Capstone Project

Play store App Review Analysis

Team Members

Kishor Kumar

Aishwarya K P

Akshat Raj Kumawat

Mourvika Shirode

Soumyadip Paul

AI

Procedure Follows For Data Analysis

1. Data Summary

2. Data Cleaning

3.Data Manipulation

4. Data Visualization

5. Problem Statement

6. Conclusion

Data Summary

1. **Data Set Name :** Play_Store_Data.csv
- **Shape =** (10841, 13)
- **Columns of Data =** App, Category, Rating, Reviews, Size, Installs, Type, Price, Content Rating, Genres, Last updated, Current version, Android Version

2. **Data Set Name :** User_Reviews.csv
- **Shape =** (64295, 5)
- **Columns of Data =** App, Translated_Review, Sentiment, Sentiment_Polarity, Sentiment_Subjectivity

Data Cleaning

```
#Checking the null values for other columns:
df_app.isnull().sum()
```

```
App                0
Category           0
Rating             0
Reviews            0
Size               0
Installs           0
Type               1
Price              0
Content Rating     0
Genres             0
Last Updated       0
Current Ver        8
Android Ver        2
dtype: int64
```

Very few rows have null values so we can drop these!

```
df_app.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
```

AI

Process Of Data Cleaning

In the process of Data Cleaning some NAN values need to be removed from data set. Here Data Cleaning is done to remove NAN values from the following -

1. **Removing NAN values from Type**
2. **Removing NAN values from Current Version**
3. **Removing NAN values from Android Version**

As continuing Data Cleaning Process changing the type of Reviews to Int type

```
[ ]   df_app['Reviews'].describe()

count     10829
unique     5999
top           0
freq        594
Name: Reviews, dtype: object
```

```
[ ]   #Lets change the type of the Reviews.Int type is the best option.

      df_app['Reviews'] = df_app['Reviews'].astype('int')
      df_app['Reviews'].head()

0       159
1       967
2     87510
3    215644
4       967
Name: Reviews, dtype: int64
```

Changing ‘dtype : object’ to ‘dtype : int64

Data Manipulation
on
‘ SIZE ‘

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

```
array(['19M', '14M', '8.7M', '25M', '2.8M', '5.6M', '29M', '33M', '3.1M',
       '28M', '12M', '20M', '21M', '37M', '5.5M', '17M', '39M', '31M',
       '4.2M', '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', '2.7M', '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', '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',
```

In ‘SIZE’ the following Data Manipulation done

1. **Replace ‘ , ‘ from Size.**
2. **Convert Data type from ‘Object’ to ‘Float’**
3. **Convert strings of different MB & KB to float type MB**

Using the following function we did Data Manipulation on ‘ SIZE ‘

```
df_app['Size'] = df_app['Size'].apply(lambda x: str(x).replace('M', '') if 'M' in str(x) else x)
df_app['Size'] = df_app['Size'].apply(lambda x: str(x).replace(',', '') if ',' in str(x) else x)
df_app['Size'] = df_app['Size'].apply(lambda x: float(str(x).replace('k', '')) / 1024 if 'k' in str(x) else x) # 1 MB is equal to 1024 KB
df_app['Size'] = df_app['Size'].apply(lambda x: str(x).replace('+','') if '+' in str(x) else x)
df_app['Size'] = df_app['Size'].astype('float')
```

By using these line of code we did the Data Manipulation

Data Manipulation
on
‘ PRICE ‘

```
df_app['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', '$9.00', '$5.49',
       '$10.00', '$24.99', '$11.99', '$79.99', '$16.99', '$14.99',
       '$1.00', '$29.99', '$12.99', '$2.49', '$10.99', '$1.50', '$19.99',
       '$15.99', '$33.99', '$74.99', '$39.99', '$3.95', '$4.49', '$1.70',
       '$8.99', '$2.00', '$3.88', '$25.99', '$399.99', '$17.99',
       '$400.00', '$3.02', '$1.76', '$4.84', '$4.77', '$1.61', '$2.50',
       '$1.59', '$6.49', '$1.29', '$5.00', '$13.99', '$299.99', '$379.99',
       '$37.99', '$18.99', '$389.99', '$19.90', '$8.49', '$1.75',
       '$14.00', '$4.85', '$46.99', '$109.99', '$154.99', '$3.08',
       '$2.59', '$4.80', '$1.96', '$19.40', '$3.90', '$4.59', '$15.46',
       '$3.04', '$4.29', '$2.60', '$3.28', '$4.60', '$28.99', '$2.95',
       '$2.90', '$1.97', '$200.00', '$89.99', '$2.56', '$30.99', '$3.61',
       '$394.99', '$1.26', '$1.20', '$1.04'], dtype=object)
```

In ‘PRICE’ the following Data Manipulation done

→ There are $ symbols in the price column and those must be removed

Using the following function we did Data Manipulation on ‘ SIZE ‘

```
df_app['Price'] = df_app['Price'].apply(lambda x: str(x).replace('$','') if '$' in str(x) else x)
df_app['Price'] = df_app['Price'].apply(lambda x: str(x).replace(' ','') if ' ' in str(x) else x)
df_app['Price'] = df_app['Price'].astype('float')
```

After doing Data Manipulation the result seems like this

```
[ ]   df_app['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])
```

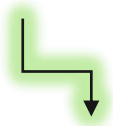
AI

Using the following function we did Data Manipulation on ‘ Installs‘

```
df_app['Installs'] = df_app['Installs'].apply(lambda x: str(x).replace('+', '') if '+' in str(x) else x)
df_app['Installs'] = df_app['Installs'].apply(lambda x: str(x).replace(',', '') if ',' in str(x) else x)
df_app['Installs'] = df_app['Installs'].apply(lambda x: float(x))
```

While studying data we just came across that there are some apps with duplicate values

Let’s check for app that called ‘ROBLOX’



```
df_app[df_app['App'] == 'ROBLOX']
```

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver
1653	ROBLOX	GAME	4.5	4447388	67.0	100000000.0	Free	0.0	Everyone 10+	Adventure;Action & Adventure	July 31, 2018	2.347.225742	4.1 and up
1701	ROBLOX	GAME	4.5	4447346	67.0	100000000.0	Free	0.0	Everyone 10+	Adventure;Action & Adventure	July 31, 2018	2.347.225742	4.1 and up
1748	ROBLOX	GAME	4.5	4448791	67.0	100000000.0	Free	0.0	Everyone 10+	Adventure;Action & Adventure	July 31, 2018	2.347.225742	4.1 and up
1841	ROBLOX	GAME	4.5	4449882	67.0	100000000.0	Free	0.0	Everyone 10+	Adventure;Action & Adventure	July 31, 2018	2.347.225742	4.1 and up
1870	ROBLOX	GAME	4.5	4449910	67.0	100000000.0	Free	0.0	Everyone 10+	Adventure;Action & Adventure	July 31, 2018	2.347.225742	4.1 and up
2016	ROBLOX	FAMILY	4.5	4449910	67.0	100000000.0	Free	0.0	Everyone 10+	Adventure;Action & Adventure	July 31, 2018	2.347.225742	4.1 and up
2088	ROBLOX	FAMILY	4.5	4450855	67.0	100000000.0	Free	0.0	Everyone 10+	Adventure;Action & Adventure	July 31, 2018	2.347.225742	4.1 and up
2206	ROBLOX	FAMILY	4.5	4450890	67.0	100000000.0	Free	0.0	Everyone 10+	Adventure;Action & Adventure	July 31, 2018	2.347.225742	4.1 and up
4527	ROBLOX	FAMILY	4.5	4443407	67.0	100000000.0	Free	0.0	Everyone 10+	Adventure;Action & Adventure	July 31, 2018	2.347.225742	4.1 and up

As we can see there are more Duplicate values for ‘ ROBLOX ‘

So we decide to keep one row from each category and drop the rest

```
df_app = df_app.drop_duplicates(['App','Category'])
df_app['App'].value_counts()
```

```
Netflix                                          2
LEGO® TV                                         2
DC Super Hero Girls™                             2
Princess Coloring Book                           2
Chess Free                                       2
                                                ..
F-Sim Space Shuttle                              1
Weather Live                                     1
Norwegian For Kids & Babies F                    1
R+F PULSE                                        1
iHoroscope - 2018 Daily Horoscope & Astrology    1
Name: App, Length: 9648, dtype: int64
```

Now let's check for app like ' NETFLIX '

```
df_app[df_app['App'] == 'Netflix']
```

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver
855	Netflix	ENTERTAINMENT	4.4	5456208	NaN	100000000.0	Free	0.0	Teen	Entertainment	July 31, 2018	Varies with device	Varies with device
3889	Netflix	FAMILY	4.4	5453997	NaN	100000000.0	Free	0.0	Teen	Entertainment	July 31, 2018	Varies with device	Varies with device

Now it just show only 2 values

While reviewing Review.csv we came across that there are several NAN values

We just took random section to see that whether there are NAN values or not

```
[ ]  df_review[1000:1010]
```

	App	Translated_Review	Sentiment	Sentiment_Polarity	Sentiment_Subjectivity
1000	4K Wallpapers and Ultra HD Backgrounds	NaN	NaN	NaN	NaN
1001	4K Wallpapers and Ultra HD Backgrounds	NaN	NaN	NaN	NaN
1002	4K Wallpapers and Ultra HD Backgrounds	NaN	NaN	NaN	NaN
1003	4K Wallpapers and Ultra HD Backgrounds	NaN	NaN	NaN	NaN
1004	4K Wallpapers and Ultra HD Backgrounds	NaN	NaN	NaN	NaN
1005	4K Wallpapers and Ultra HD Backgrounds	NaN	NaN	NaN	NaN
1006	4K Wallpapers and Ultra HD Backgrounds	NaN	NaN	NaN	NaN
1007	4K Wallpapers and Ultra HD Backgrounds	NaN	NaN	NaN	NaN
1008	4K Wallpapers and Ultra HD Backgrounds	Superb love wallpapers give confidence wheneve...	Positive	0.75	0.8
1009	4K Wallpapers and Ultra HD Backgrounds	NaN	NaN	NaN	NaN

And yes we found that yes there are really NAN values

We decide to drop all the NAN values for review.csv

```
[ ]    df_review.dropna(inplace = True)
```

```
[ ]    df_review.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 37427 entries, 0 to 64230
Data columns (total 5 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   App                     37427 non-null  object 
 1   Translated_Review       37427 non-null  object 
 2   Sentiment               37427 non-null  object 
 3   Sentiment_Polarity      37427 non-null  float64
 4   Sentiment_Subjectivity  37427 non-null  float64
dtypes: float64(2), object(3)
memory usage: 1.7+ MB
```

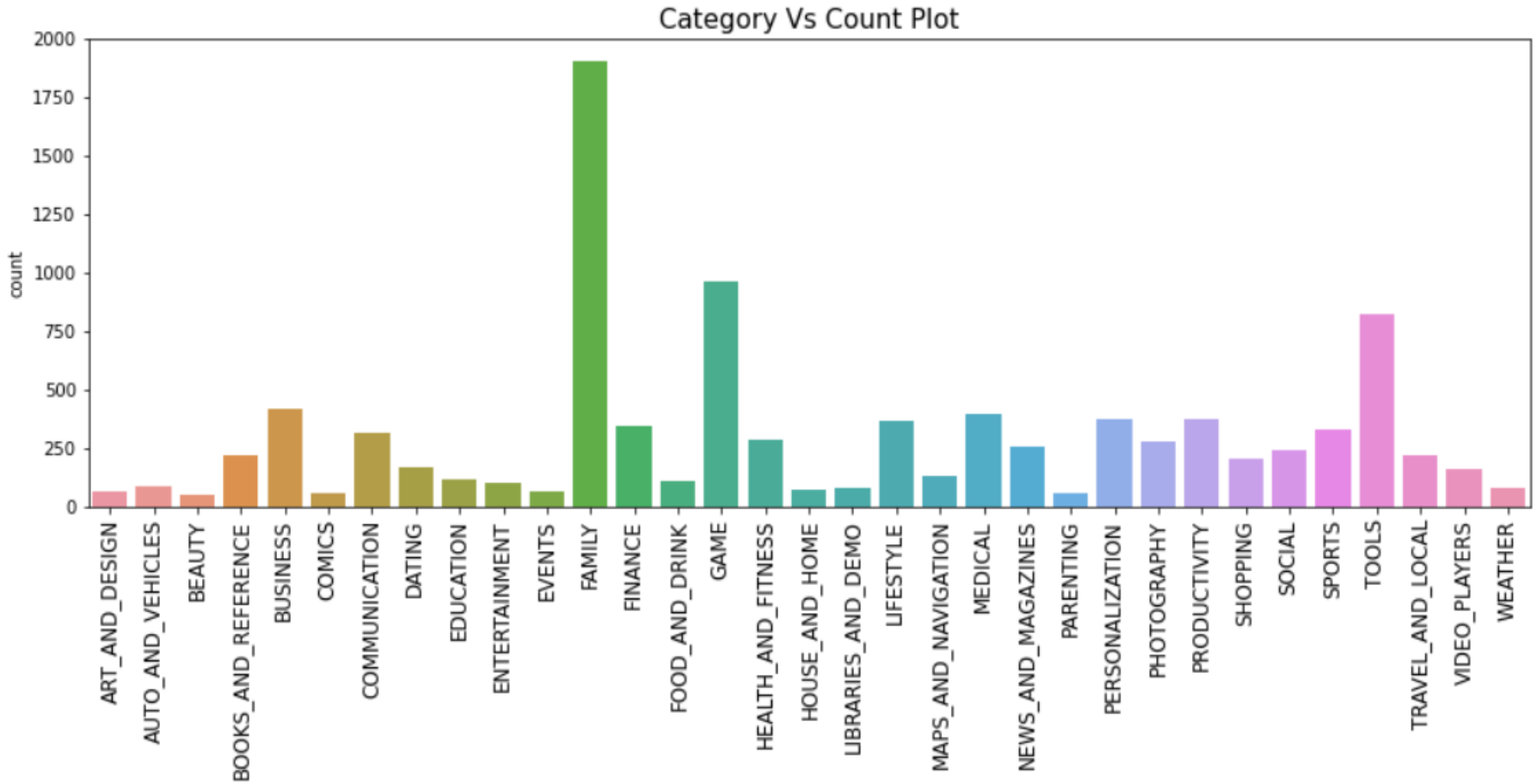
Let's combine the two data sets on the "App" column

```
merged_df = pd.merge(df_review,df_app, on='App', how='outer')
merged_df.head()
```

	App	Translated_Review	Sentiment	Sentiment_Polarity	Sentiment_Subjectivity	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver
0	10 Best Foods for You	I like eat delicious food. That's I'm cooking ...	Positive	1.00	0.533333	HEALTH_AND_FITNESS	4.0	2490.0	3.8	500000.0	Free	0.0	Everyone 10+	Health & Fitness	February 17, 2017	1.9	2.3.3 and up
1	10 Best Foods for You	This help eating healthy exercise regular basis	Positive	0.25	0.288462	HEALTH_AND_FITNESS	4.0	2490.0	3.8	500000.0	Free	0.0	Everyone 10+	Health & Fitness	February 17, 2017	1.9	2.3.3 and up
2	10 Best Foods for You	Works great especially going grocery store	Positive	0.40	0.875000	HEALTH_AND_FITNESS	4.0	2490.0	3.8	500000.0	Free	0.0	Everyone 10+	Health & Fitness	February 17, 2017	1.9	2.3.3 and up
3	10 Best Foods for You	Best idea us	Positive	1.00	0.300000	HEALTH_AND_FITNESS	4.0	2490.0	3.8	500000.0	Free	0.0	Everyone 10+	Health & Fitness	February 17, 2017	1.9	2.3.3 and up
4	10 Best Foods for You	Best way	Positive	1.00	0.300000	HEALTH_AND_FITNESS	4.0	2490.0	3.8	500000.0	Free	0.0	Everyone 10+	Health & Fitness	February 17, 2017	1.9	2.3.3 and up

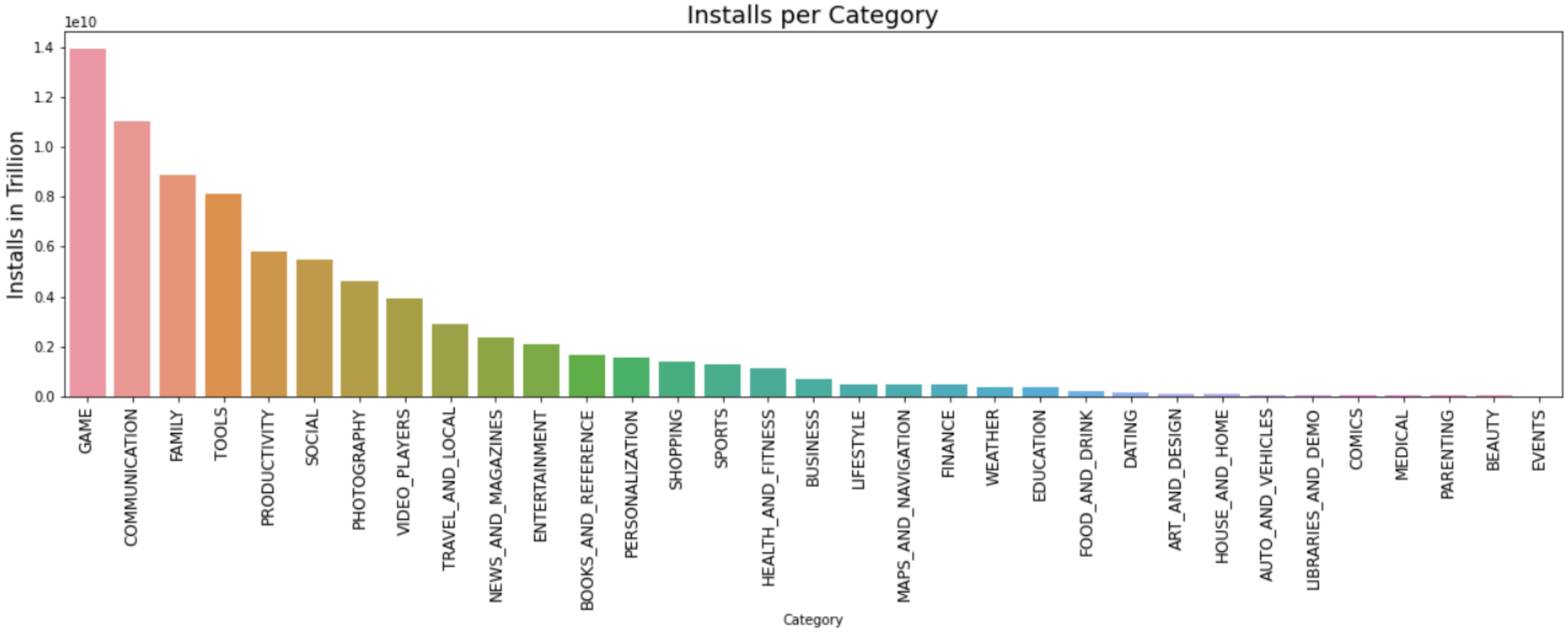
Data Visualization

Category that has the MAX number of Apps in the Play store



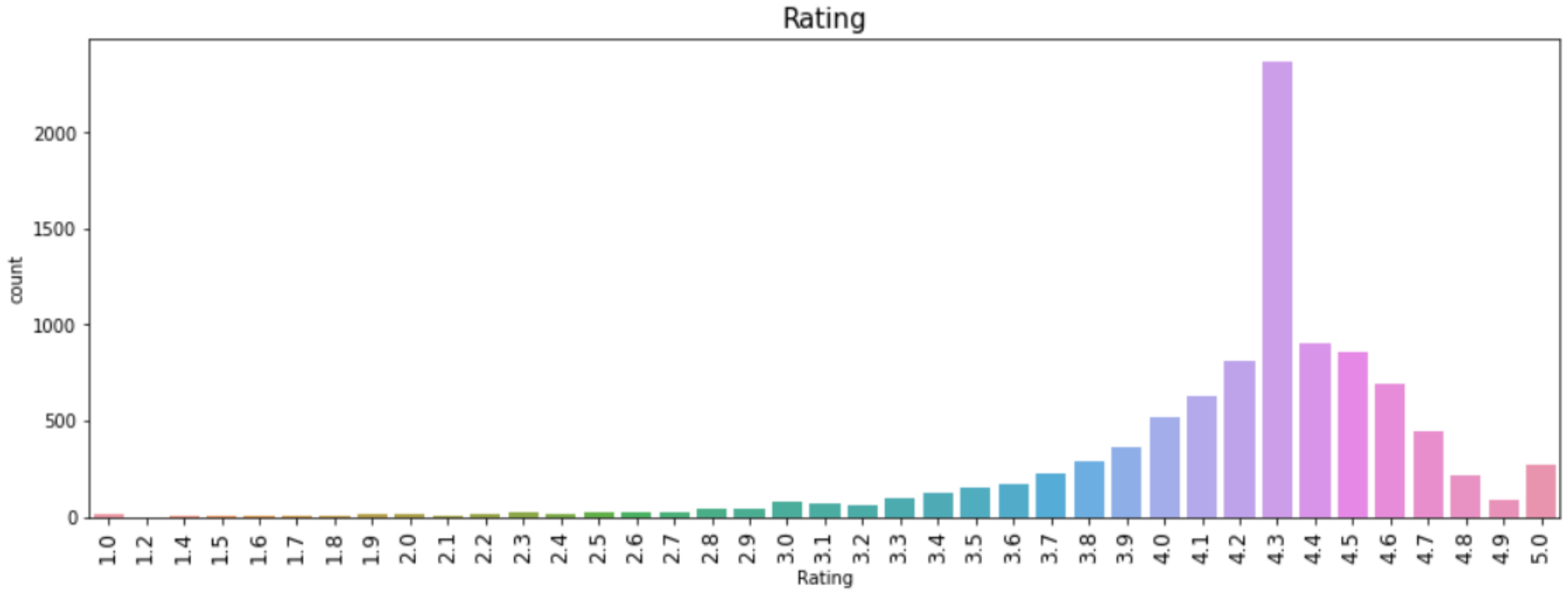
The category 'Family' has the most number of apps available followed by 'Games' and 'Tools' category

Now let's check the category that has highest installs



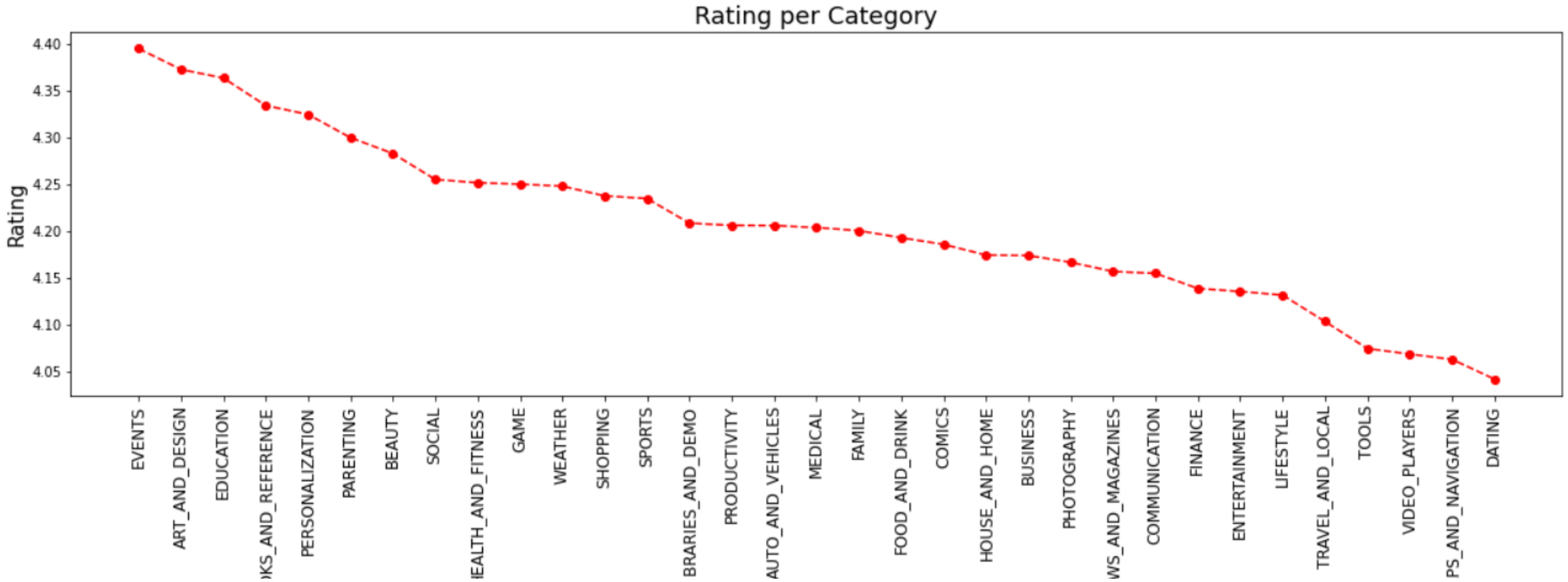
The highest installed category of Play store apps are GAME followed by COMMUNICATION, FAMILY and TOOLS respectively

Now the next is count of Rating



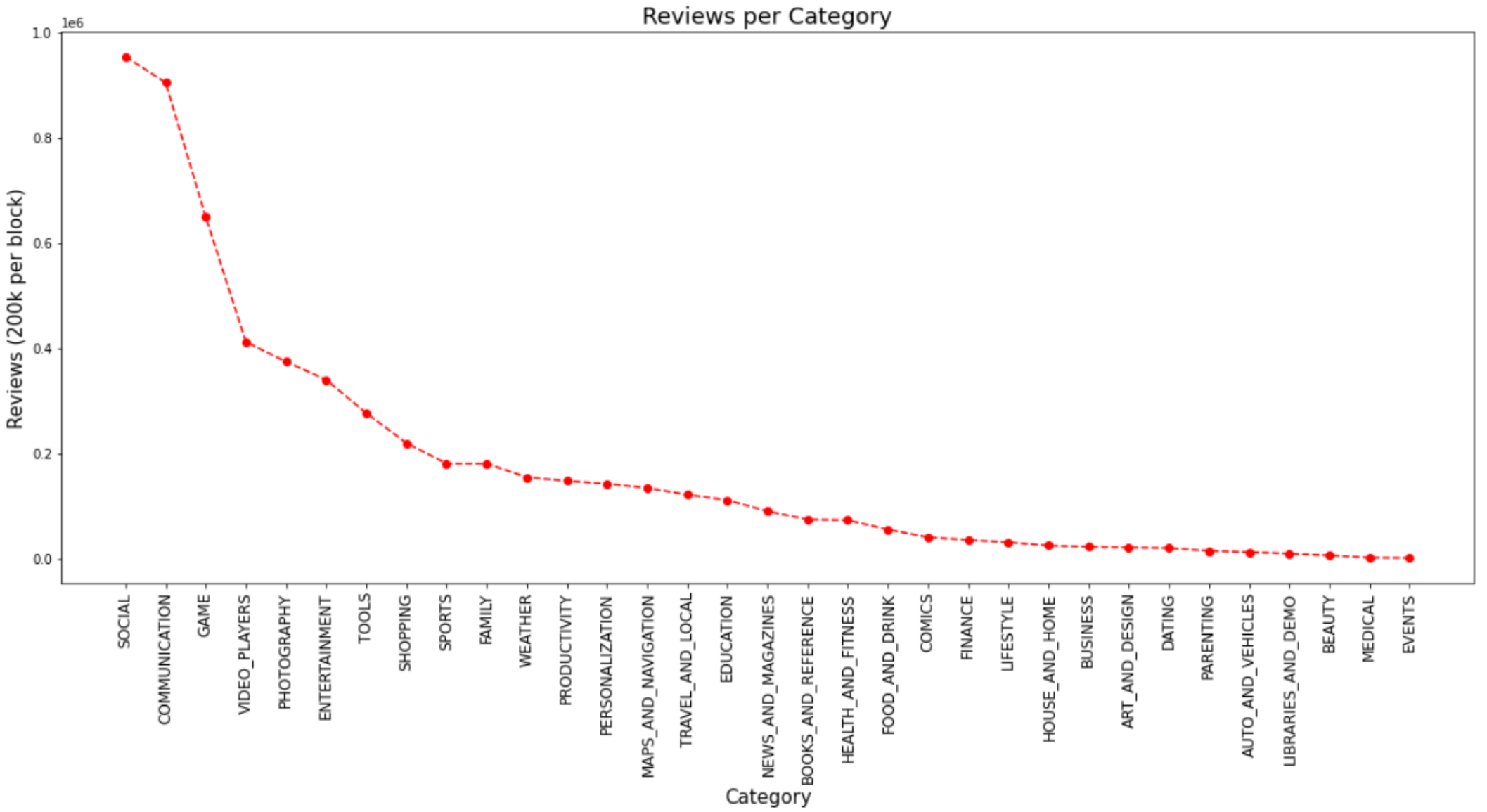
We can see that 4.3 rating has more count because we replace the NaN values in the 'RATING' column with MEDAIN values during our Data cleaning process.

Now we are looking for Rating Per Category



The category of EVENTS has the Highest Average Rating of 4.4 and the Category DATING has the lowest average ratio of 4.04

Reviews Per Category



The category of SOCIAL has the Highest no. of reviews of 953.67k and the Category EVENT has the lowest no. of reviews of 2.52k

Now we are checking the top 10 apps which has been reviewed most

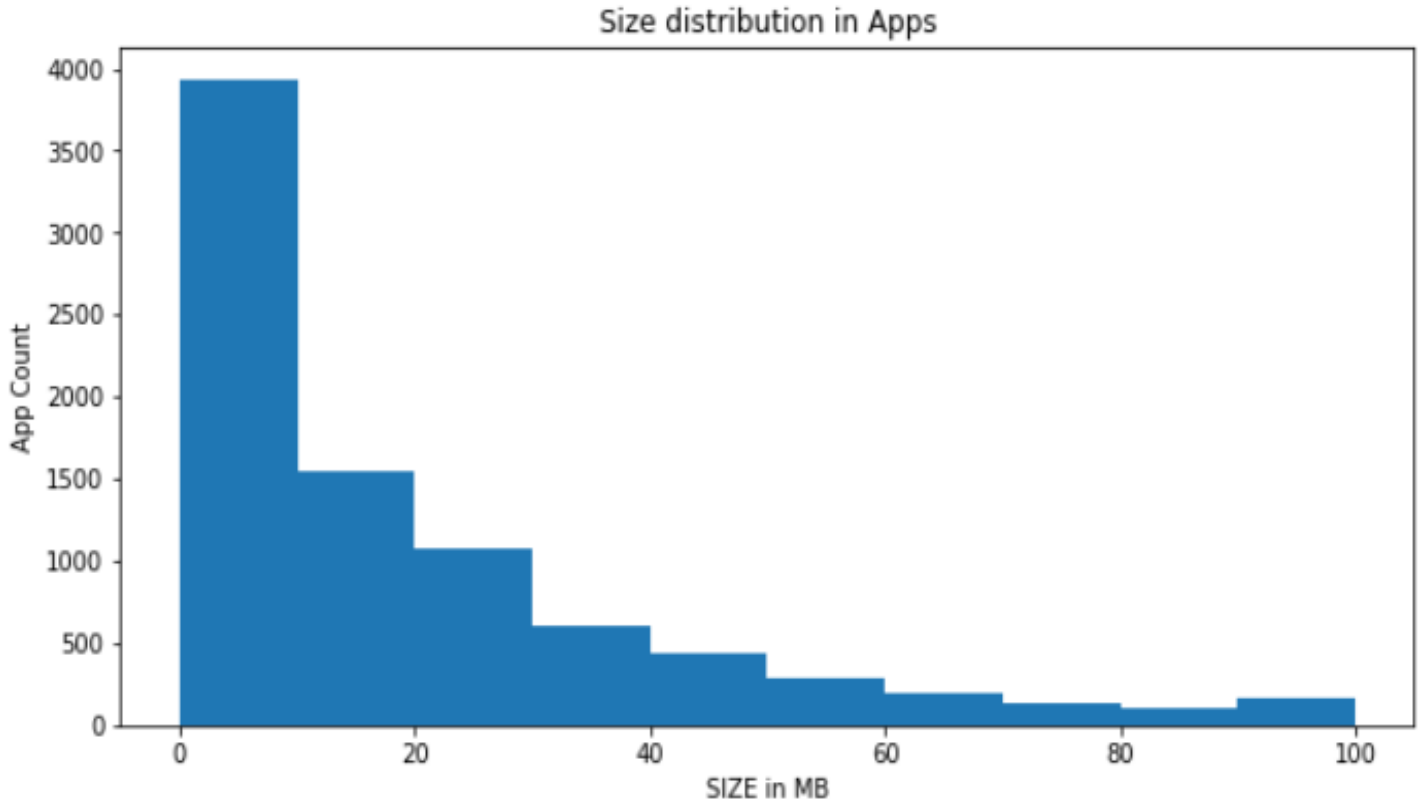
```
[ ]   #Let's check the Category of the Apps:
      df_app[['Reviews', 'App','Category']].sort_values('Reviews', ascending=False).head(10)
```

	Reviews	App	Category
2544	78158306	Facebook	SOCIAL
336	69119316	WhatsApp Messenger	COMMUNICATION
2545	66577313	Instagram	SOCIAL
335	56642847	Messenger – Text and Video Chat for Free	COMMUNICATION
1670	44891723	Clash of Clans	GAME
3986	44881447	Clash of Clans	FAMILY
4005	42916526	Clean Master- Space Cleaner & Antivirus	TOOLS
1654	27722264	Subway Surfers	GAME
3665	25655305	YouTube	VIDEO_PLAYERS
7536	24900999	Security Master - Antivirus, VPN, AppLock, Boo...	TOOLS

The top four reviewed apps belongs to the category of Social Media and Comunication.

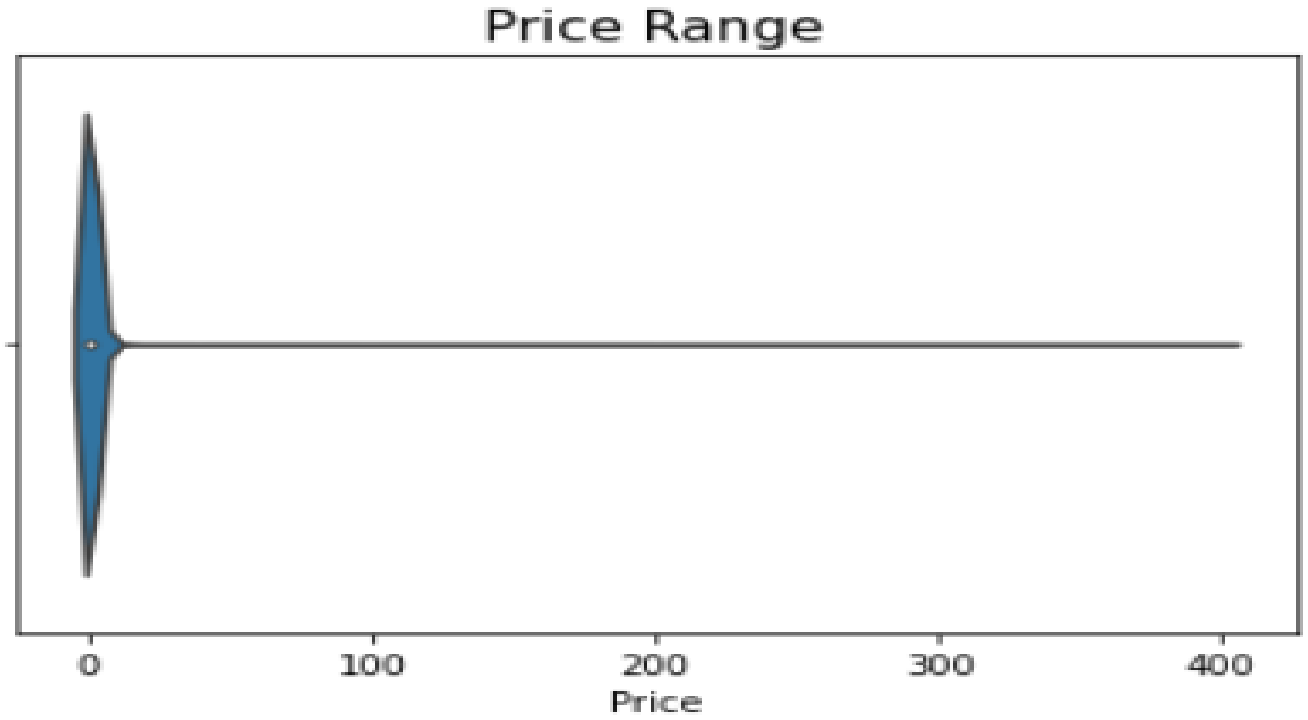
Highest app is Facebook with 78158306 reviews

Now we are analyzing the average size of the apps (in MB)



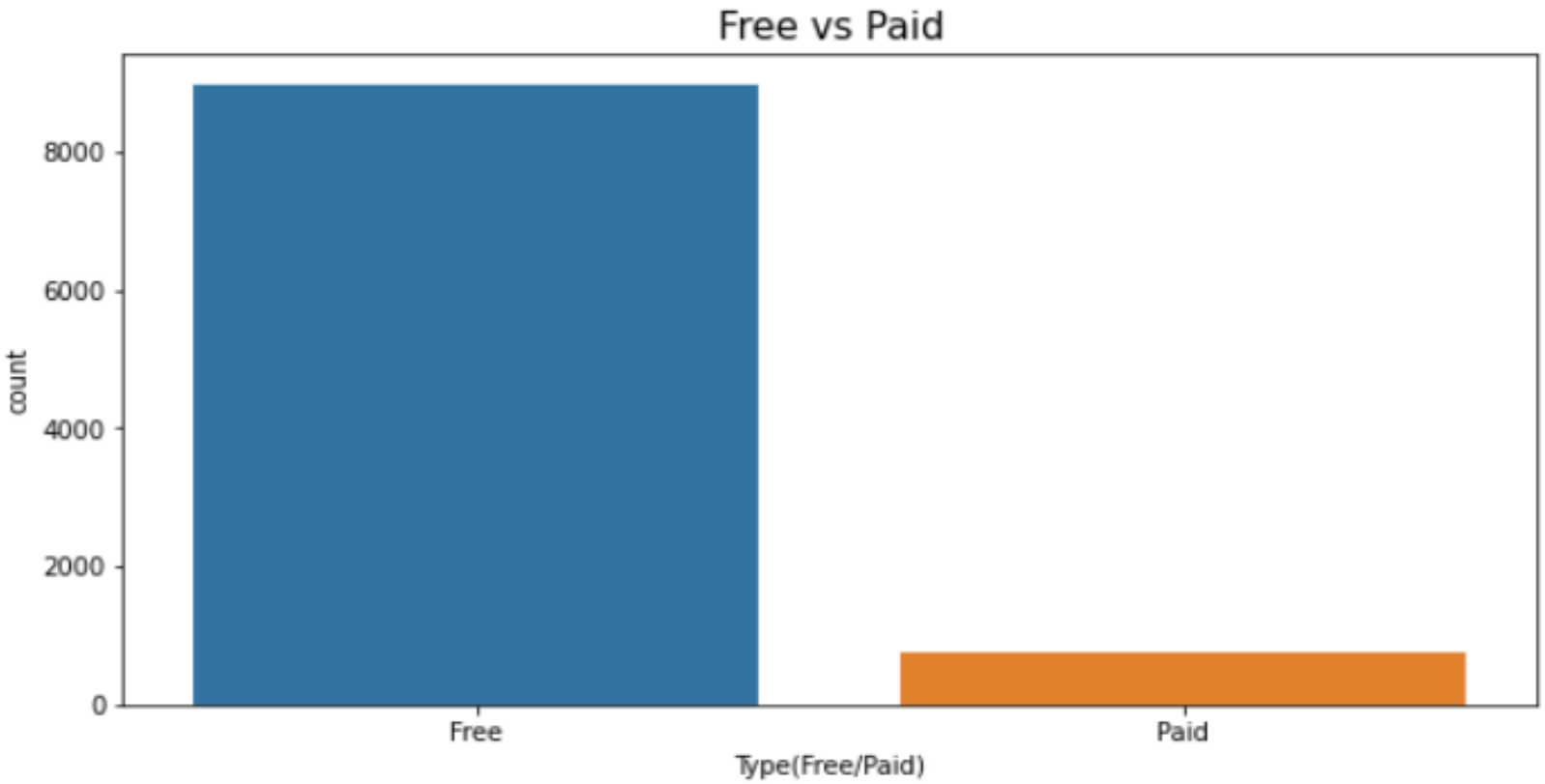
Most of the app have the size between 0MB - 40MB in the play store.
The average size of apps in Play store is 20.54.
72.92% of the apps are smaller in size than 40MB.
54.99% of the apps are smaller in size than 20MB.

Now we are analyzing the average price of the apps



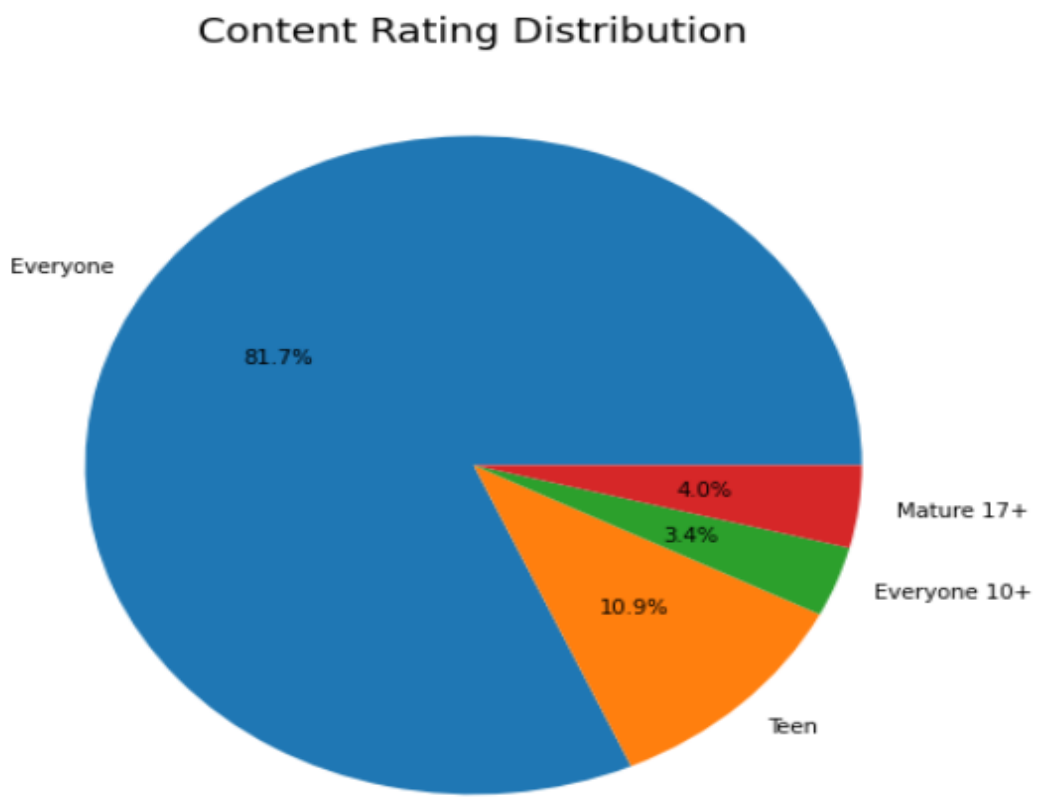
Most of the price is between 0-5 USD but it seems like some apps have a price as High as 400 USD , lets compare the amount of free apps vs amount of paid apps

Now here we are comparing Free apps VS Paid apps



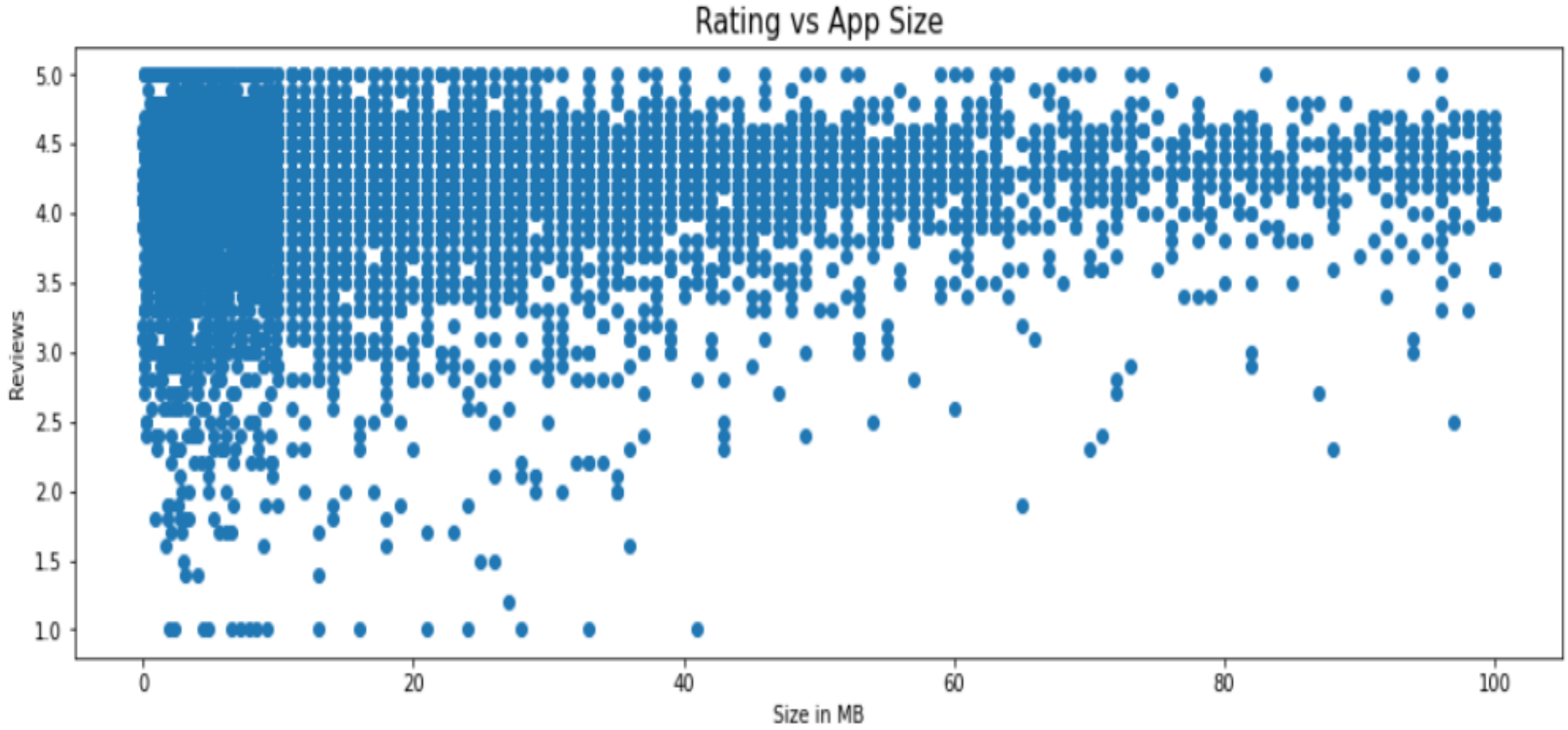
Around 92.24% apps in the Play store is free
Only 1.65% app are 5$ or more
I'm Rich - Trump Edition is by Far the most expensive app which costs around 400$

Analyzing the Content Rating across all apps



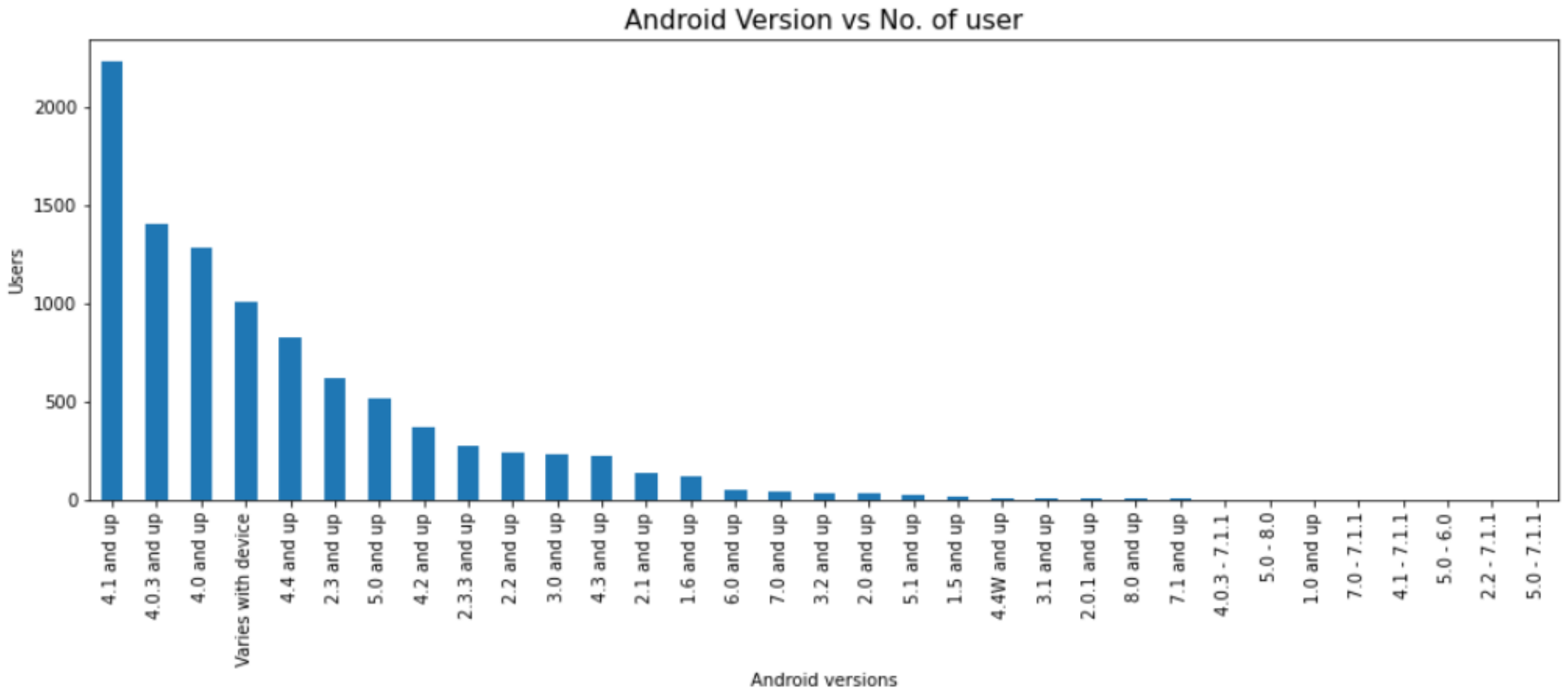
Most of the content is for everyone!

Analysis of Rating VS App size



It seems like Ratings slightly improve with the increase in Size!

Analysis of Android version VS number of users



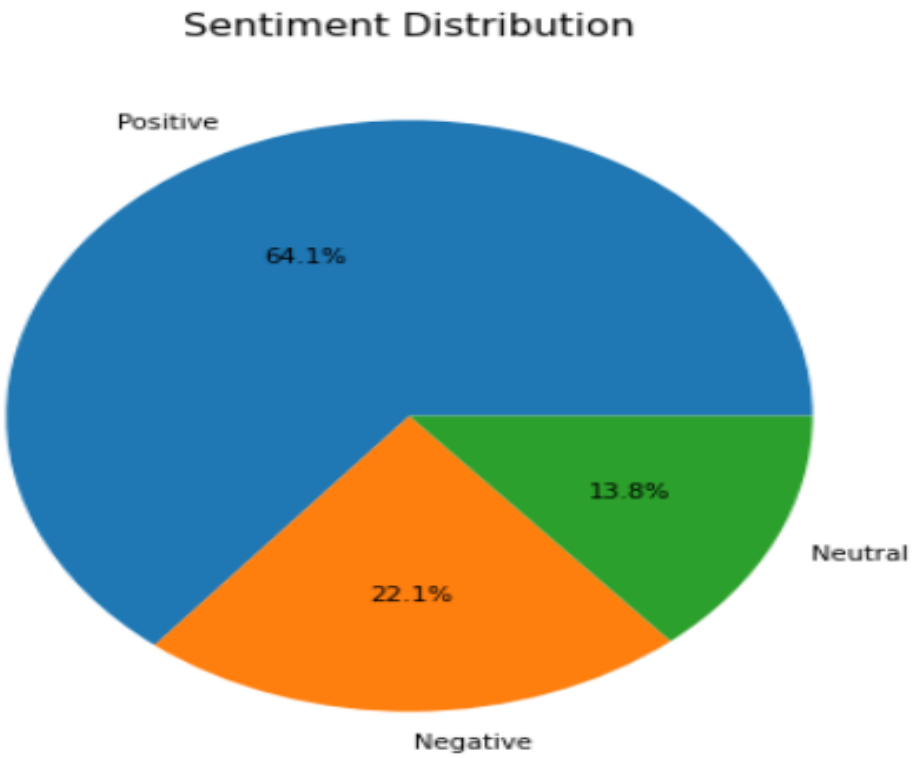
The top three Android Versions used today are:

4.1 and up 2234

4.0.3 and up 1404

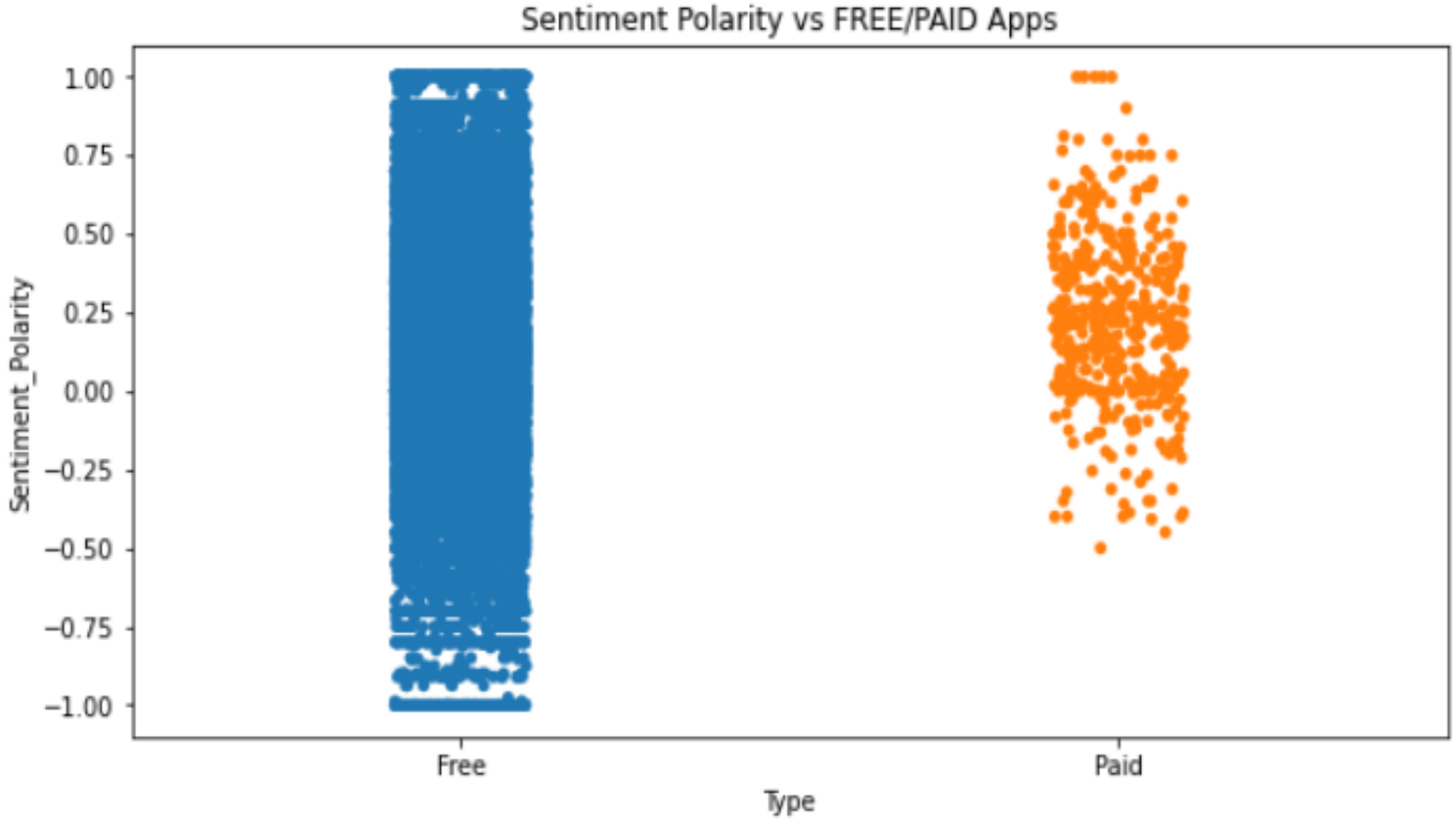
4.0 and up 1288

Analysis of user sentiments



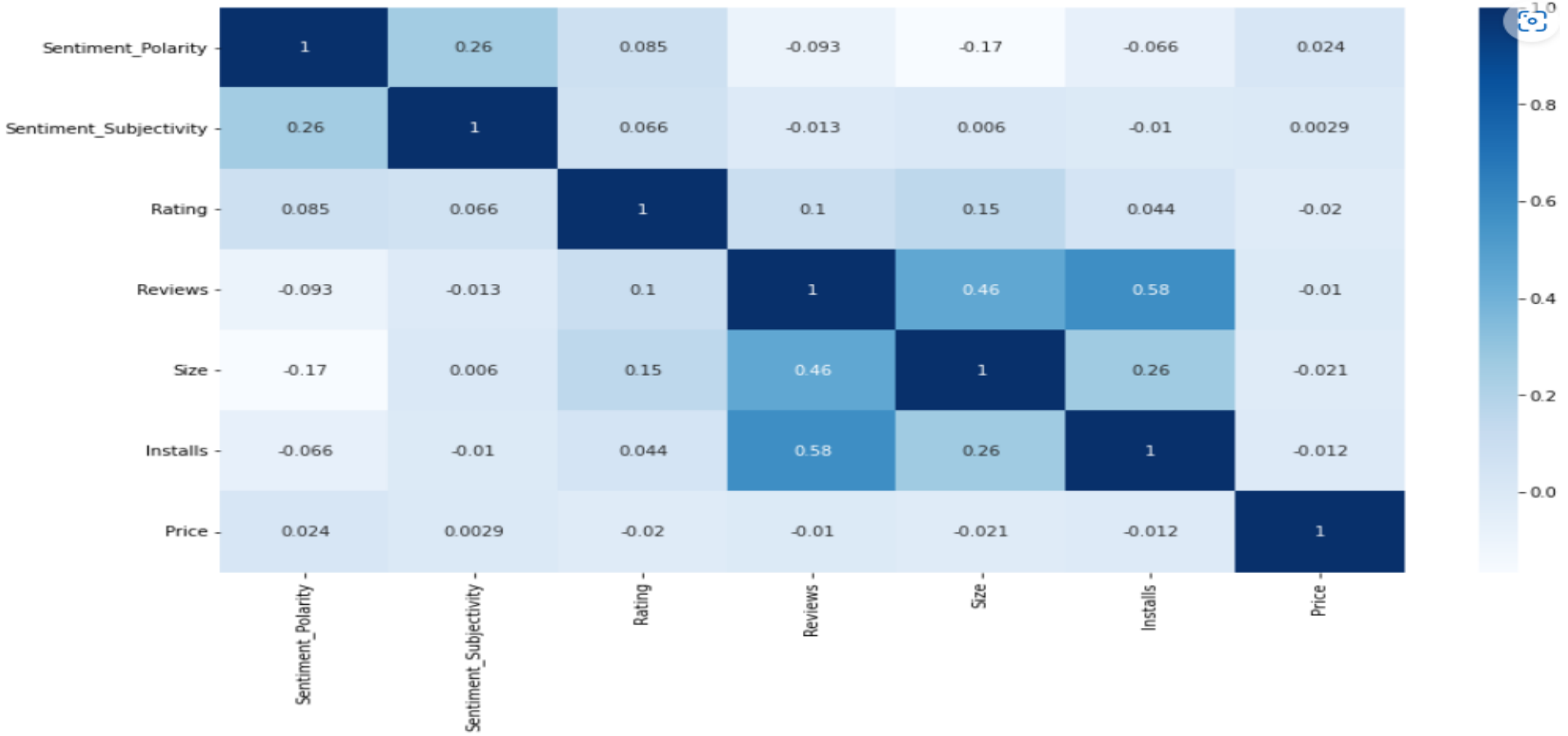
- We can se that 64% of the expressed sentiments are positive
- 22% is Negative and 14% is Neutral
- What about the type vs sentiment polarity let's see that!

Let's Check out the sentiment polarity when the app is paid vs the app is free



Here we can clearly see that paid apps are likely to get better review than free apps The sentiment polarity is uniformly distributed in free apps where as in case of paid apps it lingers mostly in the positive side

Correlation Graph



1. Installs and Reviews have a High positive Correlation
2. Price is Negatively Correlated with every other variable
3. Size has a slight positive correlation with Installs

Conclusion

The dataset contains possibilities to deliver insights to understand customer demands better and thus help developers to popularize the product. With the following dataset we have concluded that the developer must keep the following things in mind:

- Free apps have a greater number of installs as compared to paid apps.
- But Paid Apps are likely to get better reviews than free apps
- User sentiments are 64% positive, 22% Negative and 14% Neutral.
- If the size of the app increases, ratings tend to improve
- Users like to use the apps with android version 4.1 or above.
- Size of the app should be small preferably between 0mb-40mb.
- Highest installed category of play store apps is GAME, it is a high demanding category for developing an app followed by COMMUNICATIONS category.
- Size is Negatively correlated with Sentiment Polarity

THANK YOU!