Abstract:

Google Play Store, which is operated and developed by Google, is a digital sharing service that serves as official app store for the certified Android operating system users. It allows user to download and browse various application along with digital media. Google play store become one of the leading application for android over last 12 years which motivates me for further investigate. In this project I aimed to perform detail exploratory analysis on 'googleplaystore' data set to predict different trends (app category/review/rating/installs/sales) of consumer market. I am also intend to find co-related apps those has most demand. Based on the exploratory analysis, this project aims to visualize forecast past selling trend. In conclusion, I could able to anticipate what type production strategies Google has followed to reach current position and weather there is any noticeable trend from audience within this timeline.

Table of Content:Content Page

List of Figures

Appendix A

Appendix B

Appendix C

1.0 Introduction:

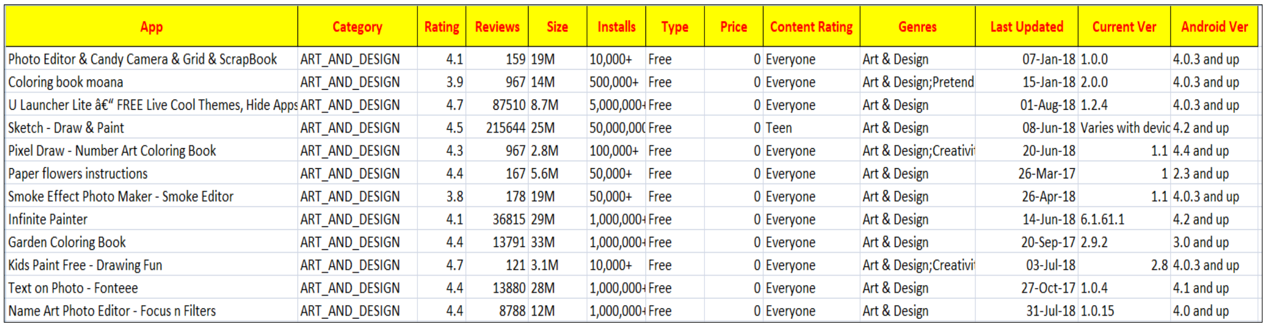
Online retailers like Google play store can increase their sales and profits faster than a brick and mortar shop as selling online offers the advantage of having an open store, twenty-four hours a day, and seven days a week. This advantage allows online retailers to expand their market to global proportions or target an extremely focused segment. It is important to set a dynamic business strategies focused on customer buying attitude along with sales trend to reach maximum profit margin.

In terms of research googleplaystore dataset, the scope is actually remarkable as all the categorical apps have record contains their total installions, reviews, rating, prices distinctly. And, this research was conducted for simulation and evaluation with well known real world android/pc application.

Challenging part will be interpret business financial trend as the resource is limited and dataset is mainly based on customer operated attributes. My plan is to cover categorical apps evaluation in terms audiences, sales, timeframe with interactive approach.

1.1 Dataset Details:

The ‘googleplaystore’ data set (available at https://www.kaggle.com/lava18/google-play-store-apps) contains summarize information about google apps consumer market circumstances.



App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver
Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19M	10,000+	Free	0	Everyone	Art & Design	07-Jan-18	1.0.0	4.0.3 and up
Coloring book moana	ART_AND_DESIGN	3.9	967	14M	500,000+	Free	0	Everyone	Art & Design;Pretend	15-Jan-18	2.0.0	4.0.3 and up
U Launcher Lite â€“ FREE Live Cool Themes, Hide Apps	ART_AND_DESIGN	4.7	87510	8.7M	5,000,000+	Free	0	Everyone	Art & Design	01-Aug-18	1.2.4	4.0.3 and up
Sketch - Draw & Paint	ART_AND_DESIGN	4.5	215644	25M	50,000,000	Free	0	Teen	Art & Design	08-Jun-18	Varies with devic	4.2 and up
Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	967	2.8M	100,000+	Free	0	Everyone	Art & Design;Creativi	20-Jun-18	1.1	4.4 and up
Paper flowers instructions	ART_AND_DESIGN	4.4	167	5.6M	50,000+	Free	0	Everyone	Art & Design	26-Mar-17	1	2.3 and up
Smoke Effect Photo Maker - Smoke Editor	ART_AND_DESIGN	3.8	178	19M	50,000+	Free	0	Everyone	Art & Design	26-Apr-18	1.1	4.0.3 and up
Infinite Painter	ART_AND_DESIGN	4.1	36815	29M	1,000,000+	Free	0	Everyone	Art & Design	14-Jun-18	6.1.61.1	4.2 and up
Garden Coloring Book	ART_AND_DESIGN	4.4	13791	33M	1,000,000+	Free	0	Everyone	Art & Design	20-Sep-17	2.9.2	3.0 and up
Kids Paint Free - Drawing Fun	ART_AND_DESIGN	4.7	121	3.1M	10,000+	Free	0	Everyone	Art & Design;Creativi	03-Jul-18	2.8	4.0.3 and up
Text on Photo - Fonteee	ART_AND_DESIGN	4.4	13880	28M	1,000,000+	Free	0	Everyone	Art & Design	27-Oct-17	1.0.4	4.1 and up
Name Art Photo Editor - Focus n Filters	ART_AND_DESIGN	4.4	8788	12M	1,000,000+	Free	0	Everyone	Art & Design	31-Jul-18	1.0.15	4.0 and up

Figure 1: Dataset from googleplaystore

1.2 Attribute Information:

- App: String, Uniquely assigned to each record.
- Category: String, consumer targeted area.
- Rating: Float value. Each value is mean value of total rating for that specific app.
- Reviews: Integer Value. Contain total customer reviews for any particular app.
- Size: String, float value ends with ‘M’ where as it depicts counts in million.
- Installs: String, Integer value ends with ‘+’ character to describe greater than range.
- Type:String. Describe free or paid app.
- Price: String. ‘$’ has used for price.
- Content Rating: String
- Genres: String
- Last Updated: String. Date value have month symbol.
- Current Ver: String. Current version.
- Android ver: String. Android version.

1.3 Problem Statement:

Research area will cover with following problems and the platform will be **Python** and **Tableau**.

- What is the most popular Categorical Apps?
- Which group of customers are most profitable for Google App Store business?
- Does Google App Store business invest recently?

2.1 Explanatory Data Analysis:

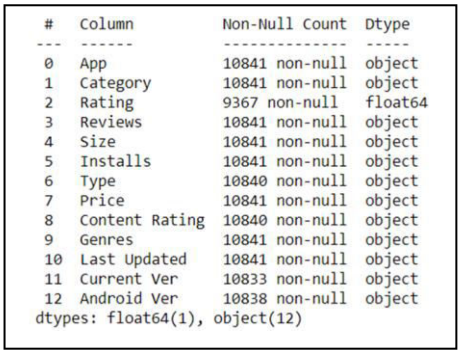
2.1.1 Data Cleaning:

I have loaded the dataset after download. The noticeable case in first Look was that, there was only one numeric float value presented in dataset named ‘Rating’, though my expectation was to find out atleast 4 to 5 numeric values from dataset. As Reviews, installations, Price should be numeric.

Secondly I plotted barplot of dataset that how the numeric value distributed. There an outlier was detected. So I have filtered ‘Rating’ by condition in range of ‘0’ to ‘5’ and drop wrong row.

After dropping, I plotted the boxplot again and find that the data mostly concentrated from ‘4’ to ‘4.5’.

In the mean time, histogram was shown ambiguous visibility at first, but later it describe corresponding distribution of ‘Rating’ as well



```
 #   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)
```

Figure 2: Dataset details

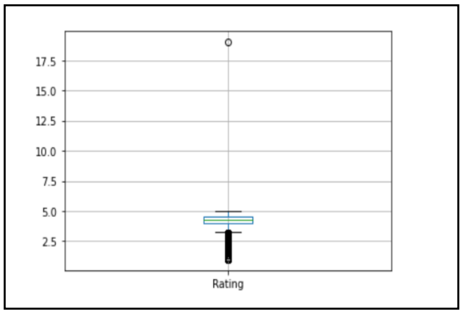


Figure 3: Initial Boxplot

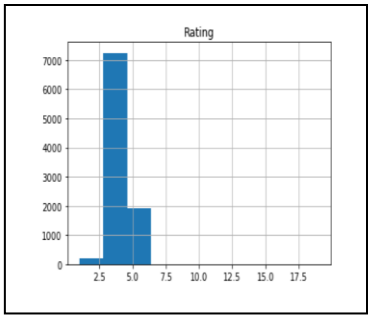


Figure 4: Initial Histogram

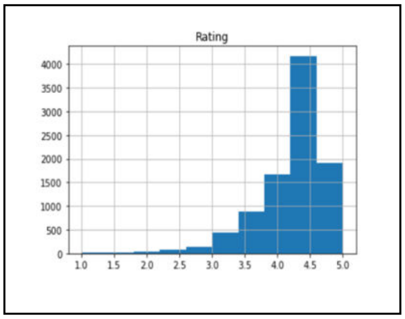


Figure 5: Histogram after cleaning

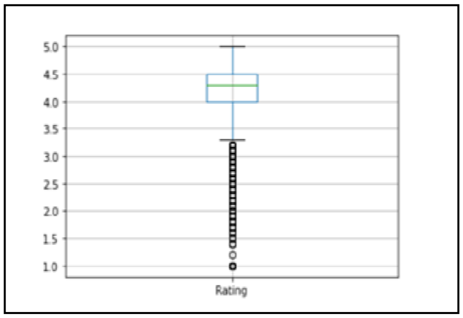


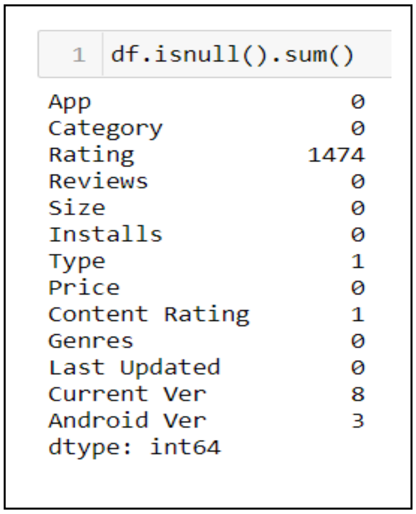
Figure 6: Boxplot after cleaning

2.1.2 Data Imputation and Manipulation:

At first I checked sum of null values inside the data set, then I got ‘Rating’, ‘Type’, ‘Android Ver’ and ‘Current Ver’ attributes contain null values.

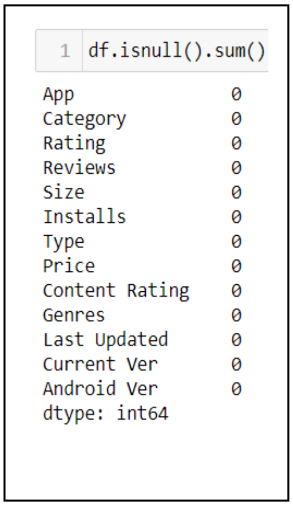
I fill up null values in ‘Rating’ by its median value and for the other categorical attributes I used their mode value for fill up.

Then I got a dataset full of values at the end.



```
df.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
```

Figure 7: Initial null values

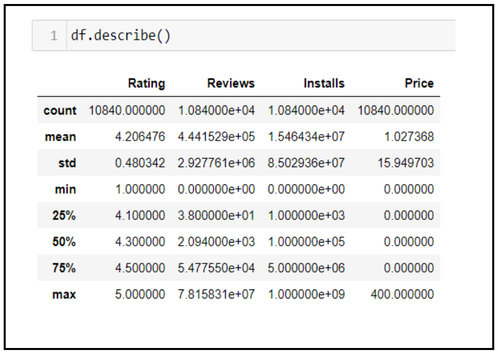


```
df.isnull().sum()

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
```

Figure 8: Null values after cleaning

2.1.3 Categorical to Numeric Conversion:

I changed 'Price' from categorical to numeric float value with replacement of '$' and ',' by ''. Similarly for 'Installations' I deleted unwanted character '+' and ',' and changed to numeric as well. I did same operation on 'Review' attribute also. Moreover, when I checked for unique value in attributes then 'Type' column shows wrong entry and replaced by its correct classification. After cleaning I created new dataset named 'googleplaystore_clean.csv' for future research.



```
df.describe()
```

	Rating	Reviews	Installs	Price
count	10840.000000	1.084000e+04	1.084000e+04	10840.000000
mean	4.206476	4.441529e+05	1.546434e+07	1.027368
std	0.480342	2.927761e+06	8.502936e+07	15.949703
min	1.000000	0.000000e+00	0.000000e+00	0.000000
25%	4.100000	3.800000e+01	1.000000e+03	0.000000
50%	4.300000	2.094000e+03	1.000000e+05	0.000000
75%	4.500000	5.477550e+04	5.000000e+06	0.000000
max	5.000000	7.815831e+07	1.000000e+09	400.000000

Figure 9: Numerical values in dataset

2.1.4 Numerical Analysis:

As I got 4 numeric attributes so for further analysis a conduction has done for correlation analysis. All the attributes values has scaled with 'minmaxscaler' and plotted correlation matrix.

Here a positive correlation has detected between 'Reviews' and 'Installs' .

For better understanding of these correlated data a pair plot has created with classification of 'Type' which are 'Free' and 'Paid'

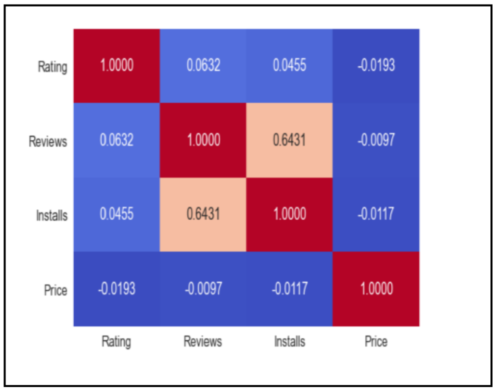


Figure 10: Correlation Matrix of dataset

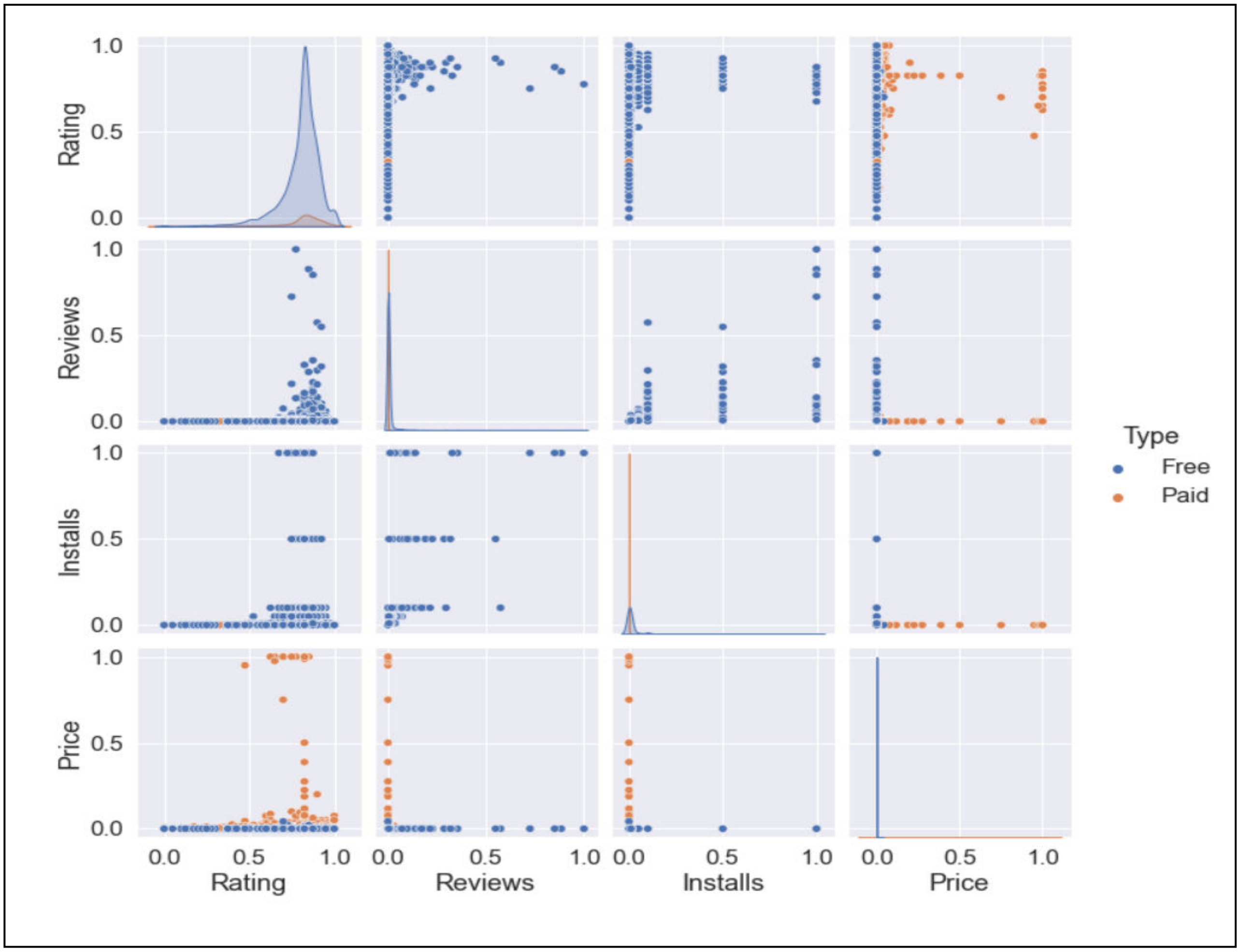


Figure 11: Pairplot of dataset classify by type

2.1.5 General Statistics:

A Pie plot has created for view and understand the distribution into the dataset for the major categories by descending order. It has detected that 'Family' and 'Game' category contain high value than other averagely distributed categories.

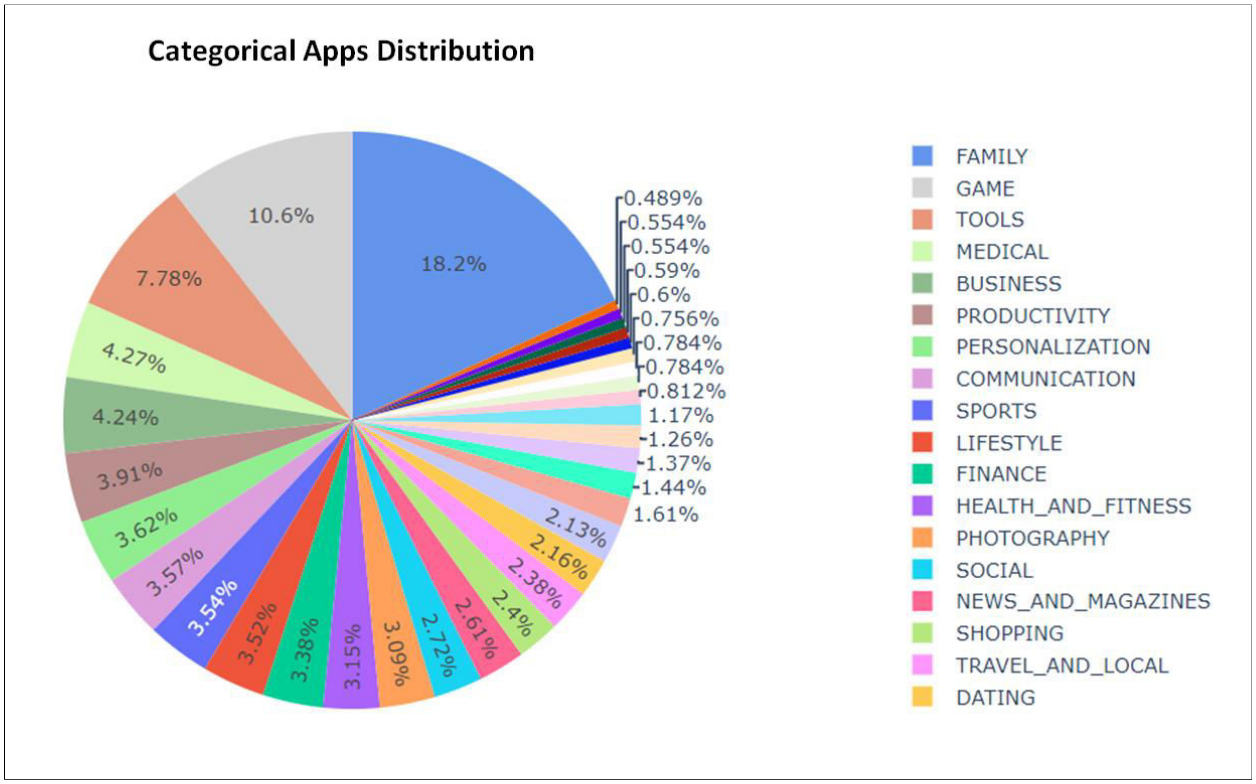


Figure 12: Categorical apps distribution

Similarly, another pie plot has created to overview the distribution of attribute 'Type' for further analysis. It has been detected that mostly apps are free which is more then 90%.

Moreover, Audience from 'Contant Rating' is shown by pie plot as well, though theoretically the audience range is overlapped here but from dataset it has been shown that they considered the audience types distinctly.

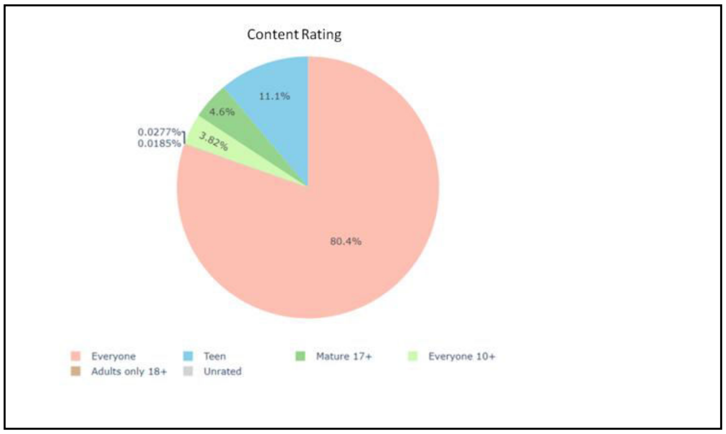


Figure 13: Content Rating distribution

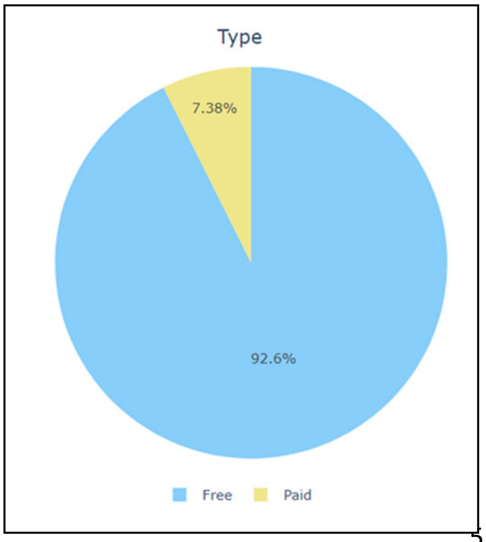


Figure 14: Type distribution

From the kde plot from 'Rating' the most density detected between '4' to '4.5' . And, from the distribution of 'Installs' a noticeable trend come that, from 0 to 0.1 (short scale billion X-axis) has the highest density by apps.

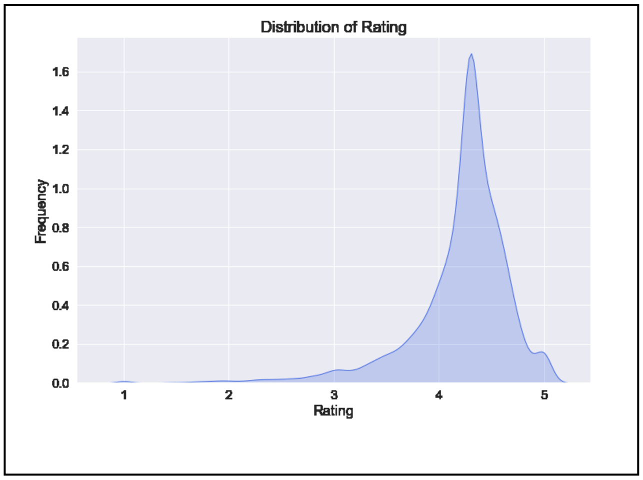


Figure 15: Rating KDE plot

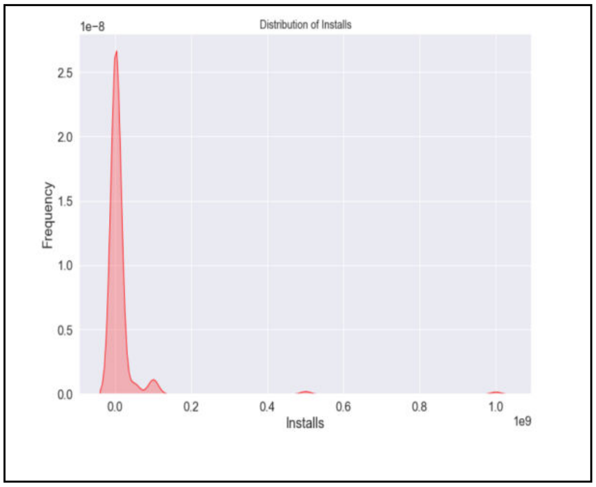


Figure 16: Installs KDE plot

3.1 Problem Statement 1:

The research problem was to derive most popular categorical apps. For this research, 'Category' 'Installs', 'Review' and 'Rating' attributes has selected. The reason behind that is, as any app is then popular when it is more installed, getting more reviews and rating. Popularity has positive impact or correlated with this parameters. So, a dashboard by tableau has created where as a scatter bauble plot and horizontal bar graph were shown side by side in interactive approach. The link is (https://public.tableau.com/profile/md.mozahidur.rahman#!/vizhome/GooglePlayStoreDataVisualization1/DashboardA)

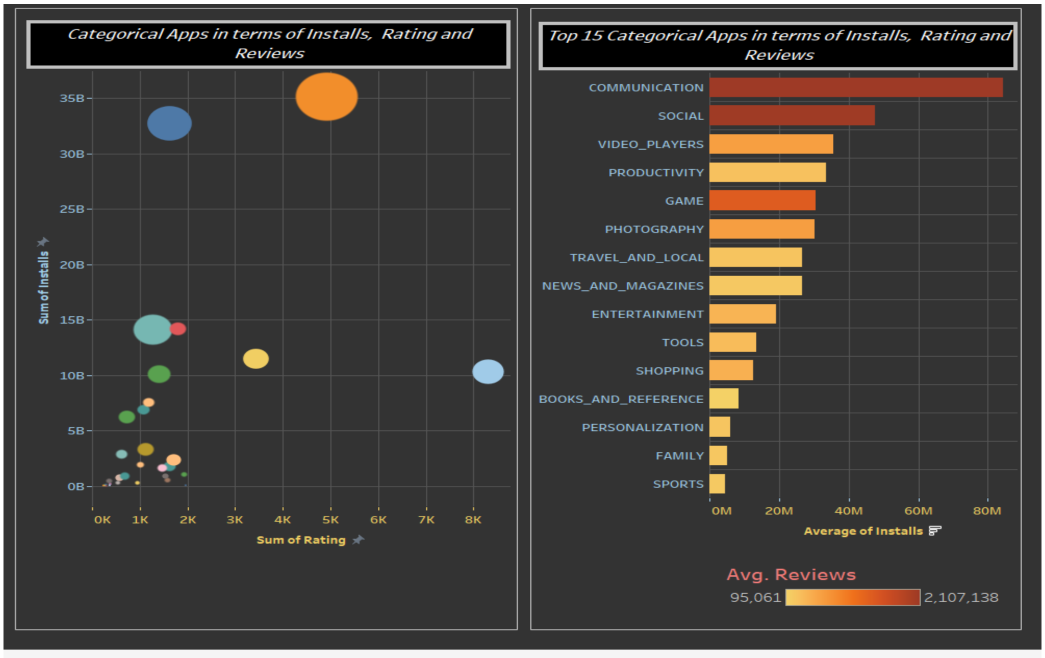


Figure 17: Dashboard A- Most popular apps

Here in first plot, ‘Sum of Rating’ in thousands has used as X-axis unit whereas ‘Sum of Installs’ in billions has used in Y-axis unit. Categorical apps is detected by color and size of the bubble depends on the sum of ‘Reviews’. As sum of parameter gives the exact view of dataset of all time from past to present so here summation operation is used. However, viewer can see the position of categorical apps by installs summation also. So, it could be easily visible which apps getting high installs and which is low. In addition size of the bubble can shown bigger circle with popularity difference among categories as well.

On the other hand second bar graph is based on average values to view their present situation in different manner. Here average of installs in million has used as unit in X-axis where top 15 categorical apps plotted in Y-axis. Average rating is used to plot the bar size and continuous color from gold to red has used for show average review. The reason behind use of average value of parameter is to analysis their popularity from another view.

3.1.1 Finding:

It has been determined that most popular categorical app is ‘Game’ and second most popular app is ‘Communication’. The popularity has determined from first plot. If we want to determine the relationship of this two apps from second graph then average installs of ‘Game’ is viewed lower where as total installs is higher then other apps. The reason is, there are more ‘Game’ apps then other category so the average getting down.

In addition, rating gave almost uniform value as the size of the bar looks uniformly distributed. And, a noticeable finding is ‘Parents’ apps have the higher sum of rating comparatively to all other apps though the total installs and reviews is lower than others.

The color concentration of ‘Communication’ and ‘Social’ is very high because there apps number is lower but the reviews are high comparatively to others.

3.1.2 Interactive Evaluation:

3.1.2.1 Case One:

When we choose Game then in second graph it shows full concentration of color. As there is no opponent here without ‘Game’ so the interactive view shown like this.

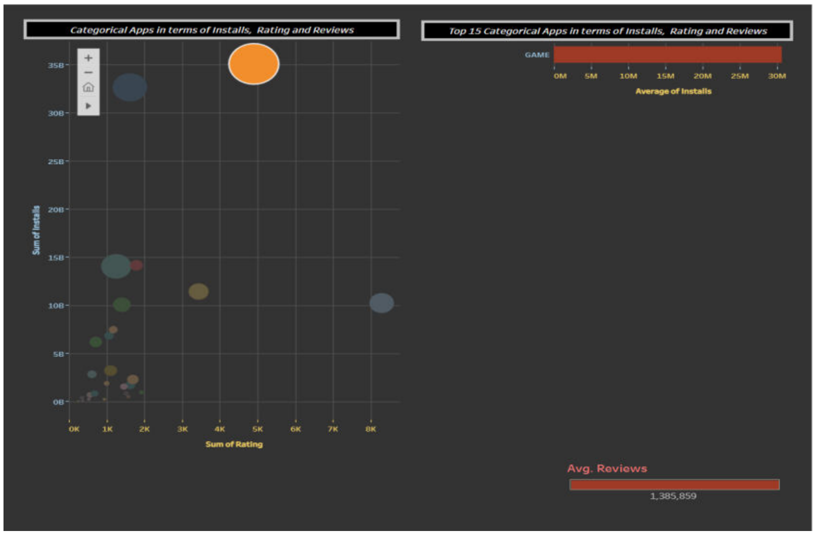


Figure 18: Dashboard A case one

3.1.2.2 Case Two:

When we choose more than one categorical apps in first graph it shows corresponding bar in second graph as well. The color concentration and X-axis unit has changed by their average values as well.

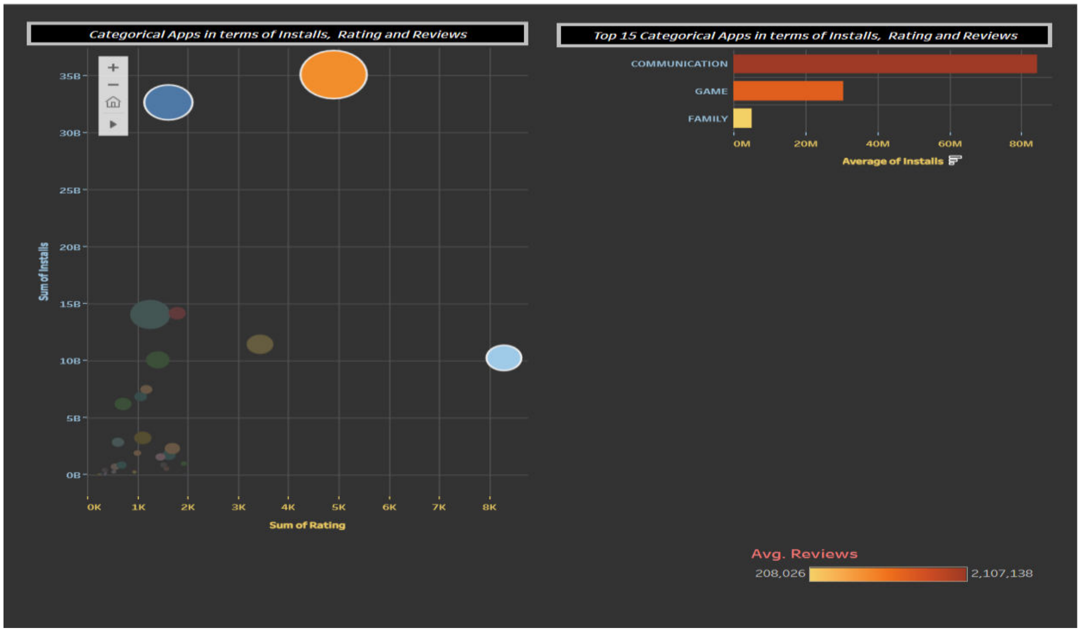


Figure 18: Dashboard A case two

3.2.1.3 Case Three:

If we want to derive most unpopular app as well, we get only the apps reside in top 15. The color, size, and bar size are also changed interactively corresponding to their categories from first graph.

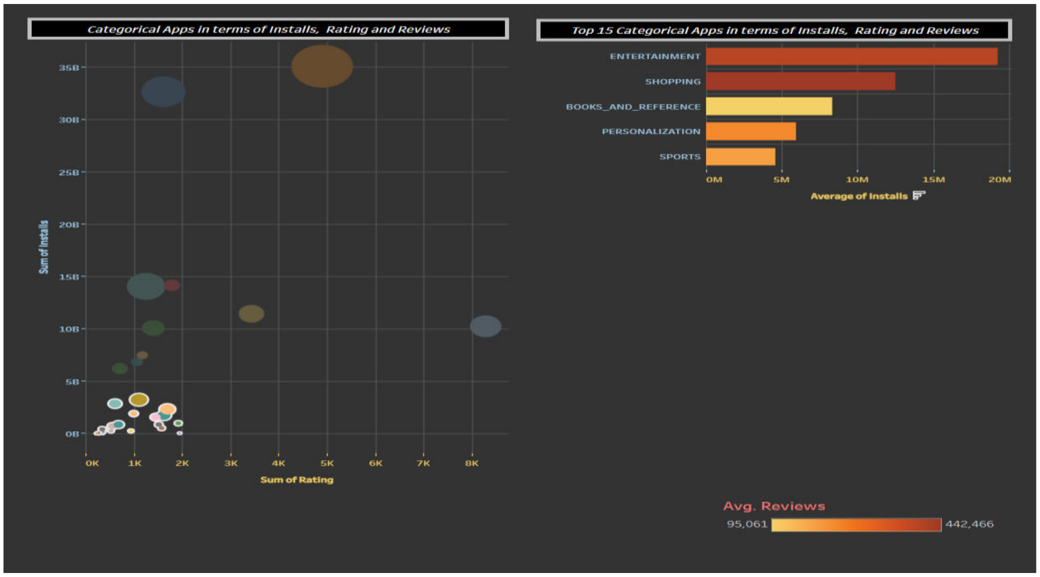


Figure 19: Dashboard A case three

3.2 Problem Statement 2:

It was quit challenging that the dataset doesn't contain any financial value like sales, purchase, invest, expenses. So after careful observation it was derived that it needed extra operation on dataset to determine most profitable customer for google play store. So, I filtered 'Type' category and took price for paid apps, then multiply with installs to get sales value. As every install need to purchase first for paid apps so the sales value can be derive from existing attributes. Lastly created a dashboard based on 'Sales', 'Category', 'Content Rating'. The analysis for dashboard was to view audiences which is derived from 'Content Rating' by ranking manner. That means I wanted to view for top category apps based on ranked audiences.

For derivation of this criteria, top categorical apps has been filtered by sales value and used in X-axis, which has later used in top like menu. 'Content Rating' categories has plotted by sales value in line gaph. Later the line graph is ranked. Another line graph has used for node creation which has designed by overlap. A horizontal bar graph categories by 'Content Rating' has created by filtered with 'Category' plotted with sum of 'Sales'. Lastly an interactive dashboard was designed to show sales value for categorical apps and differential customer by color.

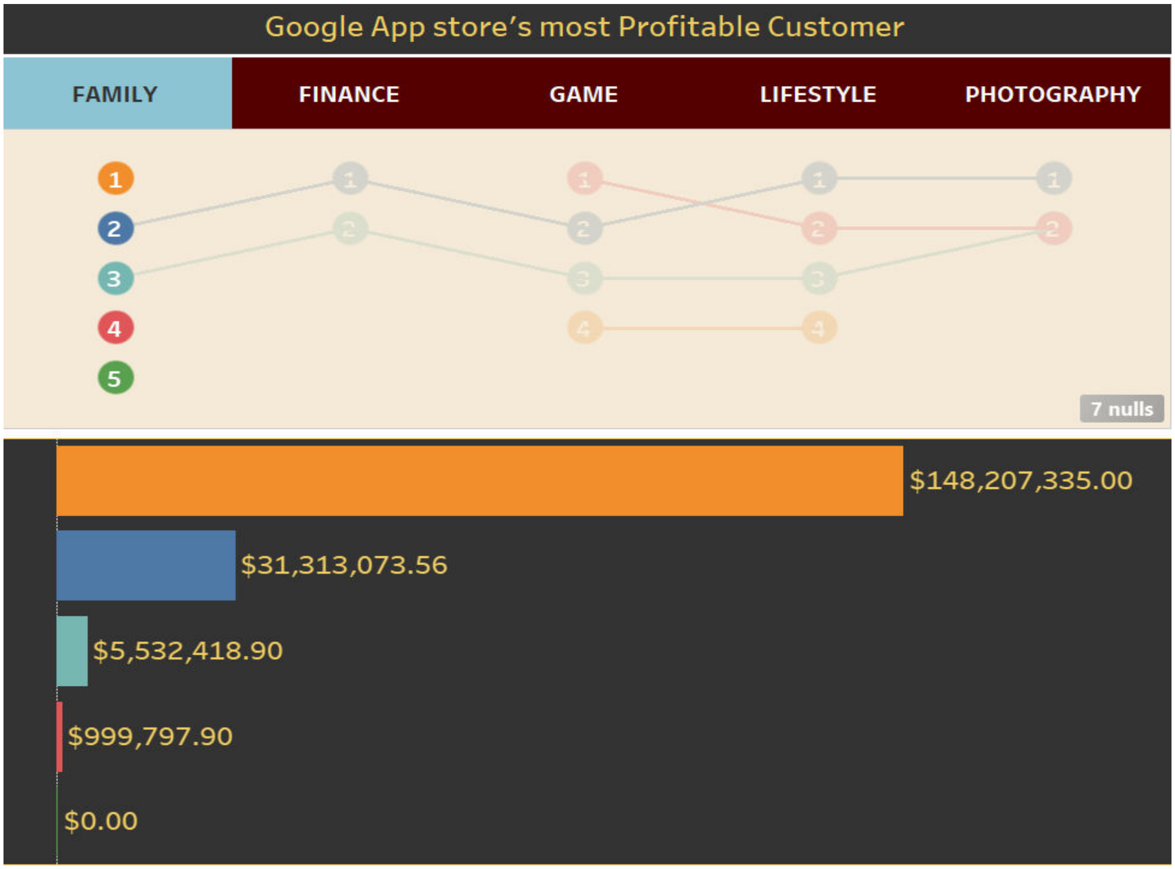


Figure 20: Dashboard B- Most profitable customer

3.2.1 Findings:

Here it is derived that top five profitable categories are ‘FAMILY’,’FINANCE’,’GAME’,’LIFESTYLE’ and ’PHOTOGRAPHY’ orderly. For family categorical apps the sales has shown in ranked manner by ‘Content Rating’. ‘Everyone 10+’ which has colored by orange is the most profitable customer for ‘FAMILY’ apps. In addition, not all group of the customer is involved with google play store paid apps. Comparatively for second profitable app ‘FINANCE’ has very low sales with first one. And, for ‘GAME’ apps the sales is even lower though the participants are higher. Though ‘GAME’ is the most popular app but it is not the profitable one.

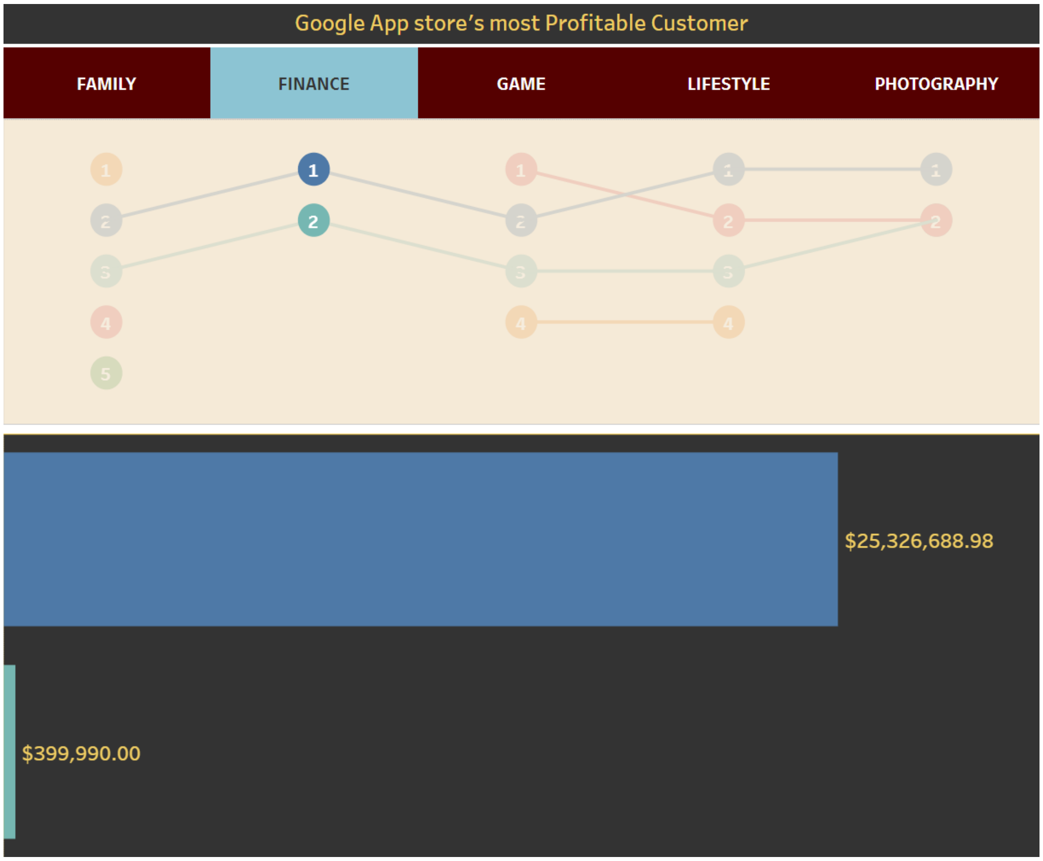


Figure 21: Dashboard B- Finance income

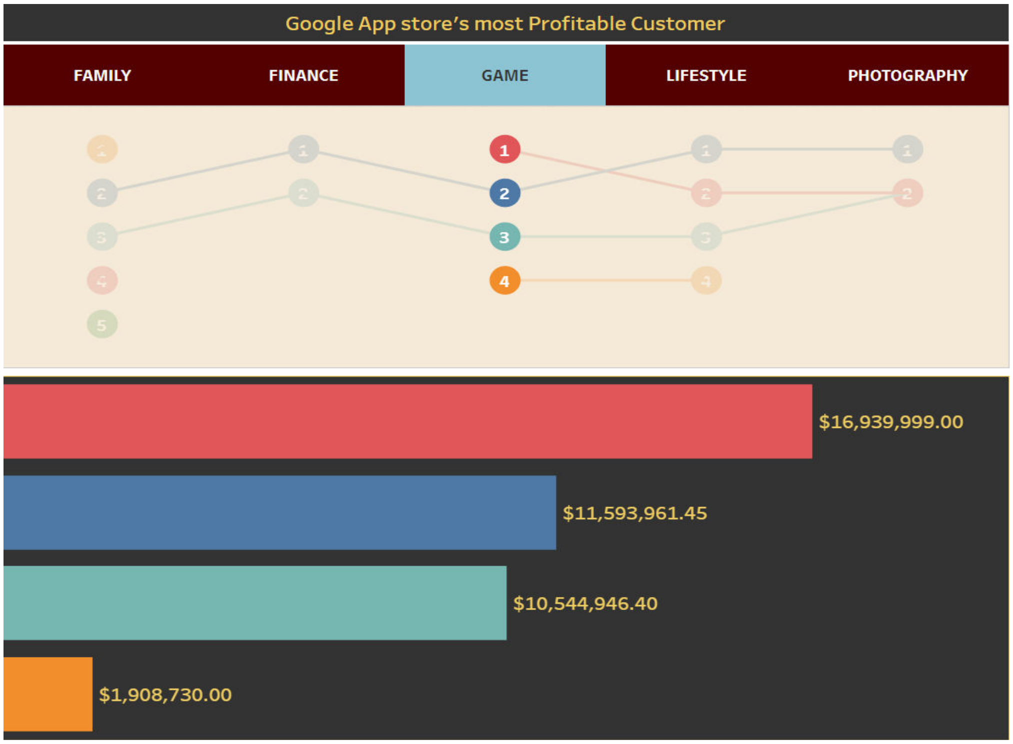
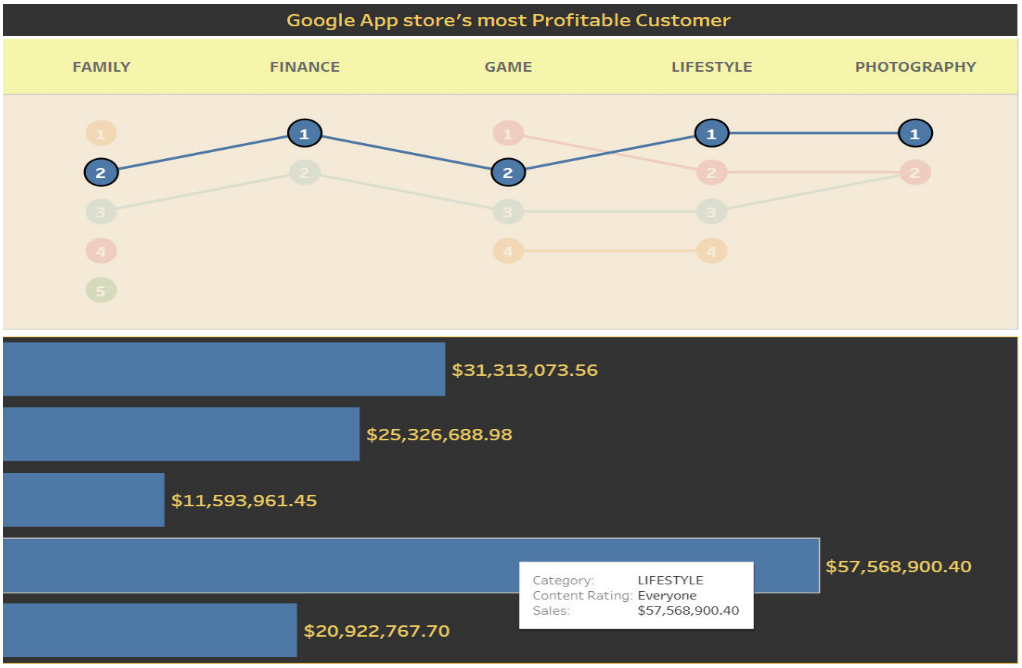


Figure 22: Dashboard B- Game income

3.2.2 Interactive Evaluation:

3.2.2.1 Case One : For a particular customer group in 'Content Rating' the dashboard could show sales by each top five category apps.

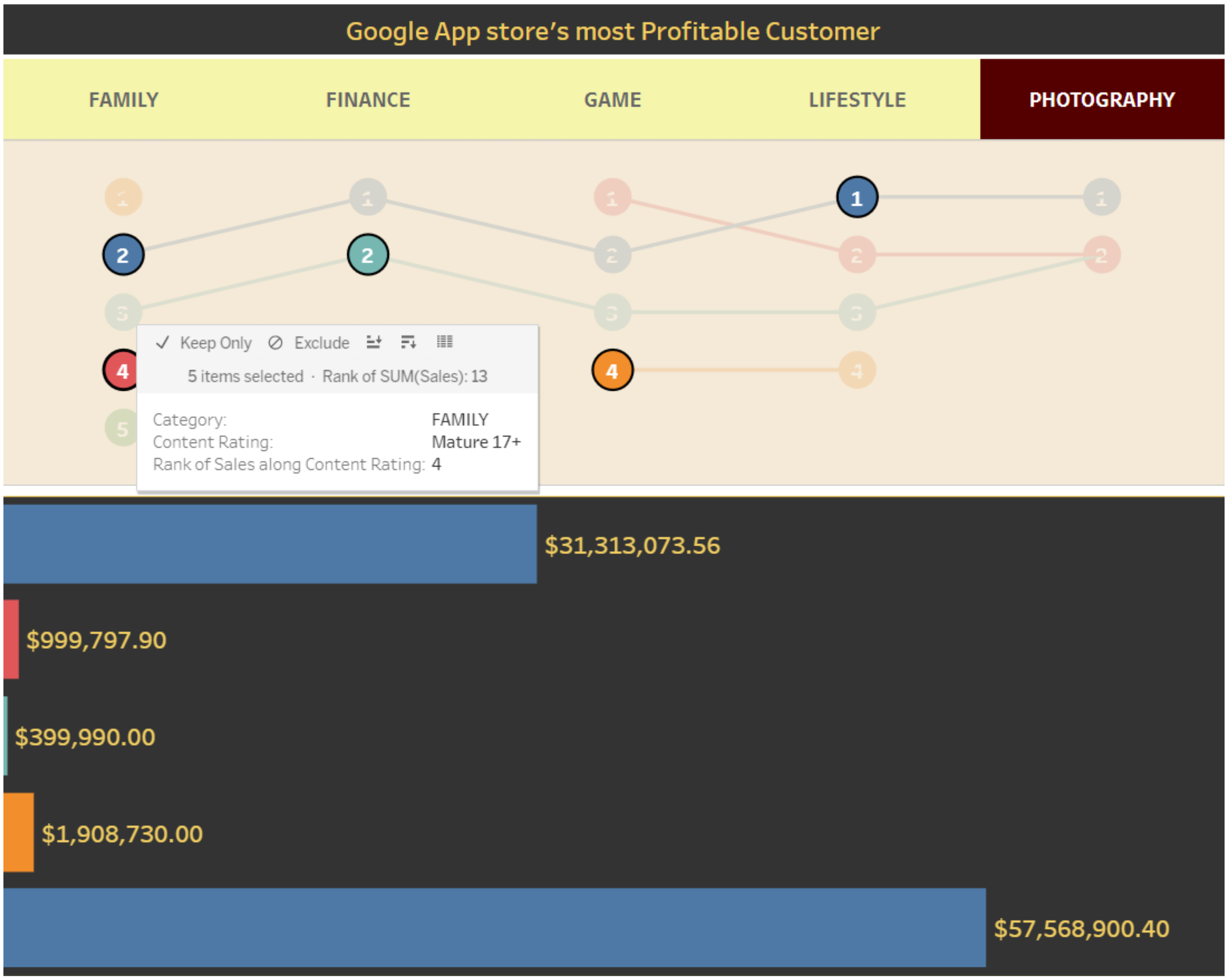


Category: LIFESTYLE
Content Rating: Everyone
Sales: $57,568,900.40

Figure 23: Dashboard B- case one

3.2.2.2 Case two:

It is possible to view sales by any customer group from any categorical apps.



5 items selected · Rank of SUM(Sales): 13

Category:	FAMILY
Content Rating:	Mature 17+
Rank of Sales along Content Rating:	4

Figure 24: Dashboard B- case two

3.3 Problem Statement 3:

For understanding 'Google Play Store ' business strategies about invest, I wanted to determine their business service in first hand. Because, when they boost their service then they invest more and got sales proportionally. On the other hand sales is positively correlated with installs, prices, ratings and reviews. So I consider all of this data to analysis their investment strategies. Moreover, to analysis timeframe 'Last Updates' has used by split manner. The data of updates distributed from 2012 to 2018 so I divided the timeframe monthly and count the sum of installs per month by rank aggregate function. In the following dashboard the first figure is racing bar diagram where as I have chosen top ten categorical apps in terms of sum of installs based on monthly slot. Continuous color range gold to red shows sum of reviews on that particular bar. Sum of 'Installs' is used for X-axis unit counted by billions.

In the second graph top five categorical apps has shown for consecutive three years whereas average of installs has plotted as Y-axis and year in X-axis. Categories were differentiate with colors and shown in line graph. Sum of reviews corresponding to that time line shown by width of the line.

Third graph is constructed by four attributes corresponding with timeline. Average installs, reviews and ratings have plotted in line graph with corresponding width of value concentration. In addition sales is also plotted with timeline here.

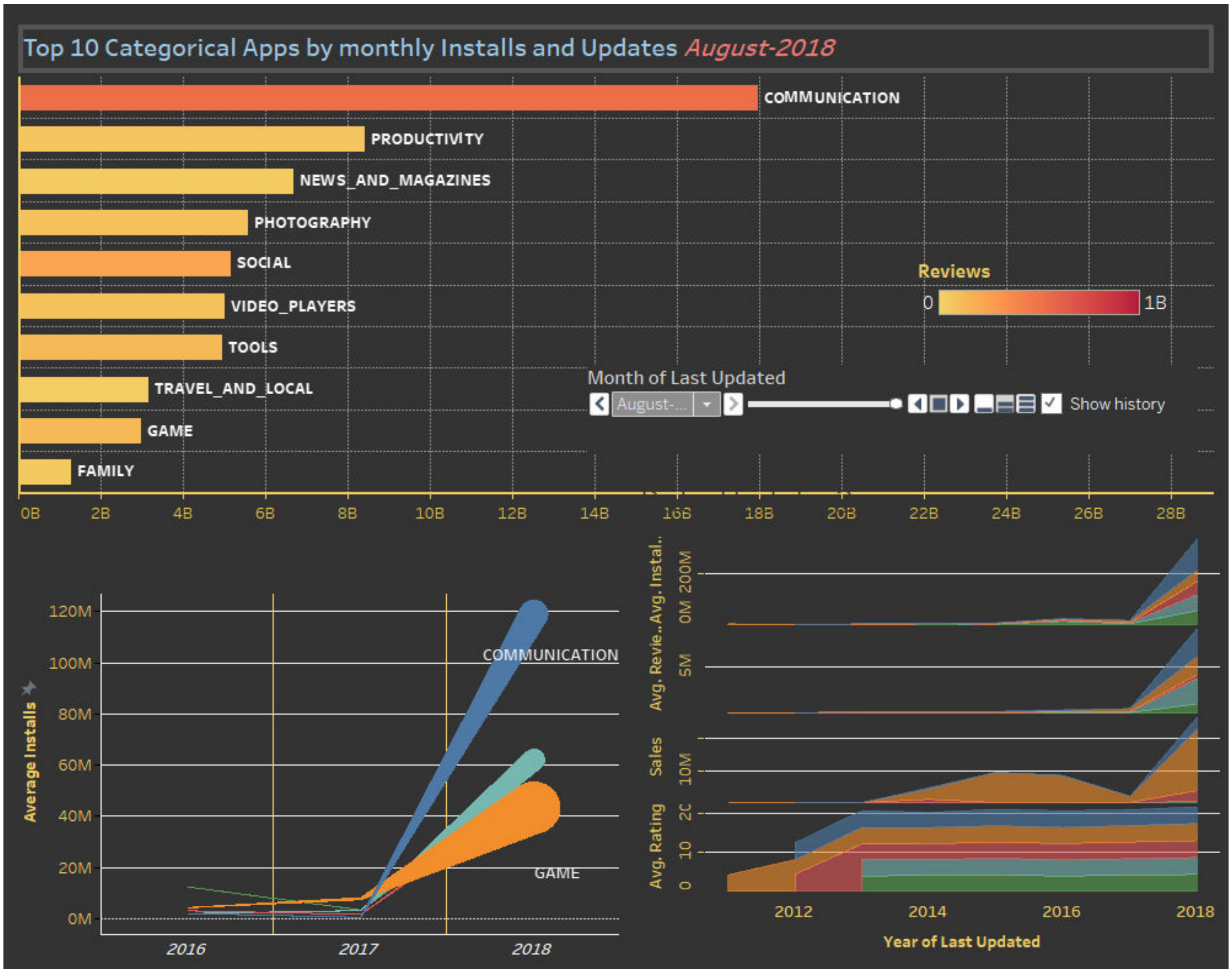
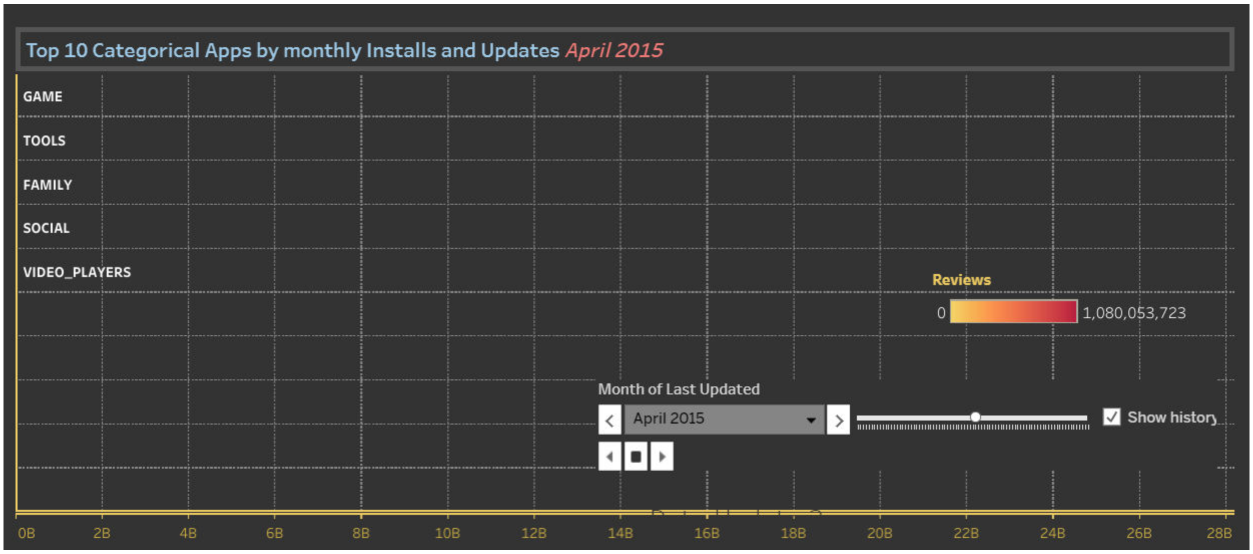


Figure 25: Dashboard C-google invest scenario

3.3.2 Findings:

In the racing bar an interesting fact has came out. It has shown that the number of installs was too low in 2012-2014. And from 2017 it getting increased gradually, it hit its maximum range in 2018. So theoretically we can say that 'Google Play Store ' got its highest sales in 2018. As sales is proportional to their service so the investment must be happened in their time frame.

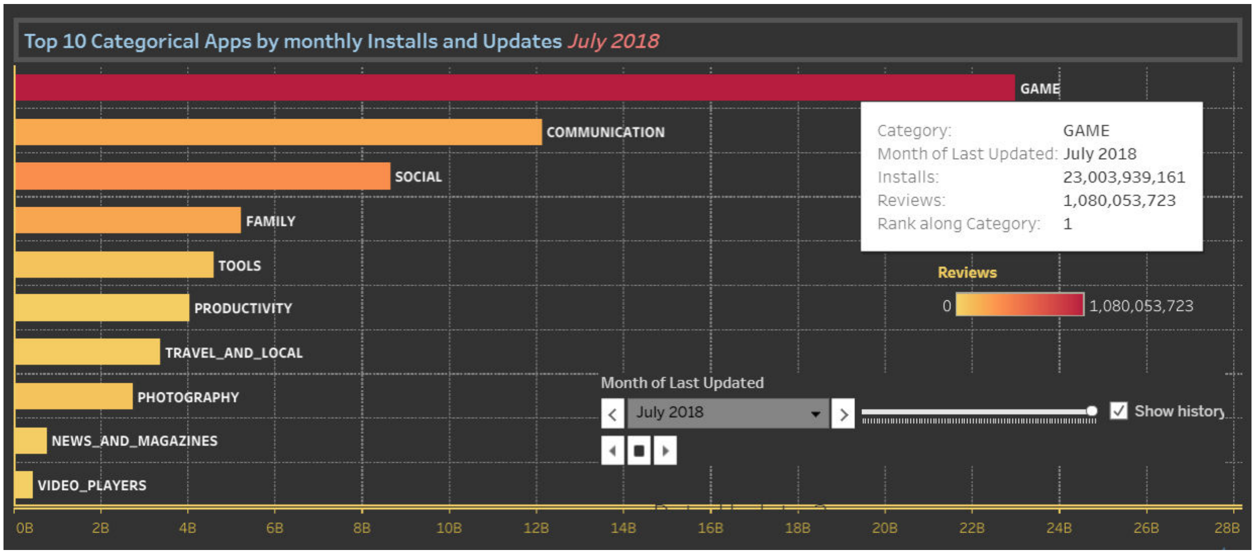


Month of Last Updated: April 2015

☑ Show history

Figure 26: Dashboard C- Racing bar 1

However, it has been noticed that the installs amount in july,2018 was very high with reviews corresponding to the last month august,2018. The reason could be that dataset might not contain full data in august for 'GAME' apps. It could be related to most popular apps as well.



Category: GAME
Month of Last Updated: July 2018
Installs: 23,003,939,161
Reviews: 1,080,053,723
Rank along Category: 1

Month of Last Updated: July 2018

☑ Show history

Figure 27: Dashboard C- Racing bar 2

In the second figure, most perceptible fact derived average installs data changing from 2017 to 2018 dramatically. Even the width of the line is interesting observation. Game apps have the highest width though the average installs is lower then others. It is because game category holds highest number of apps than others.

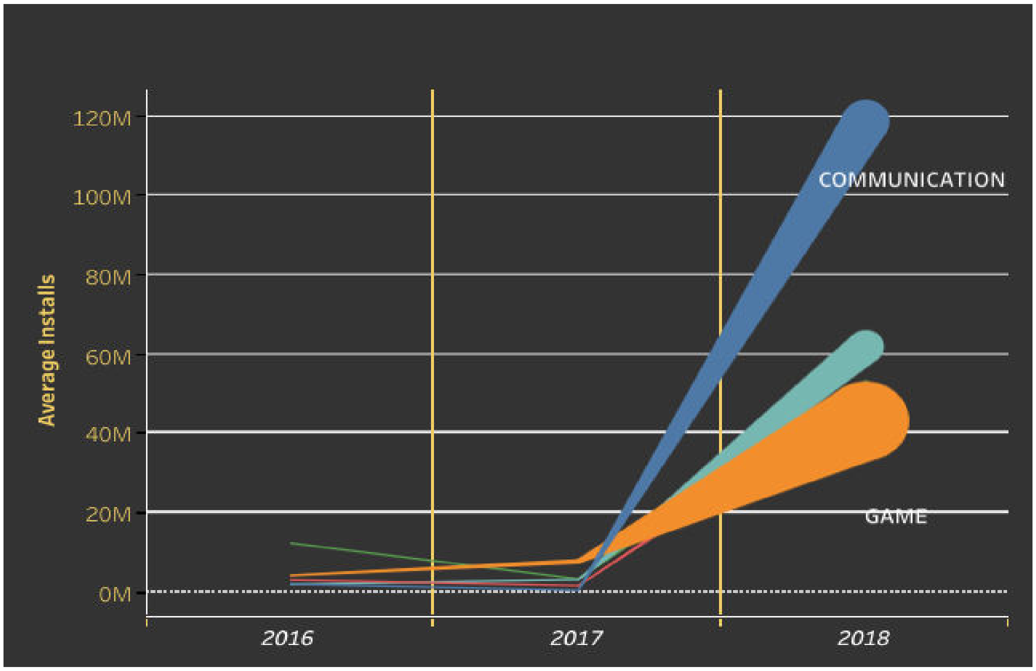


Figure 28: Dashboard C- Line graph

From the third figure it was a similar trend that detected from 2017 for top five categorical apps. But in sales vs time graph the most noticeable point is that 'GAME' app sales got increased from 2015 and little bit down at 2017 but later it hits its peak value. In addition, average rating was quit low in the beginning of 2011 to 2012 but later the business receive uniform rating for rest of the years.

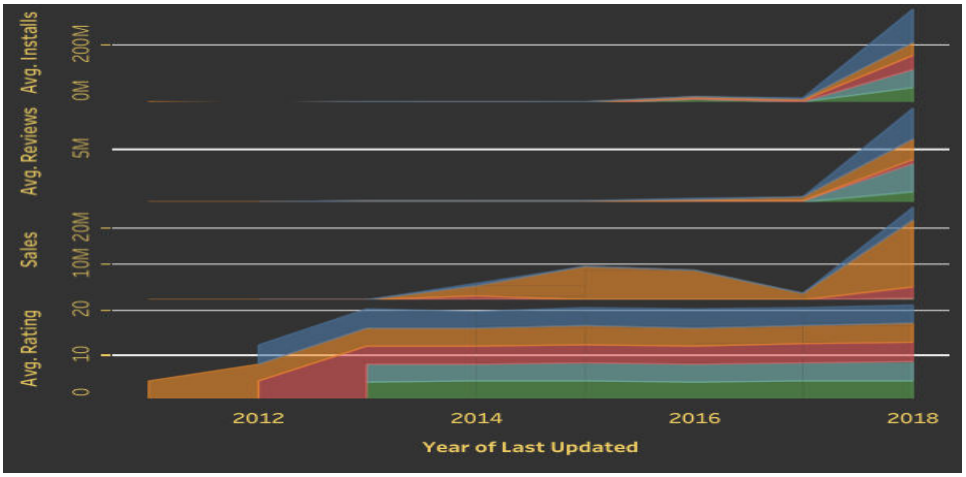


Figure 29: Dashboard C- Line graph 2

3.3.3 Interactive Evaluation:

3.3.3.1 Case One:

For any particular app or more than one app we could observe racing bar as well. The color concentration will be changed by their value also. Moreover for any app or number of top five apps we can view sum of installs by time frame as well.

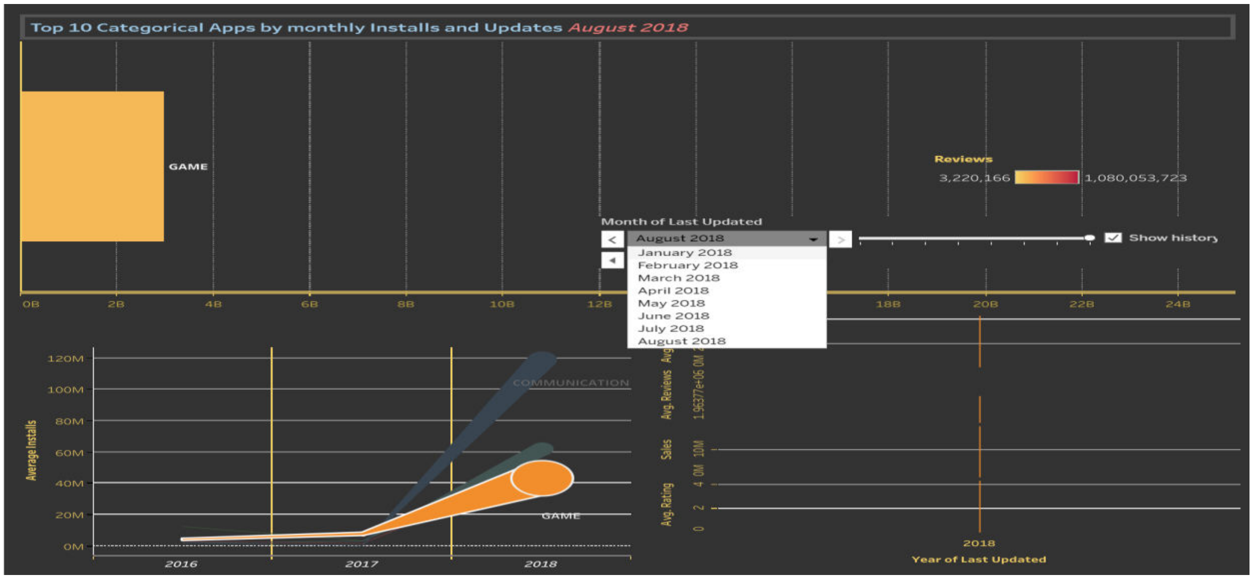


Figure 30: Dashboard C case one

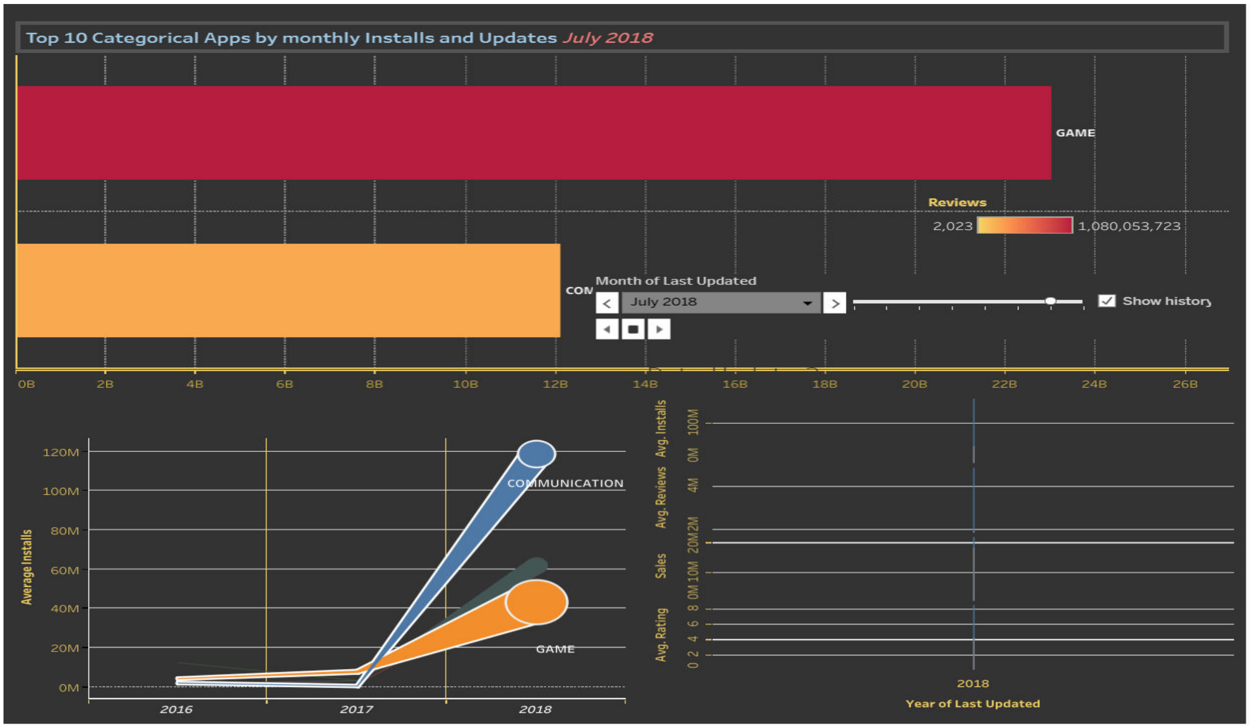


Figure 31: Dashboard C case one_

3.3.3.2 Case Two:

From graph three we can view interactive racing bar also. In this circumstances the width of the bar depends on plot size. Number of reviews in color could also describe. When there is no opponent racing bar shows full color concentration.

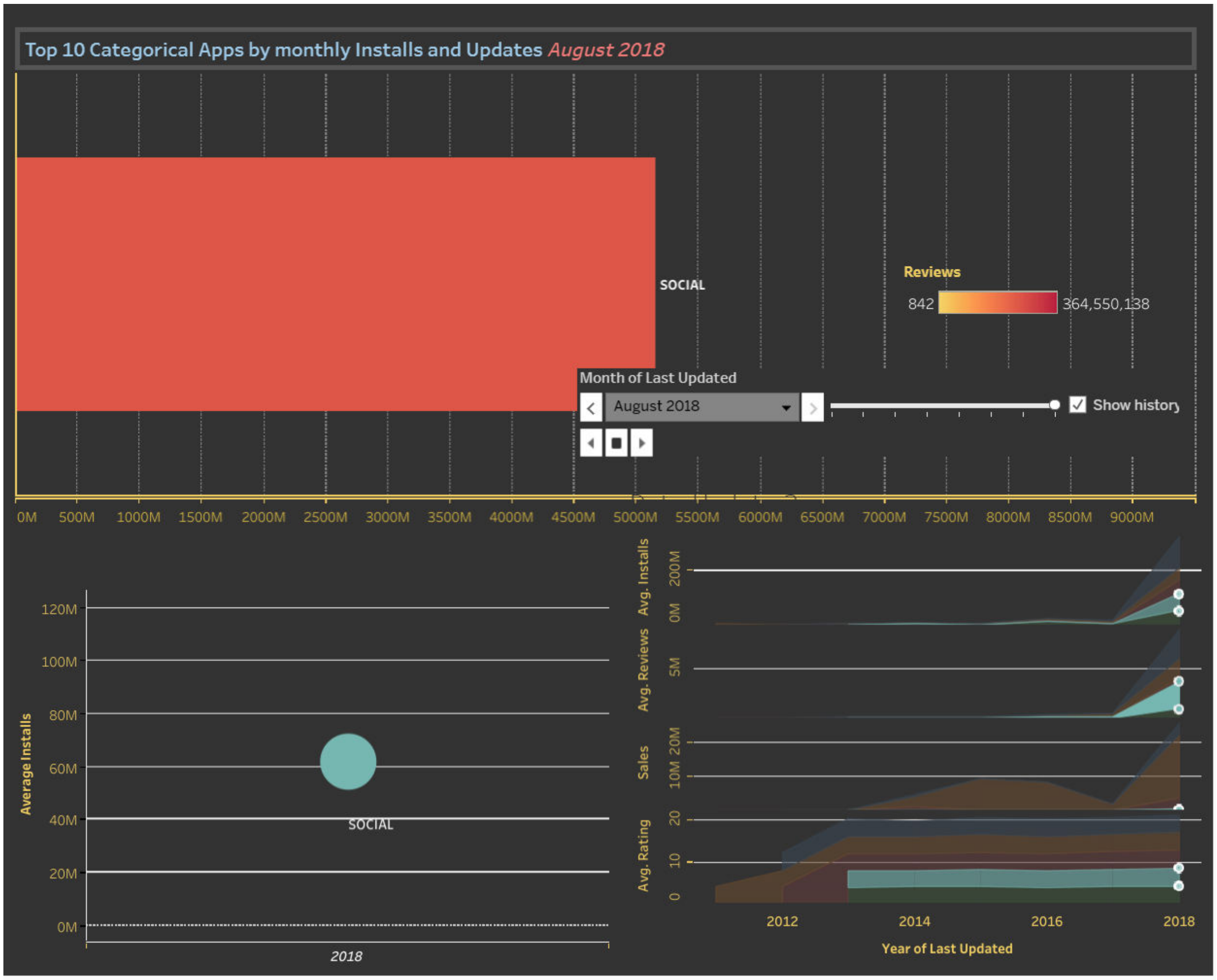

Top 10 Categorical Apps by monthly Installs and Updates August 2018
SOCIAL
Reviews
842
364,550,138
Month of Last Updated
August 2018
Show history
0M 500M 1000M 1500M 2000M 2500M 3000M 3500M 4000M 4500M 5000M 5500M 6000M 6500M 7000M 7500M 8000M 8500M 9000M
Average Installs
120M 100M 80M 60M 40M 20M 0M
SOCIAL
2018
Avg. Installs
200M 0M
Avg. Reviews
5M
Sales
20M 10M
Avg. Rating
20 10 0
2012 2014 2016 2018
Year of Last Updated

Figure 32: Dashboard C case two

4.0 Conclusion:

Google has developed their business in very efficient way from 2017. They have to reach in their highest point though they need to emphasis in some apps category which are not popular and profitable yet.

5.1 References:

1. Tableau 2021.1, computer software downloaded 1 April 2021, < https://www.tableau.com/>
2. Edureka(2019). *Tableau Full Course - Learn Tableau in 6 Hours | Tableau Training for Beginners | Edureka.* Available at: https://www.youtube.com/watch?v=aHaOIvR00So (Accessed: 1 April 2021).
3. Simplilearn(2020).*Tableau Full Course - Learn Tableau In 6 Hours | Tableau Training for Beginners | Simplilearn.* Available at: https://www.youtube.com/watch?v=HGMrIZq5dq0&t=3939s (Accessed: 2 April 2021).
4. Abhishek Agarrwal (2019). Tableau Training for Beginners | Tableau Complete Tutorial for Beginners [Full Course] [2020]. Available at: https://www.youtube.com/watch?v=xB2SO2hHc8g (Accessed: 3 April 2021).

Appendix A:

1. Tableau Workbook Screenshots (Category Bubble)

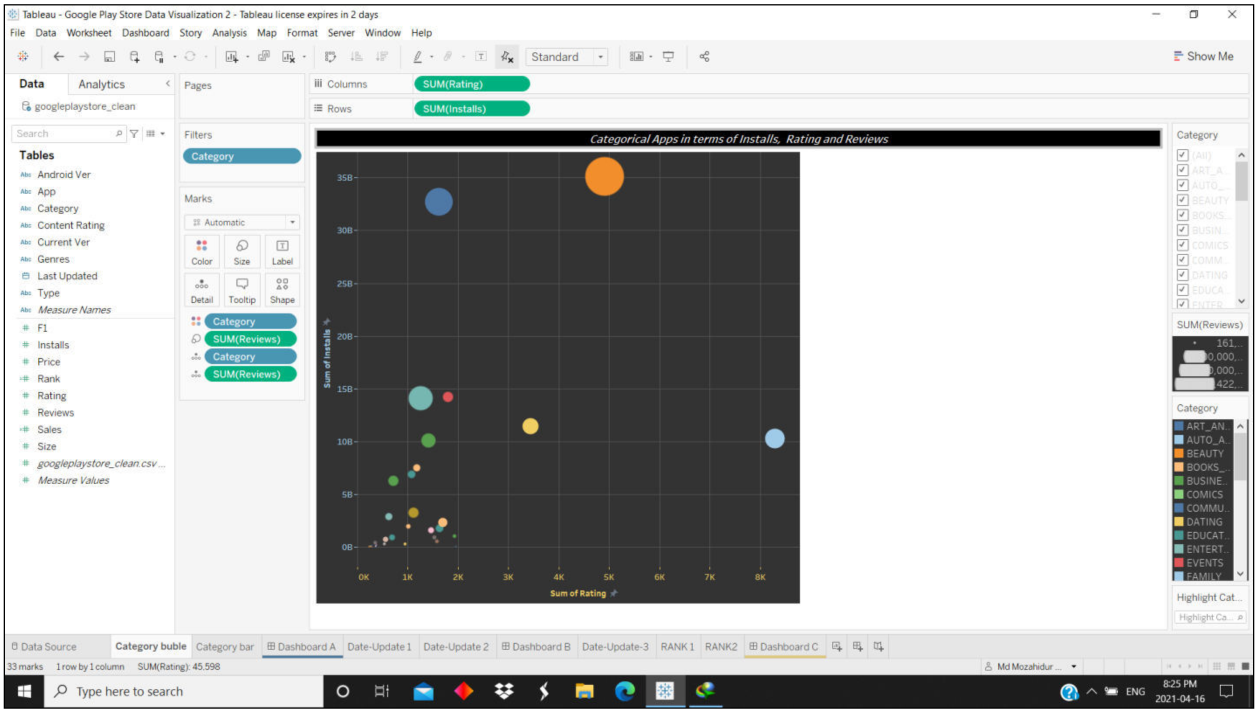


Tableau - Google Play Store Data Visualization 2 - Tableau license expires in 2 days

Columns: SUM(Rating)

Rows: SUM(Installs)

Filters: Category

Marks: Automatic; Category; SUM(Reviews); Category; SUM(Reviews)

33 marks 1 row by 1 column SUM(Rating): 45.598

2. Tableau Workbook Screenshots (Category Bar)

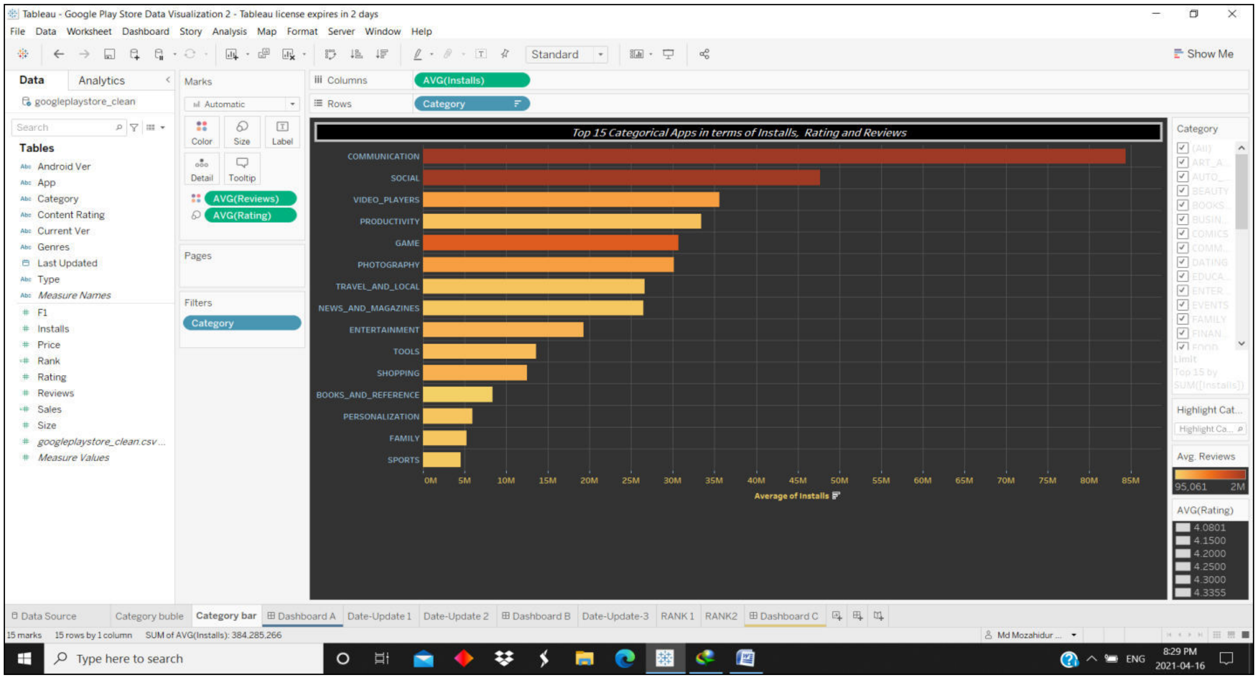


Tableau - Google Play Store Data Visualization 2 - Tableau license expires in 2 days

Columns: AVG(Installs)

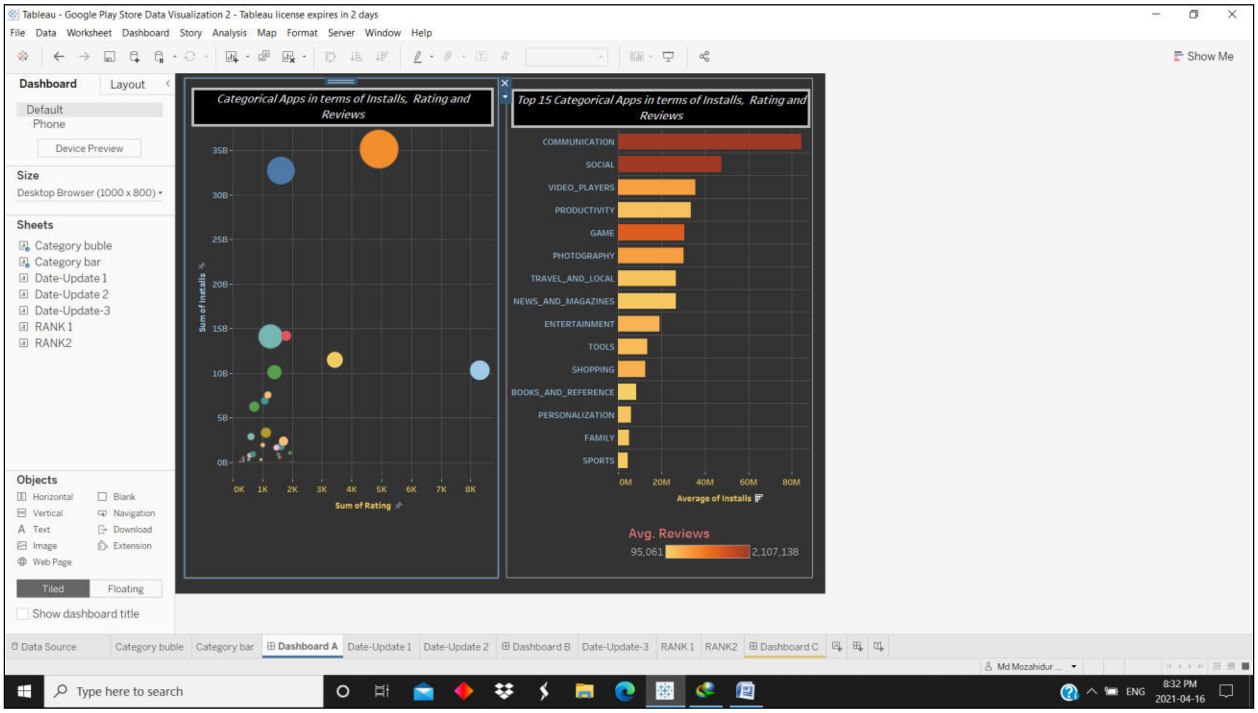
Rows: Category

Marks: Automatic; AVG(Reviews); AVG(Rating)

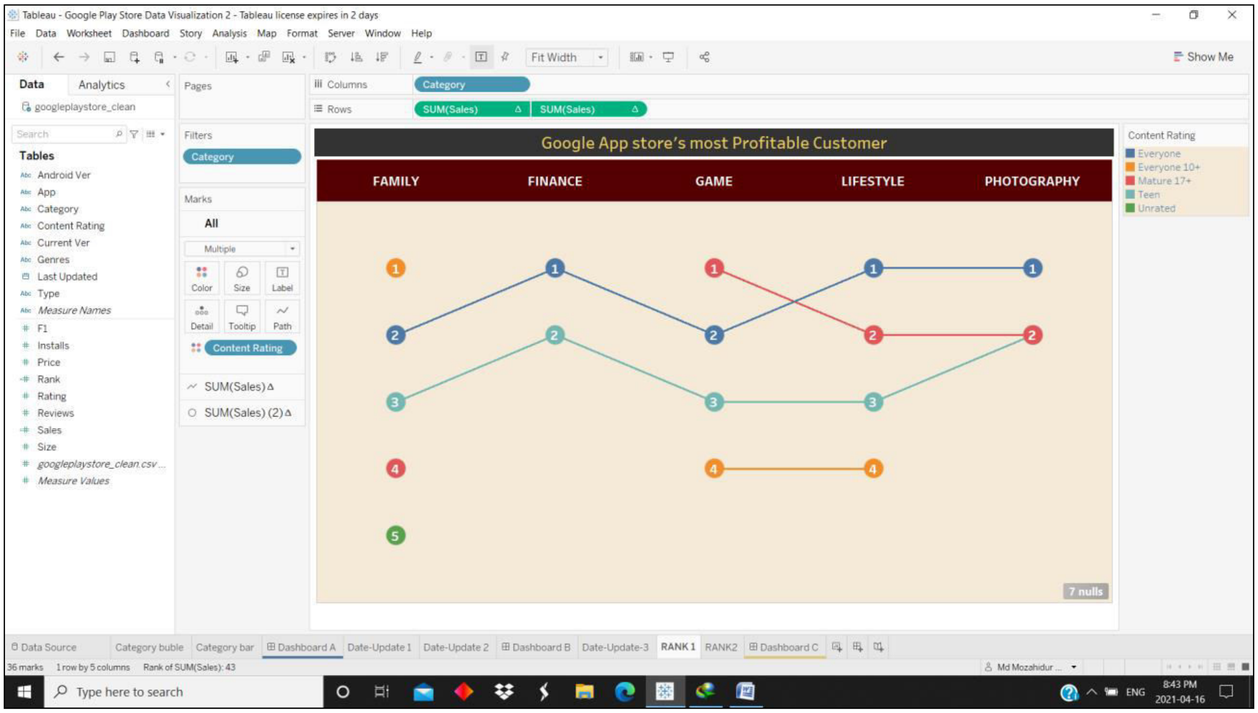
Filters: Category

15 marks 15 rows by 1 column SUM of AVG(Installs): 384,285,266

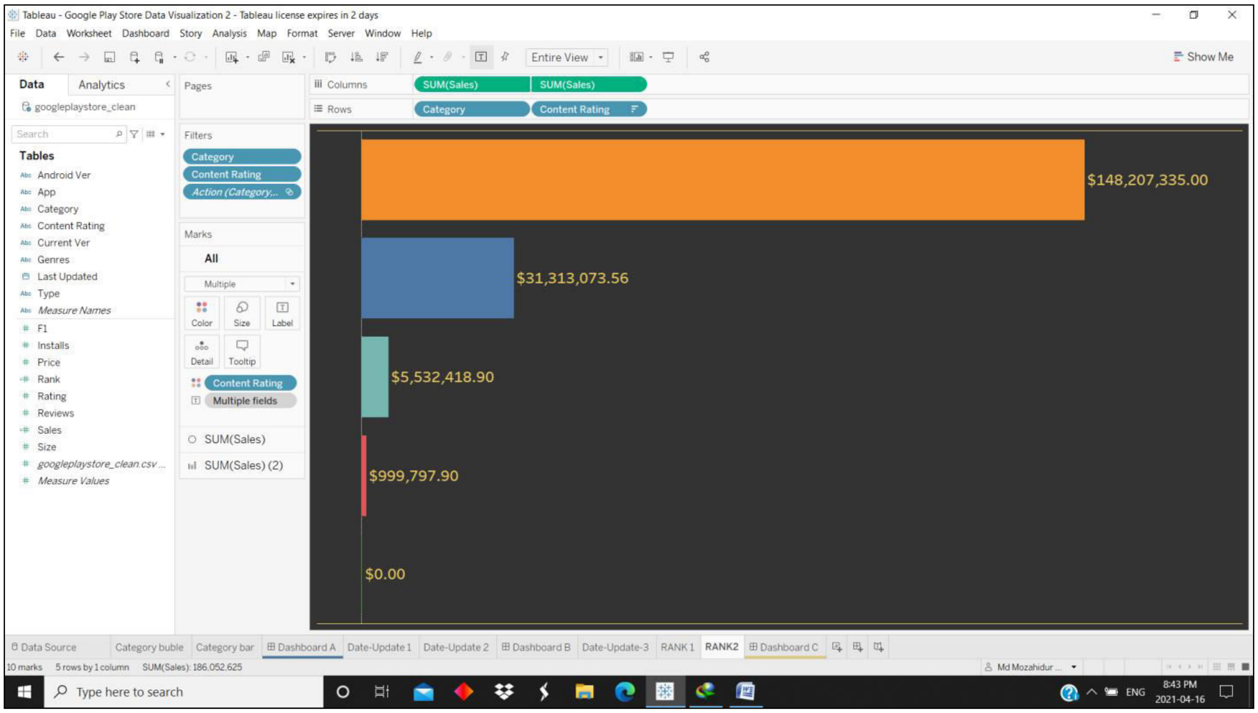
3. Tableau Workbook Screenshots (Dashboard A)



4. Tableau Workbook Screenshots (Rank 1)



5. Tableau Workbook Screenshot (Rank 2)



6. Tableau Workbook Screenshot(Dashboard B)

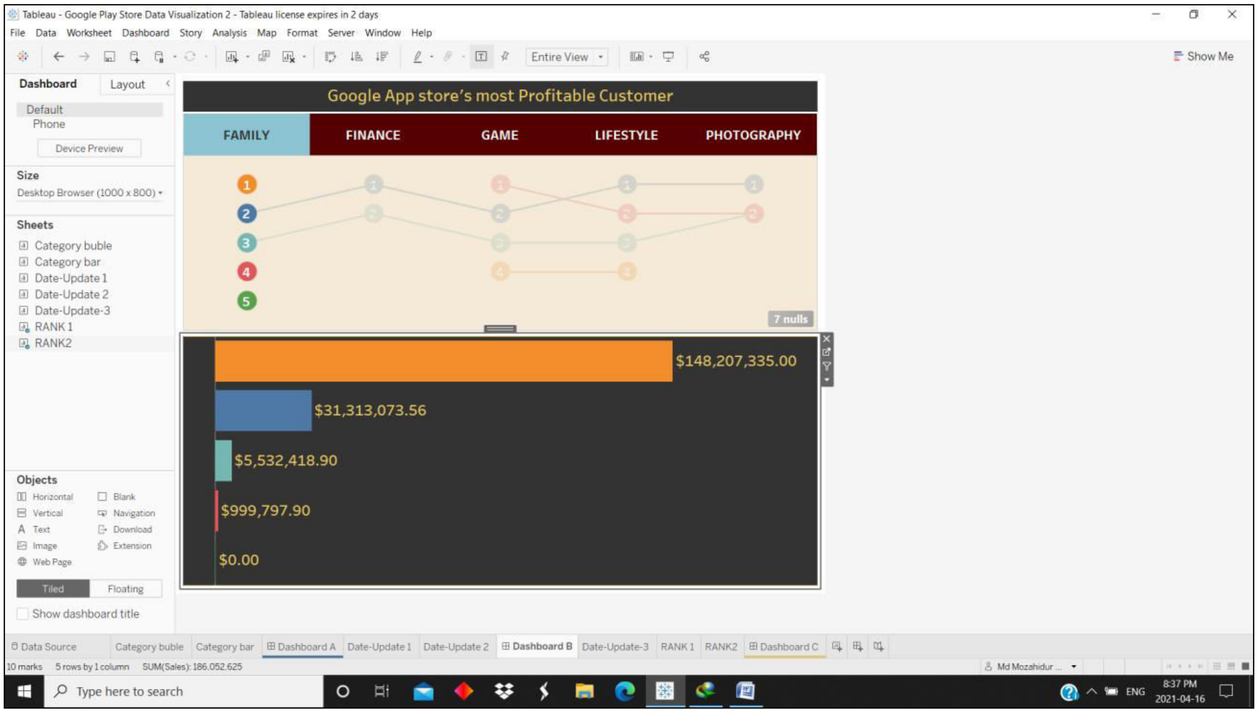


Tableau - Google Play Store Data Visualization 2 - Tableau license expires in 2 days
File Data Worksheet Dashboard Story Analysis Map Format Server Window Help
Entire View
Show Me
Dashboard Layout
Default
Phone
Device Preview
Size
Desktop Browser (1000 x 800)
Sheets
Category buble
Category bar
Date-Update 1
Date-Update 2
Date-Update-3
RANK 1
RANK2
Objects
Horizontal Blank
Vertical Navigation
Text Download
Image Extension
Web Page
Tiled Floating
Show dashboard title
Google App store's most Profitable Customer
FAMILY FINANCE GAME LIFESTYLE PHOTOGRAPHY
1
2
3
4
5
7 nulls
$148,207,335.00
$31,313,073.56
$5,532,418.90
$999,797.90
$0.00
Data Source Category buble Category bar Dashboard A Date-Update 1 Date-Update 2 Dashboard B Date-Update-3 RANK 1 RANK2 Dashboard C
10 marks 5 rows by 1 column SUM(Sales): 186,052,625
Md Mozahidur ...
Type here to search
ENG 8:37 PM 2021-04-16

7. Tableau Workbook Screenshots (Date-Update 1)

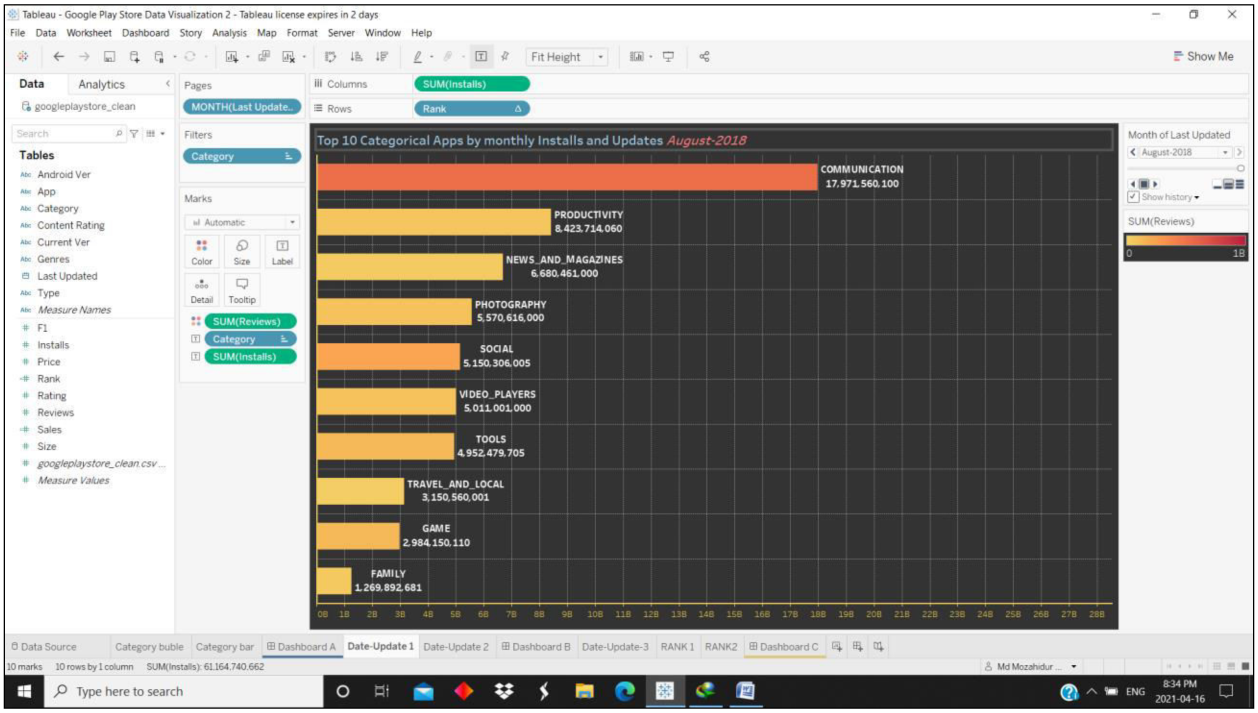


Tableau - Google Play Store Data Visualization 2 - Tableau license expires in 2 days

Pages: MONTH(Last Update..

Columns: SUM(Installs)

Rows: Rank

Filters: Category

Marks: Automatic; SUM(Reviews), Category, SUM(Installs)

Month of Last Updated: August-2018

10 marks 10 rows by 1 column SUM(Installs): 61,164,740,662

8. Tableau Workbook Screenshot (Date Update 2)

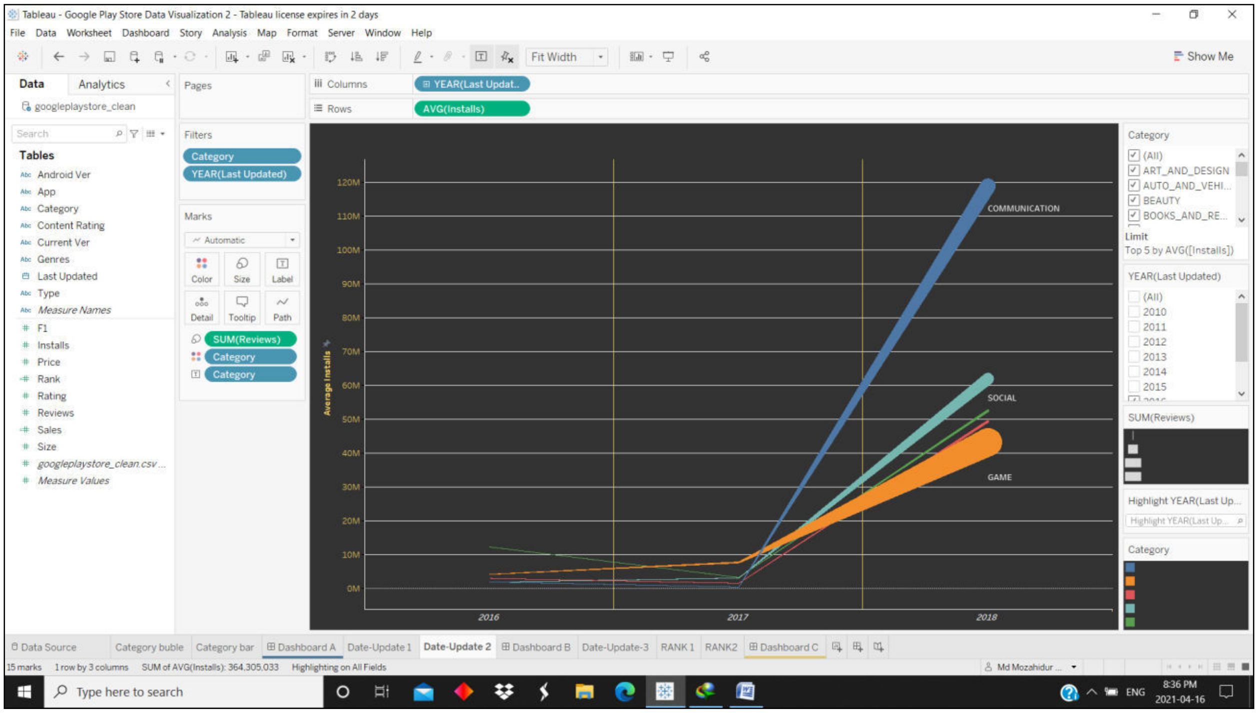


Tableau - Google Play Store Data Visualization 2 - Tableau license expires in 2 days

Columns: YEAR(Last Updat..

Rows: AVG(Installs)

Filters: Category, YEAR(Last Updated)

Marks: Automatic; SUM(Reviews), Category, Category

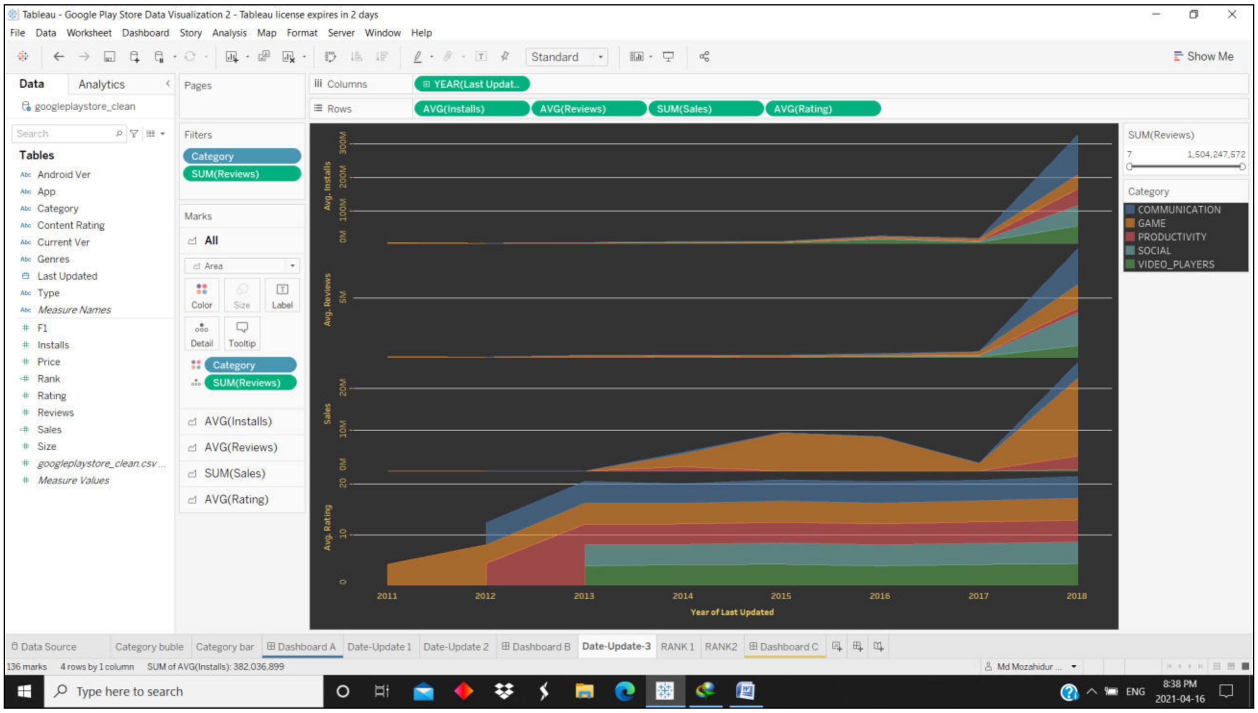
Category: ☑ (All) ☑ ART_AND_DESIGN ☑ AUTO_AND_VEHI... ☑ BEAUTY ☑ BOOKS_AND_RE...

Limit: Top 5 by AVG([Installs])

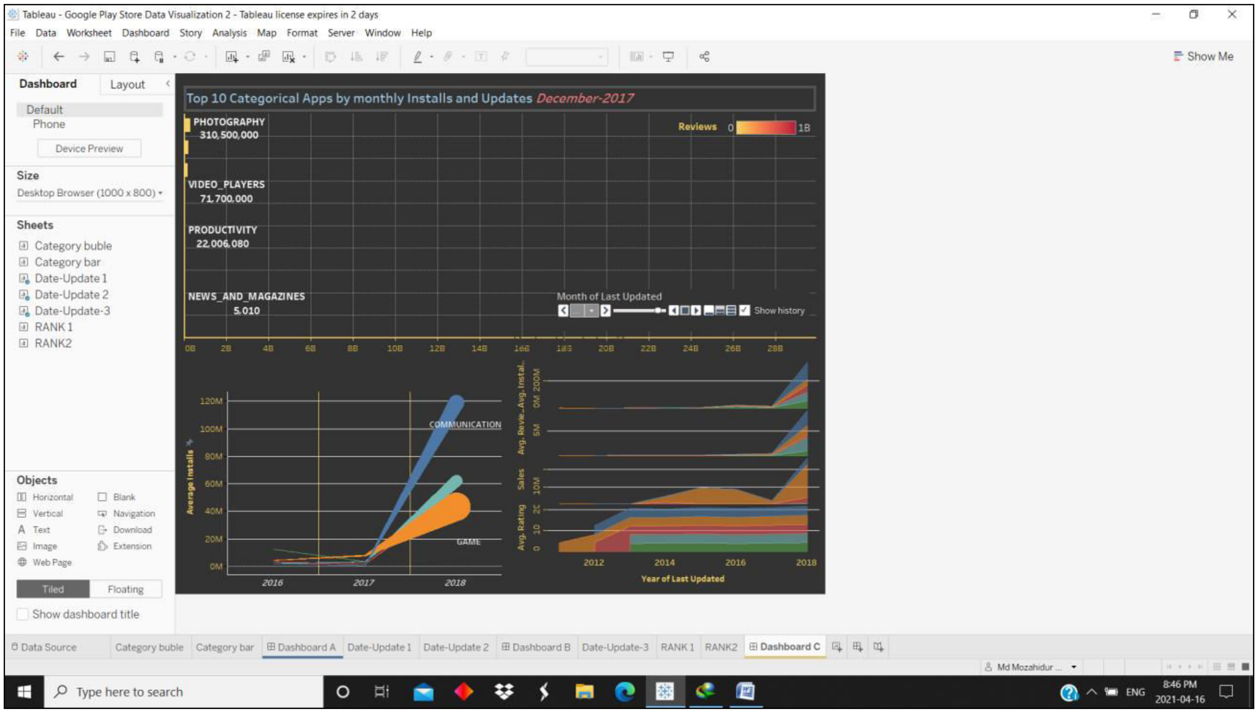
YEAR(Last Updated): □ (All) □ 2010 □ 2011 □ 2012 □ 2013 □ 2014 □ 2015

15 marks 1 row by 3 columns SUM of AVG(Installs): 364,305,033 Highlighting on All Fields

9. Tableau Workbook Screenshot (Date Update 3)

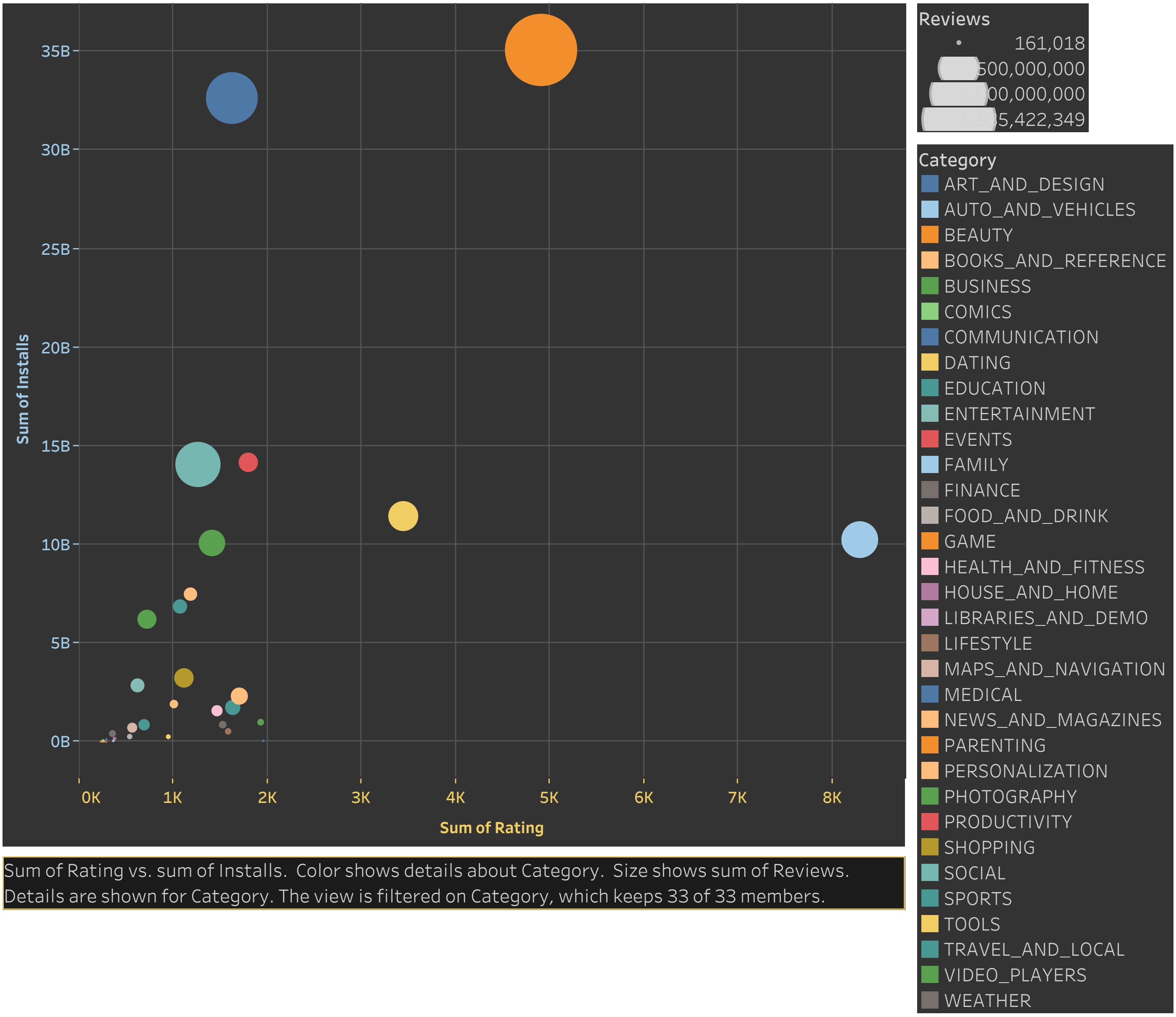


10. Tableau Workbook Screenshot (Dashboard C)



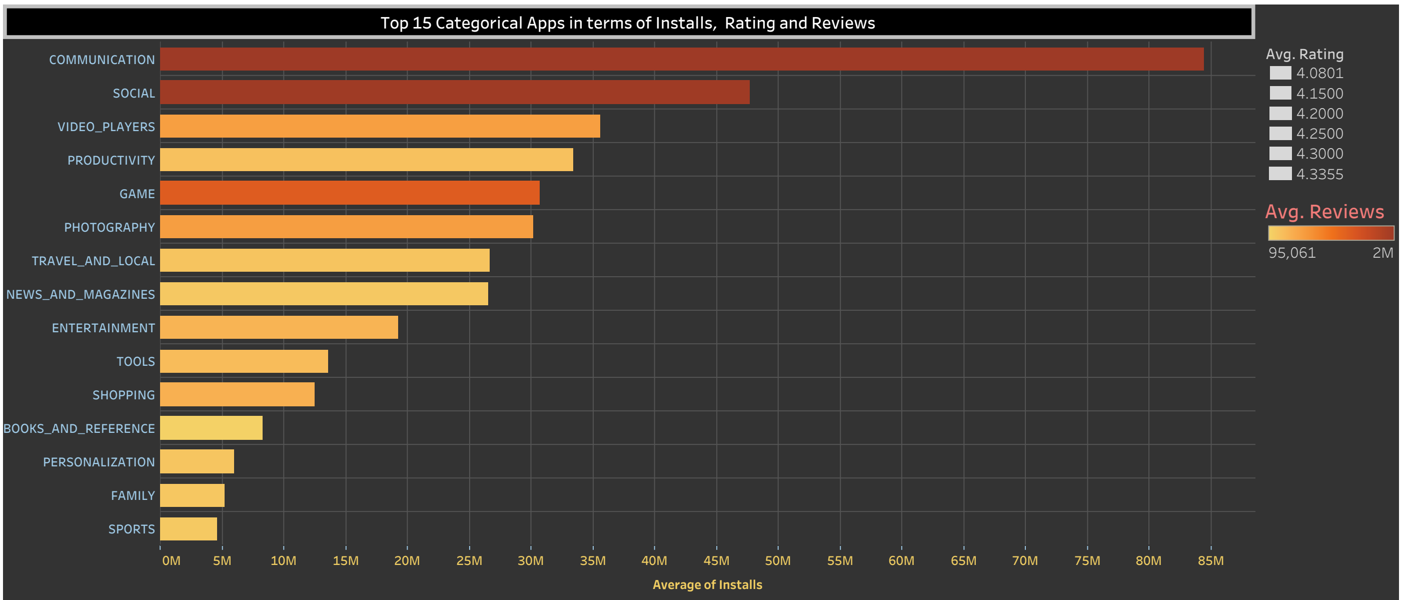
Appendix B: Tableau Generated Report

Categorical Apps in terms of Installs, Rating and Reviews



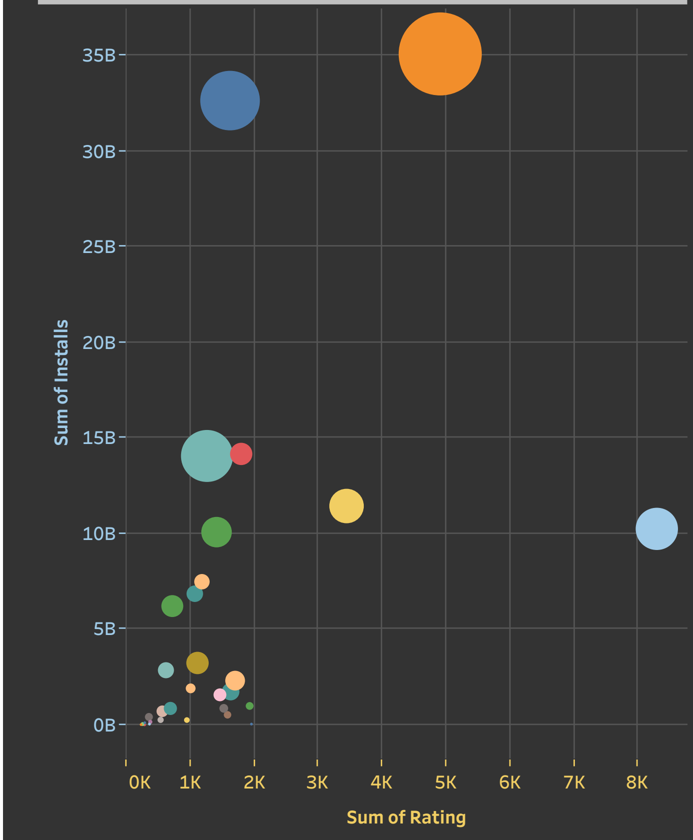
Sum of Rating vs. sum of Installs. Color shows details about Category. Size shows sum of Reviews. Details are shown for Category. The view is filtered on Category, which keeps 33 of 33 members.

Top 15 Categorical Apps in terms of Installs, Rating and Reviews

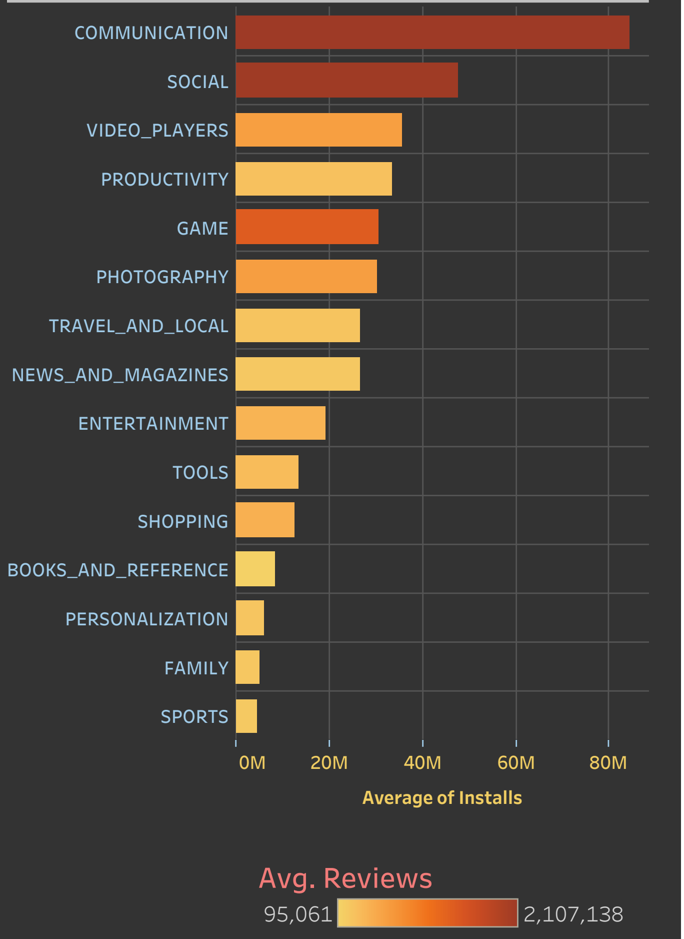


Average of Installs for each Category. Color shows average of Reviews. Size shows average of Rating. The view is filtered on Category, which keeps 15 of 33 members.

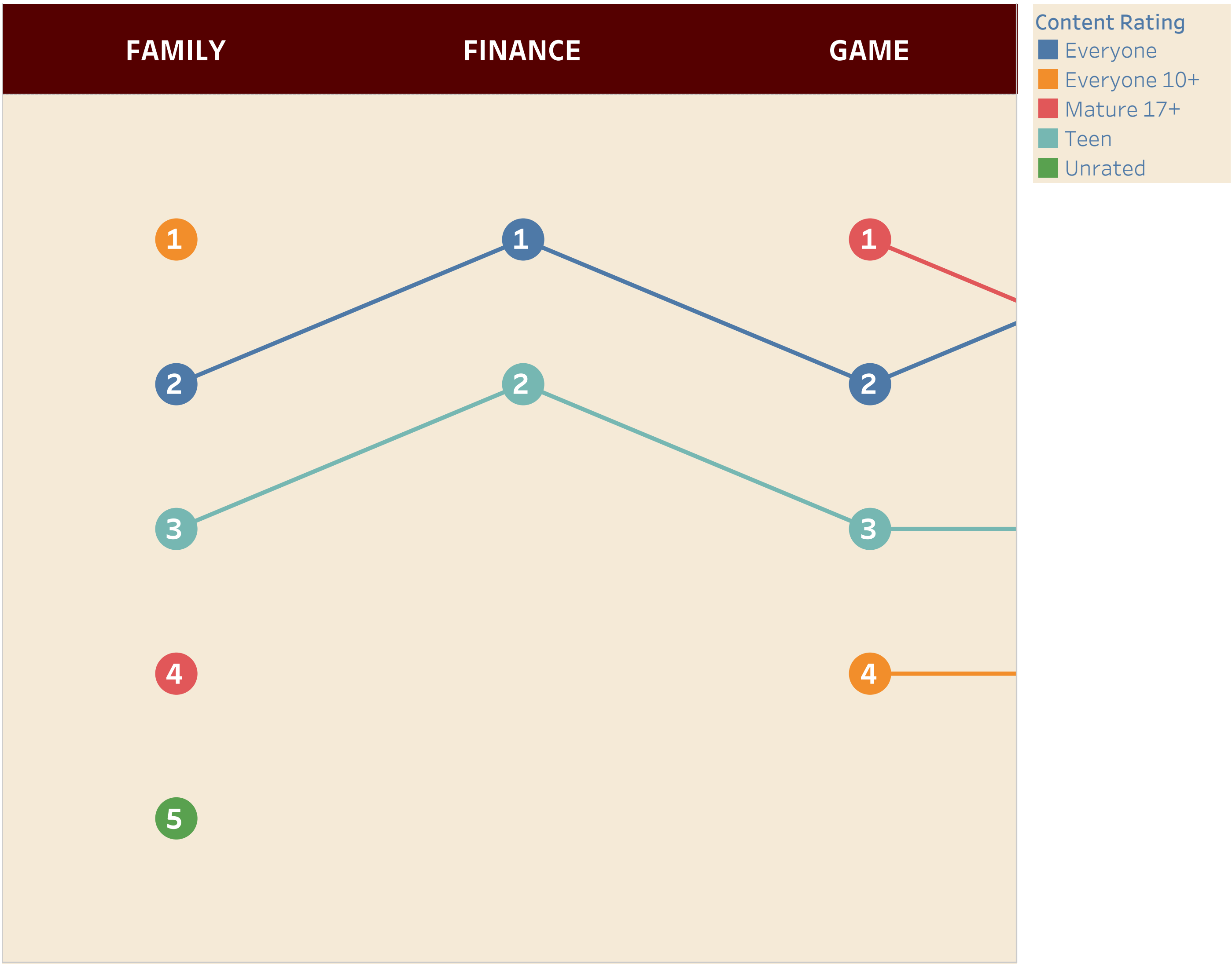
Categorical Apps in terms of Installs, Rating and Reviews



Top 15 Categorical Apps in terms of Installs, Rating and Reviews

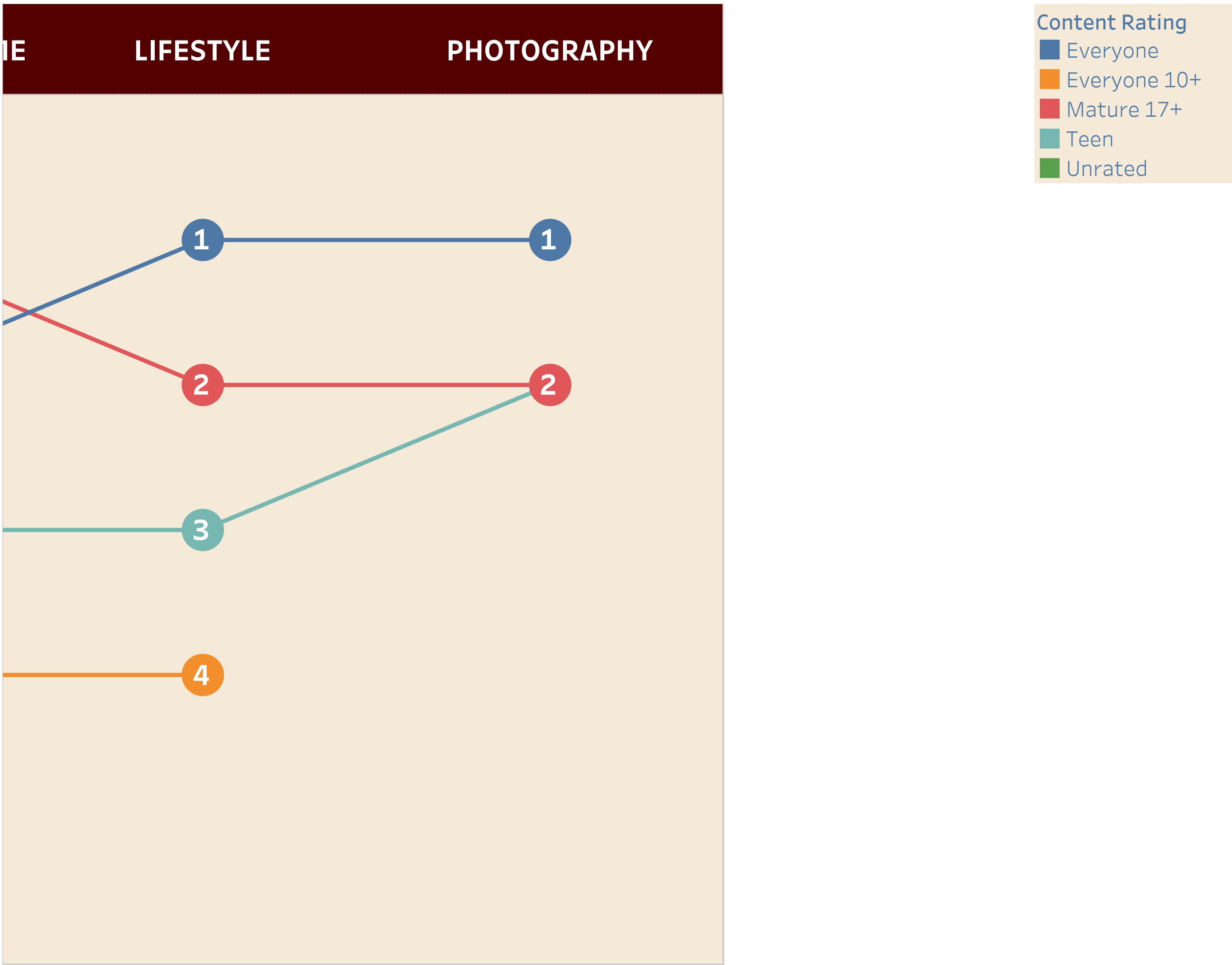


Google App store’s most Profitable Customer

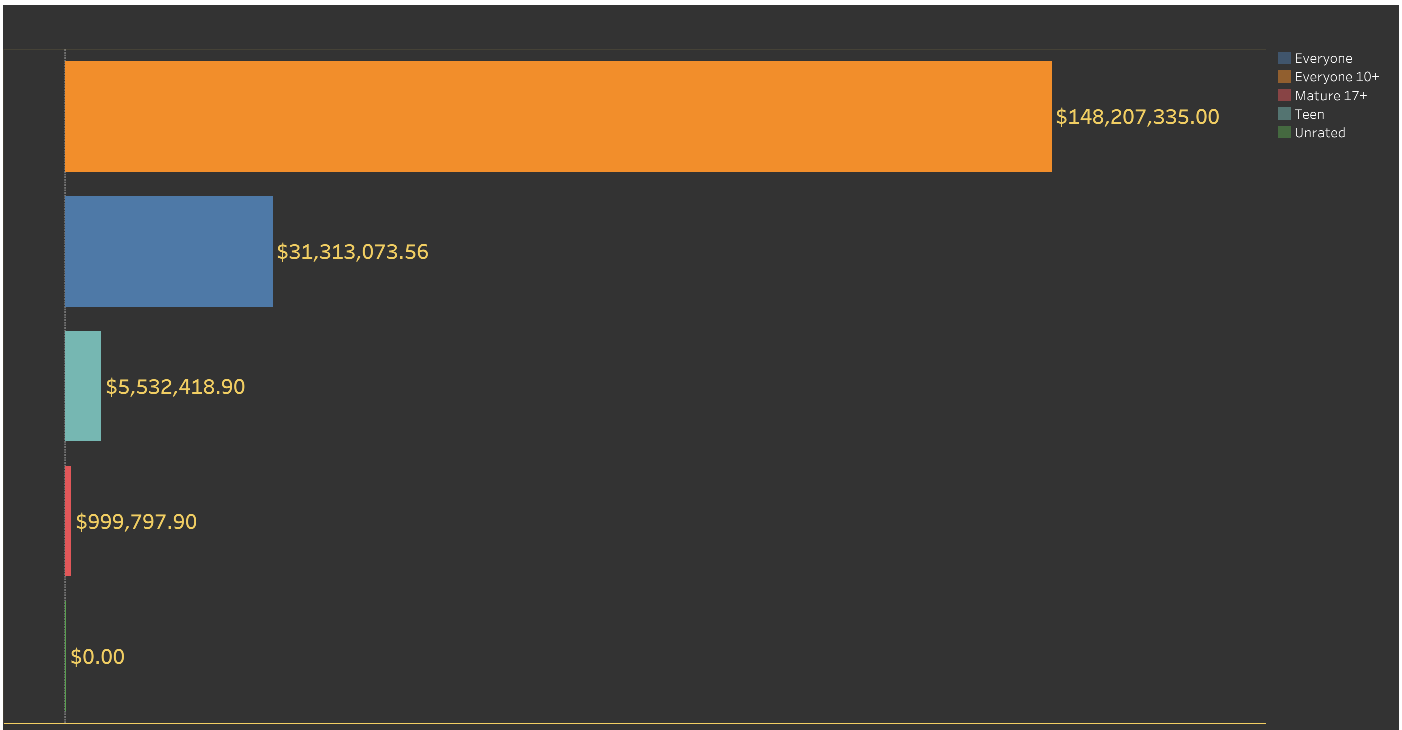


The trends of Rank of Sales and Rank of Sales for Category. Color shows details about Content Rating. The view is filtered on Category, which keeps FAMILY, FINANCE, GAME, LIFESTYLE and PHOTOGRAPHY.

Google App store’s most Profitable Customer



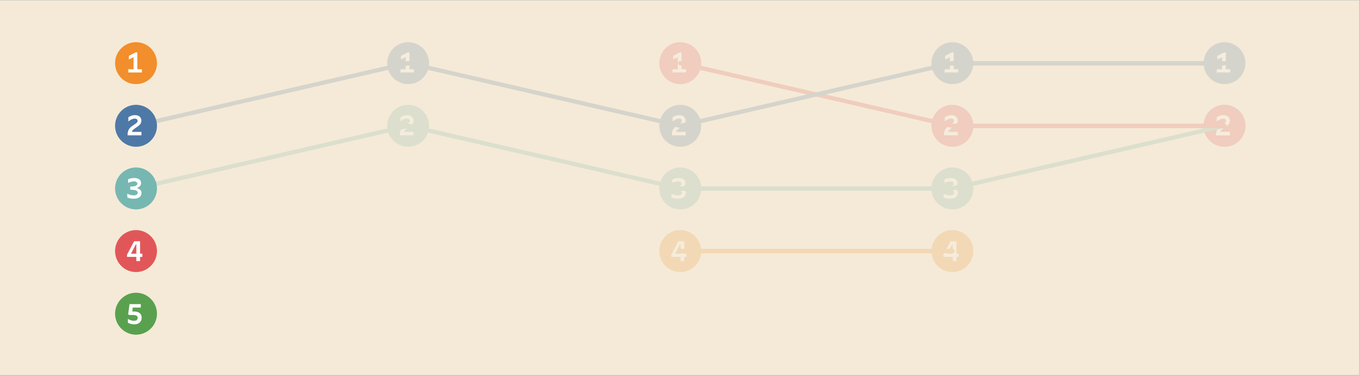
The trends of Rank of Sales and Rank of Sales for Category. Color shows details about Content Rating. The view is filtered on Category, which keeps FAMILY, FINANCE, GAME, LIFESTYLE and PHOTOGRAPHY.

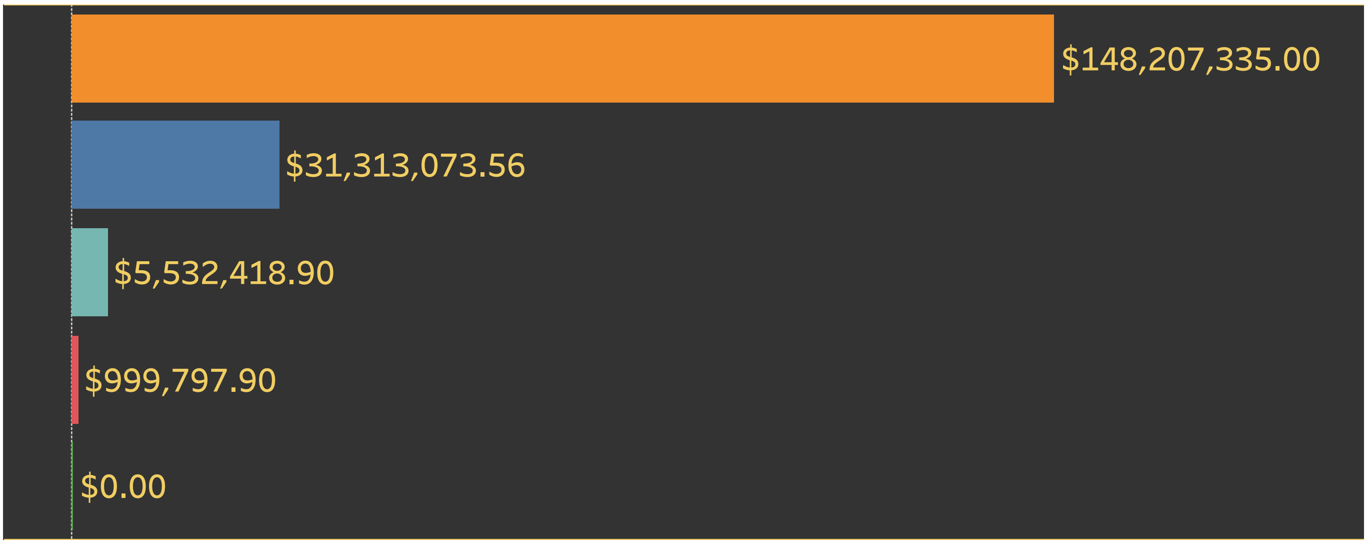


Sum of Sales and sum of Sales for each Content Rating broken down by Category. Color shows details about Content Rating. The data is filtered on Action (Category,Content Rating), which keeps 5 members. The view is filtered on Category and Content Rating. The Category filter keeps FAMILY, FINANCE, GAME, LIFESTYLE and PHOTOGRAPHY. The Content Rating filter keeps multiple members.

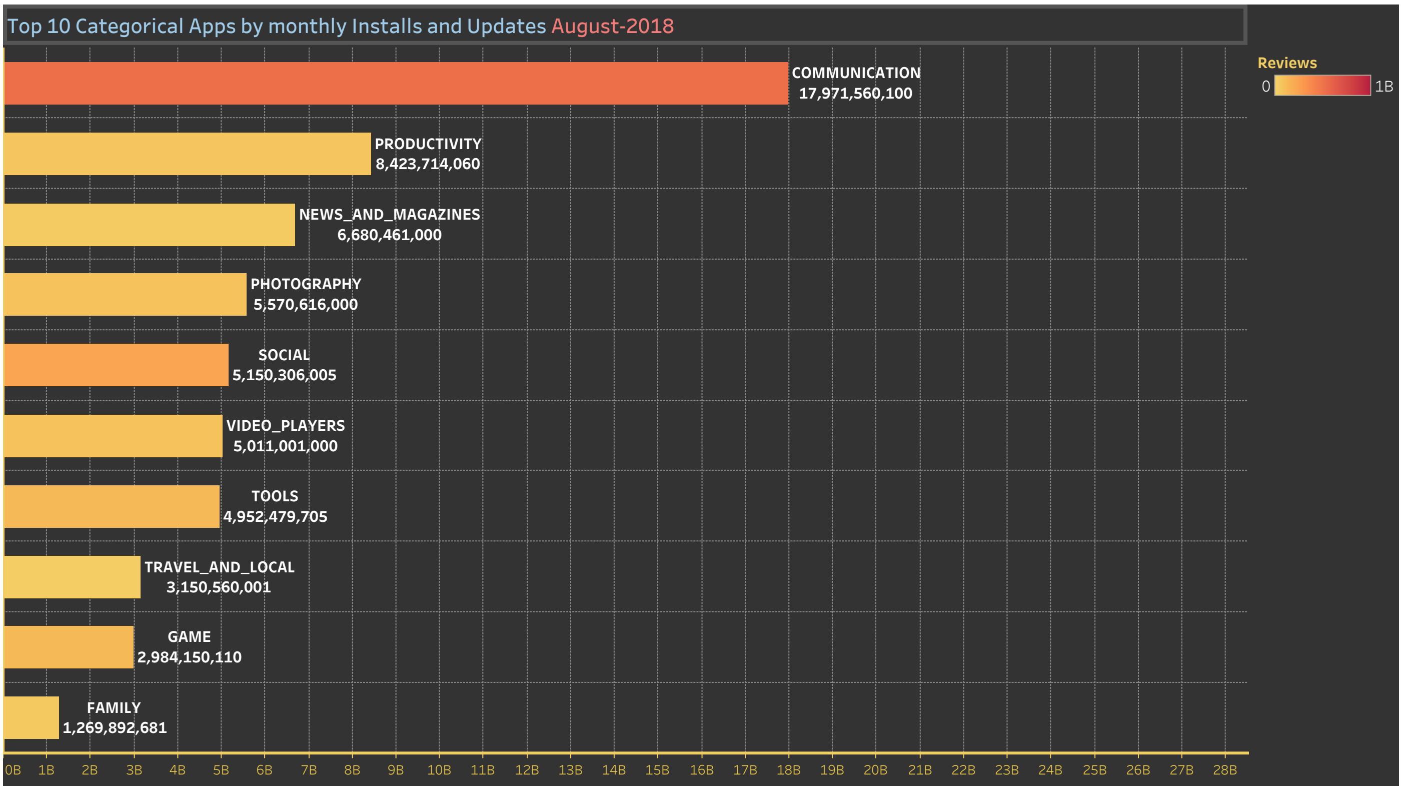
Google App store’s most Profitable Customer

FAMILY FINANCE GAME LIFESTYLE PHOTOGRAPHY

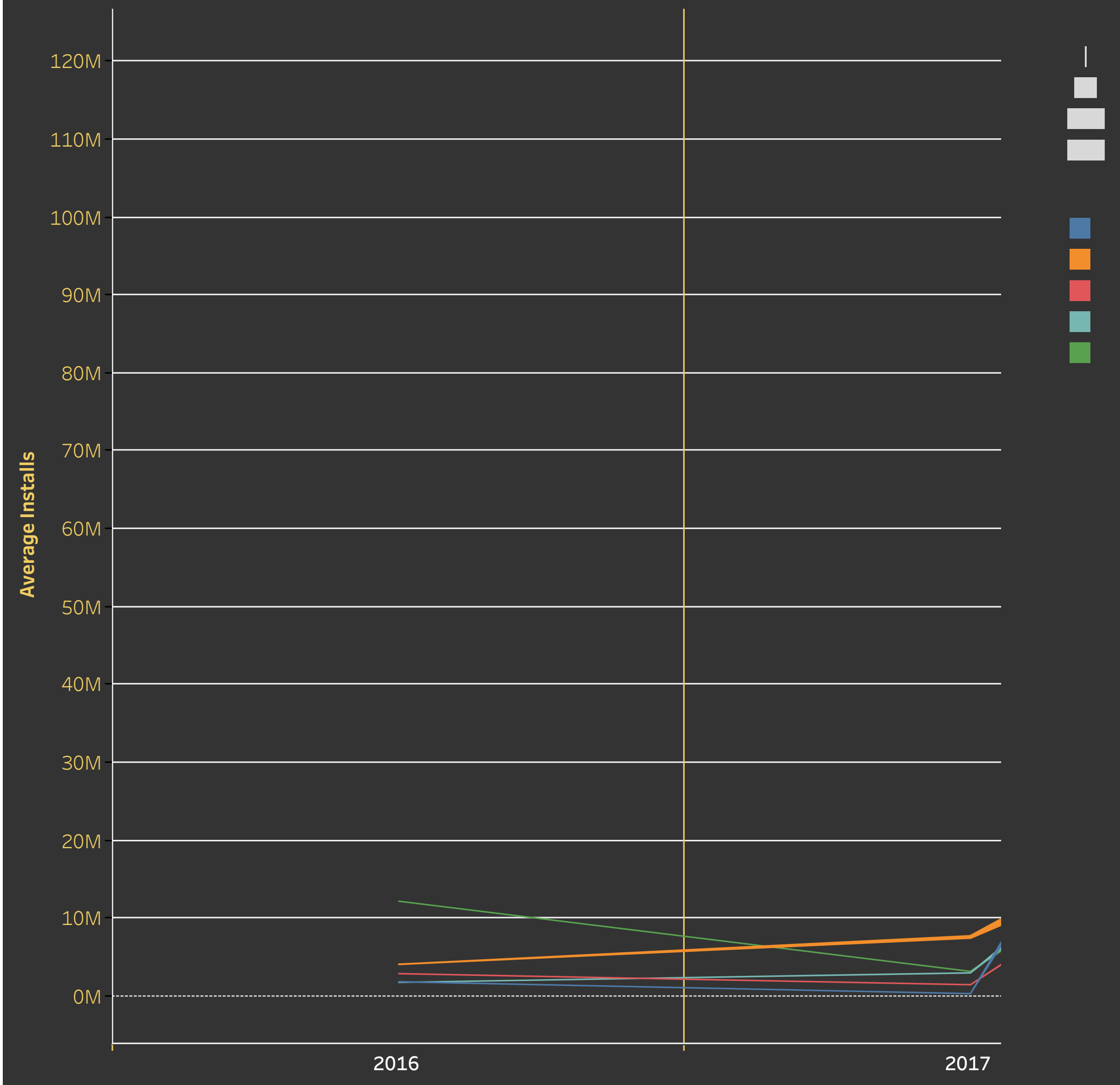


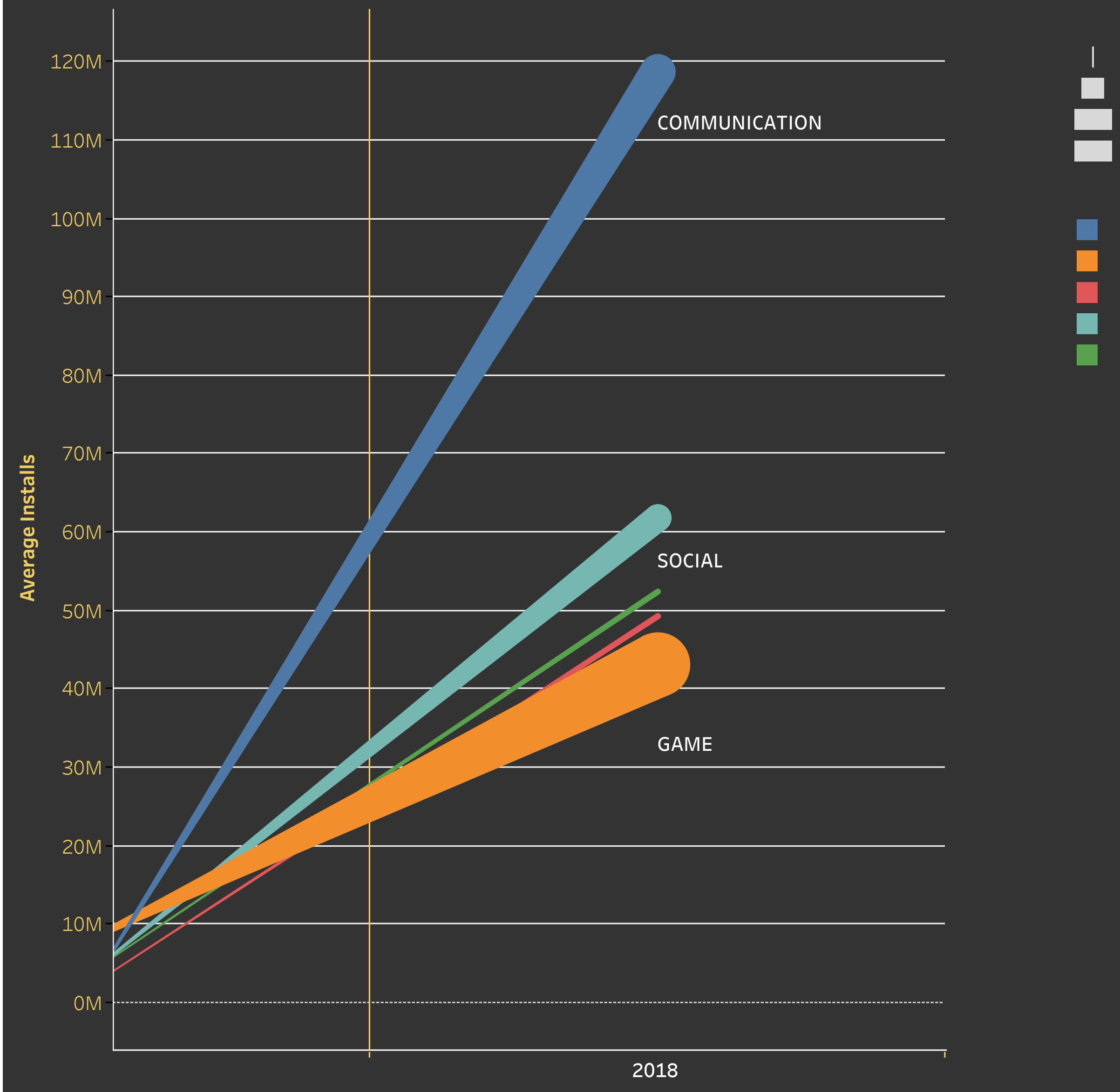


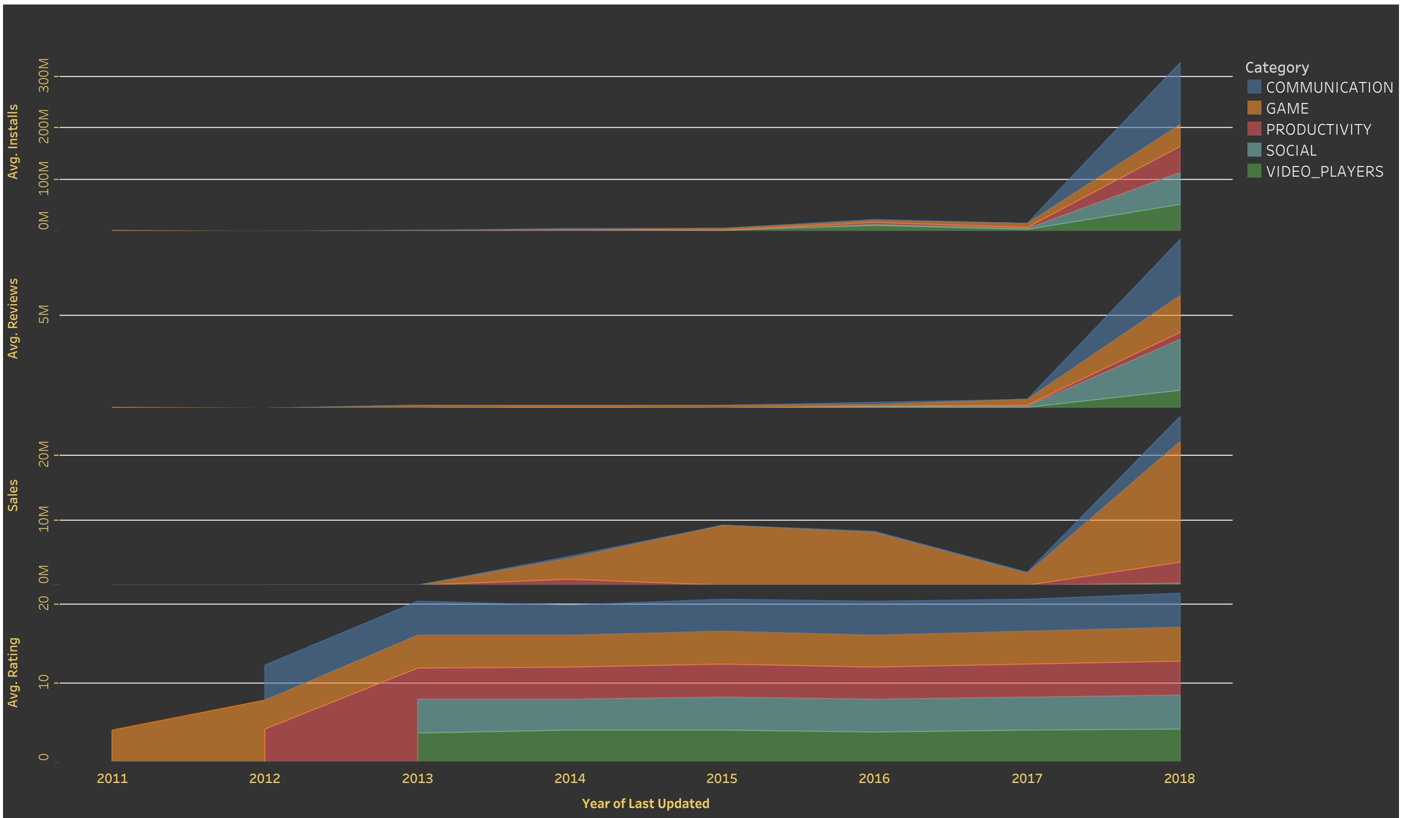
Top 10 Categorical Apps by monthly Installs and Updates August-2018



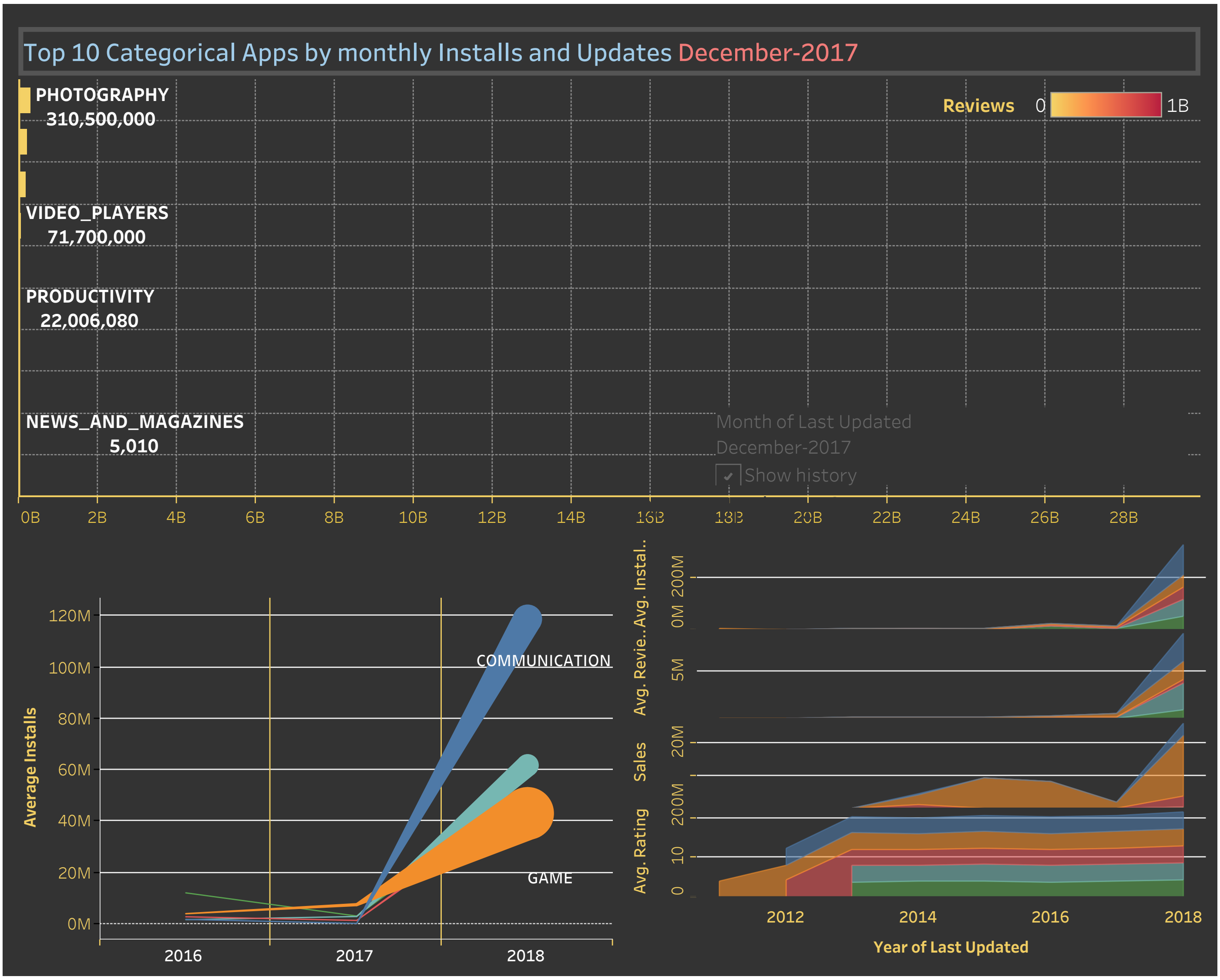
Sum of Installs for each Rank. Color shows sum of Reviews. The marks are labeled by Category and sum of Installs. The view is filtered on Category, which keeps 10 of 33 members.


Average Installs
120M
110M
100M
90M
80M
70M
60M
50M
40M
30M
20M
10M
0M
2016
2017


Average Installs
120M
110M
100M
90M
80M
70M
60M
50M
40M
30M
20M
10M
0M
COMMUNICATION
SOCIAL
GAME
2018



The plots of average of Installs, average of Reviews, sum of Sales and average of Rating for Last Updated Year. Color shows details about Category. The view is filtered on Category and sum of Reviews. The Category filter keeps COMMUNICATION, GAME, PRODUCTIVITY, SOCIAL and VIDEO_PLAYERS. The sum of Reviews filter includes everything.



Top 10 Categorical Apps by monthly Installs and Updates December-2017

Appendix C:

Code Report by Google Colab

```
# Name: Md Mozahidur Rahman (ID: 501002626)
### Project Name: Advance Data Analysis and Visualization
### Project Dataset : Google Play store
### Dataset Source : https://www.kaggle.com/lava18/google-play-store-apps
```

```
## EDA
```

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
#ignore warning messages
import warnings
warnings.filterwarnings('ignore')
import plotly
plotly.offline.init_notebook_mode(connected=True)
import plotly.graph_objs as go
from wordcloud import WordCloud
```

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

```
df.head(5)
```

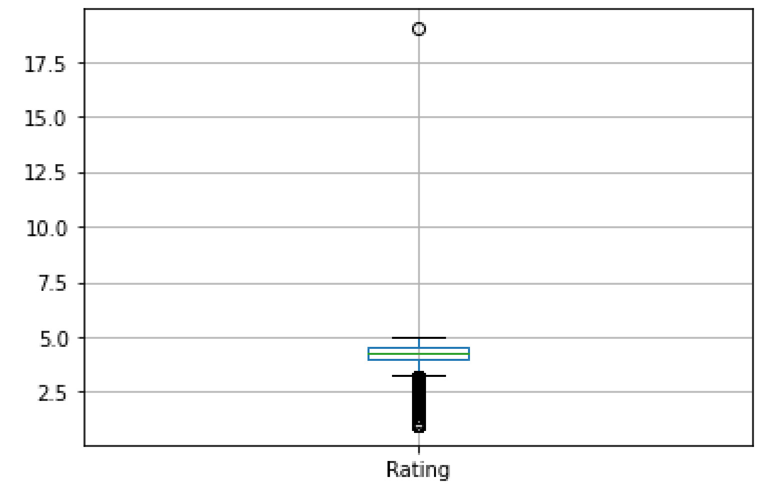
	App	Category	Rating	Reviews	Size	Installs	Type	Price	Conten Ratin
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19M	10,000+	Free	0	Everyon
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14M	500,000+	Free	0	Everyon
	U Launcher								

```
df.describe()
```

	Rating
count	9367.000000
mean	4.193338
std	0.537431
min	1.000000
25%	4.000000

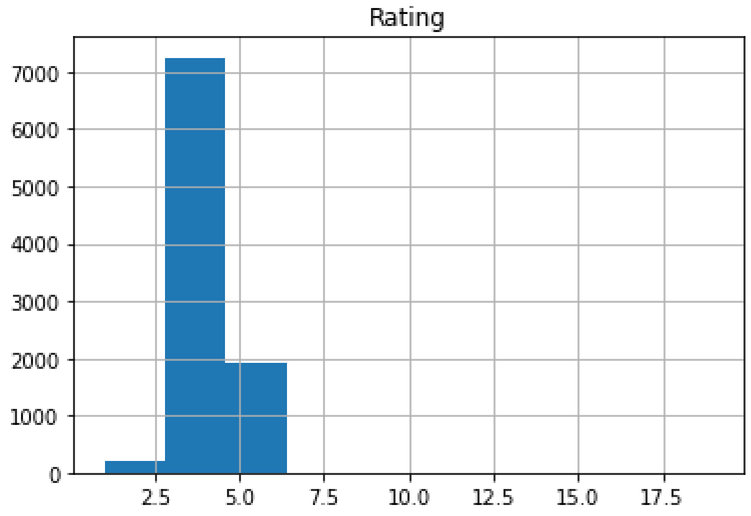
```
df.boxplot()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f34ce9bb190>
```



```
df.hist()
```

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x7f34ce938610>]],
      dtype=object)
```



```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10841 entries, 0 to 10840
```

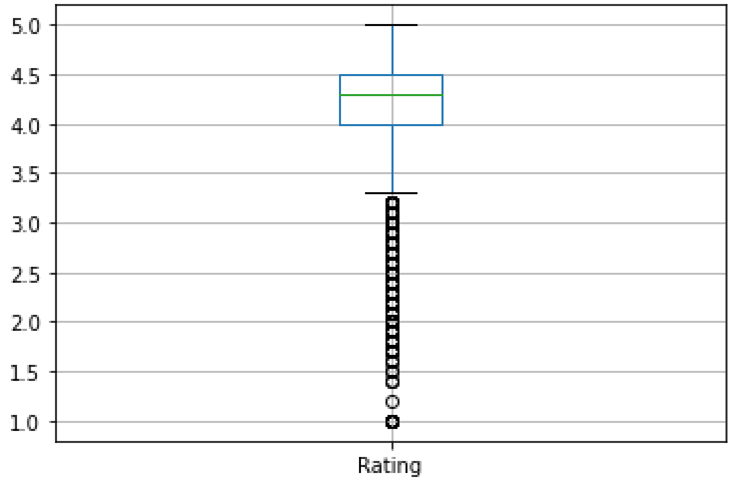
```
Data columns (total 13 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   App             10841 non-null  object 
 1   Category        10841 non-null  object 
 2   Rating          9367 non-null   float64
 3   Reviews         10841 non-null  object 
 4   Size            10841 non-null  object 
 5   Installs        10841 non-null  object 
 6   Type            10840 non-null  object 
 7   Price           10841 non-null  object 
 8   Content Rating  10840 non-null  object 
 9   Genres          10841 non-null  object 
 10  Last Updated    10841 non-null  object 
 11  Current Ver     10833 non-null  object 
 12  Android Ver     10838 non-null  object 
dtypes: float64(1), object(12)
memory usage: 1.1+ MB
```

```
df.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
```

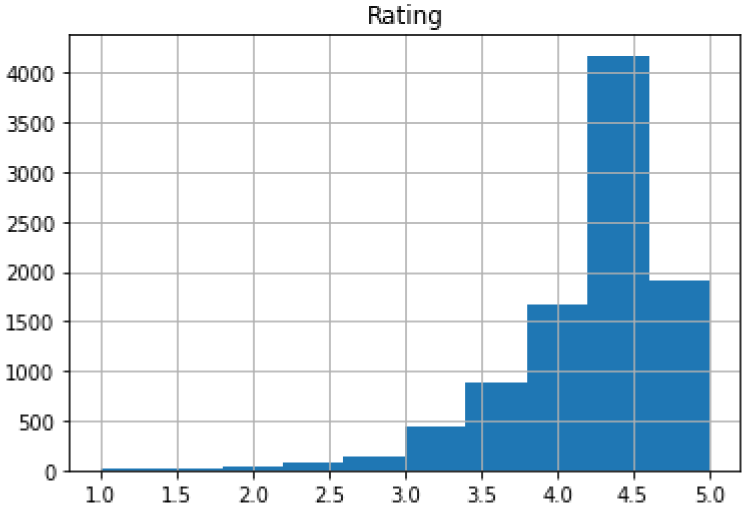
```
df.boxplot()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f34cd671d50>
```



```
df.hist()
```

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x7f34cd62d850>]],
      dtype=object)
```



```
#Define a function for impute median
def impute_median(series):
    return series.fillna(series.median())
```

```
df.Rating=df['Rating'].transform(impute_median)
```

```
# Count Null again
df.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
```

```
# Impute mode to the categical value
# Here 'Type','Current Ver' and 'Android Ver' contains null value
# Print mode value of these three column
print(df['Type'].mode())
print(df['Current Ver'].mode())
print(df['Android Ver'].mode())
```

```
0    Free
```

```
dtype: object
0    Varies with device
dtype: object
0    4.1 and up
dtype: object
```

```
# Filling the categorical null value with their mode value
df['Type'].fillna(str(df['Type'].mode()),inplace=True)
df['Current Ver'].fillna(str(df['Current Ver'].mode()),inplace=True)
df['Android Ver'].fillna(str(df['Android Ver'].mode()),inplace=True)
# Data transformation from categorical to numerical
df['Price']=df['Price'].apply(lambda x:str(x).replace('$',''))
df['Price']=df['Price'].apply(lambda x:float(x))
df['Reviews']=pd.to_numeric(df['Reviews'],errors='coerce')
df['Installs']=df['Installs'].apply(lambda x:str(x).replace('+',''))
df['Installs']=df['Installs'].apply(lambda x:str(x).replace(',',''))
df['Installs']=df['Installs'].apply(lambda x:float(x))
```

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

```
array(['Free', 'Paid', '0    Free\ndtype: object'], dtype=object)
```

```
df['Type']=df['Type'].apply(lambda x:str(x).replace('0    Free\ndtype: object','Free'))
```

```
from scipy.stats import pearsonr
from scipy.stats import spearmanr
from sklearn.preprocessing import MinMaxScaler
import scipy.stats as ss
import math
```

```
subset_columns = ["Rating","Reviews","Installs","Price"]
df_subset = df[subset_columns]
df_subset.head()
```

	Rating	Reviews	Installs	Price
0	4.1	159	10000.0	0.0
1	3.9	967	500000.0	0.0
2	4.7	87510	5000000.0	0.0
3	4.5	215644	50000000.0	0.0
4	4.3	967	100000.0	0.0

```
# create a scaler object
scaler = MinMaxScaler()
# fit and transform the data
df_subset_scaled = pd.DataFrame(scaler.fit_transform(df_subset), columns=df_subset.columns)
```

```
df_subset_scaled.head()
```

	Rating	Reviews	Installs	Price
0	0.775	0.000002	0.00001	0.0
1	0.725	0.000012	0.00050	0.0
2	0.925	0.001120	0.00500	0.0
3	0.875	0.002759	0.05000	0.0
4	0.825	0.000012	0.00010	0.0

```
df_subset_scaled.corr('pearson')
```

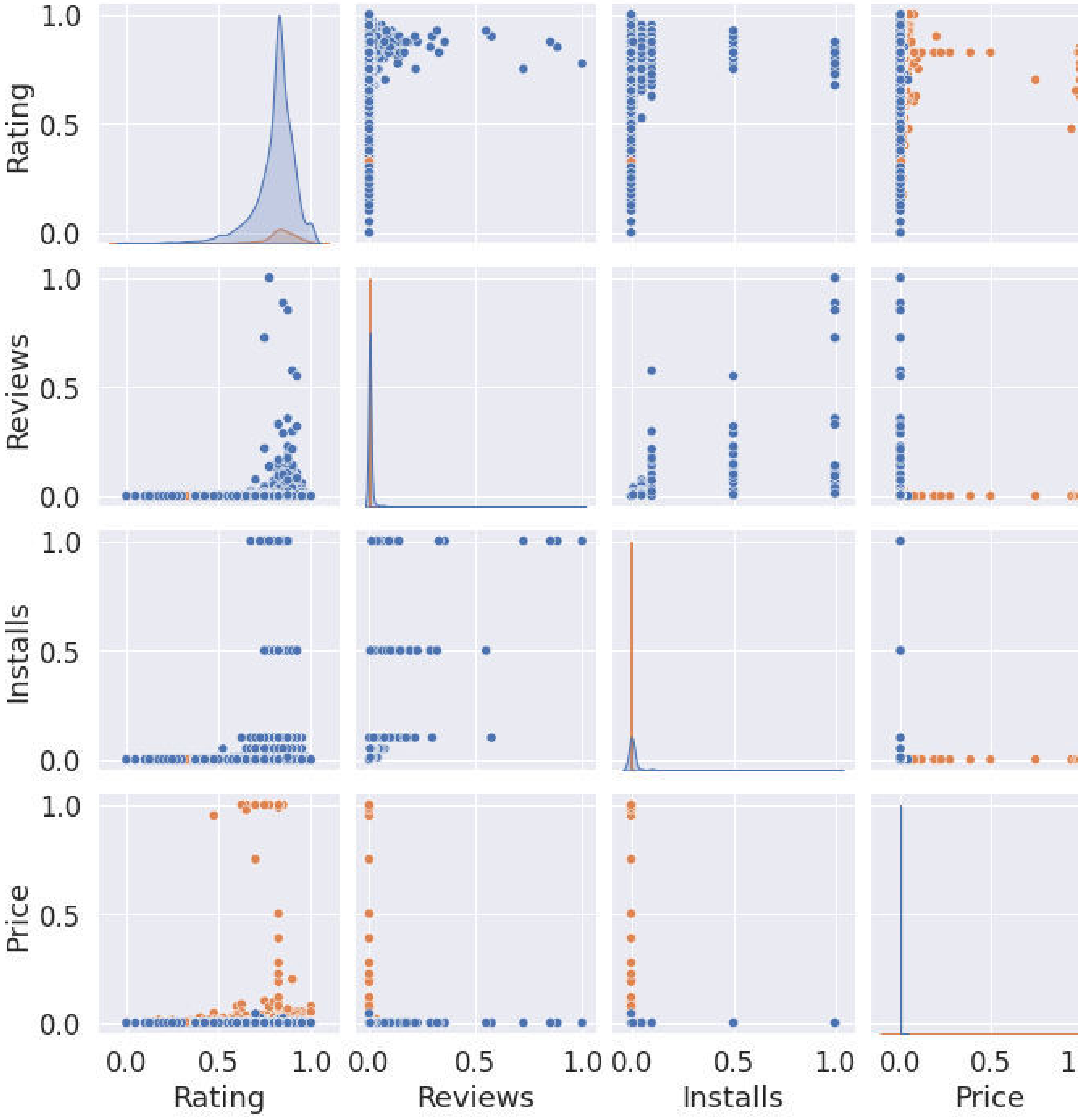
	Rating	Reviews	Installs	Price
Rating	1.000000	0.063166	0.045496	-0.019318
Reviews	0.063166	1.000000	0.643122	-0.009667
Installs	0.045496	0.643122	1.000000	-0.011689
Price	-0.019318	-0.009667	-0.011689	1.000000

```
sns.set(font_scale=1.5)
fig, ax = plt.subplots()
fig.set_size_inches(10, 6)
sns.heatmap(df_subset_scaled.corr(method='pearson'), annot=True, fmt='.4f',
            cmap=plt.get_cmap('coolwarm'), cbar=False, ax=ax)
ax.set_yticklabels(ax.get_yticklabels(), rotation="horizontal")
ax.set_xticklabels(ax.get_xticklabels())
```

```
[Text(0.5, 0, 'Rating'),
 Text(1.5, 0, 'Reviews'),
 Text(2.5, 0, 'Installs'),
 Text(3.5, 0, 'Price')]
```

```
subset_columns2 = ["Type"]
df_subset2 = df[subset_columns2]
df_new = df_subset_scaled.join(df_subset2, how='outer', lsuffix='_df_subset_scaled', rsuffix=
sns.pairplot(df_new,hue='Type')
```

```
<seaborn.axisgrid.PairGrid at 0x7f34cc207550>
```



```
import matplotlib.pyplot as plt
import seaborn as sns
import seaborn as sns
import plotly
import plotly.offline as py
py.init_notebook_mode(connected=True)
import plotly.graph_objs as go
from plotly import tools
import plotly.figure_factory as ff
from plotly.offline import init_notebook_mode, iplot
%matplotlib inline
```

```
colors = ['CornflowerBlue', 'lightgrey', 'DarkSalmon', '#D0F9B1', 'DarkSeaGreen', 'RosyBrown'
Category = df['Category'].value_counts()[:]
label = Category.index
size = Category.values
trace = go.Pie(labels=label,
               values=size,
               marker=dict(colors=colors))
data = [trace]
layout = go.Layout(title=' App Category Distribution')

fig = go.Figure(data=data, layout=layout)
py.iplot(fig)
```

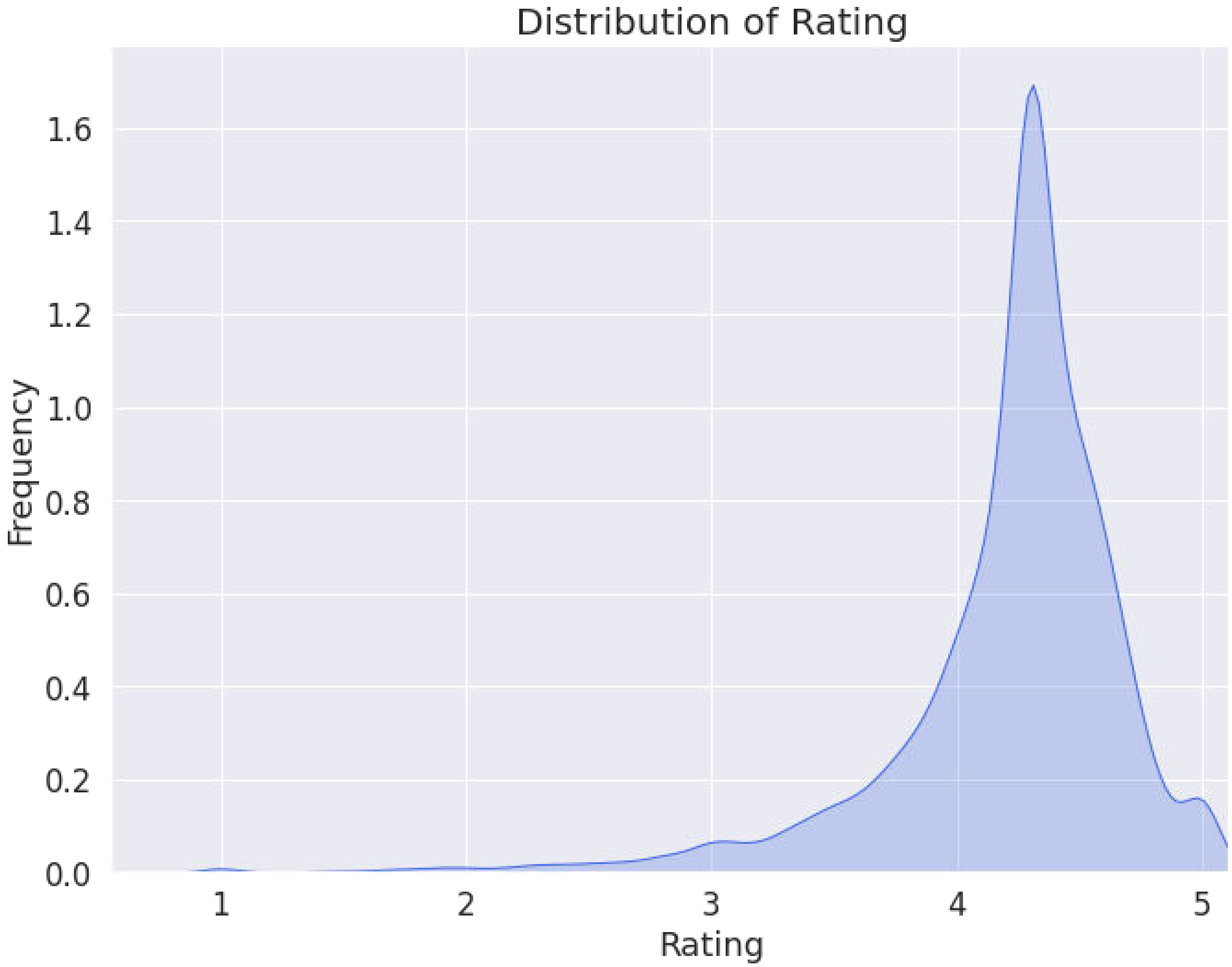
```
Type = df['Type'].value_counts()
label = Type.index
size = Type.values
colors = ['LightSkyBlue', 'Khaki']

trace = go.Pie(labels=label,
               values=size,
               marker=dict(colors=colors)
              )

data = [trace]
layout = go.Layout(title='Type',
                   legend=dict(orientation="h")
                  )

fig = go.Figure(data=data, layout=layout)
py.iplot(fig)
```

```
Content = df['Content Rating'].str.split(',')
Content_set = []
```

```
for i in Content.dropna():
    Content_set.extend(i)
    Content = pd.Series(Content_set).value_counts()[:6]

label = Content.index
size = Content.values

colors = ['#FEBFB3', 'skyblue', '#96D38C', '#D0F9B1', 'tan', 'lightgrey']

trace = go.Pie(labels=label,
              values=size,
              marker=dict(colors=colors)
             )

data = [trace]
layout = go.Layout(
    title='User Audience ',
    legend=dict(orientation="h")
)

fig = go.Figure(data=data, layout=layout)
py.iplot(fig)
```

```
from pylab import rcParams
# rating distibution
rcParams['figure.figsize'] = 11.7,8.27
g = sns.kdeplot(df.Rating, color="RoyalBlue", shade = True)
g.set_xlabel("Rating")
g.set_ylabel("Frequency")
plt.title('Distribution of Rating',size = 20)
```

```
Text(0.5, 1.0, 'Distribution of Rating')
```

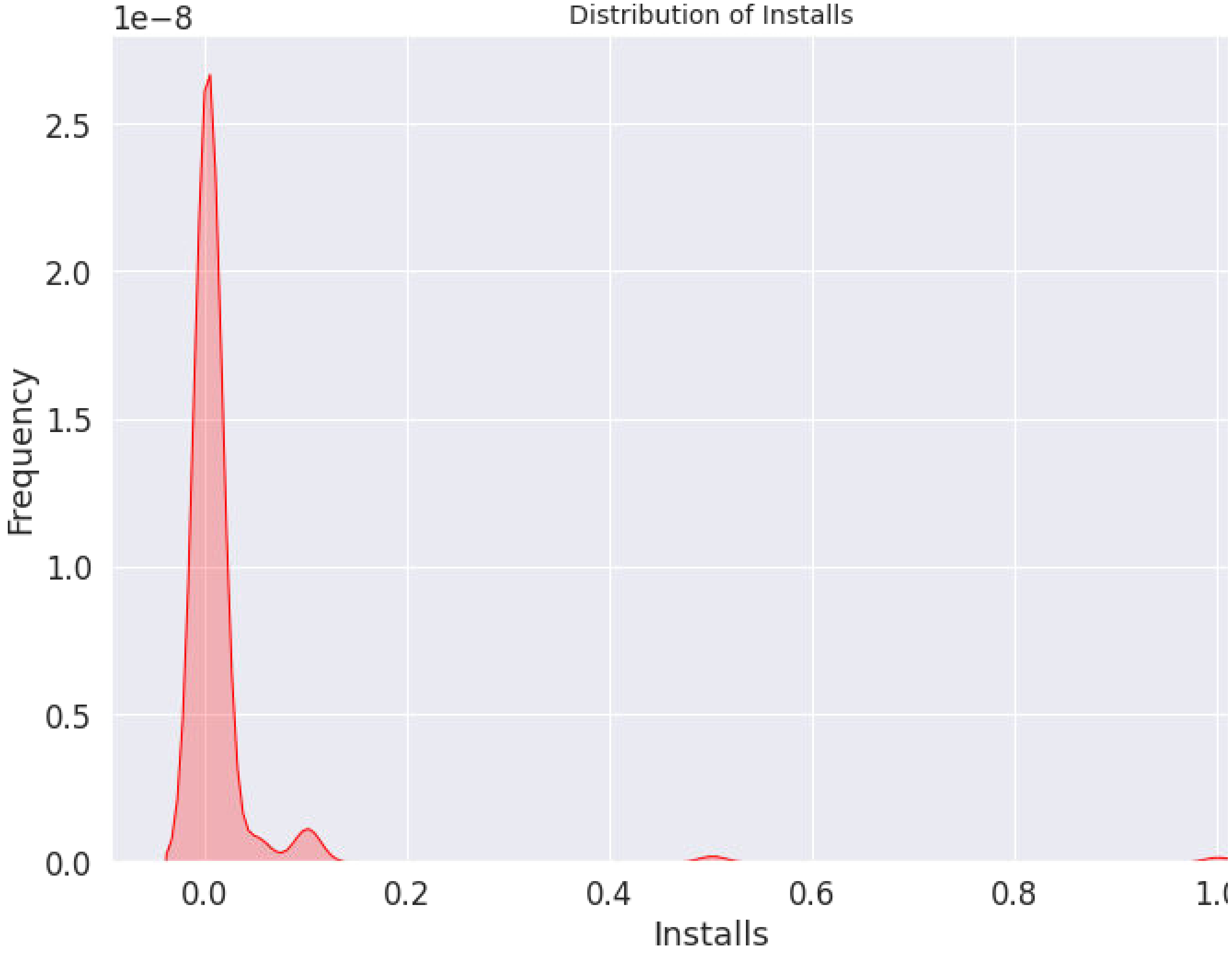


```
Content = df['Content Rating'].str.split(',')
Content_set = []
for i in Content.dropna():
    Content_set.extend(i)
    Content = pd.Series(Content_set).value_counts()[:6]

label = Content.index
size = Content.values

colors = ['#FEBFB3', 'skyblue', '#96D38C', '#D0F9B1', 'tan', 'lightgrey']
```

```
trace = go.Pie(labels=label,
               values=size,
               marker=dict(colors=colors)
              )

data = [trace]
layout = go.Layout(
    title='User Audience ',
    legend=dict(orientation="h")
)

fig = go.Figure(data=data, layout=layout)
py.iplot(fig)
```

```
from pylab import rcParams
# rating distibution
rcParams['figure.figsize'] = 11.7,8.27
g = sns.kdeplot(df.Installs, color="Red", shade = True)
g.set_xlabel("Installs")
g.set_ylabel("Frequency")
plt.title('Distribution of Installs',size = 14)
```

```
Text(0.5, 1.0, 'Distribution of Installs')
```


Distribution of Installs
1e−8
Frequency
Installs