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


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

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

```
data.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  
```

```
data.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
```

```python
data.columns
```

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

```python
data.isnull().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
```

```python
def printinfo():
    temp = pd.DataFrame(index = data.columns)
    temp ['data_type'] = data.dtypes
    temp ['null_count'] = data.isnull().sum()
    temp ['unique_count'] = data.nunique()
    return temp

printinfo()

               data_type  null_count  unique_count
App               object           0          9660
Category          object           0            34
Rating           float64        1474            40
Reviews           object           0          6002
Size              object           0           462
Installs          object           0            22
Type              object           1             3
Price             object           0            93
Content Rating    object           1             6
Genres            object           0           120
Last Updated      object           0          1378
Current Ver       object           8          2832
Android Ver       object           3            33
```

Column Rating Having Null Values

```
data[data.Rating.isnull()]

                                       App             Category
Rating  \
23                  Mcqueen Coloring pages       ART_AND_DESIGN
NaN
113              Wrinkles and rejuvenation               BEAUTY
NaN
123                 Manicure - nail design               BEAUTY
NaN
126           Skin Care and Natural Beauty               BEAUTY
NaN
129    Secrets of beauty, youth and health               BEAUTY
NaN
...                                    ...                  ...     ..
.
10824                            Cardio-FR              MEDICAL
NaN
10825                   Naruto & Boruto FR               SOCIAL
NaN
10831             payermonstationnement.fr  MAPS_AND_NAVIGATION
NaN
10835                             FR Forms             BUSINESS
NaN
10838               Parkinson Exercices FR              MEDICAL
NaN
```

```
      Reviews  Size  Installs  Type Price Content Rating  \
23         61  7.0M  100,000+  Free     0       Everyone
113       182  5.7M  100,000+  Free     0   Everyone 10+
123       119  3.7M   50,000+  Free     0       Everyone
126       654  7.4M  100,000+  Free     0           Teen
129        77  2.9M   10,000+  Free     0     Mature 17+
...       ...   ...       ...   ...   ...            ...
10824      67   82M   10,000+  Free     0       Everyone
10825       7  7.7M      100+  Free     0           Teen
10831      38  9.8M    5,000+  Free     0       Everyone
10835       0  9.6M       10+  Free     0       Everyone
10838       3  9.5M    1,000+  Free     0       Everyone

                                 Genres        Last Updated Current Ver
\
23     Art & Design;Action & Adventure       March 7, 2018       1.0.0

113                              Beauty  September 20, 2017         8.0

123                              Beauty       July 23, 2018         1.3

126                              Beauty       July 17, 2018        1.15

129                              Beauty      August 8, 2017         2.0

...                                 ...                 ...         ...

10824                           Medical       July 31, 2018       2.2.2

10825                            Social    February 2, 2018         1.0

10831                Maps & Navigation       June 13, 2018   2.0.148.0

10835                          Business  September 29, 2016       1.1.5

10838                           Medical    January 20, 2017         1.0


      Android Ver
23     4.1 and up
113    3.0 and up
123    4.1 and up
126    4.1 and up
129    2.3 and up
...           ...
10824  4.4 and up
10825  4.0 and up
10831  4.0 and up
10835  4.0 and up
10838  2.2 and up
```

```
[1474 rows x 13 columns]
```

Column Type Having Null Values

```
data[data.Type.isnull()]
```

```
                            App Category  Rating Reviews
Size  \
9148  Command & Conquer: Rivals   FAMILY     NaN       0  Varies with
device

     Installs Type Price Content Rating    Genres   Last Updated  \
9148        0  NaN     0   Everyone 10+  Strategy  June 28, 2018

             Current Ver         Android Ver
9148  Varies with device  Varies with device
```

```
data['Type'].fillna("Free",inplace = True)
```

```
data.isnull().sum()
```

```
App                  0
Category             0
Rating            1474
Reviews              0
Size                 0
Installs             0
Type                 0
Price                0
Content Rating       1
Genres               0
Last Updated         0
Current Ver          8
Android Ver          3
dtype: int64
```

Column Content Rating Having null values

```
data[data['Content Rating'].isnull()]
```

```
                                            App Category  Rating
Reviews  \
10472  Life Made WI-Fi Touchscreen Photo Frame      1.9    19.0
3.0M

         Size Installs Type     Price Content Rating
```

```
Genres  \
10472  1,000+     Free    0  Everyone            NaN  February 11,
2018

      Last Updated Current Ver Android Ver
10472       1.0.19  4.0 and up         NaN
```

```python
data.loc[10468:10477, :]
```

```
                                           App         Category
Rating  \
10468                         Tassa.fi Finland        LIFESTYLE
3.6
10469              TownWiFi | Wi-Fi Everywhere    COMMUNICATION
3.9
10470                               Jazz Wi-Fi    COMMUNICATION
3.4
10471                         Xposed Wi-Fi-Pwd  PERSONALIZATION
3.5
10472  Life Made WI-Fi Touchscreen Photo Frame              1.9
19.0
10473                  osmino Wi-Fi: free WiFi            TOOLS
4.2
10474                             Sat-Fi Voice    COMMUNICATION
3.4
10475                         Wi-Fi Visualizer            TOOLS
3.9
10476                    Lennox iComfort Wi-Fi        LIFESTYLE
3.0
10477              Sci-Fi Sounds and Ringtones  PERSONALIZATION
3.6

      Reviews    Size     Installs  Type     Price Content Rating  \
10468     346    7.5M      50,000+  Free         0       Everyone
10469    2372     58M     500,000+  Free         0       Everyone
10470      49    4.0M      10,000+  Free         0       Everyone
10471    1042    404k     100,000+  Free         0       Everyone
10472    3.0M  1,000+         Free     0  Everyone            NaN
10473  134203    4.1M  10,000,000+  Free         0       Everyone
10474      37     14M       1,000+  Free         0       Everyone
10475     132    2.6M      50,000+  Free         0       Everyone
10476     552    7.6M      50,000+  Free         0       Everyone
10477     128     11M      10,000+  Free         0       Everyone

                  Genres        Last Updated Current Ver   Android Ver

10468          Lifestyle        May 22, 2018         5.5    4.0 and up

10469      Communication      August 2, 2018       4.2.1    4.2 and up
```

```
10470      Communication   February 10, 2017         0.1    2.3 and up
10471    Personalization      August 5, 2014       3.0.0  4.0.3 and up
10472  February 11, 2018              1.0.19  4.0 and up           NaN
10473              Tools      August 7, 2018     6.06.14    4.4 and up
10474      Communication   November 21, 2014     2.2.1.5    2.2 and up
10475              Tools        May 17, 2017       0.0.9    2.3 and up
10476          Lifestyle      March 22, 2017      2.0.15  2.3.3 and up
10477    Personalization  September 27, 2017         4.0    4.0 and up
```

```
# Droping the rows from Content Rating Column
data.dropna(subset =['Content Rating'], inplace =True)

data.drop(['Current Ver','Last Updated', 'Android Ver'],
axis=1,inplace = True)

data.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
0               Art & Design
1  Art & Design;Pretend Play
2               Art & Design
3               Art & Design
4    Art & Design;Creativity
```

```python
# lets Replace the missing values in Rating Column using Mode value
for that entire column

modeValueRating = data['Rating'].mode()

modeValueRating[0]

4.4

data['Rating'].fillna(value =modeValueRating[0], inplace = True)

printinfo()
```

```
               data_type  null_count  unique_count
App               object           0          9659
Category          object           0            33
Rating           float64           0            39
Reviews           object           0          6001
Size              object           0           461
Installs          object           0            21
Type              object           0             2
Price             object           0            92
Content Rating    object           0             6
Genres            object           0           119
```

Now we are done with the data cleansing part and in a state to start the work for data preparation

Converting Datatypes of Column

Converting the Reviews column in integer

```python
data['Reviews'] = data.Reviews.astype(int)

printinfo()
```

```
               data_type  null_count  unique_count
App               object           0          9659
Category          object           0            33
Rating           float64           0            39
Reviews            int32           0          6001
Size              object           0           461
Installs          object           0            21
Type              object           0             2
Price             object           0            92
Content Rating    object           0             6
Genres            object           0           119
```

Column Size

```
#Converting the Size column in integer

data['Size'] = data.Size.apply(lambda x: x.strip('+')) # Removing +
sign

data['Size'] = data.Size.apply(lambda x : x.replace(',', '')) #
removing the ','

data['Size'] = data.Size.apply(lambda x: x.replace('M', 'e+6'))# For
converting the M to Mega

data['Size'] = data.Size.apply(lambda x : x.replace('k','e+3'))  #
Converting k to kilo

data['Size'] = data.Size.replace('Varies with device', np.NaN)

# Converting to Numeric Type

printinfo()
```

```
               data_type  null_count  unique_count
App               object           0          9659
Category          object           0            33
Rating           float64           0            39
Reviews            int32           0          6001
Size              object        1695           460
Installs          object           0            21
Type              object           0             2
Price             object           0            92
Content Rating    object           0             6
Genres            object           0           119
```

```
# Converting to numeric
data['Size'] = pd.to_numeric(data['Size'])

printinfo()
```

```
               data_type  null_count  unique_count
App               object           0          9659
Category          object           0            33
Rating           float64           0            39
Reviews            int32           0          6001
Size             float64        1695           459
Installs          object           0            21
Type              object           0             2
Price             object           0            92
Content Rating    object           0             6
Genres            object           0           119
```

```
data.dropna(subset =['Size'],inplace = True)
```

```
printinfo()

               data_type  null_count  unique_count
App               object           0          8434
Category          object           0            33
Rating           float64           0            39
Reviews            int32           0          4680
Size             float64           0           459
Installs          object           0            20
Type              object           0             2
Price             object           0            87
Content Rating    object           0             6
Genres            object           0           116
```

Columns Installs

```
data['Installs'] = data.Installs.apply(lambda x:x.strip('+'))  # 
removing + sign

data['Installs'] = data.Installs.apply(lambda x: x.replace(',','')) # 
Removing ',' Sign

# Converting to Numeric

data['Installs'] = pd.to_numeric(data['Installs'])

printinfo()

               data_type  null_count  unique_count
App               object           0          8434
Category          object           0            33
Rating           float64           0            39
Reviews            int32           0          4680
Size             float64           0           459
Installs           int64           0            20
Type              object           0             2
Price             object           0            87
Content Rating    object           0             6
Genres            object           0           116
```

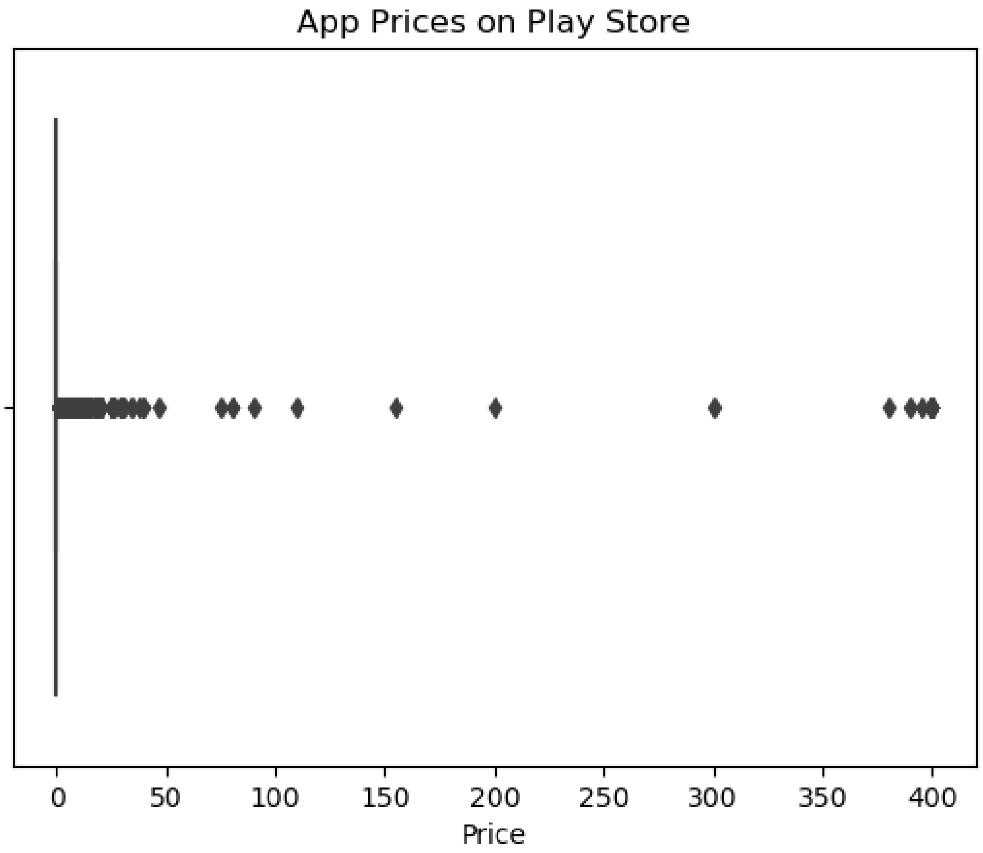
Column Price

```
data['Price'].value_counts()

0          8421
$0.99       145
$2.99       114
$1.99        66
$4.99        65
           ...
```

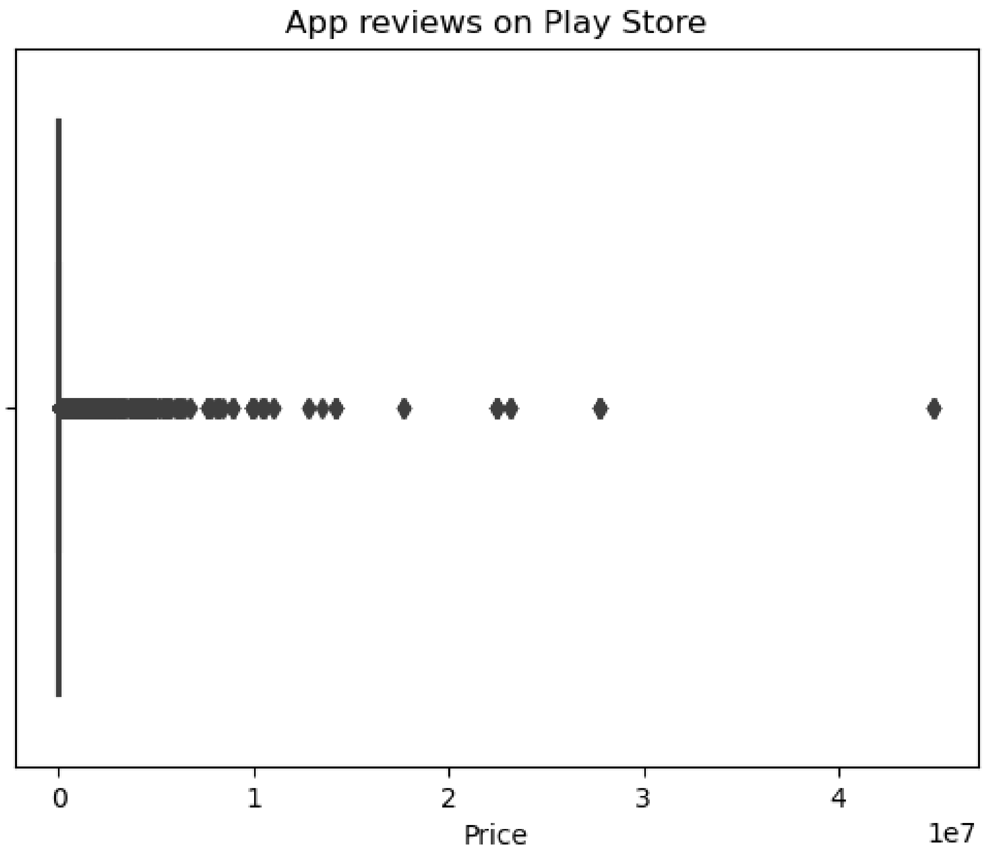
```
$389.99       1
$19.90        1
$1.75         1
$14.00        1
$1.04         1
Name: Price, Length: 87, dtype: int64

data['Price'] = data.Price.apply(lambda x: x.strip('$'))  # removing $
sign

# Converting to Numeric
data['Price'] = pd.to_numeric(data['Price'])

printinfo()

               data_type  null_count  unique_count
App               object           0          8434
Category          object           0            33
Rating           float64           0            39
Reviews            int32           0          4680
Size             float64           0           459
Installs           int64           0            20
Type              object           0             2
Price            float64           0            87
Content Rating    object           0             6
Genres            object           0           116

data.describe()

            Rating       Reviews          Size      Installs
Price
count  9145.000000  9.145000e+03  9.145000e+03  9.145000e+03
9145.000000
mean      4.208868  2.490487e+05  2.151653e+07  7.114842e+06
1.184366
std       0.507267  1.716211e+06  2.258875e+07  4.619357e+07
17.355754
min       1.000000  0.000000e+00  8.500000e+03  0.000000e+00
0.000000
25%       4.100000  2.200000e+01  4.900000e+06  1.000000e+03
0.000000
50%       4.400000  7.420000e+02  1.300000e+07  1.000000e+05
0.000000
75%       4.500000  2.503700e+04  3.000000e+07  1.000000e+06
0.000000
max       5.000000  4.489389e+07  1.000000e+08  1.000000e+09
400.000000

# Done with the Data Preparation & Cleaning
```

Performing Exploratory Data Analysis

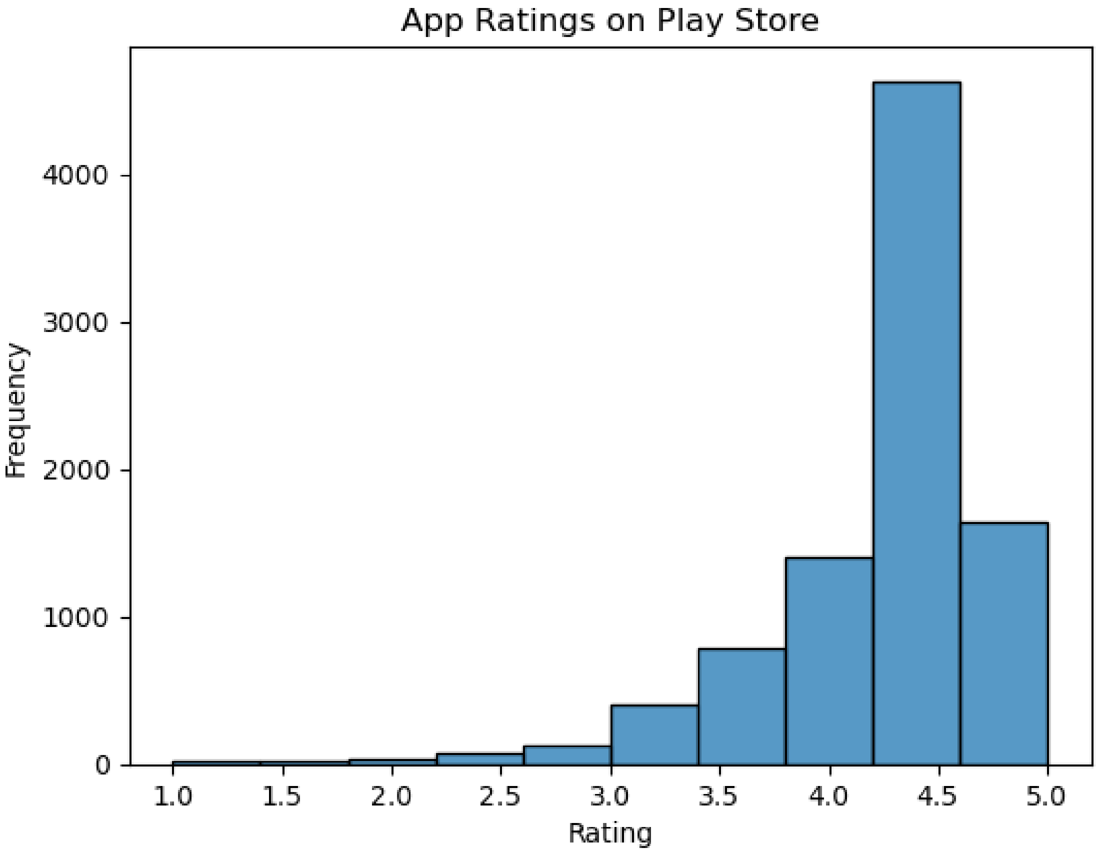
```
# Box plot for price Column

App_price = data.Price

# creating Boxplot
sns.boxplot(x=App_price)
plt.title('App Prices on Play Store')
plt.xlabel('Price')
plt.show()
```



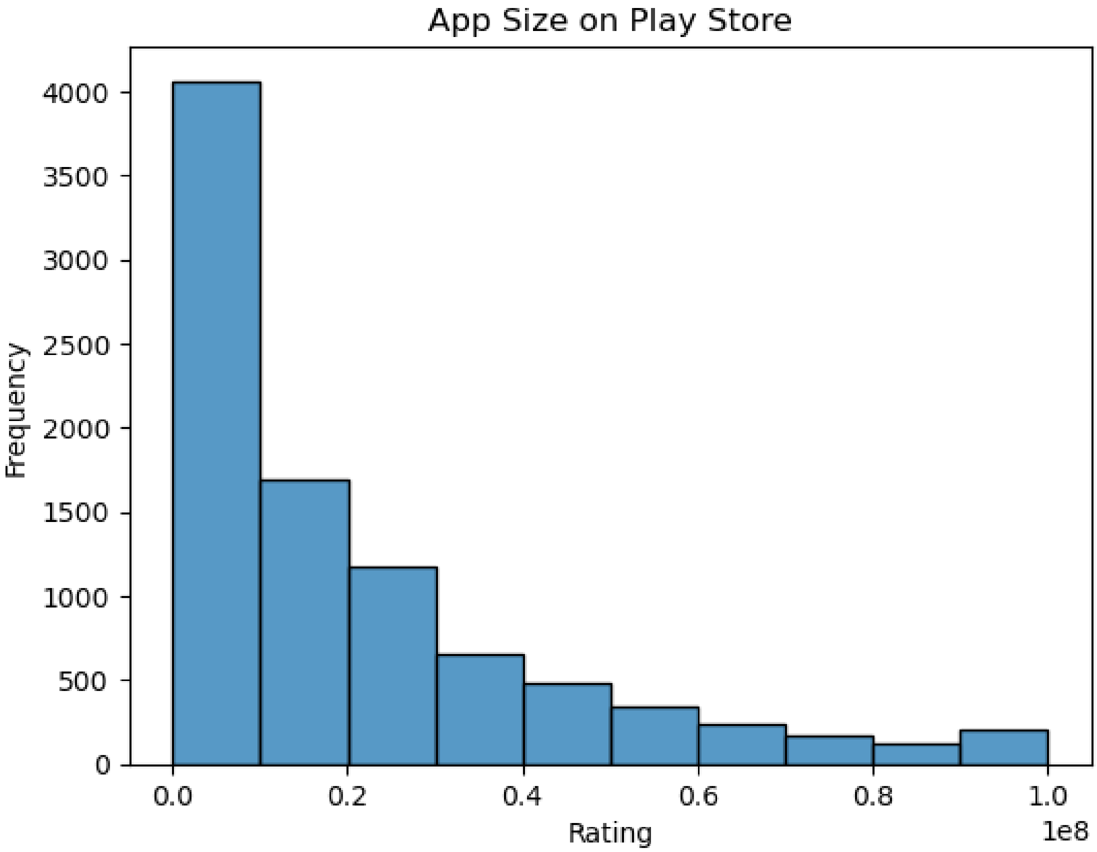
```
# Box plot for Reviews column

App_reviews = data.Reviews

# creating Boxplot
sns.boxplot(x=App_reviews)
plt.title('App reviews on Play Store')
plt.xlabel('Price')
plt.show()
```



```
# histogram for Rating

App_rating = data.Rating

sns.histplot(App_rating, bins=10)
plt.title('App Ratings on Play Store')
plt.xlabel('Rating')
plt.ylabel('Frequency')
plt.show()
```



```
# Histogram for size

App_size = data.Size

sns.histplot(App_size, bins=10)
plt.title('App Size on Play Store')
plt.xlabel('Rating')
plt.ylabel('Frequency')
plt.show()
```



```
filter_data = data[data['Price']>200]
# fitltering out the records which are greater than 200
data = data.drop(filter_data.index)
app_price = data
print(app_price)
```

```
                                                     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
...                                                  ...             .
..
10835                                           FR Forms
BUSINESS
10836                                   Sya9a Maroc - FR
```

```
FAMILY
10837                   Fr. Mike Schmitz Audio Teachings
FAMILY
10838                             Parkinson Exercices FR
MEDICAL
10840      iHoroscope - 2018 Daily Horoscope & Astrology
LIFESTYLE

       Rating  Reviews        Size  Installs  Type  Price Content
Rating  \
0         4.1      159  19000000.0     10000  Free    0.0
Everyone
1         3.9      967  14000000.0    500000  Free    0.0
Everyone
2         4.7    87510   8700000.0   5000000  Free    0.0
Everyone
3         4.5   215644  25000000.0  50000000  Free    0.0
Teen
4         4.3      967   2800000.0    100000  Free    0.0
Everyone
...       ...      ...         ...       ...   ...    ...            .
..
10835     4.4        0   9600000.0        10  Free    0.0
Everyone
10836     4.5       38  53000000.0      5000  Free    0.0
Everyone
10837     5.0        4   3600000.0       100  Free    0.0
Everyone
10838     4.4        3   9500000.0      1000  Free    0.0
Everyone
10840     4.5   398307  19000000.0  10000000  Free    0.0
Everyone

                          Genres
0                   Art & Design
1      Art & Design;Pretend Play
2                   Art & Design
3                   Art & Design
4        Art & Design;Creativity
...                          ...
10835                   Business
10836                  Education
10837                  Education
10838                    Medical
10840                  Lifestyle

[9128 rows x 10 columns]

# Filtering out records with more than 2 million reviews
data_filter = data[data['Reviews']<=2000000]
```

```
data = data.drop(data_filter.index)
print(data)

                                               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  16000000.0  100000000  Free    0.0        Everyone   
347     4785892  11000000.0  500000000  Free    0.0        Everyone   
366     3648120   3300000.0  100000000  Free    0.0            Teen   
378    17712922  40000000.0  500000000  Free    0.0            Teen   
383     4785988  11000000.0  500000000  Free    0.0        Everyone   
...         ...         ...        ...   ...    ...             ...   
9142    3344300  22000000.0   50000000  Free    0.0    Everyone 10+   
9166    2903386  58000000.0  100000000  Free    0.0      Mature 17+   
10186   7615646  71000000.0  100000000  Free    0.0        Everyone   
10190   2721923  25000000.0   10000000  Free    0.0            Teen   
10327   5534114  53000000.0  100000000  Free    0.0            Teen   

              Genres  
345    Communication  
347    Communication  
366    Communication  
378    Communication  
383    Communication  
...              ...  
9142          Racing  
9166          Action  
```

```
10186         Casual
10190     Simulation
10327         Action

[219 rows x 10 columns]
```

```
percentiles = [0.10,0.25,0.50,0.70,0.90,0.95,0.99]  # Calculating the different percentile
install_percentiles = data['Installs'].quantile(percentiles)
print(install_percentiles)
```

```
0.10    5.000000e+07
0.25    1.000000e+08
0.50    1.000000e+08
0.70    1.000000e+08
0.90    5.000000e+08
0.95    5.000000e+08
0.99    1.000000e+09
Name: Installs, dtype: float64
```

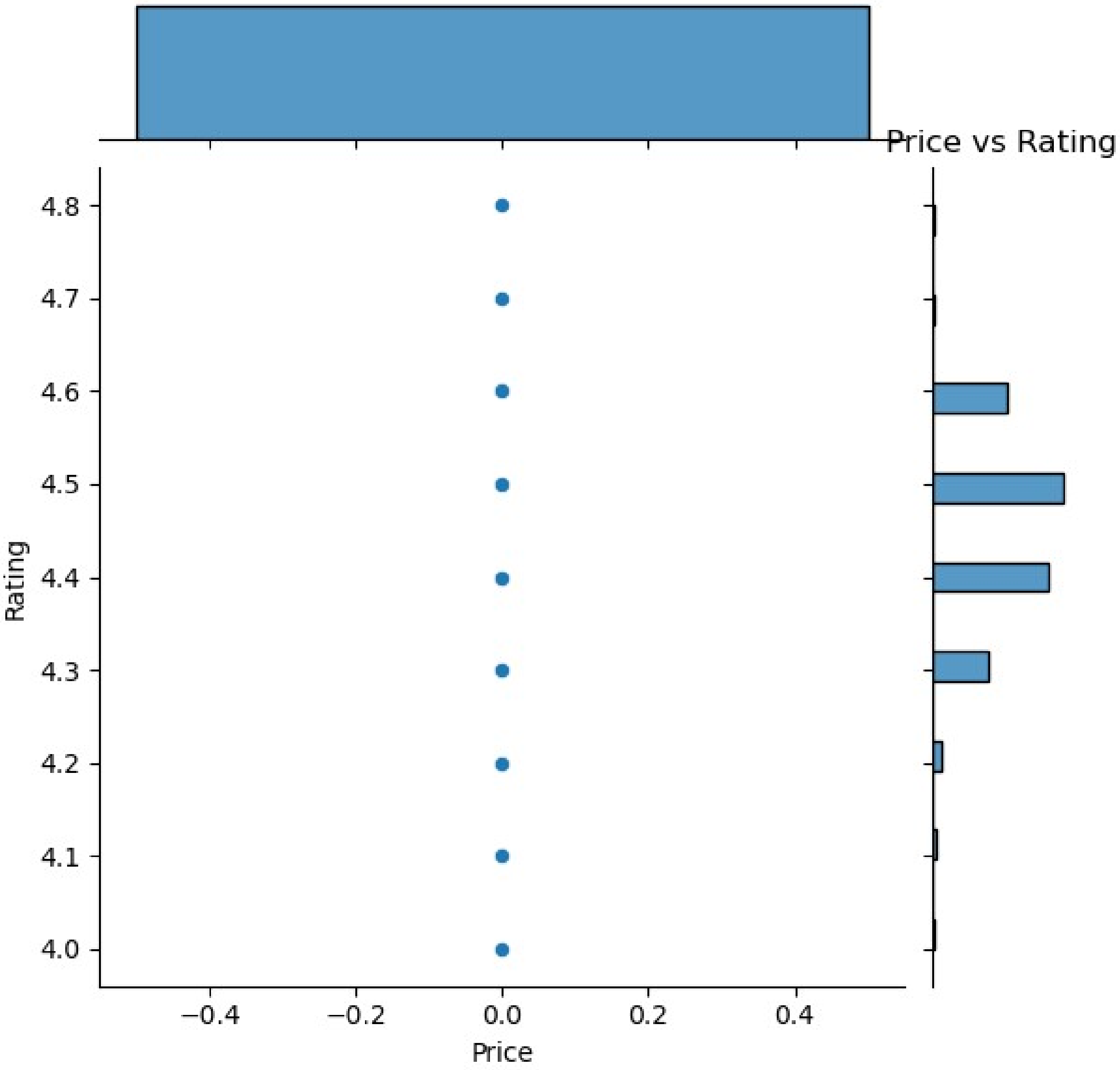
```
# Now decide the Cutoff threshold for outliers (eg .99)
cutoff_threshold = install_percentiles[0.99]

data_filter = data[data['Installs']<=cutoff_threshold]
print(data_filter)
```

```
                                               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
```

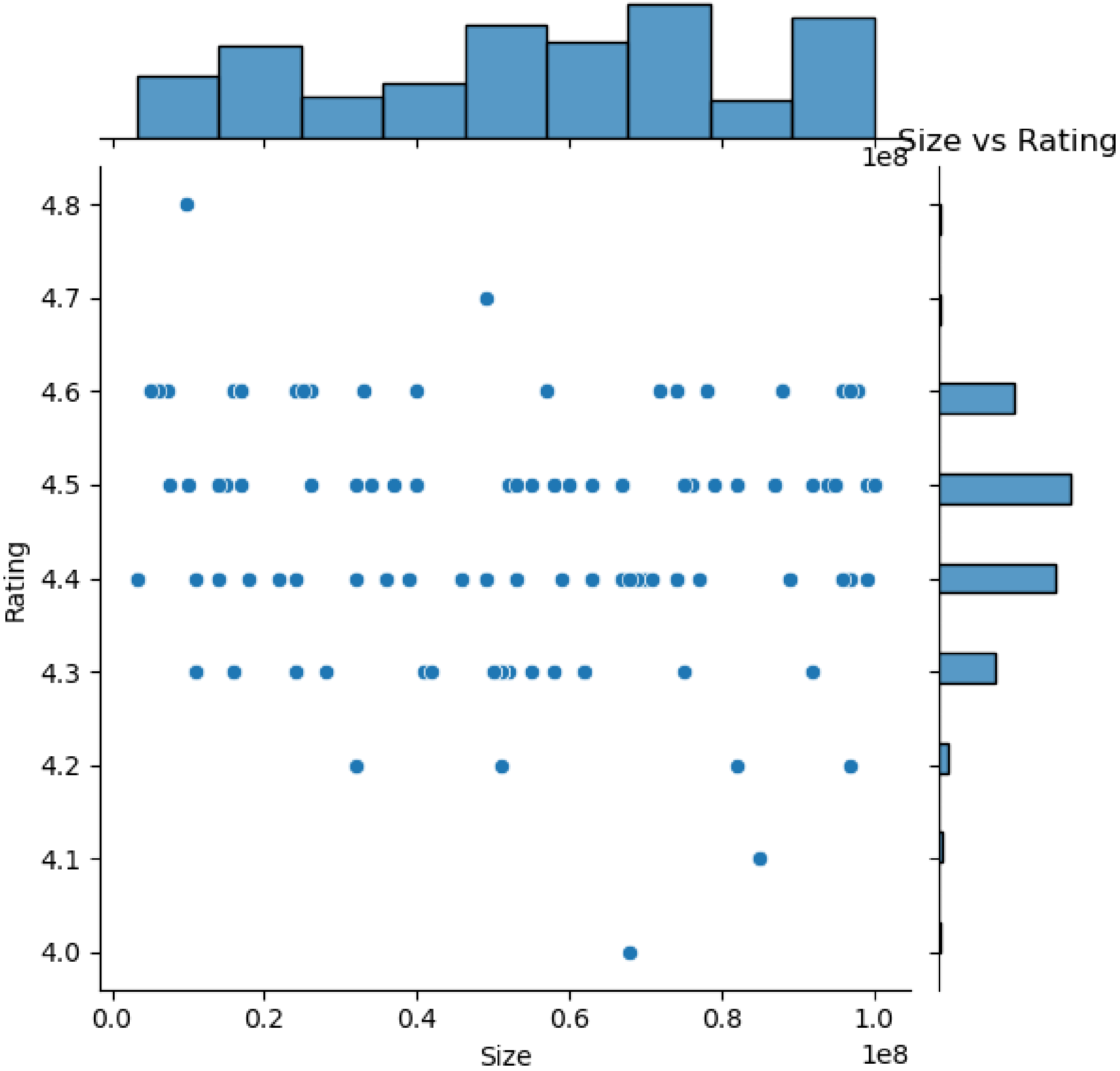
```
        Reviews        Size   Installs  Type  Price Content Rating  \
345     4187998  16000000.0  100000000  Free    0.0       Everyone   
347     4785892  11000000.0  500000000  Free    0.0       Everyone   
366     3648120   3300000.0  100000000  Free    0.0           Teen   
378    17712922  40000000.0  500000000  Free    0.0           Teen   
383     4785988  11000000.0  500000000  Free    0.0       Everyone   
...         ...         ...        ...   ...    ...            ...   
9142    3344300  22000000.0   50000000  Free    0.0   Everyone 10+   
9166    2903386  58000000.0  100000000  Free    0.0     Mature 17+   
10186   7615646  71000000.0  100000000  Free    0.0       Everyone   
10190   2721923  25000000.0   10000000  Free    0.0           Teen   
10327   5534114  53000000.0  100000000  Free    0.0           Teen   

              Genres  
345    Communication  
347    Communication  
366    Communication  
378    Communication  
383    Communication  
...              ...  
9142          Racing  
9166          Action  
10186         Casual  
10190     Simulation  
10327         Action  

[219 rows x 10 columns]
```

```
# Making  scatter plot/joinplot for Rating vs. Price
sns.jointplot(x = 'Price' , y = 'Rating', data = data)
# set plot title & label
plt.title('Price vs Rating')
plt.xlabel('Price')
plt.ylabel('Rating')
```

```
Text(463.154761904762, 0.5, 'Rating')
```

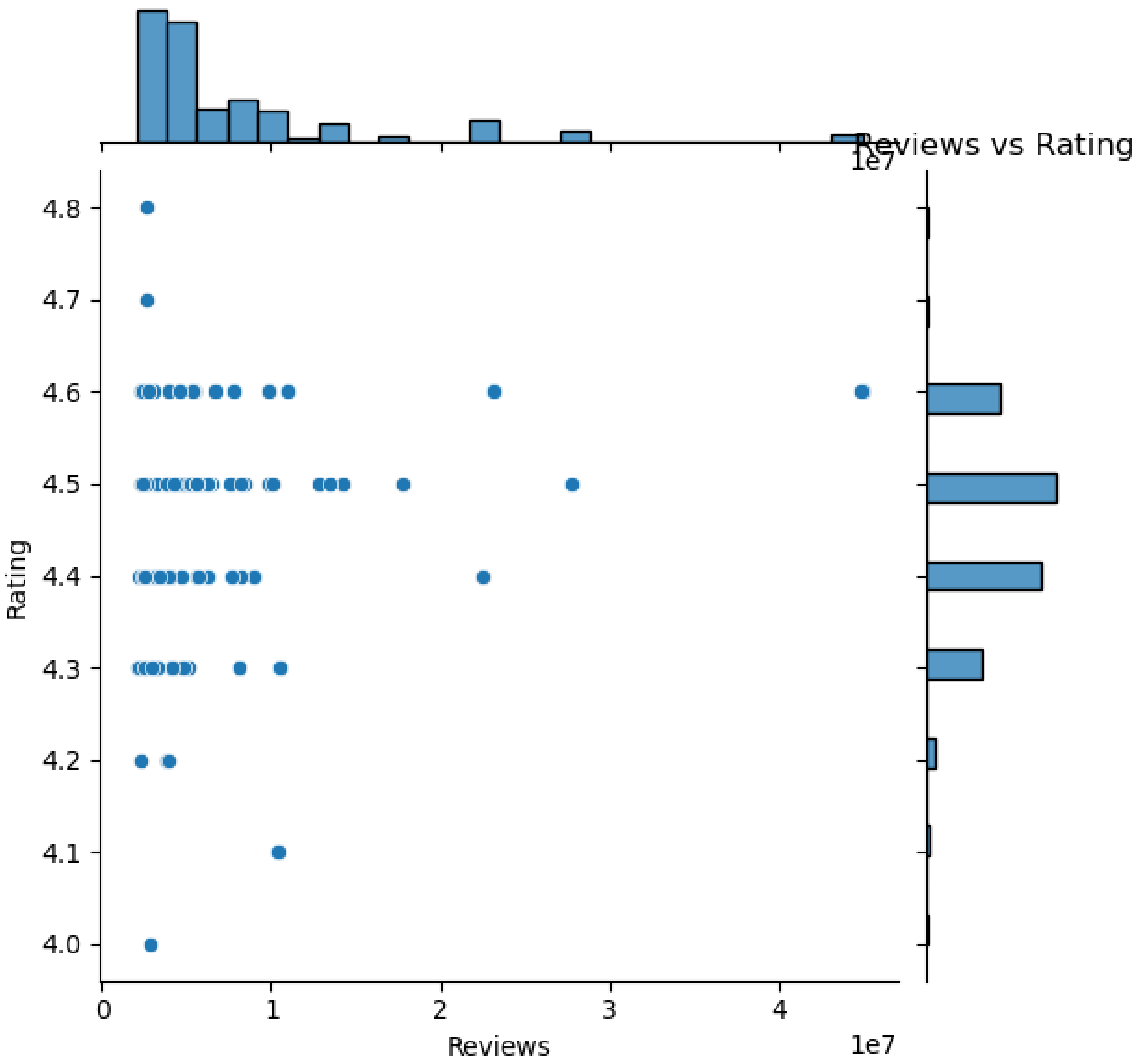


```
# Making  scatter plot/joinplot for Rating vs. Size
sns.jointplot(x = 'Size' , y = 'Rating', data = data)
# set plot title & label
plt.title('Size vs Rating')
plt.xlabel('Size')
plt.ylabel('Rating')

Text(463.154761904762, 0.5, 'Rating')
```

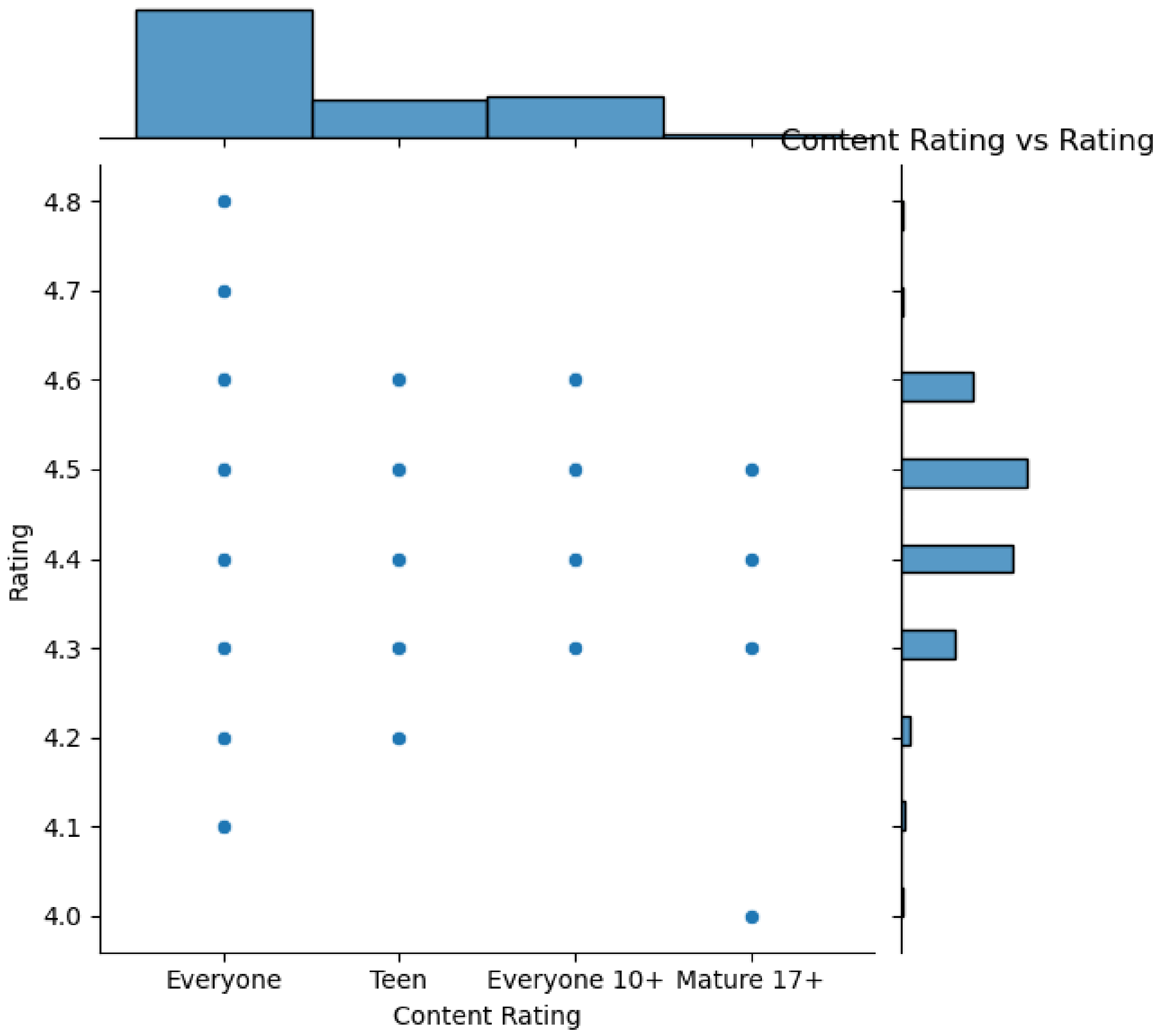


```
# Making  scatter plot/joinplot for Rating vs. Reviews
sns.jointplot(x = 'Reviews' , y = 'Rating', data = data)
# set plot title & label
plt.title('Reviews vs Rating')
plt.xlabel('Reviews')
plt.ylabel('Rating')
```

```
Text(463.154761904762, 0.5, 'Rating')
```

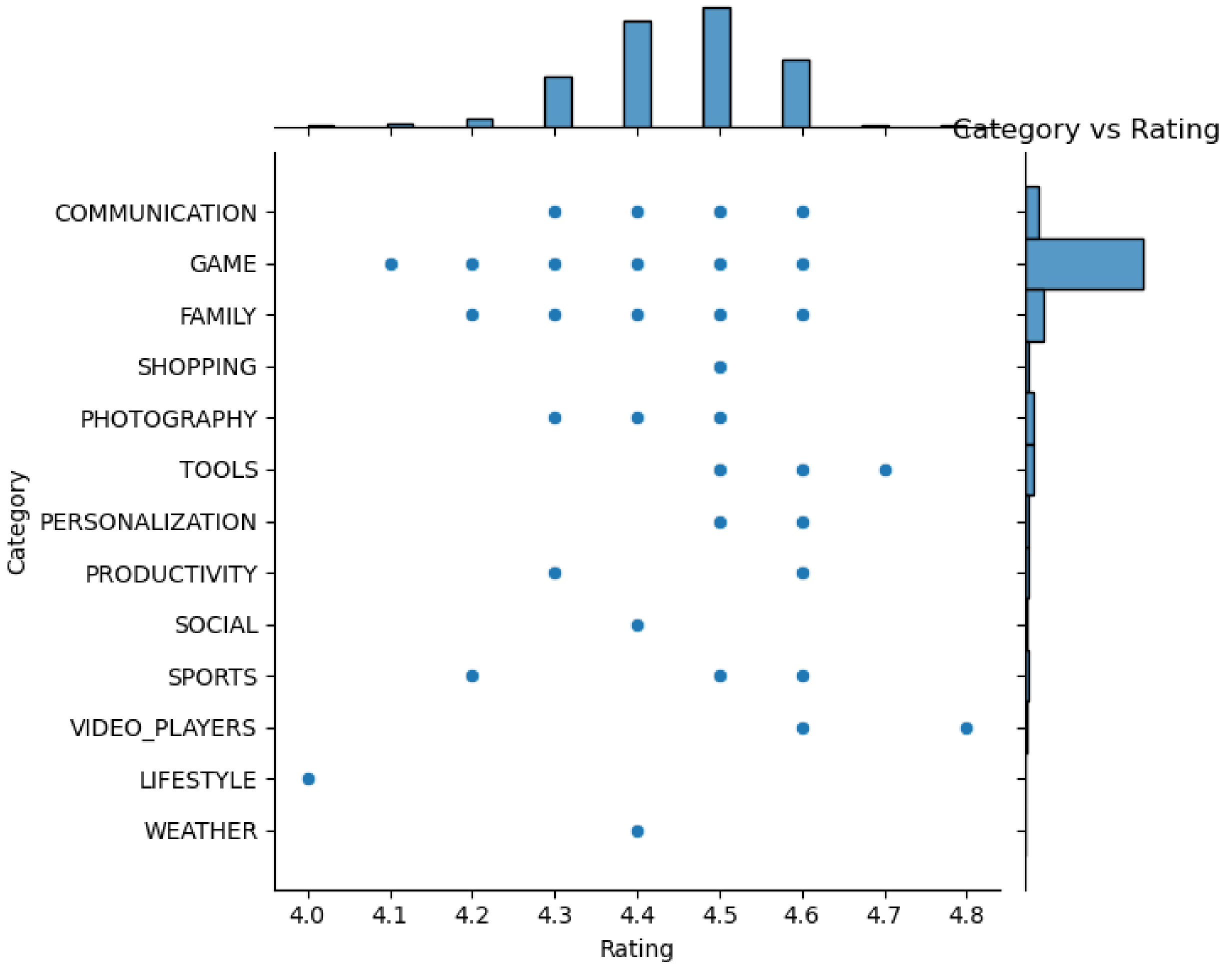


```
# Making  scatter plot/joinplot for Rating vs. Price
sns.jointplot(x = 'Content Rating' , y = 'Rating', data = data)
# set plot title & label
plt.title('Content Rating vs Rating')
plt.xlabel('Content Rating')
plt.ylabel('Rating')

Text(463.154761904762, 0.5, 'Rating')
```



```
# Making  scatter plot/joinplot for Rating vs. Price
sns.jointplot(x = 'Rating' , y = 'Category', data = data)
# set plot title & label
plt.title('Category vs Rating')
plt.ylabel('category')
plt.xlabel('Rating')
```

```
Text(0.5, 60.44444444444443, 'Rating')
```



Starting Data Preprocessing

```
# Apply log transformation to Reviews and Installs
inp1 = data.copy()

inp1['Reviews'] = np.log1p(inp1['Reviews'])
inp1['Installs'] = np.log1p(inp1['Installs'])

# Drop Unnecessary Columns
inp2 = inp1.drop(['App','Type'],axis = 1)

# get Dummies Columns For Category,Genres and Content Rating
inp2 = pd.get_dummies(inp2,columns=['Category','Genres','Content
Rating'])
print(inp2)
```

```
       Rating    Reviews         Size   Installs  Price  \
345       4.3  15.247734  16000000.0  18.420681    0.0
347       4.3  15.381183  11000000.0  20.030119    0.0
```

```
366       4.4  15.109723   3300000.0  18.420681    0.0
378       4.5  16.689805  40000000.0  20.030119    0.0
383       4.3  15.381203  11000000.0  20.030119    0.0
...       ...        ...         ...        ...    ...
9142      4.4  15.022768  22000000.0  17.727534    0.0
9166      4.3  14.881389  58000000.0  18.420681    0.0
10186     4.4  15.845716  71000000.0  18.420681    0.0
10190     4.6  14.816850  25000000.0  16.118096    0.0
10327     4.5  15.526442  53000000.0  18.420681    0.0

       Category_COMMUNICATION  Category_FAMILY  Category_GAME  \
345                         1                0              0
347                         1                0              0
366                         1                0              0
378                         1                0              0
383                         1                0              0
...                       ...              ...            ...
9142                        0                0              1
9166                        0                0              1
10186                       0                1              0
10190                       0                1              0
10327                       0                0              1

       Category_LIFESTYLE  Category_PERSONALIZATION  ...
Genres_Sports  \
345                     0                         0  ...
0
347                     0                         0  ...
0
366                     0                         0  ...
0
378                     0                         0  ...
0
383                     0                         0  ...
0
...                   ...                       ...  ...             ..
.
9142                    0                         0  ...
0
9166                    0                         0  ...
0
10186                   0                         0  ...
0
10190                   0                         0  ...
0
10327                   0                         0  ...
0

       Genres_Strategy  Genres_Tools  Genres_Trivia  \
345                  0             0              0
```

```
347                  0             0              0
366                  0             0              0
378                  0             0              0
383                  0             0              0
...                ...           ...            ...
9142                 0             0              0
9166                 0             0              0
10186                0             0              0
10190                0             0              0
10327                0             0              0

       Genres_Video Players & Editors  Genres_Weather  \
345                                 0               0
347                                 0               0
366                                 0               0
378                                 0               0
383                                 0               0
...                               ...             ...
9142                                0               0
9166                                0               0
10186                               0               0
10190                               0               0
10327                               0               0

       Content Rating_Everyone  Content Rating_Everyone 10+  \
345                          1                            0
347                          1                            0
366                          0                            0
378                          0                            0
383                          1                            0
...                        ...                          ...
9142                         0                            1
9166                         0                            0
10186                        1                            0
10190                        0                            0
10327                        0                            0

       Content Rating_Mature 17+  Content Rating_Teen
345                            0                    0
347                            0                    0
366                            0                    1
378                            0                    1
383                            0                    0
...                          ...                  ...
9142                           0                    0
9166                           1                    0
10186                          0                    0
10190                          0                    1
10327                          0                    1
```

```
[219 rows x 45 columns]

# Now Splitting the dataset
from sklearn.model_selection import train_test_split
df_train,df_test = train_test_split(inp2,test_size = 0.3,random_state
= 42)

# Now Separate the dataset into X_train,y_train,x_test,y_test
X_train = df_train.drop('Rating',axis = 1)
y_train = df_train['Rating']
X_test = df_test.drop('Rating',axis =1)
y_test = df_test['Rating']
```

Train the Algorithm

```
from sklearn.linear_model import LinearRegression
nish = LinearRegression()

nish.fit(X_train,y_train)

LinearRegression()
```

Predicting on train data

```
y_pred = nish.predict(X_train)

y_pred

array([4.46719393, 4.54649443, 4.50020016, 4.4       , 4.51873335,
       4.55478493, 4.46071994, 4.40959896, 4.46072157, 4.36964125,
       4.51087618, 4.42946928, 4.39488219, 4.5547909 , 4.38122735,
       4.40452268, 4.58644239, 4.39729398, 4.39496566, 4.46760851,
       4.42205531, 4.47842422, 4.46143535, 4.49572002, 4.39028867,
       4.42883904, 4.46378024, 4.43759283, 4.5424741 , 4.50002028,
       4.46072536, 4.5386377 , 4.46289521, 4.51711256, 4.51086719,
       4.39028867, 4.50324029, 4.55478388, 4.4       , 4.5745843 ,
       4.42915207, 4.503213  , 4.42891824, 4.41723874, 4.44752829,
       4.40251245, 4.46112827, 4.42054271, 4.47139231, 4.35911932,
       4.58644005, 4.50002028, 4.36962449, 4.42768863, 4.39026297,
       4.49571964, 4.35911932, 4.42915481, 4.6122286 , 4.47946336,
       4.49563812, 4.4404399 , 4.51499557, 4.35911834, 4.42915583,
       4.6       , 4.55479161, 4.40458367, 4.44319343, 4.554106  ,
       4.41118569, 4.43759164, 4.41779865, 4.5       , 4.47843179,
       4.43322601, 4.42884765, 4.42303047, 4.50151794, 4.49993949,
       4.46377932, 4.48334512, 4.43485231, 4.42074438, 4.39729506,
       4.36964125, 4.43889695, 4.55350557, 4.39890157, 4.59078333,
       4.46072315, 4.36964253, 4.46760048, 4.42884385, 4.51087618,
       4.42526102, 4.47893491, 4.57474287, 4.50047847, 4.43332338,
       4.42074967, 4.42914821, 4.4509518 , 4.4475319 , 4.57055294,
       4.39729219, 4.43884638, 4.5547962 , 4.45619104, 4.4611284 ,
       4.50077313, 4.57389551, 4.43013182, 4.1       , 4.43801811,
       4.42915207, 4.50019256, 4.33820321, 4.45072725, 4.50800536,
       4.50322835, 4.44826682, 4.58644005, 4.45096996, 4.41621648,
       4.61222918, 4.56909691, 4.44626105, 4.50087036, 4.47842461,
       4.42525056, 4.51772709, 4.35911834, 4.43801576, 4.43799338,
       4.57390952, 4.57889761, 4.42303175, 4.42699653, 4.50001996,
       4.4045933 , 4.56469133, 4.45356856, 4.39028536, 4.43383215,
       4.45356438, 4.4225775 , 4.50324059, 4.40252189, 4.55410333,
       4.36964255, 4.53916793, 4.55479522])
```

```
y_test
```

```
4017     4.4
1923     4.6
10186    4.4
10190    4.6
1661     4.3
        ...
395      4.4
3883     4.4
1781     4.5
2016     4.5
3987     4.6
Name: Rating, Length: 66, dtype: float64
```

Evaluating the Model

```
from sklearn.metrics import r2_score

r2 = r2_score(y_train,y_pred)
print(r2)
```

```
0.42354747585008967
```

Predicting on test data

```
y_pred = nish.predict(X_test)
```

```
y_pred
```

```
array([4.4482539 , 4.40459343, 4.4563243 , 4.53901289, 4.36963928,
       4.44826698, 4.4400602 , 4.47920499, 4.55411012, 4.47086602,
       4.43332338, 4.38876883, 4.3972885 , 4.44044157, 4.43483551,
       4.42766004, 4.34908763, 4.46760058, 4.57446886, 4.45097811,
       4.61483287, 4.54512255, 4.39028707, 4.46377842, 4.50696183,
       4.42074438, 4.46108145, 4.3887687 , 4.53863196, 4.42160124,
       4.4607014 , 4.38048417, 4.41118594, 4.41590532, 4.46072536,
       4.43759373, 4.40960369, 4.41834306, 4.099983  , 4.42915583,
       4.43759356, 4.44044393, 4.50002844, 4.35604812, 4.388772  ,
       4.4600439 , 4.46378019, 4.50151833, 4.46760851, 4.41835818,
       4.45660418, 4.45356687, 4.44826556, 4.46120353, 4.41118555,
       4.50321254, 4.4172529 , 4.41209311, 4.54513491, 4.51879379,
       4.40134589, 4.42269557, 4.43758453, 4.43361932, 4.4957093 ,
       4.54655472])
```

```
y_test
```

```
4017     4.4
1923     4.6
10186    4.4
10190    4.6
1661     4.3
        ...
395      4.4
3883     4.4
1781     4.5
2016     4.5
3987     4.6
Name: Rating, Length: 66, dtype: float64
```

Evaluating the model

```
r2 = r2_score(y_pred,y_test)
print(r2)
```

```
-1.0816981846117333
```