<matplotlib.lines.Line2D at 0x7f6be9384e90>],
 'means': []}
```



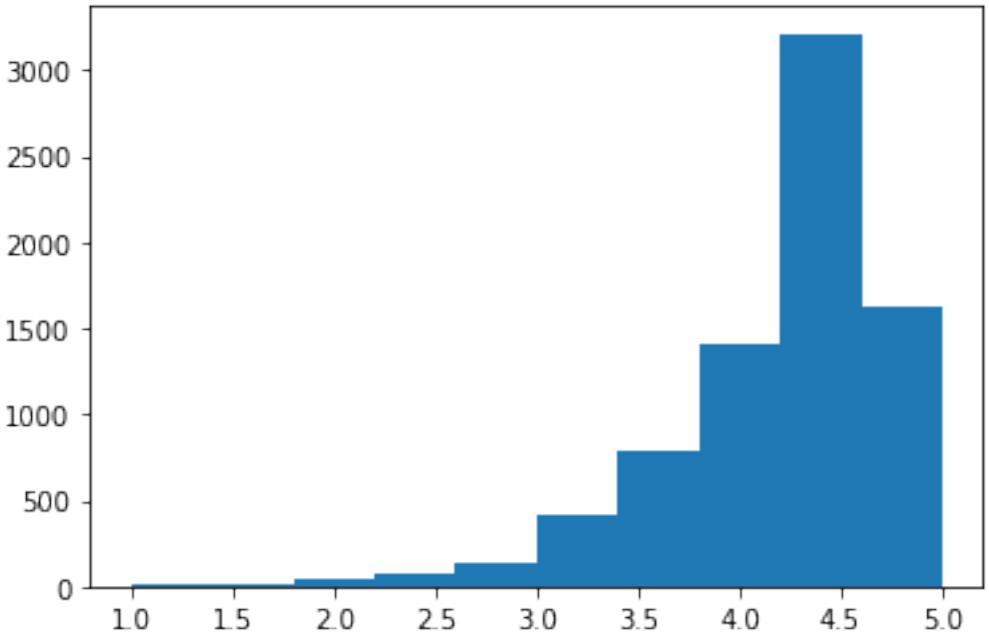
[ ]:

[64]:
```
df['Reviews'].describe()
```

[64]:
```
count    7.717000e+03
mean     2.951275e+05
std      1.864640e+06
min      1.000000e+00
25%      1.090000e+02
50%      2.351000e+03
75%      3.910900e+04
max      4.489389e+07
Name: Reviews, dtype: float64
```

[65]:
```
plt.hist(df['Rating'])
```

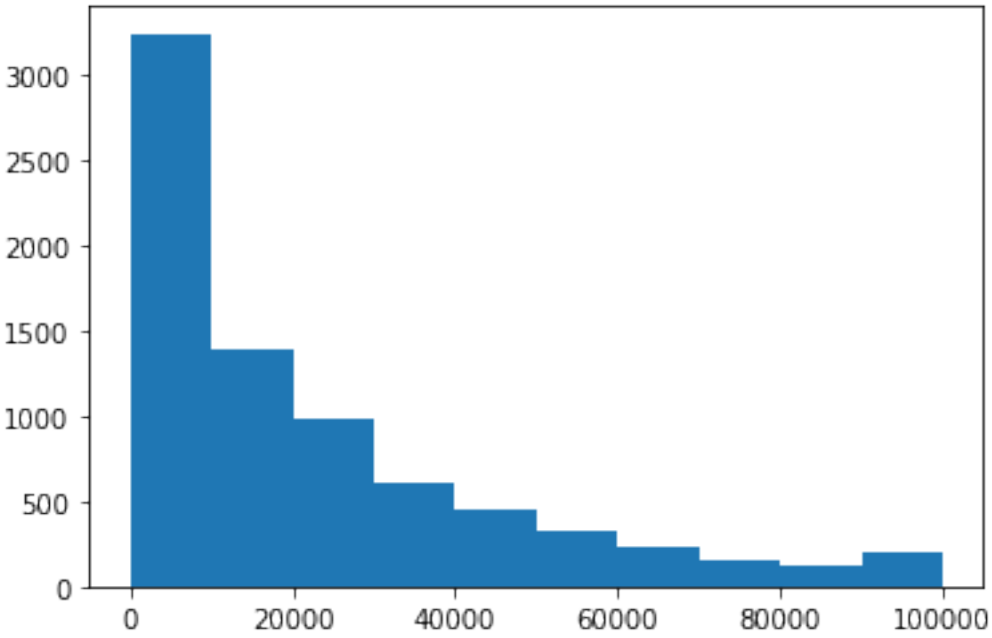
[65]:
```
(array([  17.,   18.,   39.,   72.,  132.,  408.,  781., 1406., 3212.,
        1632.]),
 array([1. , 1.4, 1.8, 2.2, 2.6, 3. , 3.4, 3.8, 4.2, 4.6, 5. ]),
 <BarContainer object of 10 artists>)
```



[66]:
```
plt.hist(df['Size'])
```

[66]:
```
(array([3245., 1398.,  991.,  606.,  449.,  325.,  226.,  161.,  117.,
         199.]),
 array([8.500000e+00, 1.000765e+04, 2.000680e+04, 3.000595e+04,
        4.000510e+04, 5.000425e+04, 6.000340e+04, 7.000255e+04,
        8.000170e+04, 9.000085e+04, 1.000000e+05]),
```

```
<BarContainer object of 10 artists>)
```



[67]:
```
df.loc[df.Price > 100]
```

[67]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated
4197	most expensive app (H)	FAMILY	4.3	6	1500.0	100	Paid	399.99	Everyone	Entertainment	July 16, 2018
4362	I'm rich	LIFESTYLE	3.8	718	26000.0	10000	Paid	399.99	Everyone	Lifestyle	March 11, 2018
4367	I'm Rich - Trump Edition	LIFESTYLE	3.6	275	7300.0	10000	Paid	400.00	Everyone	Lifestyle	May 3, 2018
5351	I am rich	LIFESTYLE	3.8	3547	1800.0	100000	Paid	399.99	Everyone	Lifestyle	January 12, 2018
5354	I am Rich Plus	FAMILY	4.0	856	8700.0	10000	Paid	399.99	Everyone	Entertainment	July 6, 2018
5355	I am rich VIP	LIFESTYLE	3.8	411	2600.0	10000	Paid	299.99	Everyone	Lifestyle	July 31, 2018
5356	I Am Rich Premium	FINANCE	4.1	1867	4700.0	50000	Paid	399.99	Everyone	Finance	November 12, 2017
5357	I am extremely Rich	LIFESTYLE	2.9	41	2900.0	1000	Paid	379.99	Everyone	Lifestyle	July 1, 2018
5358	I am Rich!	FINANCE	3.8	93	22000.0	1000	Paid	399.99	Everyone	Finance	December 11, 2017
5359	I am rich(premium)	FINANCE	3.5	472	965.0	5000	Paid	399.99	Everyone	Finance	May 1, 2017
5362	I Am Rich Pro	FAMILY	4.4	201	2700.0	5000	Paid	399.99	Everyone	Entertainment	January 19, 2018
5364	I am rich (Most expensive app)	FINANCE	4.1	129	2700.0	1000	Paid	399.99	Everyone	Finance	December 6, 2017
5366	I Am Rich	FAMILY	3.6	217	4900.0	10000	Paid	399.99	Everyone	Entertainment	June 22, 2018
5369	I am Rich	FINANCE	4.3	180	3800.0	5000	Paid	399.99	Everyone	Finance	March 22, 2018
5373	I AM RICH PRO PLUS	FINANCE	4.0	36	41000.0	1000	Paid	399.99	Everyone	Finance	May 30, 2017

```
5354     10000  Paid  399.99       Everyone  Entertainment       May 19, 2018
5355     10000  Paid  299.99       Everyone      Lifestyle      July 21, 2018
5356     50000  Paid  399.99       Everyone        Finance  November 12, 2017
5357      1000  Paid  379.99       Everyone      Lifestyle       July 1, 2018
5358      1000  Paid  399.99       Everyone        Finance  December 11, 2017
5359      5000  Paid  399.99       Everyone        Finance        May 1, 2017
5362      5000  Paid  399.99       Everyone  Entertainment       May 30, 2017
5364      1000  Paid  399.99           Teen        Finance   December 6, 2017
5366     10000  Paid  389.99       Everyone  Entertainment      June 22, 2018
5369      5000  Paid  399.99       Everyone        Finance     March 22, 2018
5373      1000  Paid  399.99       Everyone        Finance      June 25, 2018

     Current Ver   Android Ver
4197           1    7.0 and up
4362       1.0.0    4.4 and up
4367       1.0.1    4.1 and up
5351           2  4.0.3 and up
5354           3    4.4 and up
5355       1.1.1    4.3 and up
5356         1.6    4.0 and up
5357           1    4.0 and up
5358           1    4.1 and up
5359         3.4    4.4 and up
5362        1.54    1.6 and up
5364           2  4.0.3 and up
5366         1.5    4.2 and up
5369           1    4.2 and up
5373       1.0.2    4.1 and up
```

[68]:
```
df[df.Price>100]
```

[68]:
```
                                App   Category  Rating  Reviews     Size  \
4197          most expensive app (H)     FAMILY     4.3        6   1500.0
4362                       I'm rich  LIFESTYLE     3.8      718  26000.0
4367        I'm Rich - Trump Edition  LIFESTYLE     3.6      275   7300.0
5351                      I am rich  LIFESTYLE     3.8     3547   1800.0
5354                I am Rich Plus     FAMILY     4.0      856   8700.0
5355                 I am rich VIP  LIFESTYLE     3.8      411   2600.0
5356              I Am Rich Premium    FINANCE     4.1     1867   4700.0
5357            I am extremely Rich  LIFESTYLE     2.9       41   2900.0
5358                    I am Rich!    FINANCE     3.8       93  22000.0
5359             I am rich(premium)    FINANCE     3.5      472    965.0
5362                  I Am Rich Pro     FAMILY     4.4      201   2700.0
5364  I am rich (Most expensive app)    FINANCE     4.1      129   2700.0
5366                      I Am Rich     FAMILY     3.6      217   4900.0
5369                      I am Rich    FINANCE     4.3      180   3800.0
5373             I AM RICH PRO PLUS    FINANCE     4.0       36  41000.0
```

	Installs	Type	Price	Content Rating	Genres	Last Updated	\
4197	100	Paid	399.99	Everyone	Entertainment	July 16, 2018	
4362	10000	Paid	399.99	Everyone	Lifestyle	March 11, 2018	
4367	10000	Paid	400.00	Everyone	Lifestyle	May 3, 2018	
5351	100000	Paid	399.99	Everyone	Lifestyle	January 12, 2018	
5354	10000	Paid	399.99	Everyone	Entertainment	May 19, 2018	
5355	10000	Paid	299.99	Everyone	Lifestyle	July 21, 2018	
5356	50000	Paid	399.99	Everyone	Finance	November 12, 2017	
5357	1000	Paid	379.99	Everyone	Lifestyle	July 1, 2018	
5358	1000	Paid	399.99	Everyone	Finance	December 11, 2017	
5359	5000	Paid	399.99	Everyone	Finance	May 1, 2017	
5362	5000	Paid	399.99	Everyone	Entertainment	May 30, 2017	
5364	1000	Paid	399.99	Teen	Finance	December 6, 2017	
5366	10000	Paid	389.99	Everyone	Entertainment	June 22, 2018	
5369	5000	Paid	399.99	Everyone	Finance	March 22, 2018	
5373	1000	Paid	399.99	Everyone	Finance	June 25, 2018	

	Current Ver	Android Ver
4197	1	7.0 and up
4362	1.0.0	4.4 and up
4367	1.0.1	4.1 and up
5351	2	4.0.3 and up
5354	3	4.4 and up
5355	1.1.1	4.3 and up
5356	1.6	4.0 and up
5357	1	4.0 and up
5358	1	4.1 and up
5359	3.4	4.4 and up
5362	1.54	1.6 and up
5364	2	4.0.3 and up
5366	1.5	4.2 and up
5369	1	4.2 and up
5373	1.0.2	4.1 and up

[69]:
```
df[df.Price>100].shape
```

[69]: (15, 13)

[70]:
```
df.shape
```

[70]: (7717, 13)

[71]:
```
df[df.Price <=100].shape
```

[71]: (7702, 13)

[72]:
```
df = df[df.Price <=100]
```

[73]:
```
df.shape
```

[73]:
```
(7702, 13)
```

[74]:
```
df.describe()
```

[74]:

	Rating	Reviews	Size	Installs	Price
count	7702.000000	7.702000e+03	7702.000000	7.702000e+03	7702.000000
mean	4.173890	2.957011e+05	23004.020709	8.447011e+06	0.368802
std	0.544481	1.866409e+06	23466.178824	5.022383e+07	2.348127
min	1.000000	1.000000e+00	8.500000	5.000000e+00	0.000000
25%	4.000000	1.090000e+02	5300.000000	1.000000e+04	0.000000
50%	4.300000	2.374500e+03	14000.000000	1.000000e+05	0.000000
75%	4.500000	3.949125e+04	33000.000000	1.000000e+06	0.000000
max	5.000000	4.489389e+07	100000.000000	1.000000e+09	79.990000

[75]:
```
df['Reviews'].describe()
```

[75]:
```
count    7.702000e+03
mean     2.957011e+05
std      1.866409e+06
min      1.000000e+00
25%      1.090000e+02
50%      2.374500e+03
75%      3.949125e+04
max      4.489389e+07
Name: Reviews, dtype: float64
```

[76]:
```
df.loc[df.Reviews > 2000000]
```

[76]:

	App	Category	Rating	\
345	Yahoo Mail - Stay Organized	COMMUNICATION	4.3	
347	imo free video calls and chat	COMMUNICATION	4.3	
366	UC Browser Mini -Tiny Fast Private & Secure	COMMUNICATION	4.4	
378	UC Browser - Fast Download Private & Secure	COMMUNICATION	4.5	
383	imo free video calls and chat	COMMUNICATION	4.3	
…	…	…	…	
9142	Need for Speed No Limits	GAME	4.4	
9166	Modern Combat 5: eSports FPS	GAME	4.3	
10186	Farm Heroes Saga	FAMILY	4.4	
10190	Fallout Shelter	FAMILY	4.6	
10327	Garena Free Fire	GAME	4.5	

	Reviews	Size	Installs	Type	Price	Content Rating	\
345	4187998	16000.0	100000000	Free	0.0	Everyone	

```
347     4785892  11000.0  500000000  Free    0.0       Everyone
366     3648120   3300.0  100000000  Free    0.0           Teen
378    17712922  40000.0  500000000  Free    0.0           Teen
383     4785988  11000.0  500000000  Free    0.0       Everyone
…         …       …         …    …    …              …
9142    3344300  22000.0   50000000  Free    0.0   Everyone 10+
9166    2903386  58000.0  100000000  Free    0.0     Mature 17+
10186   7615646  71000.0  100000000  Free    0.0       Everyone
10190   2721923  25000.0   10000000  Free    0.0           Teen
10327   5534114  53000.0  100000000  Free    0.0           Teen

              Genres    Last Updated       Current Ver   Android Ver
345    Communication   July 18, 2018             5.29.3    4.4 and up
347    Communication    June 8, 2018  9.8.000000010501    4.0 and up
366    Communication   July 18, 2018             11.4.0    4.0 and up
378    Communication  August 2, 2018       12.8.5.1121    4.0 and up
383    Communication    June 8, 2018  9.8.000000010501    4.0 and up
…                …              …                 …           …
9142          Racing   July 24, 2018             2.12.1    4.1 and up
9166          Action   July 24, 2018             3.2.1c    4.0 and up
10186         Casual  August 7, 2018              5.2.6    2.3 and up
10190     Simulation   June 11, 2018            1.13.12    4.1 and up
10327         Action  August 3, 2018             1.21.0  4.0.3 and up

[219 rows x 13 columns]
```

[77]:
```
df[df.Reviews > 2000000]
```

[77]:
```
                                                App       Category  Rating  \
345                     Yahoo Mail - Stay Organized  COMMUNICATION     4.3
347                   imo free video calls and chat  COMMUNICATION     4.3
366    UC Browser Mini -Tiny Fast Private & Secure  COMMUNICATION     4.4
378    UC Browser - Fast Download Private & Secure  COMMUNICATION     4.5
383                   imo free video calls and chat  COMMUNICATION     4.3
…                                               …            …     …
9142                    Need for Speed  No Limits           GAME     4.4
9166                  Modern Combat 5: eSports FPS           GAME     4.3
10186                              Farm Heroes Saga         FAMILY     4.4
10190                               Fallout Shelter         FAMILY     4.6
10327                               Garena Free Fire           GAME     4.5

        Reviews     Size   Installs  Type  Price Content Rating  \
345     4187998  16000.0  100000000  Free    0.0       Everyone
347     4785892  11000.0  500000000  Free    0.0       Everyone
366     3648120   3300.0  100000000  Free    0.0           Teen
378    17712922  40000.0  500000000  Free    0.0           Teen
383     4785988  11000.0  500000000  Free    0.0       Everyone
```

```
…         …      …         …    …     …              …
9142    3344300  22000.0   50000000  Free    0.0   Everyone 10+
9166    2903386  58000.0  100000000  Free    0.0     Mature 17+
10186   7615646  71000.0  100000000  Free    0.0       Everyone
10190   2721923  25000.0   10000000  Free    0.0           Teen
10327   5534114  53000.0  100000000  Free    0.0           Teen

              Genres    Last Updated       Current Ver   Android Ver
345    Communication   July 18, 2018             5.29.3    4.4 and up
347    Communication    June 8, 2018  9.8.000000010501    4.0 and up
366    Communication   July 18, 2018             11.4.0    4.0 and up
378    Communication  August 2, 2018       12.8.5.1121    4.0 and up
383    Communication    June 8, 2018  9.8.000000010501    4.0 and up
…                …               …                 …           …
9142          Racing   July 24, 2018             2.12.1    4.1 and up
9166          Action   July 24, 2018             3.2.1c    4.0 and up
10186         Casual  August 7, 2018              5.2.6    2.3 and up
10190     Simulation   June 11, 2018            1.13.12    4.1 and up
10327         Action  August 3, 2018             1.21.0  4.0.3 and up

[219 rows x 13 columns]
```

[78]:
```
df[df.Reviews > 2000000].shape
```

[78]:
```
(219, 13)
```

[79]:
```
df.shape
```

[79]:
```
(7702, 13)
```

[80]:
```
df = df[df.Reviews <= 2000000]
```

[81]:
```
df.shape
```

[81]:
```
(7483, 13)
```

[82]:
```
df.describe()
```

[82]:
```
             Rating       Reviews           Size      Installs        Price
count  7483.000000  7.483000e+03    7483.000000  7.483000e+03  7483.000000
mean      4.165789  7.260651e+04   22027.284177  3.947465e+06     0.379595
std       0.549946  2.123720e+05   22582.977041  2.781831e+07     2.381384
min       1.000000  1.000000e+00       8.500000  5.000000e+00     0.000000
25%       4.000000  9.900000e+01    5100.000000  1.000000e+04     0.000000
50%       4.300000  2.026000e+03   14000.000000  1.000000e+05     0.000000
75%       4.500000  3.238600e+04   31000.000000  1.000000e+06     0.000000
max       5.000000  1.986068e+06  100000.000000  1.000000e+09    79.990000
```

[83]:
```
df['Installs'].describe()
```

[83]:
```
count    7.483000e+03
mean     3.947465e+06
std      2.781831e+07
min      5.000000e+00
25%      1.000000e+04
50%      1.000000e+05
75%      1.000000e+06
max      1.000000e+09
Name: Installs, dtype: float64
```

[84]:
```
np.arange(0,1,0.05)
```

[84]:
```
array([0.  , 0.05, 0.1 , 0.15, 0.2 , 0.25, 0.3 , 0.35, 0.4 , 0.45, 0.5 ,
       0.55, 0.6 , 0.65, 0.7 , 0.75, 0.8 , 0.85, 0.9 , 0.95])
```

[85]:
```
df['Installs'].quantile(q = np.arange(0,1,0.05))
```

[85]:
```
0.00           5.0
0.05         100.0
0.10        1000.0
0.15        1000.0
0.20        5000.0
0.25       10000.0
0.30       10000.0
0.35       10000.0
0.40       50000.0
0.45      100000.0
0.50      100000.0
0.55      100000.0
0.60      500000.0
0.65     1000000.0
0.70     1000000.0
0.75     1000000.0
0.80     5000000.0
0.85     5000000.0
0.90    10000000.0
0.95    10000000.0
Name: Installs, dtype: float64
```

[86]:
```
df['Installs'].quantile(0.99)
```

[86]:
```
50000000.0
```

[87]:
```
df.shape
```

[87]: (7483, 13)

[88]:
```
df[df.Installs > 10000000]
```

[88]:

```
                                        App        Category  Rating  \
3                      Sketch - Draw & Paint  ART_AND_DESIGN     4.5
194    OfficeSuite : Free Office + PDF Editor        BUSINESS     4.3
225                            Secure Folder        BUSINESS     3.8
293    OfficeSuite : Free Office + PDF Editor        BUSINESS     4.3
346             imo beta free calls and text   COMMUNICATION     4.3
…                                        …              …     …
10378                               BMX Boy            GAME     4.2
10408               Shoot Hunter-Gun Killer            GAME     4.3
10429            Talking Tom Bubble Shooter          FAMILY     4.4
10513         Flight Simulator: Fly Plane 3D          FAMILY     4.0
10549                     Toy Truck Rally 3D            GAME     4.0

       Reviews     Size   Installs  Type  Price Content Rating         Genres  \
3       215644  25000.0   50000000  Free    0.0           Teen   Art & Design
194    1002861  35000.0  100000000  Free    0.0       Everyone       Business
225      14760   8600.0   50000000  Free    0.0       Everyone       Business
293    1002859  35000.0  100000000  Free    0.0       Everyone       Business
346     659395  11000.0  100000000  Free    0.0       Everyone  Communication
…          …       …         …   …    …    …            …              …
10378   839206  12000.0   50000000  Free    0.0       Everyone         Racing
10408   320334  27000.0   50000000  Free    0.0           Teen         Action
10429   687136  54000.0   50000000  Free    0.0       Everyone         Casual
10513   660613  21000.0   50000000  Free    0.0       Everyone     Simulation
10549   301895  25000.0   50000000  Free    0.0       Everyone         Racing

             Last Updated         Current Ver Android Ver
3            June 8, 2018  Varies with device  4.2 and up
194        August 2, 2018           9.7.14188  4.1 and up
225      January 31, 2018            1.1.07.6  7.0 and up
293        August 2, 2018           9.7.14188  4.1 and up
346          June 7, 2018  9.8.000000010492  4.0 and up
…                     …                 …          …
10378  September 20, 2017             1.16.33  4.1 and up
10408      August 8, 2018               1.1.2  4.1 and up
10429        May 25, 2018             1.5.3.20  4.1 and up
10513       March 1, 2017                1.32  2.3 and up
10549        May 23, 2018               1.4.4  4.1 and up

[176 rows x 13 columns]
```

[89]:
```
df.loc[df.Installs > 10000000]
```

[89]:

	App	Category	Rating	\
3	Sketch - Draw & Paint	ART_AND_DESIGN	4.5	
194	OfficeSuite : Free Office + PDF Editor	BUSINESS	4.3	
225	Secure Folder	BUSINESS	3.8	
293	OfficeSuite : Free Office + PDF Editor	BUSINESS	4.3	
346	imo beta free calls and text	COMMUNICATION	4.3	
…	…	…	…	
10378	BMX Boy	GAME	4.2	
10408	Shoot Hunter-Gun Killer	GAME	4.3	
10429	Talking Tom Bubble Shooter	FAMILY	4.4	
10513	Flight Simulator: Fly Plane 3D	FAMILY	4.0	
10549	Toy Truck Rally 3D	GAME	4.0	

	Reviews	Size	Installs	Type	Price	Content Rating	Genres	\
3	215644	25000.0	50000000	Free	0.0	Teen	Art & Design	
194	1002861	35000.0	100000000	Free	0.0	Everyone	Business	
225	14760	8600.0	50000000	Free	0.0	Everyone	Business	
293	1002859	35000.0	100000000	Free	0.0	Everyone	Business	
346	659395	11000.0	100000000	Free	0.0	Everyone	Communication	
…	…	…	…	…	…	…	…	
10378	839206	12000.0	50000000	Free	0.0	Everyone	Racing	
10408	320334	27000.0	50000000	Free	0.0	Teen	Action	
10429	687136	54000.0	50000000	Free	0.0	Everyone	Casual	
10513	660613	21000.0	50000000	Free	0.0	Everyone	Simulation	
10549	301895	25000.0	50000000	Free	0.0	Everyone	Racing	

	Last Updated	Current Ver	Android Ver
3	June 8, 2018	Varies with device	4.2 and up
194	August 2, 2018	9.7.14188	4.1 and up
225	January 31, 2018	1.1.07.6	7.0 and up
293	August 2, 2018	9.7.14188	4.1 and up
346	June 7, 2018	9.8.000000010492	4.0 and up
…	…	…	…
10378	September 20, 2017	1.16.33	4.1 and up
10408	August 8, 2018	1.1.2	4.1 and up
10429	May 25, 2018	1.5.3.20	4.1 and up
10513	March 1, 2017	1.32	2.3 and up
10549	May 23, 2018	1.4.4	4.1 and up

[176 rows x 13 columns]

[90]:
```
df[df.Installs <= 10000000].shape
```

[90]: (7307, 13)

[91]:
```
df = df[df.Installs <= 10000000]
```

[92]:
```
df.describe()
```

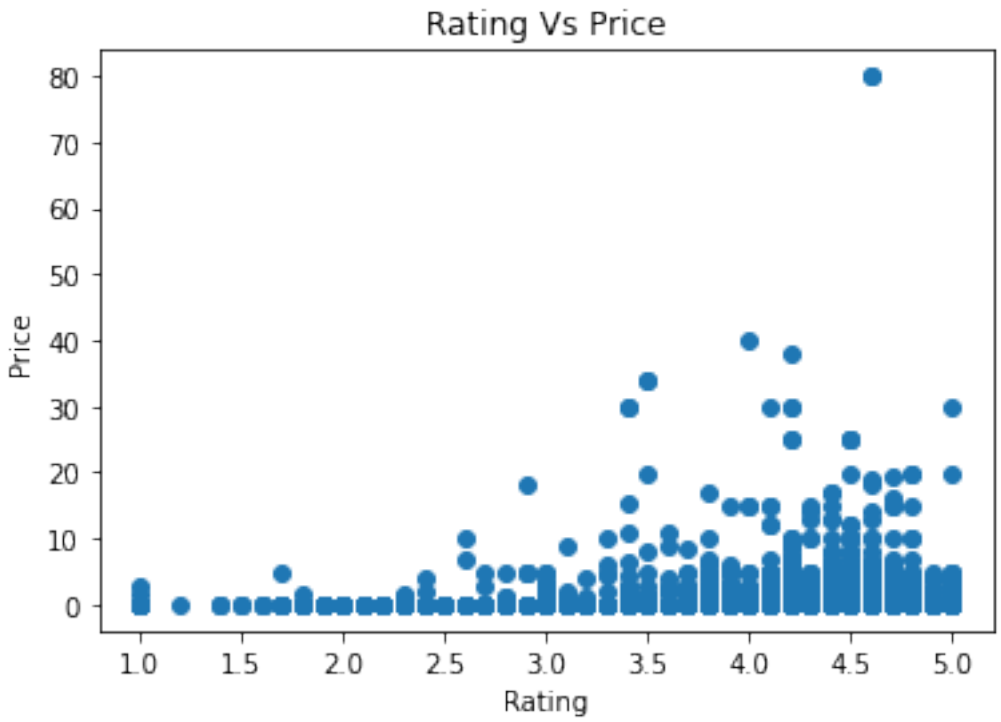
[92]:

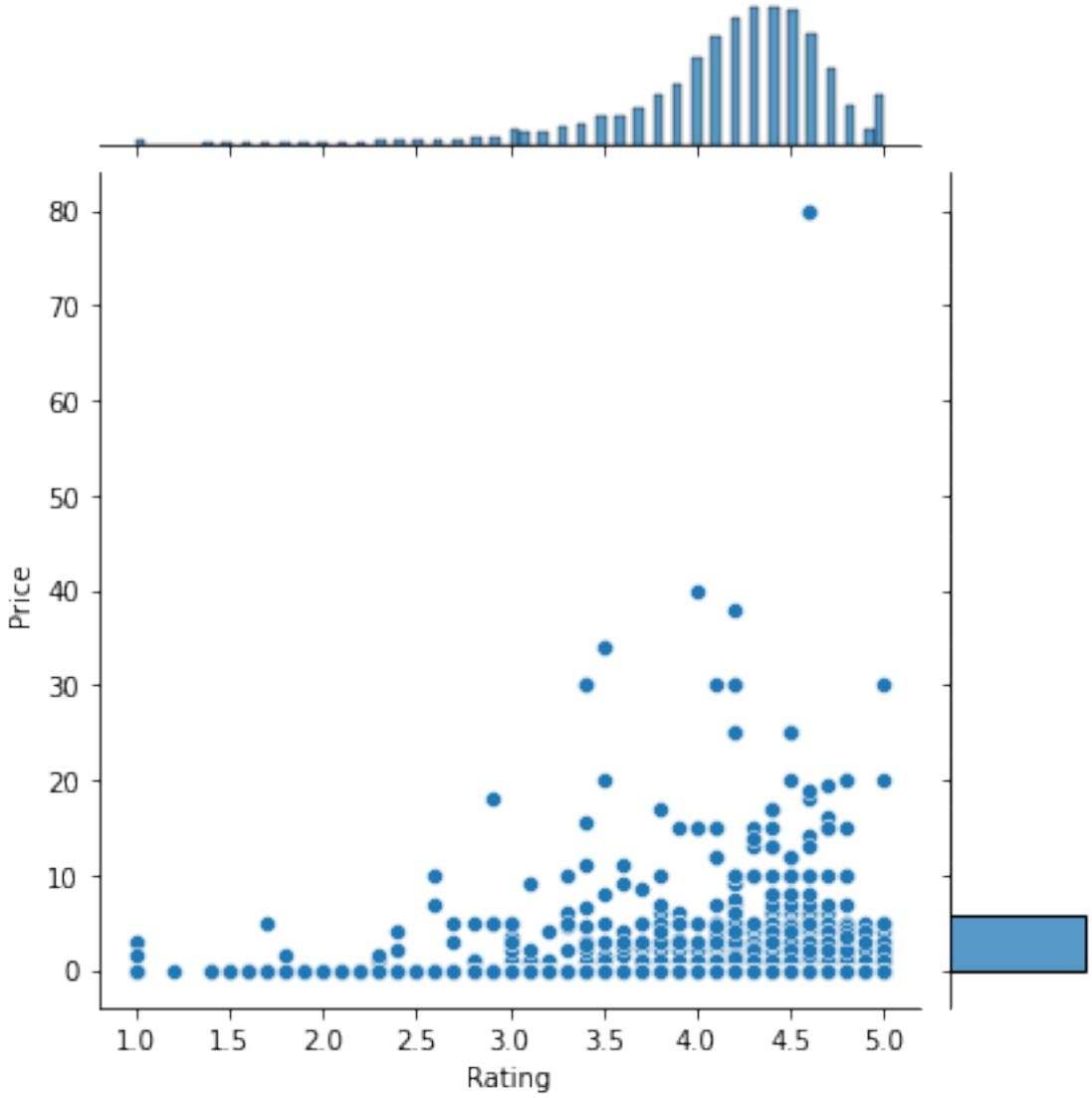
	Rating	Reviews	Size	Installs	Price
count	7307.000000	7.307000e+03	7307.000000	7.307000e+03	7307.000000
mean	4.162899	5.091109e+04	21687.801765	1.716009e+06	0.388738
std	0.555276	1.457407e+05	22460.971012	3.205978e+06	2.409159
min	1.000000	1.000000e+00	8.500000	5.000000e+00	0.000000
25%	4.000000	9.100000e+01	5000.000000	1.000000e+04	0.000000
50%	4.300000	1.749000e+03	14000.000000	1.000000e+05	0.000000
75%	4.500000	2.755850e+04	30000.000000	1.000000e+06	0.000000
max	5.000000	1.736105e+06	100000.000000	1.000000e+07	79.990000

[93]:
```
plt.scatter(df['Rating'], df['Price'])
plt.xlabel('Rating')
plt.ylabel('Price')
plt.title('Rating Vs Price')
sns.jointplot(df['Rating'], df['Price'])
print('form the plot below, rating does not increase with price')
```

```
/usr/local/lib/python3.7/site-packages/seaborn/_decorators.py:43: 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.
  FutureWarning
```

```
form the plot below, rating does not increase with price
```



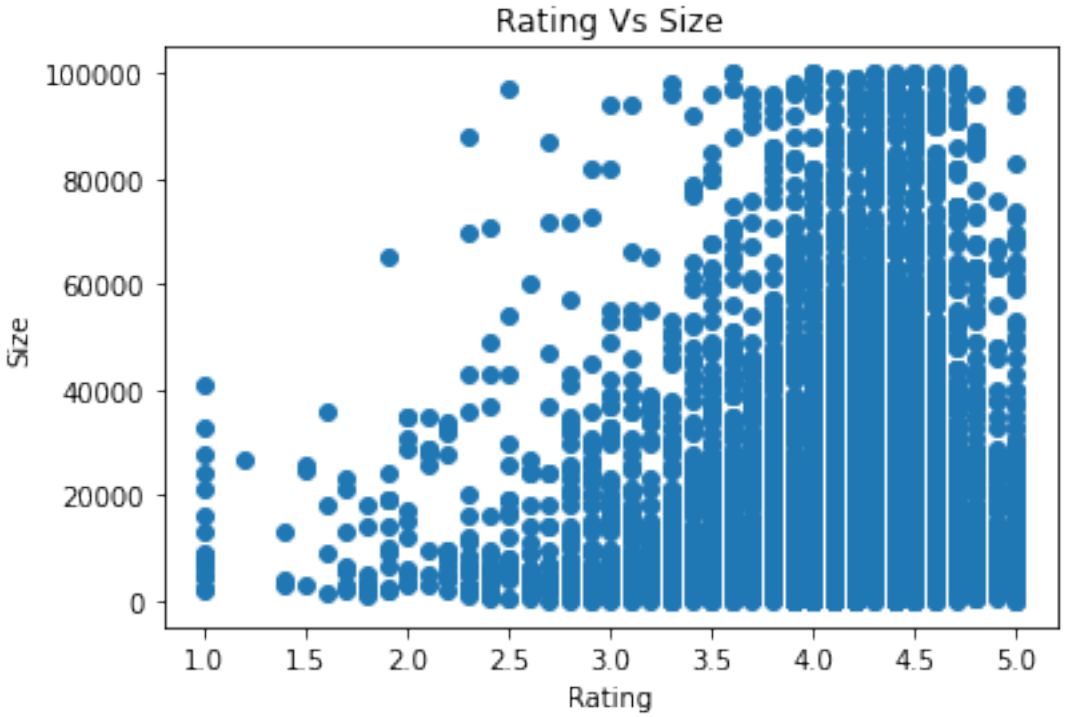


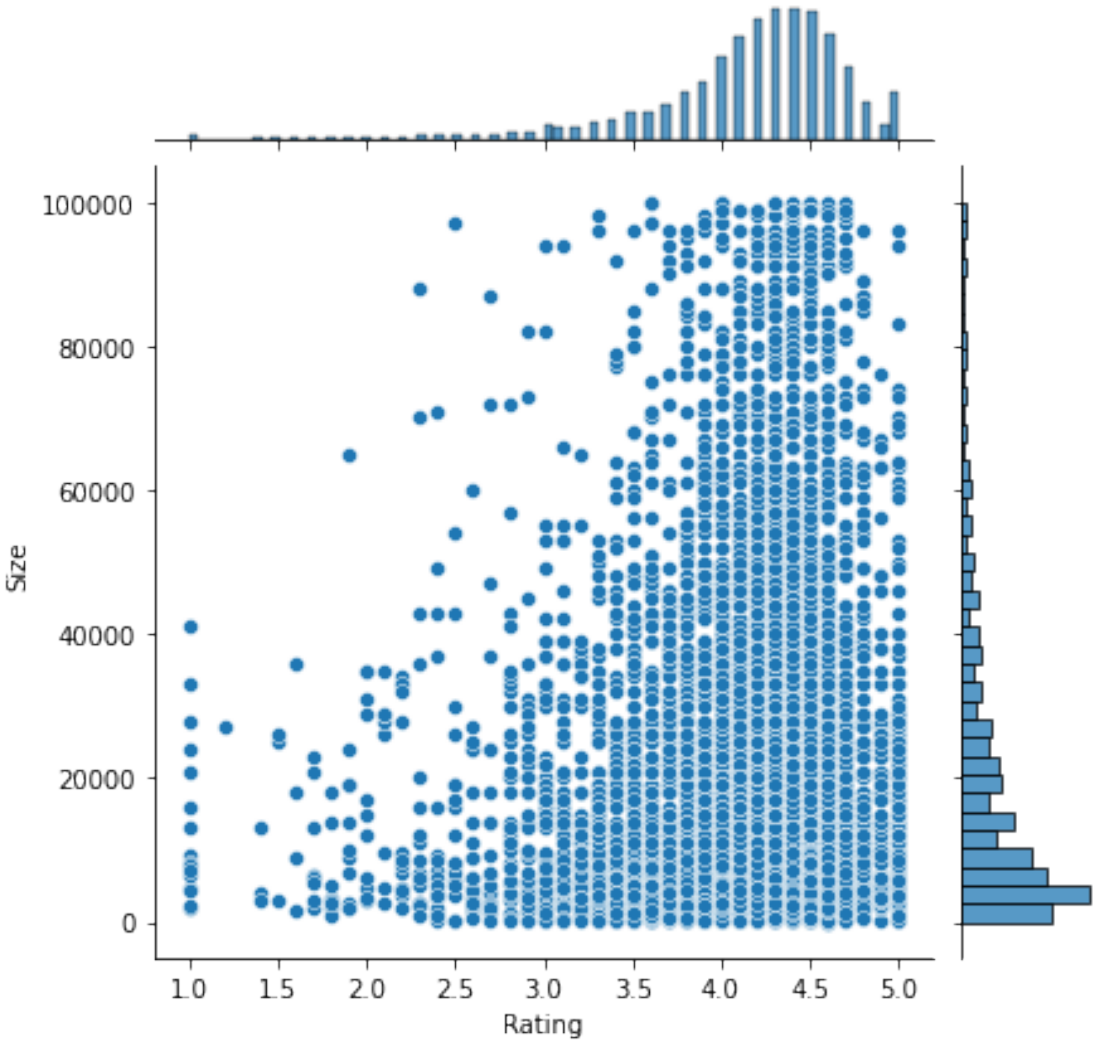
[94]:
```
plt.scatter(df['Rating'], df['Size'])
plt.xlabel('Rating')
plt.ylabel('Size')
plt.title('Rating Vs Size')
sns.jointplot(df['Rating'], df['Size'])
print('from the plot, lighter apps have less ratings than the heavier apps and␣
 ↪are likely to be rated lower')
```

```
/usr/local/lib/python3.7/site-packages/seaborn/_decorators.py:43: 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.
  FutureWarning
```

from the plot, lighter apps have less ratings than the heavier apps and are likely to be rated lower

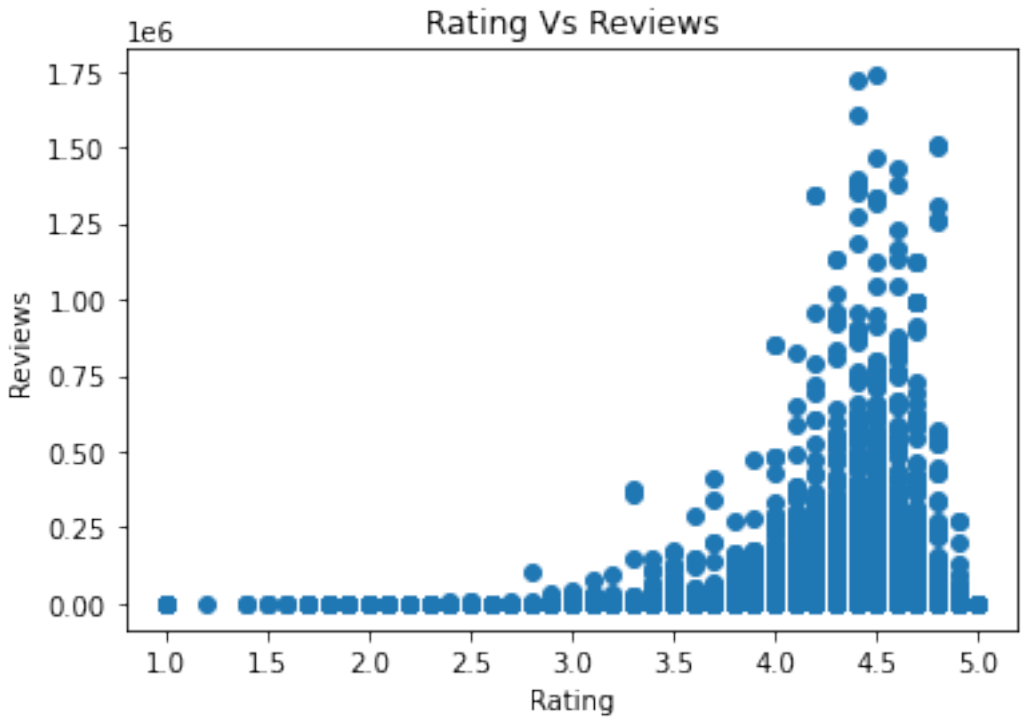


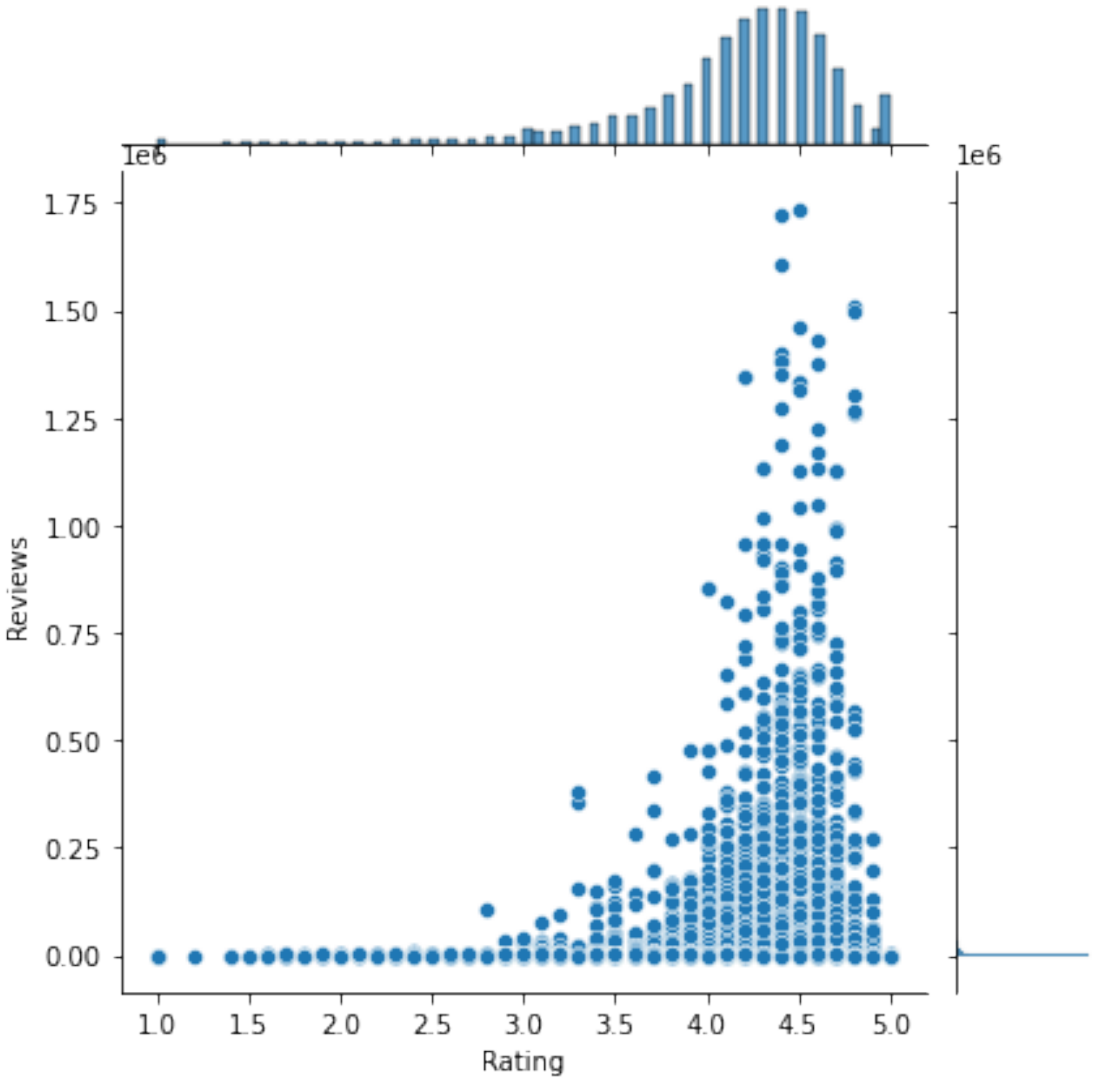


[95]:
```
plt.scatter(df['Rating'], df['Reviews'])
plt.xlabel('Rating')
plt.ylabel('Reviews')
plt.title('Rating Vs Reviews')
sns.jointplot(df['Rating'], df['Reviews'])
print('from the plot, apps with the most reviews are rated highly')
```

```
/usr/local/lib/python3.7/site-packages/seaborn/_decorators.py:43: 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.
  FutureWarning
```

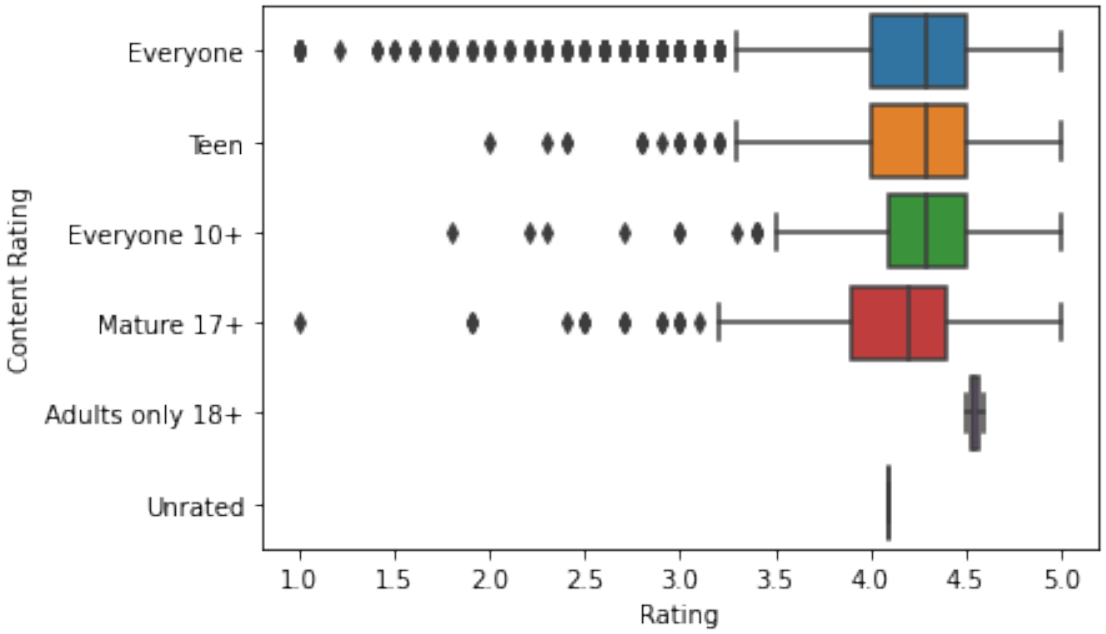
```
from the plot, apps with the most reviews are rated highly
```


Rating Vs Reviews
1e6
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
Reviews
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
Rating



[96]:
```
sns.boxplot(df['Rating'], df['Content Rating'])
print('Apps for Teens  Content Rating are generally rated higher than others,␣
 ↪while the apps for Everyone show a large variance in rating')
```

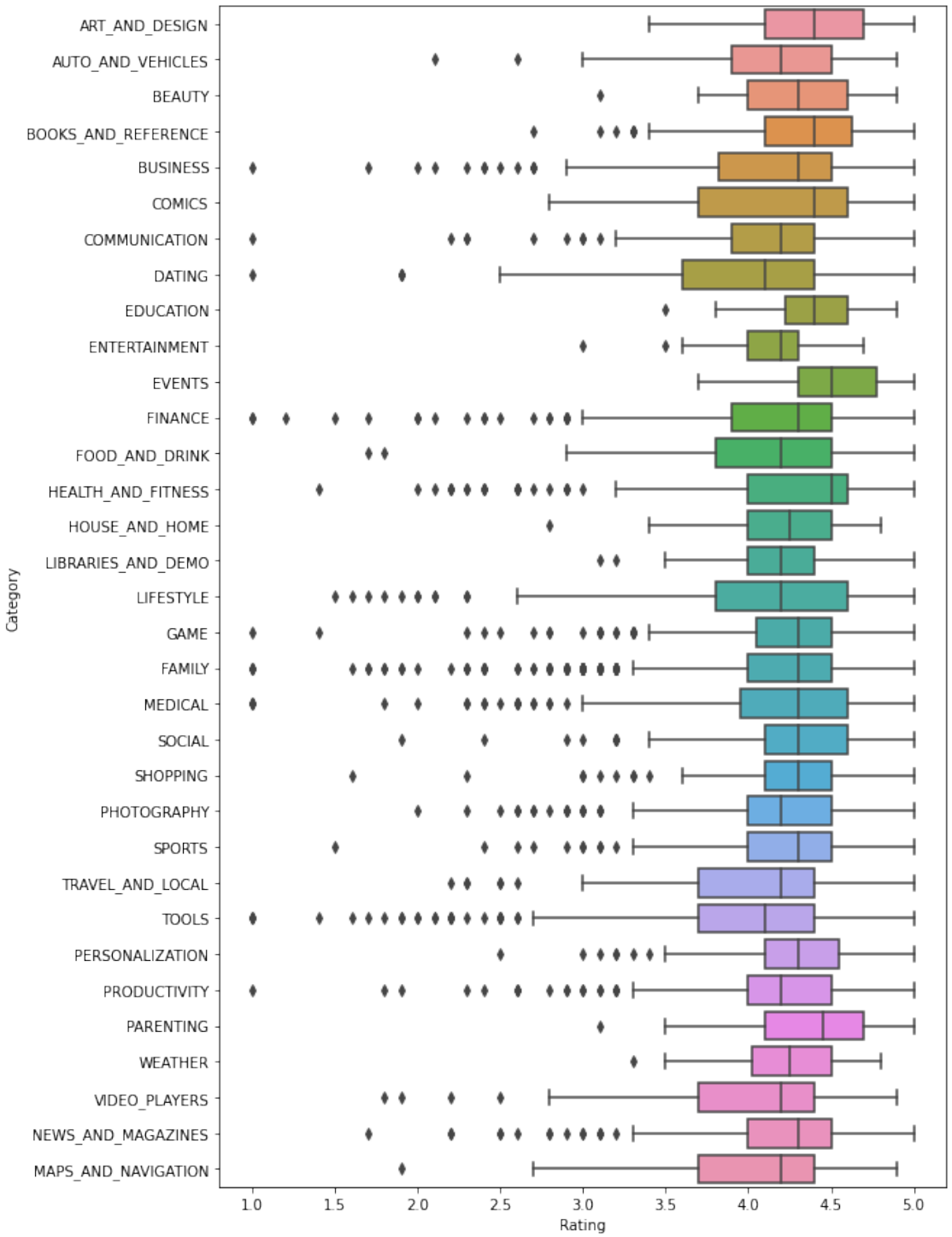
```
Apps for Teens  Content Rating are generally rated higher than others, while the
apps for Everyone show a large variance in rating

/usr/local/lib/python3.7/site-packages/seaborn/_decorators.py:43: 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.
  FutureWarning
```



[97]:
```
fig, axis = plt.subplots(figsize=(9, 15))
sns.boxplot(df['Rating'], df['Category'])
print('Apps for parenting and events show the highest ratings')
```

```
/usr/local/lib/python3.7/site-packages/seaborn/_decorators.py:43: 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.
  FutureWarning
```

```
Apps for parenting and events show the highest ratings
```



[98]:
```
inp1 = df.copy().reset_index()
```

[99]:
```
inp1['Reviews'] = np.log1p(inp1['Reviews'])
inp1['Installs'] = np.log1p(inp1['Installs'])
inp1['Size'] = np.log1p(inp1['Size'])
```

[100]:
```
inp1.drop(columns = ['index', 'App', 'Last Updated', 'Current Ver', 'Android␣
↪Ver'], axis = 1, inplace = True)
```

[101]:
```
inp1.columns
```

[101]:
```
Index(['Category', 'Rating', 'Reviews', 'Size', 'Installs', 'Type', 'Price',
       'Content Rating', 'Genres'],
      dtype='object')
```

[102]:
```
Index(['Category', 'Rating', 'Reviews', 'Size', 'Installs', 'Type', 'Price',
       'Content Rating', 'Genres'],
      dtype='object')
```

```
      ␣
↪---------------------------------------------------------------------------

       NameError                                 Traceback (most recent call␣
↪last)

       <ipython-input-102-83cbcbb9a951> in <module>
   ----> 1 Index(['Category', 'Rating', 'Reviews', 'Size', 'Installs', 'Type',␣
↪'Price',
         2        'Content Rating', 'Genres'],
         3       dtype='object')


       NameError: name 'Index' is not defined
```

[103]:
```
inp1.head()
```

[103]:

	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres
0	ART_AND_DESIGN	4.1	5.075174	9.852247	9.210440	Free	0.0	Everyone	Art & Design
1	ART_AND_DESIGN	3.9	6.875232	9.546884	13.122365	Free	0.0	Everyone	Art & Design;Pretend Play
2	ART_AND_DESIGN	4.7	11.379520	9.071193	15.424949	Free	0.0	Everyone	Art & Design
3	ART_AND_DESIGN	4.3	6.875232	7.937732	11.512935	Free	0.0		
4	ART_AND_DESIGN	4.4	5.123964	8.630700	10.819798	Free	0.0		

```
3       Everyone    Art & Design;Creativity
4       Everyone               Art & Design
```

[104]:
```
inp1.shape
```

[104]:
```
(7307, 9)
```

[105]:
```
categorical_cols = ['Category', 'Genres', 'Content Rating', 'Type']

inp2 = pd.get_dummies(inp1, columns=categorical_cols, drop_first=True)
```

[106]:
```
inp2.head()
```

[106]:

	Rating	Reviews	Size	Installs	Price	Category_AUTO_AND_VEHICLES	\
0	4.1	5.075174	9.852247	9.210440	0.0	0	
1	3.9	6.875232	9.546884	13.122365	0.0	0	
2	4.7	11.379520	9.071193	15.424949	0.0	0	
3	4.3	6.875232	7.937732	11.512935	0.0	0	
4	4.4	5.123964	8.630700	10.819798	0.0	0	

	Category_BEAUTY	Category_BOOKS_AND_REFERENCE	Category_BUSINESS	\
0	0	0	0	
1	0	0	0	
2	0	0	0	
3	0	0	0	
4	0	0	0	

	Category_COMICS	…	Genres_Video Players & Editors;Creativity	\
0	0	…	0	
1	0	…	0	
2	0	…	0	
3	0	…	0	
4	0	…	0	

	Genres_Video Players & Editors;Music & Video	Genres_Weather	Genres_Word	\
0	0	0	0	
1	0	0	0	
2	0	0	0	
3	0	0	0	
4	0	0	0	

	Content Rating_Everyone	Content Rating_Everyone 10+	\
0	1	0	
1	1	0	
2	1	0	
3	1	0	
4	1	0	

```
   Content Rating_Mature 17+  Content Rating_Teen  Content Rating_Unrated  \
0                          0                    0                       0
1                          0                    0                       0
2                          0                    0                       0
3                          0                    0                       0
4                          0                    0                       0

   Type_Paid
0          0
1          0
2          0
3          0
4          0

[5 rows x 154 columns]
```

[ ]:
```
# 10. Separate the dataframes into X_train, y_train, X_test, and y_test.

y_test = df_test.Rating
X_test = df_test.drop(['Rating'], axis=1)
# 11.1 Model building
# Use linear regression as the technique

from sklearn.linear_model import LinearRegression
lr = LinearRegression()
lr.fit(X_train, y_train)
LinearRegression()

# 11.2
# Report the R2 on the train set
from sklearn.metrics import r2_score
y_train_pred= lr.predict(X_train)
r2_score(y_train, y_train_pred)
0.06861486297278863

# 12 Make predictions on test set and report R2.
y_test_pred= lr.predict(X_test)
r2_score(y_test, y_test_pred)
0.05096091664816793
```