Hi everybody !

In this notebook, I'm gonna analyze Google Play Store datas. While I was analyzing the data, I used Python. This study is my first data analyzing study. If you liked this kernel or it was benefit to you, forgot upvotes! Good studies. :)

CONTENT

1. INTRODUCTION TO DATA

Firstly let's get to know data. While I was analyzing the data, I used Pandas library.

- info(): It informs about data columns and data types.
- head(): It returns the first five data.
- tail(): It returns the last five data.
- columns : It returns data columns

In [2]:

```
#import library
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import numpy as np # linear algebra
import matplotlib.pyplot as plt
import seaborn as sns  # visualization tool
# plotly
# import plotly.plotly as py
# from plotly.offline import init_notebook_mode, iplot
# init_notebook_mode(connected=True)
# import plotly.graph_objs as go

# word cloud library
from wordcloud import WordCloud
```

In [3]:
```
#read to csv
data = pd.read_csv("Google Apps data.csv")
data.head()
```

Out[3]:

	Unnamed: 0.1	Unnamed: 0	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Last Updated	Current Ver	Minimum Android Ver	Genres
0	0	0	Photo Editor & Candy Camera & Grid & ScrapBook	Art And Design	4.1	159	19.0	10000	Free	0.0	Others	January 7, 2018	1.0.0	4.0.3	Art & Design
1	1	1	Coloring book moana	Art And Design	3.9	967	14.0	500000	Free	0.0	Others	January 15, 2018	2.0.0	4.0.3	Art & Design
2	2	5	U Launcher Lite – FREE Live Cool Themes, Hide ...	Art And Design	4.7	87510	8.7	5000000	Free	0.0	Others	August 1, 2018	1.2.4	4.0.3	Art & Design
3	3	6	Sketch - Draw & Paint	Art And Design	4.5	215644	25.0	50000000	Free	0.0	Teen	June 8, 2018	Varies with device	4.2	Art & Design
4	4	7	Pixel Draw - Number Art Coloring Book	Art And Design	4.3	967	2.8	100000	Free	0.0	Others	June 20, 2018	1.1	4.4	Art & Design

In [4]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8276 entries, 0 to 8275
Data columns (total 15 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   Unnamed: 0.1         8276 non-null   int64  
 1   Unnamed: 0           8276 non-null   int64  
 2   App                  8276 non-null   object 
 3   Category             8276 non-null   object 
 4   Rating               8276 non-null   float64
 5   Reviews              8276 non-null   int64  
 6   Size                 8276 non-null   float64
 7   Installs             8276 non-null   int64  
 8   Type                 8276 non-null   object 
 9   Price                8276 non-null   float64
 10  Content Rating       7915 non-null   object 
 11  Last Updated         8276 non-null   object 
 12  Current Ver          8276 non-null   object 
 13  Minimum Android Ver  8276 non-null   object 
 14  Genres               8276 non-null   object 
dtypes: float64(3), int64(4), object(8)
memory usage: 970.0+ KB
```

In [5]:
```
data.columns
```

Out[5]:
```
Index(['Unnamed: 0.1', 'Unnamed: 0', 'App', 'Category', 'Rating', 'Reviews',
       'Size', 'Installs', 'Type', 'Price', 'Content Rating', 'Last Updated',
       'Current Ver', 'Minimum Android Ver', 'Genres'],
      dtype='object')
```

In [6]:
```
data.shape
```

Out[6]:
```
(8276, 15)
```

In [7]:
```
data.head()
```

Out[7]:

	Unnamed: 0.1	Unnamed: 0	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Last Updated	Current Ver	Minimum Android Ver	Genres
0	0	0	Photo Editor & Candy Camera & Grid & ScrapBook	Art And Design	4.1	159	19.0	10000	Free	0.0	Others	January 7, 2018	1.0.0	4.0.3	Art & Design
1	1	1	Coloring book moana	Art And Design	3.9	967	14.0	500000	Free	0.0	Others	January 15, 2018	2.0.0	4.0.3	Art & Design
2	2	5	U Launcher Lite – FREE Live Cool Themes, Hide ...	Art And Design	4.7	87510	8.7	5000000	Free	0.0	Others	August 1, 2018	1.2.4	4.0.3	Art & Design
3	3	6	Sketch - Draw & Paint	Art And Design	4.5	215644	25.0	50000000	Free	0.0	Teen	June 8, 2018	Varies with device	4.2	Art & Design
4	4	7	Pixel Draw - Number Art Coloring Book	Art And Design	4.3	967	2.8	100000	Free	0.0	Others	June 20, 2018	1.1	4.4	Art & Design

In [8]:
```
data.tail()
```

	Unnamed: 0.1	Unnamed: 0	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Last Updated	Current Ver	Android Ver	Genr
8271	8271	8912	FR Calculator	Family	4.0	7	2.6	500	Free	0.0	Others	June 18, 2017	1.0.0	4.1	Educati
8272	8272	8913	Sya9a Maroc - FR	Family	4.5	38	53.0	5000	Free	0.0	Others	July 25, 2017	1.48	4.1	Educati
8273	8273	8914	Fr. Mike Schmitz Audio Teachings	Family	5.0	4	3.6	100	Free	0.0	Others	July 6, 2018	1.0	4.1	Educati
8274	8274	8915	The SCP Foundation DB fr nn5n	Books And Reference	4.5	114	1.0	1000	Free	0.0	NaN	January 19, 2015	Varies with device	-1	Books Referen
8275	8275	8916	iHoroscope - 2018 Daily Horoscope & Astrology	Lifestyle	4.5	398307	19.0	10000000	Free	0.0	Others	July 25, 2018	Varies with device	-1	Lifest

We can combine tables to make it easier to see data. For this, we are gonna use "concat function" that is found Pandas library.

pd.concat([data frame parameters], axis,ignore_index) : It combines 2 tables.
axis : It adds the tables as horizontal or vertical. If axis equals 0, it adds as horizontal. If axis equals 1, it adds as vertical.
ignore_index : It ignores index values.

In [9]:
```
data1 = data.head()
data2 = data.tail()
concat_data = pd.concat([data1,data2],axis=0,ignore_index=True)
concat_data
```

			Coloring Book	Design								2018		Design
5	8271	8912	FR Calculator	Family	4.0	7	2.6	500	Free	0.0	Others	June 18, 2017	1.0.0	4.1	Education
6	8272	8913	Sya9a Maroc - FR	Family	4.5	38	53.0	5000	Free	0.0	Others	July 25, 2017	1.48	4.1	Education
7	8273	8914	Fr. Mike Schmitz Audio Teachings	Family	5.0	4	3.6	100	Free	0.0	Others	July 6, 2018	1.0	4.1	Education
8	8274	8915	The SCP Foundation DB fr nn5n	Books And Reference	4.5	114	1.0	1000	Free	0.0	NaN	January 19, 2015	Varies with device	-1	Books & Reference
9	8275	8916	iHoroscope - 2018 Daily Horoscope & Astrology	Lifestyle	4.5	398307	19.0	10000000	Free	0.0	Others	July 25, 2018	Varies with device	-1	Lifestyle

2. Cleaning Data

Dataset can contain missing data, numerical string value, various cues. If we can clean them, we can make easy our analysis.

Let's have some fun. :)

Category

In [32]:
```
data.isnull().sum()
```

Out[32]:
```
Unnamed: 0.1            0
Unnamed: 0              0
App                     0
Category                0
Rating                  0
Reviews                 0
Size                    0
Installs                0
Type                    0
Price                   0
Content Rating        361
Last Updated            0
Current Ver             0
Minimum Android Ver     0
Genres                  0
dtype: int64
```

#content Rating have 361 Nan Value

In [10]:
```
data['Category'].unique()
```

Out[10]:
```
array(['Art And Design', 'Auto And Vehicles', 'Beauty',
       'Books And Reference', 'Business', 'Comics', 'Communication',
       'Dating', 'Education', 'Entertainment', 'Events', 'Finance',
       'Food And Drink', 'Health And Fitness', 'House And Home',
       'Libraries And Demo', 'Lifestyle', 'Game', 'Family', 'Medical',
       'Social', 'Shopping', 'Photography', 'Sports', 'Travel And Local',
       'Tools', 'Personalization', 'Productivity', 'Parenting', 'Weather',
       'Video Players', 'News And Magazines', 'Maps And Navigation'],
      dtype=object)
```

Rating

In [13]:
```
data['Rating'].unique()
```

Out[13]:
```
array([4.1, 3.9, 4.7, 4.5, 4.3, 4.4, 3.8, 4.2, 4.6, 4. , 4.8, 4.9, 3.6,
       3.7, 3.2, 3.3, 3.4, 3.5, 3.1, 5. , 2.6, 3. , 1.9, 2.5, 2.8, 2.7,
       1. , 2.9, 2.3, 2.2, 1.7, 2. , 1.8, 2.4, 1.6, 2.1, 1.4, 1.5, 1.2])
```

Data type of Rating is object. If we convert from string to numeric, we can make easy.

In [14]:
```
data['Rating'] = pd.to_numeric(data['Rating'], errors='coerce')
data['Rating'].dtype
```

Out[14]:
```
dtype('float64')
```

Reviews

In [15]:
```
data['Reviews'].unique()
```

Out[15]:
```
array([   159,    967,  87510, ...,    603,   1195, 398307], dtype=int64)
```

Size

In [17]:
```
data['Size'].unique()
```

Out[17]:
```
array([1.9000e+01, 1.4000e+01, 8.7000e+00, 2.5000e+01, 2.8000e+00,
       5.6000e+00, 2.9000e+01, 3.3000e+01, 3.1000e+00, 2.8000e+01,
       1.2000e+01, 2.0000e+01, 2.1000e+01, 3.7000e+01, 5.5000e+00,
       1.7000e+01, 3.9000e+01, 3.1000e+01, 4.2000e+00, 2.3000e+01,
       6.0000e+00, 6.1000e+00, 4.6000e+00, 9.2000e+00, 5.2000e+00,
       1.1000e+01, 2.4000e+01, 1.0000e+00, 9.4000e+00, 1.5000e+01,
       1.0000e+01, 1.2000e+00, 2.6000e+01, 8.0000e+00, 7.9000e+00,
       5.6000e+01, 5.7000e+01, 3.5000e+01, 5.4000e+01, 1.9629e-01,
       3.6000e+00, 5.7000e+00, 8.6000e+00, 2.4000e+00, 2.7000e+01,
       2.7000e+00, 2.5000e+00, 7.0000e+00, 1.6000e+01, 3.4000e+00,
       8.9000e+00, 3.9000e+00, 2.9000e+00, 3.8000e+01, 3.2000e+01,
       5.4000e+00, 1.8000e+01, 1.1000e+00, 2.2000e+00, 4.5000e+00,
       9.8000e+00, 5.2000e+01, 9.0000e+00, 6.7000e+00, 3.0000e+01,
       2.6000e+00, 7.1000e+00, 2.2000e+01, 6.4000e+00, 3.2000e+00,
       8.2000e+00, 4.9000e+00, 9.5000e+00, 5.0000e+00, 5.9000e+00,
       1.3000e+01, 7.3000e+01, 6.8000e+00, 3.5000e+00, 4.0000e+00,
       2.3000e+00, 2.1000e+00, 4.2000e+01, 9.1000e+00, 5.5000e+01,
       2.2460e-02, 7.3000e+00, 6.5000e+00, 1.5000e+00, 7.5000e+00,
       5.1000e+01, 4.1000e+01, 4.8000e+01, 8.5000e+00, 4.6000e+01,
       8.3000e+00, 4.3000e+00, 4.7000e+00, 3.3000e+00, 4.0000e+01,
       7.8000e+00, 8.8000e+00, 6.6000e+00, 5.1000e+00, 6.1000e+01,
       6.6000e+01, 7.7150e-02, 8.4000e+00, 3.7000e+00, 1.1523e-01,
       4.4000e+01, 6.7871e-01, 1.6000e+00, 6.2000e+00, 5.3000e+01,
       1.4000e+00, 3.0000e+00, 7.2000e+00, 5.8000e+00, 3.8000e+00,
       9.6000e+00, 4.5000e+01, 6.3000e+01, 4.9000e+01, 7.7000e+01,
       4.4000e+00, 7.0000e+01, 9.3000e+00, 8.1000e+00, 3.6000e+01,
       6.9000e+00, 7.4000e+00, 8.4000e+01, 9.7000e+01, 2.0000e+00,
       1.9000e+00, 1.8000e+00, 5.3000e+00, 4.7000e+01, 5.4297e-01,
       5.1367e-01, 7.6000e+01, 7.6000e+00, 5.9000e+01, 9.7000e+00,
       7.8000e+01, 7.2000e+01, 4.3000e+01, 7.7000e+00, 6.3000e+00,
       3.2617e-01, 9.3000e+01, 6.5000e+01, 7.9000e+01, 1.0000e+02,
       5.8000e+01, 5.0000e+01, 6.8000e+01, 6.4000e+01, 3.4000e+01,
       6.7000e+01, 6.0000e+01, 9.4000e+01, 9.9000e+00, 2.2656e-01,
       9.9000e+01, 6.0938e-01, 9.5000e+01, 8.3000e-03, 4.0040e-02,
       2.8516e-01, 8.0000e+01, 1.7000e+00, 7.4000e+01, 6.2000e+01,
       6.9000e+01, 7.5000e+01, 9.8000e+01, 8.5000e+01, 8.2000e+01,
       9.6000e+01, 8.7000e+01, 7.1000e+01, 8.6000e+01, 9.1000e+01,
       8.1000e+01, 9.2000e+01, 8.3000e+01, 8.8000e+01, 6.8750e-01,
       8.4180e-01, 8.7793e-01, 3.6914e-01, 4.8000e+00, 2.5977e-01,
       3.6621e-01, 1.3000e+00, 9.5215e-01, 9.5703e-01, 4.1000e+00,
       8.9000e+01, 6.7969e-01, 5.3125e-01, 5.1270e-01, 8.9844e-01,
```

```
7.6074e-01, 8.3301e-01, 7.0312e-01, 6.9629e-01, 7.5391e-01,
3.1055e-01, 5.6640e-02, 2.3535e-01, 1.9141e-01, 8.3691e-01,
4.9800e-02, 9.3066e-01, 8.4473e-01, 2.4512e-01, 9.0820e-01,
5.2734e-01, 3.0566e-01, 7.2852e-01, 1.9824e-01, 2.5390e-02,
3.0664e-01, 2.3340e-01, 3.6230e-01, 2.1484e-01, 7.1289e-01,
7.3828e-01, 8.8870e-02, 2.8613e-01, 1.6600e-02, 7.2270e-02,
1.3670e-02, 3.0957e-01, 7.6170e-02, 9.0234e-01, 7.9883e-01,
7.9100e-02, 9.1699e-01, 1.6504e-01, 4.3950e-02, 9.4238e-01,
9.0000e+01, 5.3223e-01, 5.9570e-02, 2.7637e-01, 6.3965e-01,
6.9727e-01, 9.0820e-02, 8.5156e-01, 1.1816e-01, 3.1445e-01,
9.5312e-01, 2.0117e-01, 9.3164e-01, 4.3359e-01, 7.0020e-01,
2.0508e-01, 5.9473e-01, 3.0078e-01, 2.9883e-01, 1.7090e-01,
3.4180e-01, 3.7402e-01, 4.4336e-01, 6.8360e-02, 7.9297e-01,
4.3164e-01, 8.2227e-01, 4.0723e-01, 4.0234e-01, 4.4824e-01,
4.6680e-01, 3.2715e-01, 7.6367e-01, 7.0410e-01, 4.1992e-01,
4.1895e-01, 1.8750e-01, 4.4922e-01, 7.1094e-01, 4.8438e-01,
7.9688e-01, 4.0430e-01, 4.9414e-01, 8.6621e-01, 5.9863e-01,
7.5977e-01, 6.6699e-01, 5.7812e-01, 1.8164e-01, 8.2031e-01,
6.3184e-01, 3.6426e-01, 4.2676e-01, 5.8398e-01, 6.9922e-01,
5.7129e-01, 9.5898e-01, 2.1387e-01, 5.3710e-02, 3.1543e-01,
6.7480e-01, 4.9902e-01, 9.2871e-01, 9.4043e-01, 2.4410e-02,
5.4102e-01, 3.4277e-01, 2.6370e-02, 8.0080e-02, 2.0312e-01,
5.3809e-01, 2.8320e-02, 1.0059e-01, 1.1328e-01, 1.4941e-01,
2.0410e-01, 4.8730e-01, 1.6895e-01, 5.8301e-01, 7.9004e-01,
1.1914e-01, 4.0137e-01, 3.9062e-01, 7.8223e-01, 7.6855e-01,
4.8830e-02, 6.2793e-01, 9.6289e-01, 5.0391e-01, 8.1738e-01,
7.6172e-01, 1.9530e-02, 4.8633e-01, 5.8594e-01, 6.4062e-01,
2.1582e-01, 2.2266e-01, 1.7188e-01, 3.3200e-02, 2.5293e-01,
1.6016e-01, 4.4727e-01, 6.1426e-01, 2.7340e-02, 2.8125e-01,
7.5684e-01, 7.6660e-01, 6.2109e-01, 8.9453e-01, 9.7070e-01,
3.0176e-01, 4.7363e-01, 8.9258e-01, 8.8184e-01, 5.9375e-01,
4.8828e-01, 5.2730e-02, 5.4883e-01, 8.2715e-01, 9.2578e-01,
7.9199e-01, 2.6367e-01, 4.6880e-02, 5.1074e-01, 7.6562e-01,
2.7344e-01, 2.3440e-02, 8.7109e-01, 1.5039e-01, 1.7580e-02,
3.2230e-02, 8.3984e-01, 3.5547e-01, 3.7793e-01, 6.1133e-01,
1.5723e-01, 8.5840e-01, 3.8090e-02, 1.6602e-01, 1.3770e-01,
1.5625e-01, 1.4062e-01, 1.3965e-01, 1.8555e-01, 3.6719e-01,
1.8848e-01, 4.6191e-01, 2.4023e-01, 7.1290e-02, 2.4707e-01,
9.3457e-01, 4.1016e-01, 7.0310e-02, 3.9453e-01, 4.5898e-01,
2.2070e-01, 2.3438e-01, 8.6910e-02, 2.2852e-01, 2.5098e-01,
8.4082e-01, 4.5605e-01, 6.6016e-01, 5.3906e-01, 5.6836e-01,
6.0449e-01])
```

Installs

In [19]:
```
data['Installs'].unique()
```

Out[19]:
```
array([     10000,     500000,    5000000,   50000000,     100000,
            50000,    1000000,   10000000,       5000,  100000000,
       1000000000,       1000,  500000000,        100,        500,
               10,          5,         50,          1], dtype=int64)
```

Price

In [21]:
```
data['Price'].unique()
```

Out[21]:
```
array([  0.  ,   4.99,   3.99,   6.99,   7.99,   5.99,   2.99,   3.49,
         1.99,   9.99,   7.49,   0.99,   9.  ,   5.49,  10.  ,  24.99,
        11.99,  79.99,  16.99,  14.99,  29.99,  12.99,   2.49,  10.99,
         1.5 ,  19.99,  15.99,  33.99,  39.99,   3.95,   4.49,   1.7 ,
         8.99,   1.49,   3.88, 399.99,  17.99, 400.  ,   3.02,   1.76,
         4.84,   4.77,   1.61,   2.5 ,   1.59,   6.49,   1.29, 299.99,
       379.99,  37.99,  18.99, 389.99,   8.49,   1.75,  14.  ,   2.  ,
         3.08,   2.59,  19.4 ,   3.9 ,   4.59,  15.46,   3.04,  13.99,
         4.29,   3.28,   4.6 ,   1.  ,   2.95,   2.9 ,   1.97,   2.56,
         1.2 ])
```

Last Updated

In [31]:
```
data['Last Updated'].unique()
```

Out[31]:
```
array(['2018-01-07T00:00:00.000000000', '2018-01-15T00:00:00.000000000',
       '2018-08-01T00:00:00.000000000', ...,
       '2014-01-20T00:00:00.000000000', '2014-02-16T00:00:00.000000000',
       '2014-03-23T00:00:00.000000000'], dtype='datetime64[ns]')
```

In [30]:
```
data['Last Updated'] = pd.to_datetime(data['Last Updated'])
data['Last Updated']
```

Out[30]:
```
0      2018-01-07
1      2018-01-15
2      2018-08-01
3      2018-06-08
4      2018-06-20
          ...
8271   2017-06-18
8272   2017-07-25
8273   2018-07-06
8274   2015-01-19
8275   2018-07-25
Name: Last Updated, Length: 8276, dtype: datetime64[ns]
```

Exploratory Data Analysis

After, I prepared to analyze our data, somewhat let's explore the datas. :)

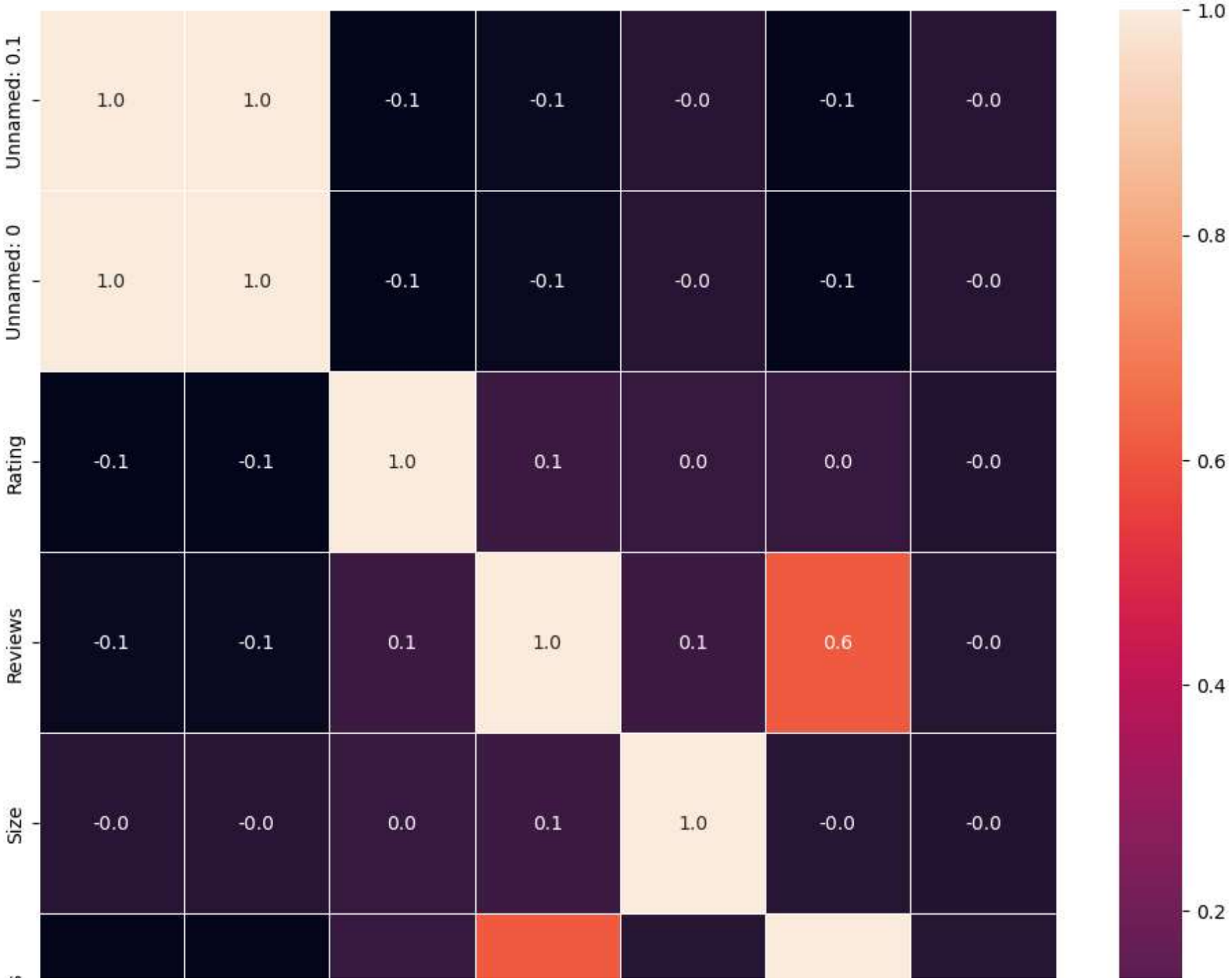
- corr() : It returns correlation.
- describe (): It returns number of entries, average of entries, outlier values, standart deviation, minimum and maximum entry.

In [23]:
```
data.corr()
```

Out[23]:

	Unnamed: 0.1	Unnamed: 0	Rating	Reviews	Size	Installs	Price
Unnamed: 0.1	1.000000	0.999571	-0.123565	-0.092450	-0.004724	-0.115358	-0.002715
Unnamed: 0	0.999571	1.000000	-0.124091	-0.092332	-0.005205	-0.114801	-0.001452
Rating	-0.123565	-0.124091	1.000000	0.059430	0.041181	0.042372	-0.021316
Reviews	-0.092450	-0.092332	0.059430	1.000000	0.056789	0.611471	-0.008080
Size	-0.004724	-0.005205	0.041181	0.056789	1.000000	-0.005860	-0.017899
Installs	-0.115358	-0.114801	0.042372	0.611471	-0.005860	1.000000	-0.009859
Price	-0.002715	-0.001452	-0.021316	-0.008080	-0.017899	-0.009859	1.000000

In [24]:
```
#correlation map
f,ax = plt.subplots(figsize=(12, 12))
sns.heatmap(data.corr(), annot=True, linewidths=.5, fmt= '.1f',ax=ax)
plt.show()
```


Unnamed: 0.1
Unnamed: 0
Rating
Reviews
Size
1.0 1.0 -0.1 -0.1 -0.0 -0.1 -0.0
1.0 1.0 -0.1 -0.1 -0.0 -0.1 -0.0
-0.1 -0.1 1.0 0.1 0.0 0.0 -0.0
-0.1 -0.1 0.1 1.0 0.1 0.6 -0.0
-0.0 -0.0 0.0 0.1 1.0 -0.0 -0.0
1.0
0.8
0.6
0.4
0.2

In [25]:
```
data.describe()
```

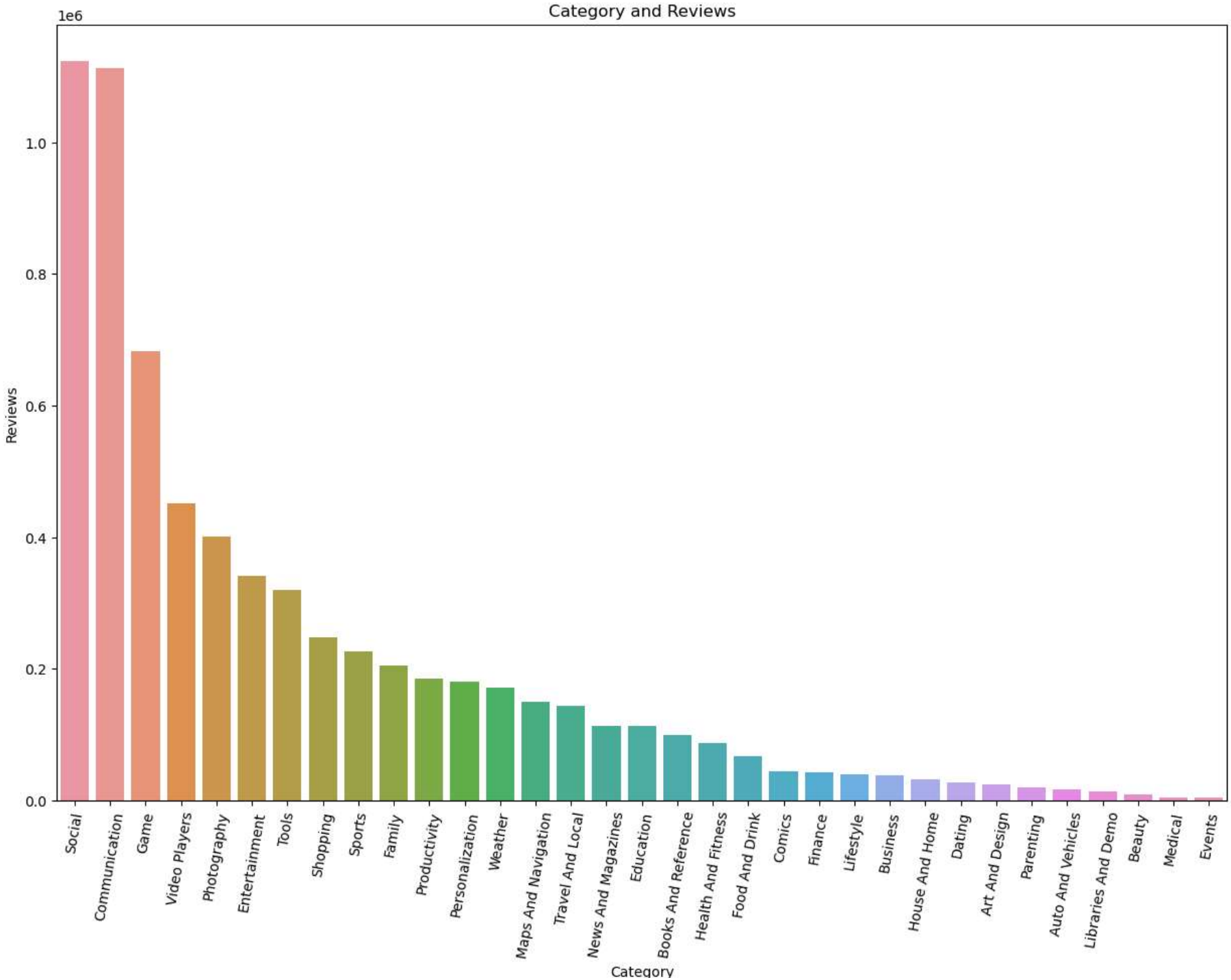
Out[25]:

	Unnamed: 0.1	Unnamed: 0	Rating	Reviews	Size	Installs	Price
count	8276.000000	8276.000000	8276.000000	8.276000e+03	8276.000000	8.276000e+03	8276.000000
mean	4137.500000	4560.609957	4.175121	2.803270e+05	18.897761	9.658206e+06	1.028758
std	2389.219747	2560.879748	0.534762	2.096170e+06	22.376521	5.986505e+07	16.776622
min	0.000000	0.000000	1.000000	1.000000e+00	0.008300	1.000000e+00	0.000000
25%	2068.750000	2459.750000	4.000000	1.290000e+02	2.800000	1.000000e+04	0.000000
50%	4137.500000	4613.500000	4.300000	3.213500e+03	9.500000	1.000000e+05	0.000000
75%	6206.250000	6765.250000	4.500000	4.627800e+04	27.000000	1.000000e+06	0.000000
max	8275.000000	8916.000000	5.000000	7.815831e+07	100.000000	1.000000e+09	400.000000

Category and Reviews

In [26]:
```
category_list = list(data['Category'].unique())
category_review = []
for i in category_list:
    x = data[data['Category'] == i]
    if(len(x)!=0):
        review = sum(x.Reviews)/len(x)
        category_review.append(review)
    else:
        review = sum(x.Reviews)
        category_review.append(review)
#sorting
data_category_reviews = pd.DataFrame({'category': category_list,'review':category_review})
new_index = (data_category_reviews['review'].sort_values(ascending=False)).index.values
sorted_data =data_category_reviews.reindex(new_index)

# visualization
plt.figure(figsize=(15,10))
sns.barplot(x=sorted_data['category'], y=sorted_data['review'])
plt.xticks(rotation=80)
plt.xlabel("Category")
plt.ylabel("Reviews")
plt.title("Category and Reviews")
plt.show()
```

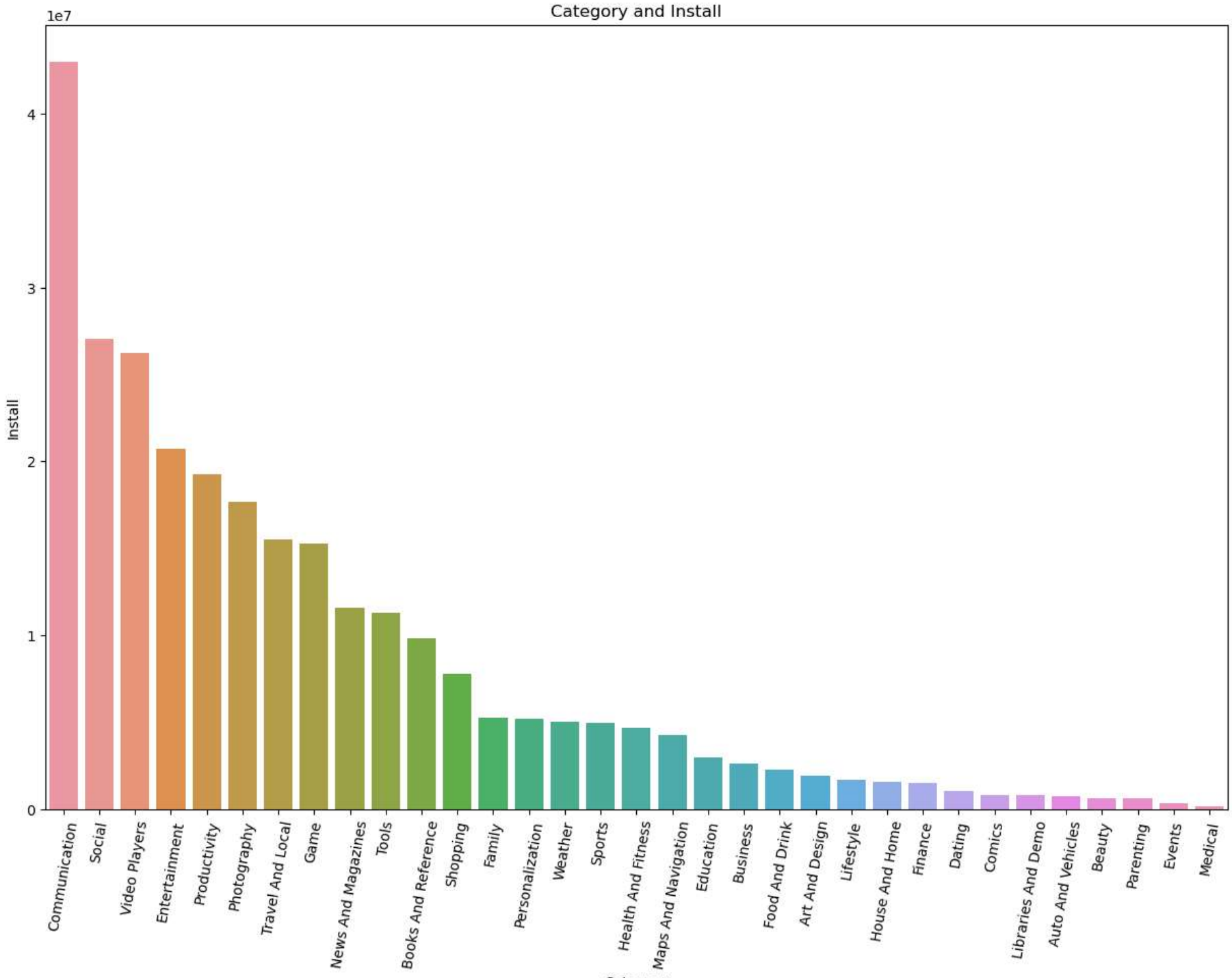

Category and Reviews
1e6
Reviews
1.0
0.8
0.6
0.4
0.2
0.0
Social
Communication
Game
Video Players
Photography
Entertainment
Tools
Shopping
Sports
Family
Productivity
Personalization
Weather
Maps And Navigation
Travel And Local
News And Magazines
Education
Books And Reference
Health And Fitness
Food And Drink
Comics
Finance
Lifestyle
Business
House And Home
Dating
Art And Design
Parenting
Auto And Vehicles
Libraries And Demo
Beauty
Medical
Events
Category

Category and Installs

In [27]:
```
category_list = list(data['Category'].unique())
category_install = []
for i in category_list:
    x = data[data['Category'] == i]
    if(len(x)!=0):
        install = sum(x.Installs)/len(x)
        category_install.append(install)
    else:
        install = sum(x.Installs)
        category_install.append(install)

#sorting
data_category_install = pd.DataFrame({'category': category_list,'install':category_install})
new_index = (data_category_install['install'].sort_values(ascending=False)).index.values
sorted_data =data_category_install.reindex(new_index)

# visualization
plt.figure(figsize=(15,10))
sns.barplot(x=sorted_data['category'], y=sorted_data['install'])
plt.xticks(rotation=80)
plt.xlabel("Category")
plt.ylabel("Install")
plt.title("Category and Install")
plt.show()
```


Category and Install
1e7
Install
0
1
2
3
4
Communication
Social
Video Players
Entertainment
Productivity
Photography
Travel And Local
Game
News And Magazines
Tools
Books And Reference
Shopping
Family
Personalization
Weather
Sports
Health And Fitness
Maps And Navigation
Education
Business
Food And Drink
Art And Design
Lifestyle
House And Home
Finance
Dating
Comics
Libraries And Demo
Auto And Vehicles
Beauty
Parenting
Events
Medical

Category

Word Cloud

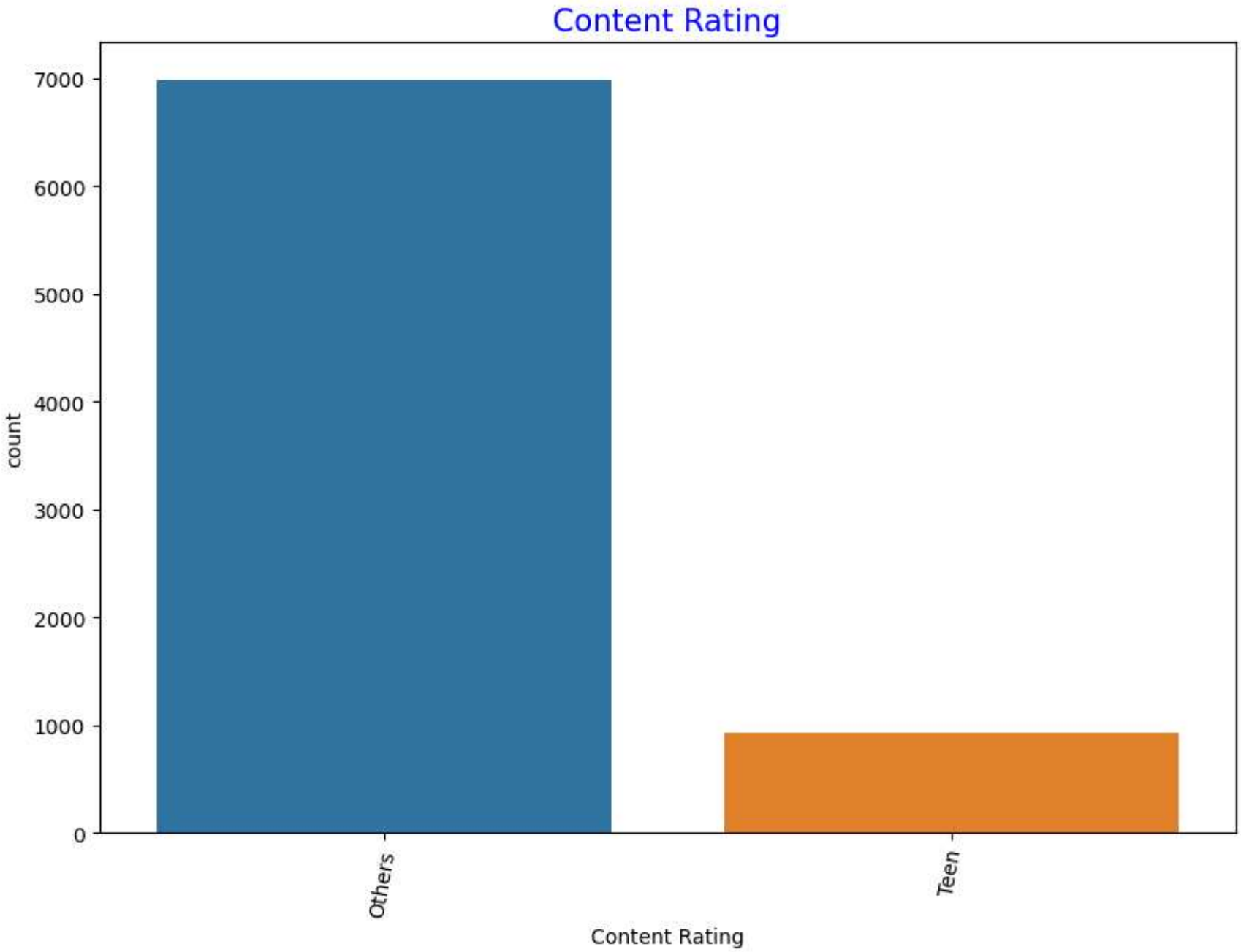
In [28]:
```
plt.subplots(figsize=(8,8))
wordcloud = WordCloud(
                           background_color='white',
                           width=512,
                           height=384
                          ).generate(" ".join(data))
plt.imshow(wordcloud)
plt.axis('off')
plt.savefig('graph.png')

plt.show()
```


Ver
Current
Size
Content
Reviews
Unnamed
Category
Updated
App
Price
Type
Rating
Genres
Android
Minimum
Installs
Last

Content Rating

In [29]:
```
plt.figure(figsize=(10,7))
sns.countplot(data=data, x='Content Rating')
plt.xticks(rotation=80)
plt.title('Content Rating',color = 'blue',fontsize=15)
plt.show()
```


Content Rating
7000
6000
5000
4000
3000
2000
1000
0
count
Others
Teen
Content Rating

Conclusion

This is the end of the story. I hope It benefits to you. You can visualize with a lot of different model. Actually, I thought that EDA study could be boring. But It's important for ML models. I'm definitely gonna myself about this subject. You can help with your comments. Wish to see you with different datasets.

**Thank you for your votes and comments* *