```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import statistics as stc
```

Load the data file using pandas

```
df=pd.read_csv('googleplaystore.csv')
```

```
df.shape
```

```
(10841, 13)
```

```
type(df)
```

```
pandas.core.frame.DataFrame
```

```
df.dtypes
```

```
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
```

```
df.describe()
```

```
             Rating
count  9367.000000
mean      4.193338
std       0.537431
min       1.000000
25%       4.000000
50%       4.300000
75%       4.500000
max      19.000000
```

```
df.head()
```

```
                                                 App        Category
Rating  \
0     Photo Editor & Candy Camera & Grid & ScrapBook  ART_AND_DESIGN
4.1
```

```
1                                Coloring book moana  ART_AND_DESIGN
3.9
2  U Launcher Lite – FREE Live Cool Themes, Hide ...  ART_AND_DESIGN
4.7
3                              Sketch - Draw & Paint  ART_AND_DESIGN
4.5
4              Pixel Draw - Number Art Coloring Book  ART_AND_DESIGN
4.3

  Reviews  Size     Installs  Type Price Content Rating  \
0     159   19M      10,000+  Free     0       Everyone
1     967   14M     500,000+  Free     0       Everyone
2   87510  8.7M   5,000,000+  Free     0       Everyone
3  215644   25M  50,000,000+  Free     0           Teen
4     967  2.8M     100,000+  Free     0       Everyone

                      Genres      Last Updated         Current Ver  \
0               Art & Design   January 7, 2018               1.0.0
1  Art & Design;Pretend Play  January 15, 2018               2.0.0
2               Art & Design    August 1, 2018               1.2.4
3               Art & Design      June 8, 2018  Varies with device
4    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

df.tail()

                                                 App
Category  \
10836                               Sya9a Maroc - FR
FAMILY
10837               Fr. Mike Schmitz Audio Teachings
FAMILY
10838                         Parkinson Exercices FR
MEDICAL
10839                   The SCP Foundation DB fr nn5n
BOOKS_AND_REFERENCE
10840  iHoroscope - 2018 Daily Horoscope & Astrology
LIFESTYLE

       Rating Reviews                Size     Installs  Type Price  \
10836     4.5      38                53M       5,000+  Free     0
10837     5.0       4               3.6M         100+  Free     0
10838     NaN       3               9.5M       1,000+  Free     0
10839     4.5     114  Varies with device       1,000+  Free     0
```

```
10840     4.5  398307                19M  10,000,000+  Free     0

      Content Rating             Genres      Last Updated
Current Ver  \
10836       Everyone          Education     July 25, 2017
1.48
10837       Everyone          Education      July 6, 2018
1.0
10838       Everyone            Medical  January 20, 2017
1.0
10839     Mature 17+  Books & Reference  January 19, 2015  Varies with
device
10840       Everyone          Lifestyle     July 25, 2018  Varies with
device

              Android Ver
10836          4.1 and up
10837          4.1 and up
10838          2.2 and up
10839  Varies with device
10840  Varies with device
```

```
df.columns
```

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

Check for null values

```
df.isna().sum()
```

```
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
```

Get the number of null values for each column

```
df= df.dropna()
df
```

```
                                                 App
Category  \
0         Photo Editor & Candy Camera & Grid & ScrapBook
ART_AND_DESIGN
1                                    Coloring book moana
ART_AND_DESIGN
2      U Launcher Lite – FREE Live Cool Themes, Hide ...
ART_AND_DESIGN
3                                  Sketch - Draw & Paint
ART_AND_DESIGN
4                  Pixel Draw - Number Art Coloring Book
ART_AND_DESIGN
...                                                ...
...
10834                                   FR Calculator
FAMILY
10836                                 Sya9a Maroc - FR
FAMILY
10837                   Fr. Mike Schmitz Audio Teachings
FAMILY
10839                     The SCP Foundation DB fr nn5n
BOOKS_AND_REFERENCE
10840      iHoroscope - 2018 Daily Horoscope & Astrology
LIFESTYLE

       Rating Reviews                Size     Installs  Type Price  \
0         4.1     159                 19M      10,000+  Free     0
1         3.9     967                 14M     500,000+  Free     0
2         4.7   87510                8.7M   5,000,000+  Free     0
3         4.5  215644                 25M  50,000,000+  Free     0
4         4.3     967                2.8M     100,000+  Free     0
...       ...     ...                 ...          ...   ...   ...
10834     4.0       7                2.6M         500+  Free     0
10836     4.5      38                 53M       5,000+  Free     0
10837     5.0       4                3.6M         100+  Free     0
10839     4.5     114  Varies with device       1,000+  Free     0
10840     4.5  398307                 19M  10,000,000+  Free     0

      Content Rating                     Genres      Last Updated  \
0           Everyone               Art & Design   January 7, 2018
1           Everyone  Art & Design;Pretend Play  January 15, 2018
2           Everyone               Art & Design    August 1, 2018
3               Teen               Art & Design      June 8, 2018
4           Everyone    Art & Design;Creativity     June 20, 2018
...              ...                        ...               ...
10834       Everyone                  Education     June 18, 2017
```

```
10836       Everyone                 Education     July 25, 2017
10837       Everyone                 Education      July 6, 2018
10839     Mature 17+         Books & Reference  January 19, 2015
10840       Everyone                 Lifestyle     July 25, 2018

              Current Ver         Android Ver
0                   1.0.0       4.0.3 and up
1                   2.0.0       4.0.3 and up
2                   1.2.4       4.0.3 and up
3      Varies with device         4.2 and up
4                     1.1         4.4 and up
...                   ...                ...
10834               1.0.0         4.1 and up
10836                1.48         4.1 and up
10837                 1.0         4.1 and up
10839  Varies with device  Varies with device
10840  Varies with device  Varies with device

[9360 rows x 13 columns]
```

```
df.isna().sum()
```

```
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
```

```
df.dropna(inplace=True);print(df.shape)
```

```
(9360, 13)
```

Convert size

```
df['Size'].unique()
```

```
array(['19M', '14M', '8.7M', '25M', '2.8M', '5.6M', '29M', '33M',
'3.1M',
       '28M', '12M', '20M', '21M', '37M', '2.7M', '5.5M', '17M',
'39M',
       '31M', '4.2M', '7.0M', '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', '2.5M', '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', '3.7M', '22M', '7.4M', '6.4M', '3.2M', '8.2M', '9.9M',
'4.9M', '9.5M', '5.0M', '5.9M', '13M', '73M', '6.8M', '3.5M',
'4.0M', '2.3M', '7.2M', '2.1M', '42M', '7.3M', '9.1M', '55M',
'23k', '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', '118k', '44M', '695k',
'1.6M',
'6.2M', '18k', '53M', '1.4M', '3.0M', '5.8M', '3.8M', '9.6M',
'45M', '63M', '49M', '77M', '4.4M', '4.8M', '70M', '6.9M',
'9.3M',
'10.0M', '8.1M', '36M', '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', '34M', '93M', '65M',
'79M',
'100M', '58M', '50M', '68M', '64M', '67M', '60M', '94M',
'232k',
'99M', '624k', '95M', '8.5k', '41k', '292k', '11k', '80M',
'1.7M',
'74M', '62M', '69M', '75M', '98M', '85M', '82M', '96M', '87M',
'71M', '86M', '91M', '81M', '92M', '83M', '88M', '704k',
'862k',
'899k', '378k', '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', '902k',
'818k',
'81k', '939k', '169k', '45k', '475k', '965k', '90M', '545k',
'61k',
'283k', '655k', '714k', '93k', '872k', '121k', '322k', '1.0M',
'976k', '172k', '238k', '549k', '206k', '954k', '444k', '717k',
'210k', '609k', '308k', '705k', '306k', '904k', '473k', '175k',
'350k', '383k', '454k', '421k', '70k', '812k', '442k', '842k',
'417k', '412k', '459k', '478k', '335k', '782k', '721k', '430k',
'429k', '192k', '200k', '460k', '728k', '496k', '816k', '414k',
'506k', '887k', '613k', '243k', '569k', '778k', '683k', '592k',
'319k', '186k', '840k', '647k', '191k', '373k', '437k', '598k',
'716k', '585k', '982k', '222k', '219k', '55k', '948k', '323k',
'691k', '511k', '951k', '963k', '25k', '554k', '351k', '27k',
'82k', '208k', '913k', '514k', '551k', '29k', '103k', '898k',
'743k', '116k', '153k', '209k', '353k', '499k', '173k', '597k',
'809k', '122k', '411k', '400k', '801k', '787k', '237k', '50k',

```
       '643k', '986k', '97k', '516k', '837k', '780k', '961k', '269k',
       '20k', '498k', '600k', '749k', '642k', '881k', '72k', '656k',
       '601k', '221k', '228k', '108k', '940k', '176k', '33k', '663k',
       '34k', '942k', '259k', '164k', '458k', '245k', '629k', '28k',
       '288k', '775k', '785k', '636k', '916k', '994k', '309k', '485k',
       '914k', '903k', '608k', '500k', '54k', '562k', '847k', '957k',
       '688k', '811k', '270k', '48k', '329k', '523k', '921k', '874k',
       '981k', '784k', '280k', '24k', '518k', '754k', '892k', '154k',
       '860k', '364k', '387k', '626k', '161k', '879k', '39k', '970k',
       '170k', '141k', '160k', '144k', '143k', '190k', '376k', '193k',
       '246k', '73k', '658k', '992k', '253k', '420k', '404k',
'1,000+',
       '470k', '226k', '240k', '89k', '234k', '257k', '861k', '467k',
       '157k', '44k', '676k', '67k', '552k', '885k', '1020k', '582k',
       '619k'], dtype=object)
```

formatting Values

```
def change_size(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
```

```
df.columns
```

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

```
df['Size'] = df['Size'].map(change_size)
```

```
df.describe()
```

```
             Rating           Size
count  9367.000000    9145.000000
mean      4.193338   21516.529524
std       0.537431   22588.747934
min       1.000000       8.500000
25%       4.000000    4900.000000
50%       4.300000   13000.000000
```

```
75%       4.500000   30000.000000
max      19.000000  100000.000000
```

```
df.dtypes
```

```
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
```

```
df.isnull().sum()
```

```
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
```

Reviews ..... Convert it to numeric (int/float)

```
df.Reviews= df.Reviews.apply(np.log1p)
```

```
df.Reviews.unique()
```

```
array([ 5.07517382,  6.87523209, 11.37951978, ...,  6.4035742 ,
        7.08673793, 12.89498085])
```

```
df.Reviews = df.Reviews.astype('int32')
```

```
df.Reviews.dtype
```

```
dtype('int32')
```

Installs

```
df['Installs']
```

```
0           10000
1          500000
2         5000000
3        50000000
4          100000
           ...
10836        5000
10837         100
10838        1000
10839        1000
10840    10000000
Name: Installs, Length: 10840, dtype: int64
```

```
df['Installs'].unique()
```

```
array([     10000,     500000,    5000000,   50000000,     100000,
            50000,    1000000,   10000000,       5000,  100000000,
       1000000000,       1000,  500000000,         50,        100,
              500,         10,          1,          5,          0])
```

```
df= df[df.Installs!='Free']
```

```
inst='100,000+'
```

```
inst.replace(",","").replace("+","")
```

```
'100000'
```

```
def remove_plusandcommas(val):
    return int(val.replace(",","").replace("+",""))
```

```
df.Installs= df.Installs.map(remove_plusandcommas)
```

```
---------------------------------------------------------------------------
-----
NameError                                 Traceback (most recent call
last)
<ipython-input-14-9054ccdbcf10> in <module>
----> 1 df.Installs= df.Installs.map(remove_plusandcommas)

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

```
df.Installs.describe()
```

```
count    1.084000e+04
mean     1.546434e+07
std      8.502936e+07
min      0.000000e+00
25%      1.000000e+03
50%      1.000000e+05
```

```
75%      5.000000e+06
max      1.000000e+09
Name: Installs, dtype: float64
```

Checking price....Convert it to numeric (int/float)

```
df['Price']
```

```
0        0.0
1        0.0
2        0.0
3        0.0
4        0.0
        ...
10836    0.0
10837    0.0
10838    0.0
10839    0.0
10840    0.0
Name: Price, Length: 10840, dtype: float64
```

```
df['Price'].unique()
```

```
array([  0.  ,   4.99,   3.99,   6.99,   1.49,   2.99,   7.99,   5.99,
         3.49,   1.99,   9.99,   7.49,   0.99,   9.  ,   5.49,  10.  ,
        24.99,  11.99,  79.99,  16.99,  14.99,   1.  ,  29.99,  12.99,
         2.49,  10.99,   1.5 ,  19.99,  15.99,  33.99,  74.99,  39.99,
         3.95,   4.49,   1.7 ,   8.99,   2.  ,   3.88,  25.99, 399.99,
        17.99, 400.  ,   3.02,   1.76,   4.84,   4.77,   1.61,   2.5 ,
         1.59,   6.49,   1.29,   5.  ,  13.99, 299.99, 379.99,  37.99,
        18.99, 389.99,  19.9 ,   8.49,   1.75,  14.  ,   4.85,  46.99,
       109.99, 154.99,   3.08,   2.59,   4.8 ,   1.96,  19.4 ,   3.9 ,
         4.59,  15.46,   3.04,   4.29,   2.6 ,   3.28,   4.6 ,  28.99,
         2.95,   2.9 ,   1.97, 200.  ,  89.99,   2.56,  30.99,   3.61,
       394.99,   1.26,   1.2 ,   1.04])
```

```
df= df[df.Price!='Everyone']
```

```
def remove_chars(val):
    return float(val.replace("$",""))
```

```
df.Price = df.Price.map(remove_chars)
```

```
---------------------------------------------------------------------------
-----
NameError                                 Traceback (most recent call
last)
<ipython-input-15-6c141ea5b504> in <module>
----> 1 df.Price = df.Price.map(remove_chars)

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

```
df.Price.describe()
```

```
count    9221.000000
mean        0.339463
std         2.184155
min         0.000000
25%         0.000000
50%         0.000000
75%         0.000000
max        79.990000
Name: Price, dtype: float64
```

Sanity check

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

len(df.index)
```

```
9366
```

```
df.Rating.describe()
```

```
count    9366.000000
mean        4.191757
std         0.515219
min         1.000000
25%         4.000000
50%         4.300000
75%         4.500000
max         5.000000
Name: Rating, dtype: float64
```

```
## There is no outlayers .. Ratings are between 1 to 5
```

Checking valid Reviews

```
df[df.Reviews<df.Installs]
```

```
                                                 App
Category  \
0         Photo Editor & Candy Camera & Grid & ScrapBook
ART_AND_DESIGN
1                                   Coloring book moana
ART_AND_DESIGN
2      U Launcher Lite – FREE Live Cool Themes, Hide ...
ART_AND_DESIGN
3                                 Sketch - Draw & Paint
ART_AND_DESIGN
4                  Pixel Draw - Number Art Coloring Book
ART_AND_DESIGN
...                                                 ...
...
10834                                  FR Calculator
FAMILY
10836                                Sya9a Maroc - FR
FAMILY
```

```
10837                   Fr. Mike Schmitz Audio Teachings
FAMILY
10839                      The SCP Foundation DB fr nn5n
BOOKS_AND_REFERENCE
10840      iHoroscope - 2018 Daily Horoscope & Astrology
LIFESTYLE

       Rating  Reviews                Size  Installs  Type  Price  \
0         4.1        5                19M     10000  Free    0.0
1         3.9        6                14M    500000  Free    0.0
2         4.7       11               8.7M   5000000  Free    0.0
3         4.5       12                25M  50000000  Free    0.0
4         4.3        6               2.8M    100000  Free    0.0
...       ...      ...                ...       ...   ...    ...
10834     4.0        2               2.6M       500  Free    0.0
10836     4.5        3                53M      5000  Free    0.0
10837     5.0        1               3.6M       100  Free    0.0
10839     4.5        4  Varies with device      1000  Free    0.0
10840     4.5       12                19M  10000000  Free    0.0

      Content Rating                     Genres      Last Updated  \
0           Everyone                Art & Design   January 7, 2018
1           Everyone  Art & Design;Pretend Play  January 15, 2018
2           Everyone                Art & Design    August 1, 2018
3               Teen                Art & Design      June 8, 2018
4           Everyone    Art & Design;Creativity     June 20, 2018
...              ...                        ...               ...
10834       Everyone                   Education     June 18, 2017
10836       Everyone                   Education     July 25, 2017
10837       Everyone                   Education      July 6, 2018
10839     Mature 17+          Books & Reference  January 19, 2015
10840       Everyone                   Lifestyle     July 25, 2018

               Current Ver         Android Ver
0                    1.0.0        4.0.3 and up
1                    2.0.0        4.0.3 and up
2                    1.2.4        4.0.3 and up
3      Varies with device          4.2 and up
4                      1.1          4.4 and up
...                    ...                 ...
10834                1.0.0          4.1 and up
10836                 1.48          4.1 and up
10837                  1.0          4.1 and up
10839  Varies with device  Varies with device
10840  Varies with device  Varies with device

[9363 rows x 13 columns]

len(df.index)
```

```
9366
```

```
df[(df.Type =='Free') & (df.Price>0)]
```

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

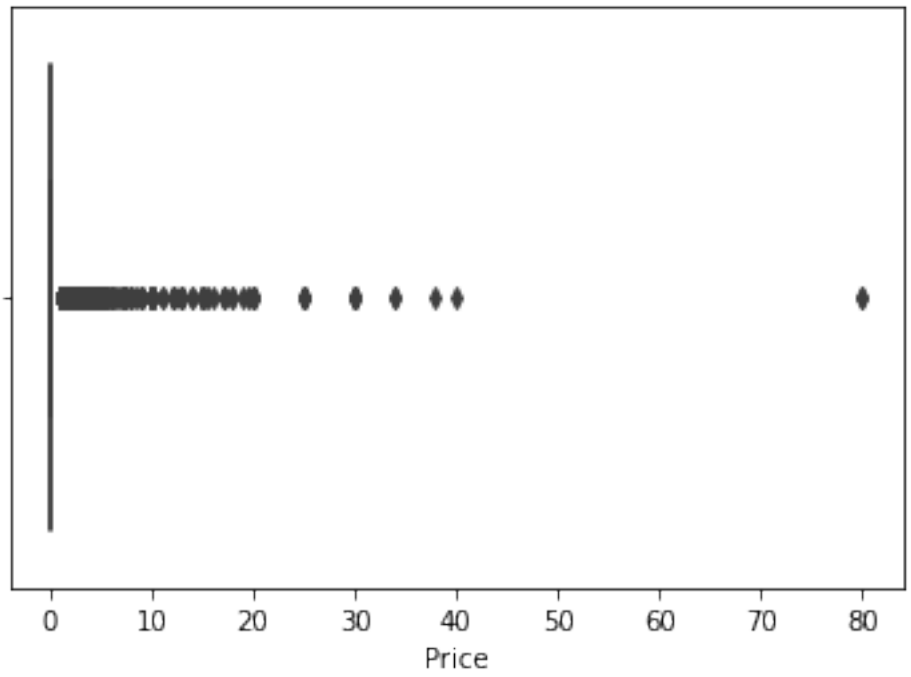
No Apps are find priced for the Type="Free"

Boxplot for Price

```
sns.boxplot(df.Price)
```

```
/usr/local/lib/python3.7/site-packages/seaborn/_decorators.py:43:
FutureWarning: Pass the following variable as a keyword arg: x. 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
```

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



Yes, There are outliers which are Type != "Free".

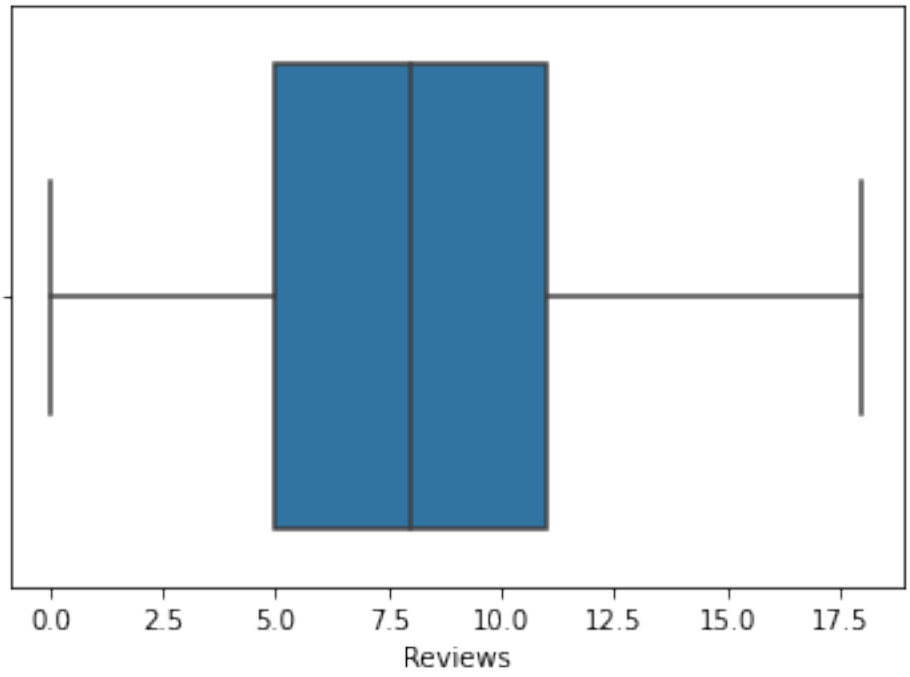
Boxplot for Reviews

```
sns.boxplot(df.Reviews)
```

```
/usr/local/lib/python3.7/site-packages/seaborn/_decorators.py:43:
FutureWarning: Pass the following variable as a keyword arg: x. 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
```

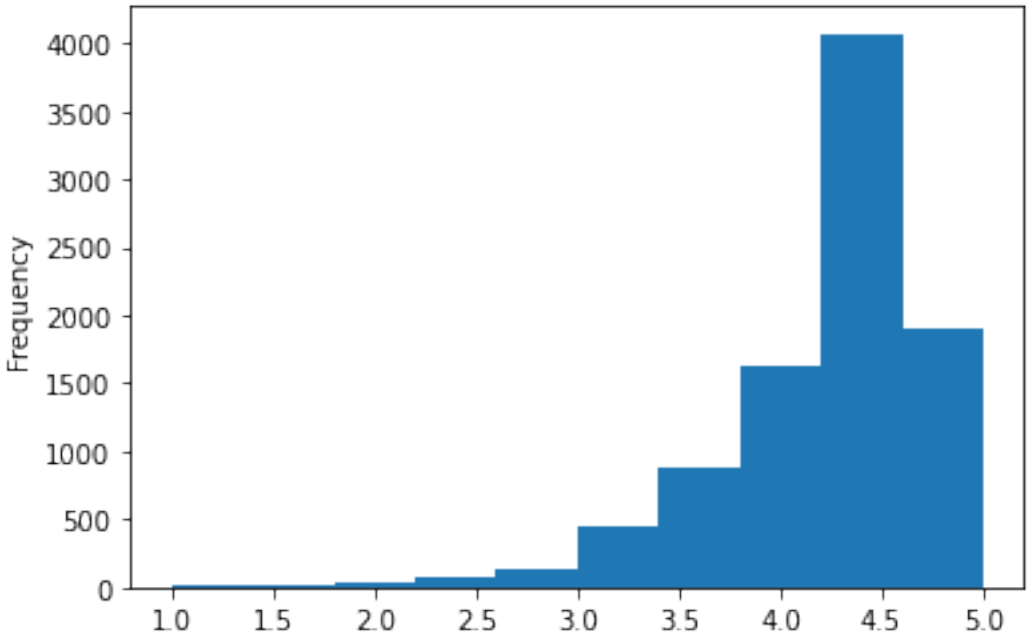
```
<AxesSubplot:xlabel='Reviews'>
```



There are higher number of reviews but they are not more than installs of respevtive apps.So, the values seem right

Histogram for Rating

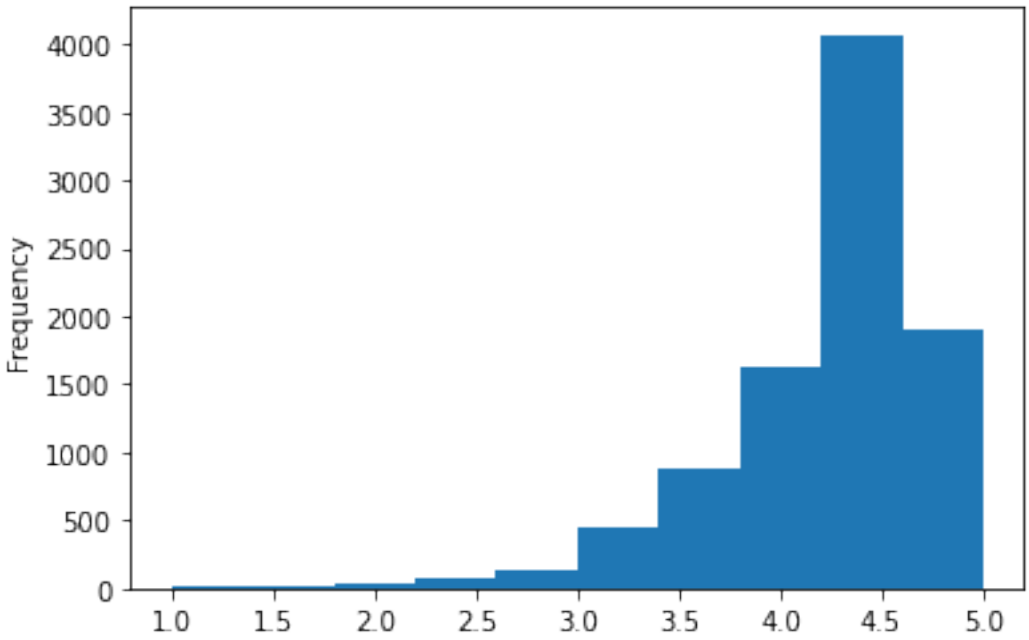
```
df.Rating.plot.hist()
plt.show()
```



Maximum number of rating disturbed towards higher rating

Histogram for Size

```
df.Rating.plot.hist()
plt.show()
```



There are outliyers..In the plot of 'Price' and 'Reviews' some outliers are observed. In 'Price',

Checking the high priced app and considering them as junk apps and dropping them.

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

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

```
len(df[df.Price>=200])
```

```
0
```

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

```
len(df.index)
```

```
9221
```

Dropping reviews records having more than 2000000 reviews

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

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

```
len(df[df.Reviews>2000000])
```

```
0
```

```
df.shape
```

```
(9221, 13)
```

Dropping high installed apps

```
df.Installs.quantile([0.1,0.25,0.5,0.7,0.9,0.95,0.99])
```

```
0.10         1000.0
0.25        10000.0
0.50       500000.0
0.70      1000000.0
0.90     10000000.0
0.95     50000000.0
0.99    100000000.0
Name: Installs, dtype: float64
```

```
### 99% of app downloads are within 100M, only 1% are more than 100M.
```

```
len(df[df.Installs>100000000])
```

```
0
```

```
df[df.Installs>100000000]
```

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

```
df=df[df.Installs<=100000000].copy()
```

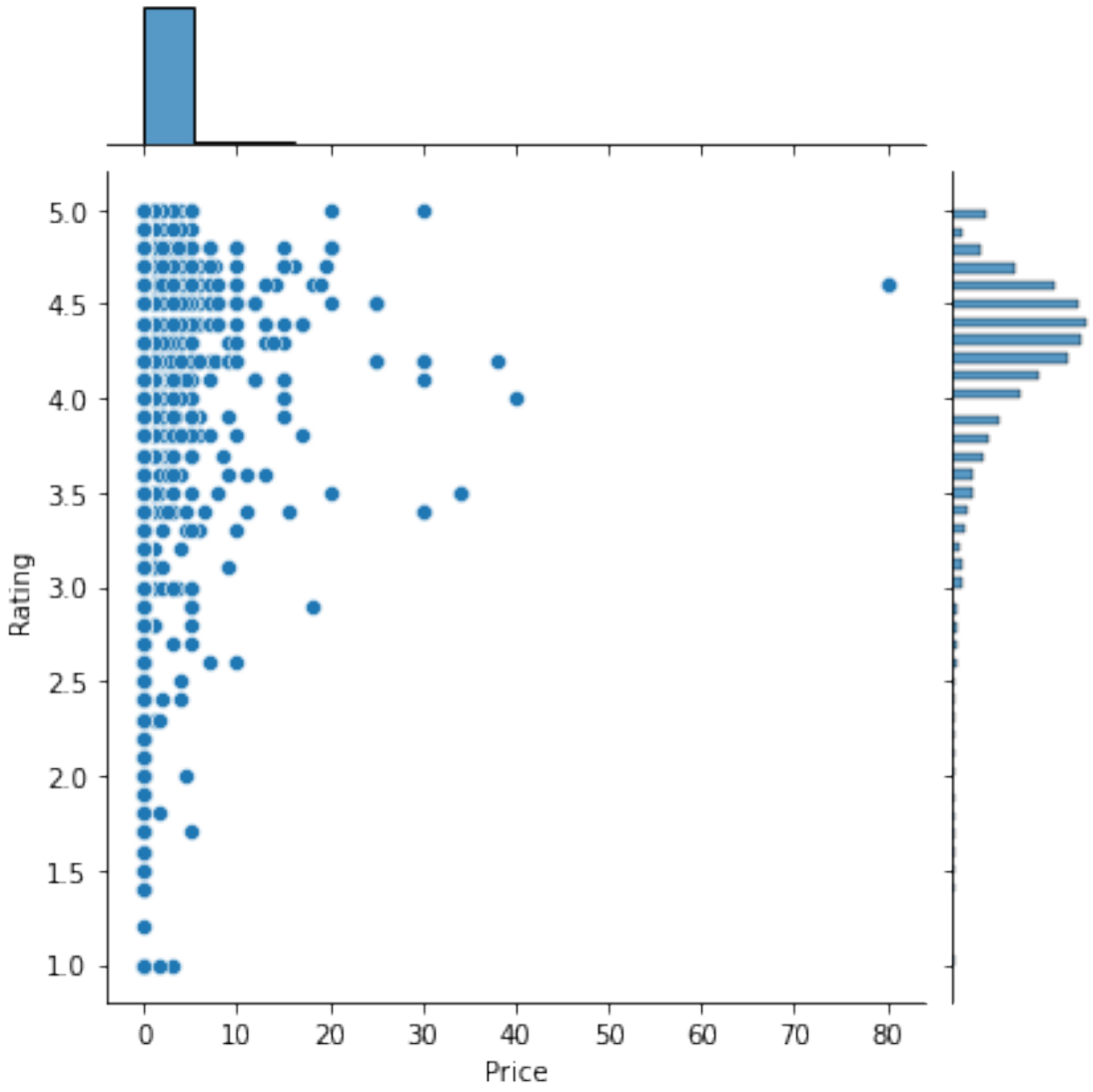
```
df.shape
```

```
(9221, 13)
```

joinplot for Rating vs. Price

```
sns.jointplot(x='Price', y='Rating' ,data=df)
```

```
<seaborn.axisgrid.JointGrid at 0x7fe311eb3890>
```

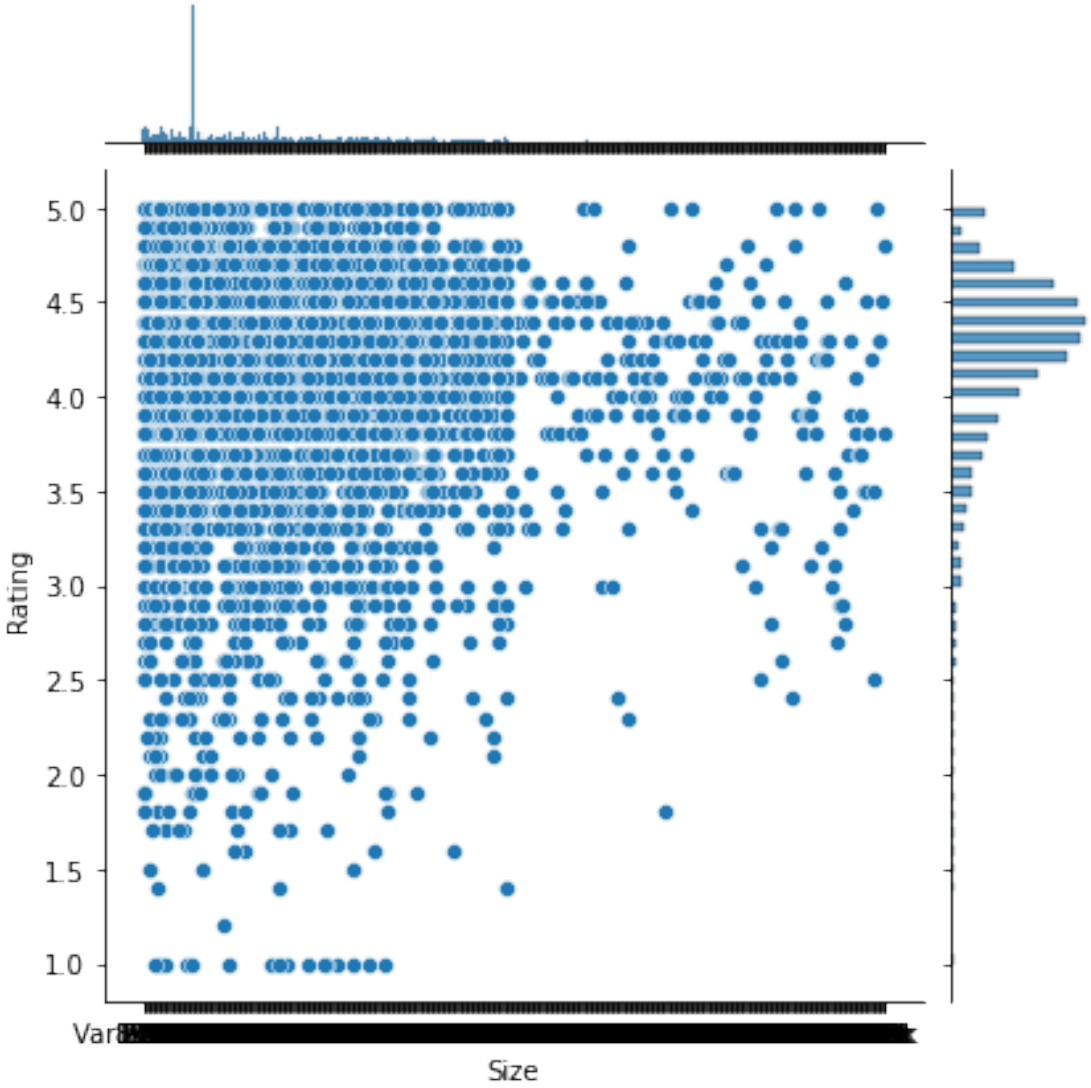



```
## Free apps also had high ratings, it shows that price doesn't
matters in the ratings.
```

joinplot for Rating vs. Size

```
sns.jointplot(x='Size',y='Rating',data=df)
```

```
<seaborn.axisgrid.JointGrid at 0x7fe311c6de50>
```

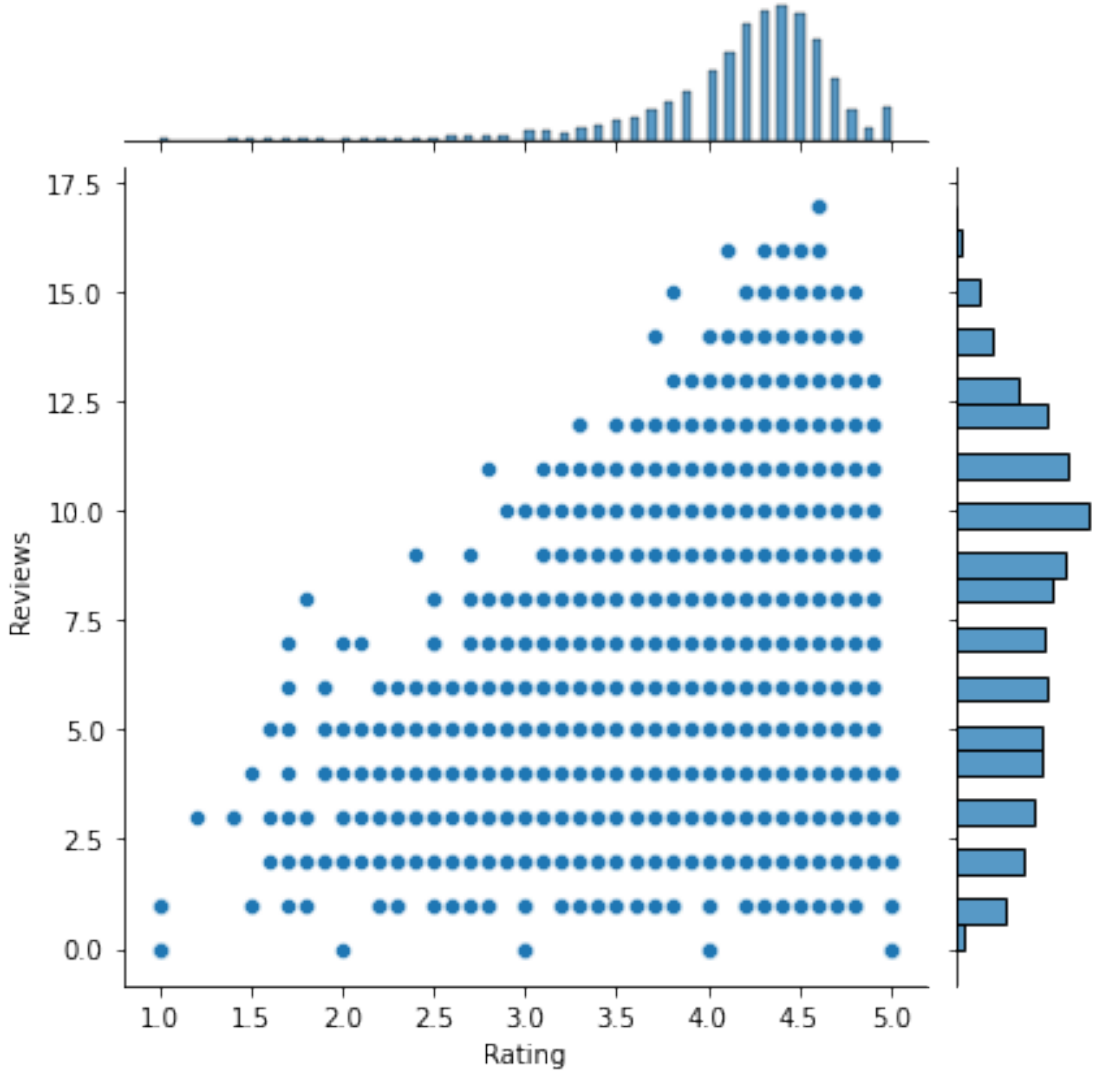


Very small difference, heavier app rated better when compared to low Size app. Because there are no low rate for the heavier Size app.

joinplot for Rating vs. Reviews

```
sns.jointplot(x='Rating',y='Reviews',data=df)
```

```
<seaborn.axisgrid.JointGrid at 0x7fe3094bc190>
```

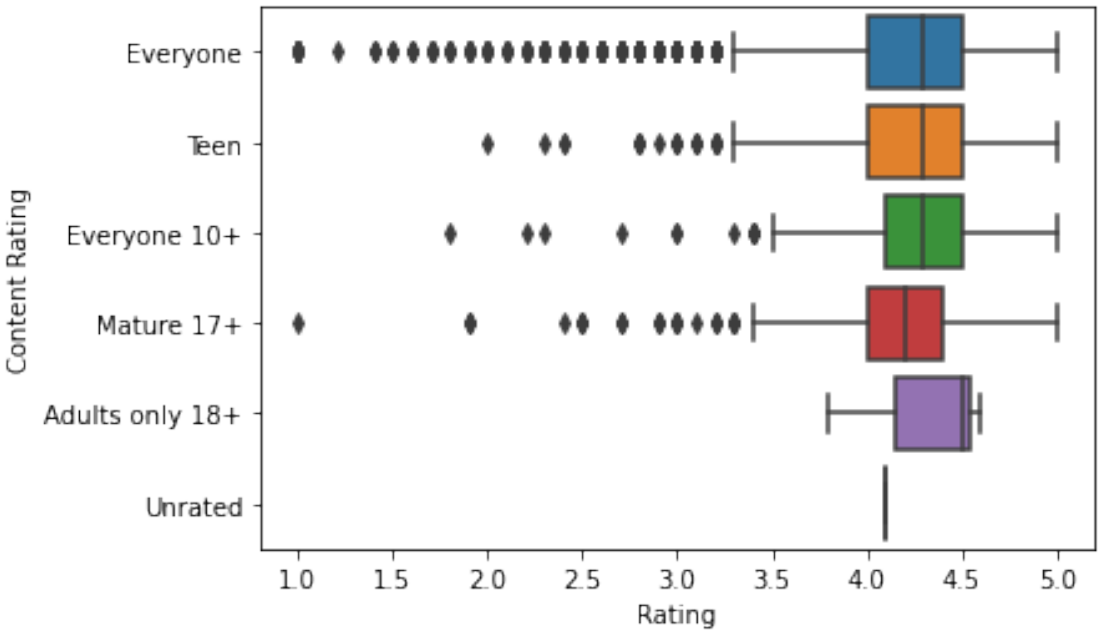


By the above plot, it shows that as higher the reviews, the average ratings also better when compare to low reviews app.

boxplot for Rating vs. Content Rating

```
sns.boxplot(x='Rating',y='Content Rating',data=df)
```

```
<AxesSubplot:xlabel='Rating', ylabel='Content Rating'>
```

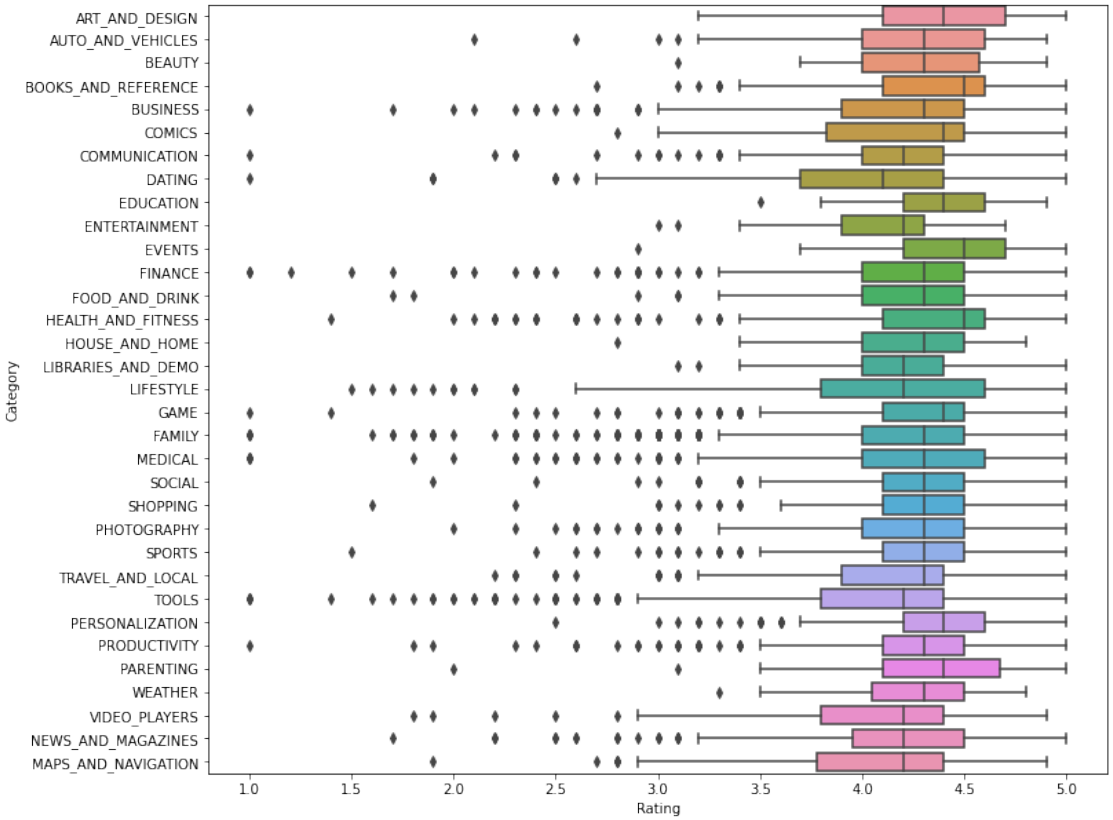


There is very small differences in ratings between Content Rating.

boxplot for Ratings vs. Category

```
a4_dimen=(11.7,10.27)
ax=plt.subplots(figsize=a4_dimen)
sns.boxplot(x='Rating',y='Category',data=df)
```

```
<AxesSubplot:xlabel='Rating', ylabel='Category'>
```



Events has best and consistent ratings, when compared to other genre apps.

Data preprocessing ....creating a new dataframe named inp1

```
inp1=df.copy()

inp1.Installs.describe()

count    9.221000e+03
mean     7.984735e+06
std      2.177167e+07
min      1.000000e+00
25%      1.000000e+04
50%      5.000000e+05
75%      5.000000e+06
max      1.000000e+08
Name: Installs, dtype: float64

inp1.Reviews.describe()

count    9221.000000
mean        7.779742
std         3.777713
min         0.000000
25%         5.000000
50%         8.000000
```

```
75%        11.000000
max        17.000000
Name: Reviews, dtype: float64
```

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

```
inp1.Installs = inp1.Installs.apply(np.log1p)
```

Drop columns

```
inp1.dtypes
```

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

```
inp1.drop(["App","Last Updated","Current Ver","Android Ver"],axis=1,
inplace=True)
```

```
inp1.shape
```

```
(9221, 9)
```

Creating dummy columns ... New data frame np2

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

```
inp2.columns
```

```
Index(['Rating', 'Reviews', 'Installs', 'Price',
'Category_AUTO_AND_VEHICLES',
       'Category_BEAUTY', 'Category_BOOKS_AND_REFERENCE',
'Category_BUSINESS',
       'Category_COMICS', 'Category_COMMUNICATION',
       ...
       '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=568)
```

Train test split and apply 70-30 split

```
from sklearn.model_selection import train_test_split

df_train, df_test = train_test_split(inp2, train_size = 0.7,
random_state = 100)

df_train.shape, df_test.shape
```

```
((6454, 568), (2767, 568))
```

Separate the dataframes into X_train, y_train, X_test, y_test

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

```
(9221, 567)
```

```
target = pd.DataFrame(inp2.Rating)
target.shape
```

```
(9221, 1)
```

Model building

```
from sklearn.linear_model import LinearRegression

x_train, x_test, y_train, y_test =
train_test_split(data,target,test_size=0.3, random_state=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  (6454, 567)
y_train shape is  (6454, 1)
x_test shape is  (2767, 567)
y_test shape is  (2767, 1)
```

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

```
LinearRegression()
```

Report the R2 on the train set

```
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  0.14620804314269054
```

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  -2.027498835820438e+17
```