Midterm Project

Discovering Google Play Store Apps and Their Properties

Team Math Kids

Nicolas Chasteler, Dat Tran

November 7, 2019

1 Introduction

The data set that we explored originates from Kaggle, which is a web scraped data from Google Play Store that contains 10842 apps that were publically available on September 2018. The attributes include genre (Category), rating (Rating), numbers of reviews (Reviews), size (Size), number of installs (Installs), either free or paid (Type), cost of purchase (Price), maturity rating (Content Rating), the last date updated (Last Updated), etc. We also needed to find out the actual number of positive and negative reviews each app has. For this purpose, one other dataset, googleplaystore_users_review, is joined together with the main googleplaystore dataset. This user review dataset contains the names of the app, English translated reviews, sentiment (positive, neutral, or negative), sentiment polarity score and sentiment subjectivity score for each review.

From this dataset, we realize that the cost of purchase, rating, and number of reviews as well as their quality and number of installs are the most important factors affected its number of installs and download itself. On the other hand, we discover that the ratings are not strictly going with the number of installs. We find out that the highest rating apps are most likely with 10,000+ install attempts, while apps with 500,000+ install attempts do not have stable ratings. Moreover, we also are curious on how developers balance their apps between the apps' cost of purchase and their number of installs, since most of the apps will be free and maintain their operation through ads, while the rest will be free of ads in the cost of premium charges.

Our goal is to analyze the apps in the Google Play store and find their relationship to the type of apps. This includes relationships between data points and its effect on the overall rating and total number of installs. This information can be used by app developers to determine the category, type and price, content rating, and genre. Our main goal is to answer the question "how does the type of app on the Google Play store is the most successful?" To answer this broad question and we must first answer the questions, "how can we use the data from reviews, ratings, installs, sentiment and sentiment polarity to determine whether an app is successful?", "which attributes affect the users opinions the most of apps such as category, type, price, and content rating?", "what similarities do apps have which have become very popular share and what is the chance of another app with similar aspects becoming popular?". The answers of the questions will provide useful data for app development by revealing some of the variables which create the most success of an app.

2 Data Cleaning

2.1 The main application dataset

We began using the googleplaystore dataset provided on Kaggle, which was downloaded through the link https://www.kaggle.com/lava18/google-play-store-apps, narrowing down the dataset by only choosing the records of applications that are either paid or free (Type = c("Free", "Paid")), since their types are known and specified. We also checked if there are any duplicate rows and removed them.

To further explore the data more efficiently, we also rebuilt several variables summarized as follows,

- Installs: It represents the number of installs each application has. We removed the "+" sign in the variable, and transform the variable into numeric form.
- Categories: It represents the category of each application. We rebuilt Categories, using the number of installs to identify 10 Categories with highest install attempts. We then grouped all other categories that are not included in that top 10 into 1 variable named "OTHER". After that we factorized Categories into 11 levels, which are the top 10 and "OTHER" variable.
- Rating: It represents the rating of each application. We rebuilt Rating, grouped the variable to only 5 rating, 1, 2, 3, 4, and 5.
- Reviews: It indicates the number of reviews each application has. We transformed the variable into characteristic form first from a factor level variable, then into numeric form after.
- Price: It represents the price of each application. We removed the sign "$" in the variable, and transformed the variable into numeric form.

Now we have narrowed down the dataset and standardized variables across all of tuples, using the code as shown in Appendix. We title this dataset as data.clean, since in our code we named the main application (googleplaystore) dataset as data, and this is the cleaned dataset followed after.

2.2 The user review dataset

We began using the googleplaystore_user_review dataset provided on Kaggle, which was downloaded through the link https://www.kaggle.com/lava18/google-play-store-apps. This dataset provides the details of each review of some particular applications, the sentiment of each review (whether it is positive, negative, or neutral), and the quantitative properties of each review, which are the sentiment polarity of each review (whether it's extremely positive (1.0), or extremely negative (-1.0), or in between (from -1.0 to 1.0)), and the sentiment subjectivity of each review (how subjective each review is, range from 0.0 to 1.0). We narrowed down this dataset by only choosing the records of reviews whose sentiment are negative, neutral or positive (Sentiment = c("Negative, "Neutral, "Positive)), since their types are known and specified. We also checked if there are any duplicate rows and removed them.

We title this dataset as subdata.clean, since in our code we named this dataset as subdata, and this is the cleaned version of it.

2.3 The merged dataset

In order to create a link between the two previous dataset, googleplaystore and googleplaystore_user_review dataset, we need to merge them together based on what they have in common. As a result, we made the mergedata.clean dataset based on the merge between the two previous dataset, googleplaystore and googleplaystore_user_review dataset by application name. By using this dataset, we could not only see the number of reviews, ratings, number of installs of each application, but also the actual quality of each review they have. Similar to the main application dataset.

3 Properties of Popular Applications

In this section, we discovered multiple aspects of what defines a popular application, and what keeps the application in the place that it is currently, including top 10 installs per category, installs on price, installs on rating, etc. Since there are 34 categories, in order to make our observations more concise and accurate, we only focused on the first 10 categories that have the most install attempts.

3.1 Top 10 Installs per Category

Figure 1 displays the most installed number of apps by category, where we can clearly see the number of installs of the category ‘Game’ and ‘Communication’ take up more than double every other app category. The difference is major between the number of installs as the ‘Game’ category makes up over 22% of all total installs. Communication is right next to Game, with over 20 billions installing attempts, around 14% of all total installs. This graph shows a very interesting finding, as the majority of users find game and communication applications essential parts when using their phones nowadays. Games in general are extremely popular, and the popularity level even goes higher when it comes to mobile games, since they are all fun and portable, which means users can play them anywhere at anytime. That goes the same with Communication apps, where users can talk with each other online, both anonymously and non-anonymously. But, does the total revenue relate to number of installs? We are eager to find out the answer behind.

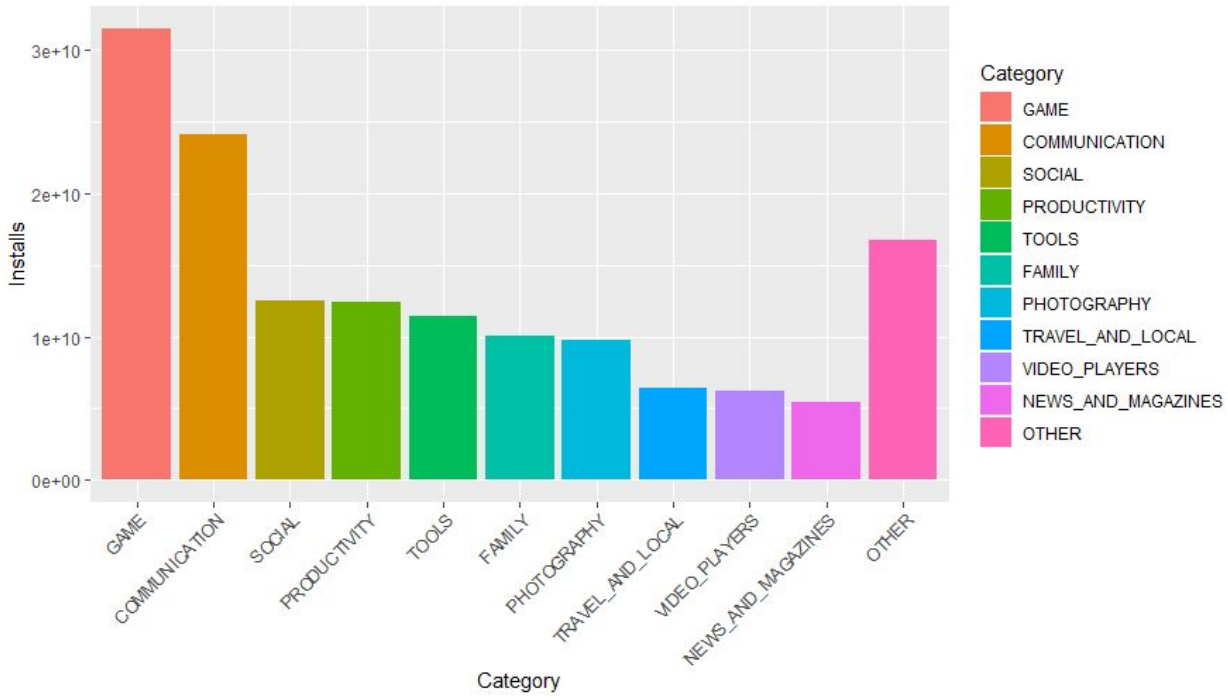

Figure 1: The top 10 categories by number of installs.

3.2 Installs and price

Figure 2 displays the price of applications in each category along with their number of installs, followed up with a curved line to show the relationship between number of installs and price. As we can see here, the majority of applications are free apps, and they also are applications that have the highest numbers of installs. The curved line of supply and demand keeps going down when the price increases from $0.0 to around $1.5 for all categories. This is understandable since compared to free applications, cheap applications don't offer that many more features, and users hate to pay even a small amount for those apps. But one interesting find occurs when the price goes up from $1.5 to about $5: the curved line meets its bottom at $1.5 point, and starts to increase again to its peak at $5 before going down constantly again. This shows that for apps that have price range from $1.5 to $5 are likely offer more features compared with normal free apps, with no ads and such. Therefore users find those apps worth spending $5, a relatively small amount of money, on. This also indicates that if developers want to make the most out of application purchase, along with premium quality, they should price their apps within the range from $1.5 to $5.

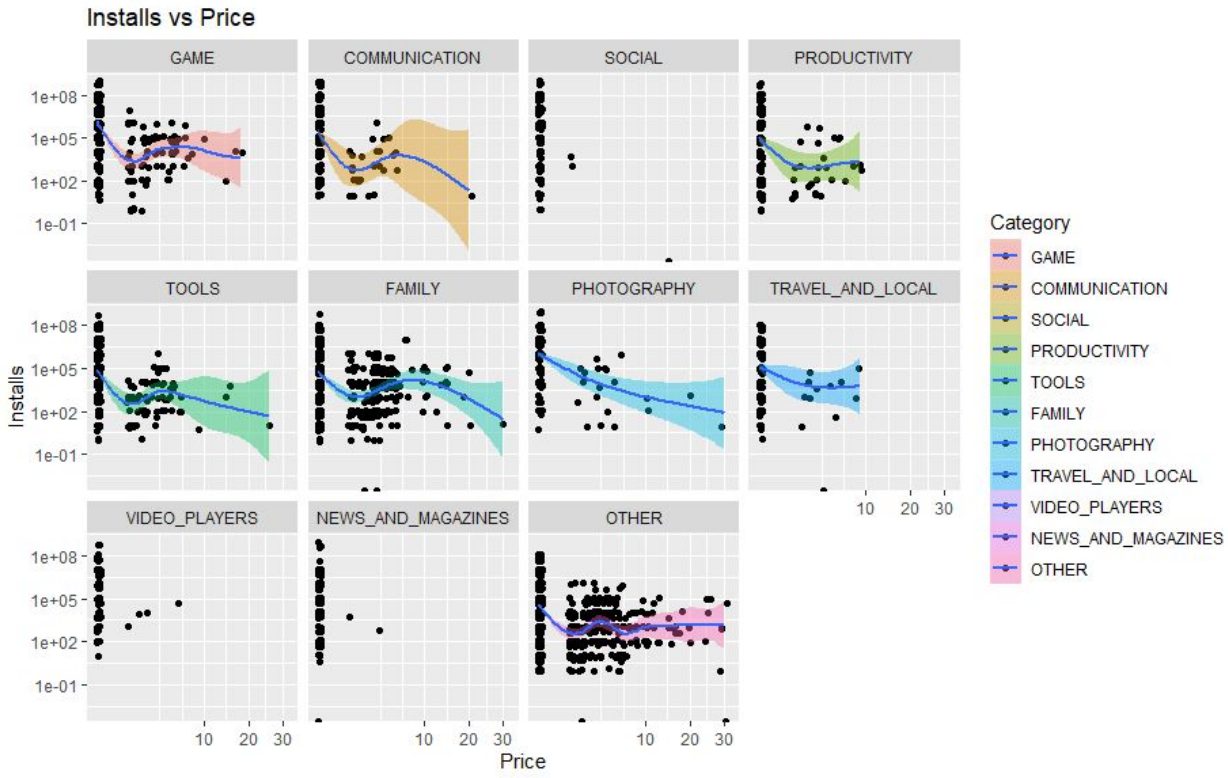

Figure 2: Amount of installs per price separated by category.

3.3 Installs and relation to rating

Figure 3 displays the distribution of rating for each category. As we can see, most of the install comes from applications that has 4 or 5 ratings. One interesting finding here is that except for Game, Tools, and Video Players categories, all other categories' highest install attempts stay on applications which have around 4.0 rating, instead of 5 as we expected. This may be because with highest install attempt applications, users first installed those applications because they were impressed by the install numbers and ratings themselves. But later they found out that the applications were not that enjoyable, and they might have some problems with those apps, and as a result they voted those apps lower than 5. That goes the same with apps with rating of 5. Since those apps don't have large user pools like 4 rating apps, the users who use them should have been enjoying features and the apps themselves. As a result, the rating of those apps stay at 5.

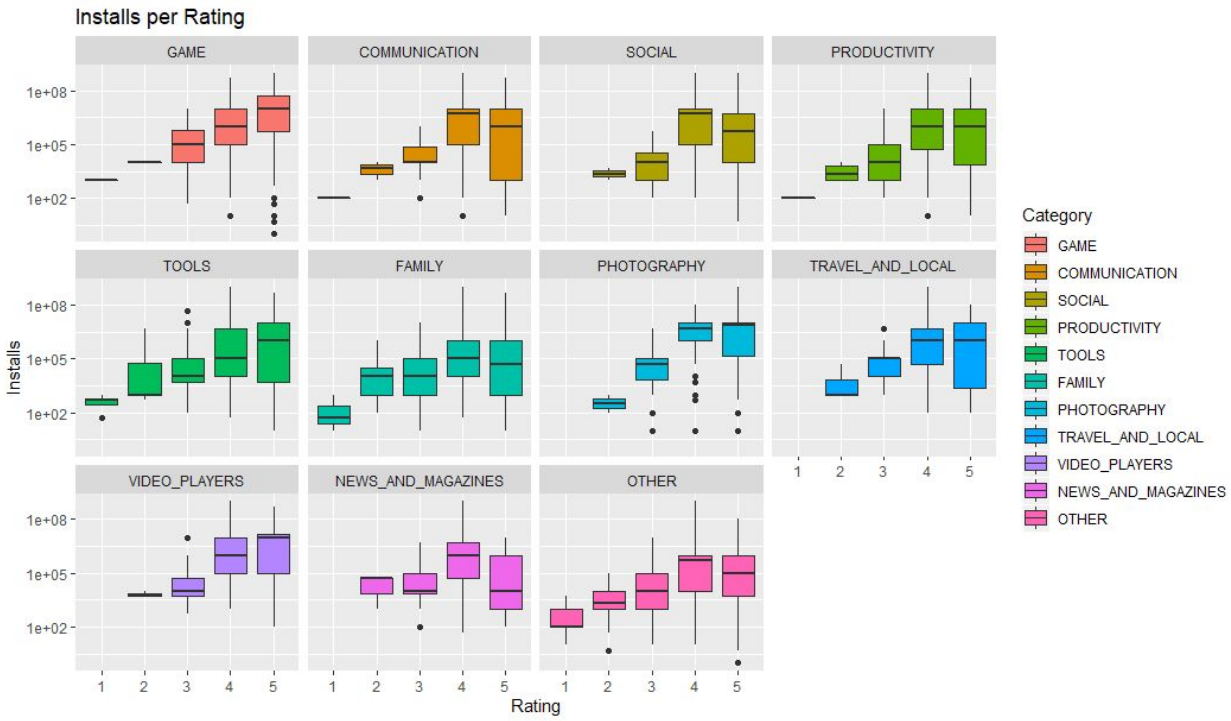

Figure 3: Amount of installs per rating separated by category

3.4 Polarity of written review

Figure 4 displays the polarity of written reviews for each category. Polarity is a quantitative variable that is the parameterization of english translated reviews in numeric format. Polarity ranges from -1.0 (extremely negative) to 1.0 (extremely positive). As we can see here, most of the reviews are in the range of 0.0 to 0.6, from neutral to positive. Most negative reviews are in the range of 0.0 to -0.15. We can see that the ranges of positive reviews for all categories are much larger than the ranges of negative reviews. This interesting finding can give explanation for many of our questions. First of all, if the users are about to write a review, it's most likely neutral or positive/highly positive, and if the review is a negative one, its negative level is substantially low. This means the negative reviews are not extremely or even highly negative, and the users who wrote that may disagree with some of the features in the applications, or have some problems with the apps, and not totally dislike the apps. Developers can use these reviews to see what the drawbacks of their apps and improve them for the better.

Secondly, we can see that Game has generally lower polarity compared to other categories. Its polarity is much more grouped up around 0.0, to the positive side. This makes sense since for games, it's extremely hard to make every player happy with every feature or content of the game. Even the most dedicated players to those games will still have some disagreement with the game itself or on how developers manage the games. This leads to a substantially lower polarity rate for Game categories applications.

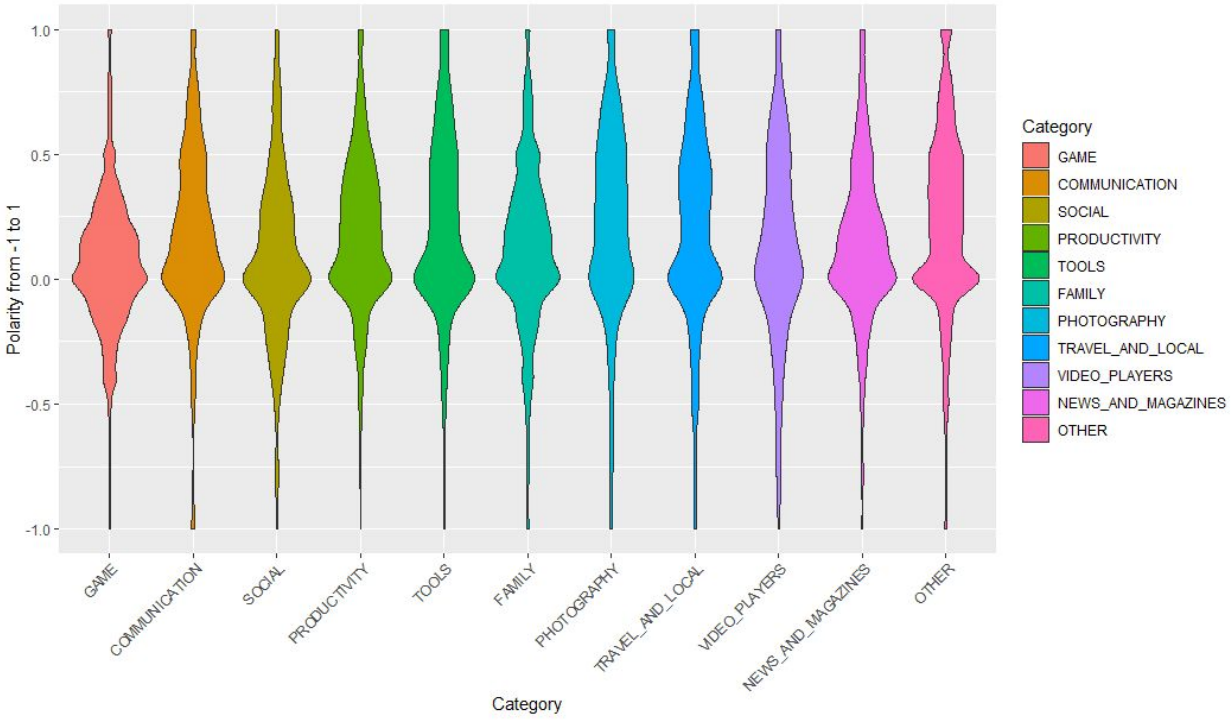

Figure 4: Polarity of written review by category

3.5 Further dead end questions

We failed to come up with the answers to our questions: “What effect does sentiment subjectivity of users’ review have on users’ decision to install the app?”, “How does Android version make people less accessible to the applications?”, “How frequent developers should update their app in order to get the most users to their apps?”, “Do ads have a huge impact on the users’ experience when using the apps”, “What size should the developers make their app to make the most users install their apps?”, “What genres are most popular in each category?”. We failed because we either did not have enough data to answer our questions (advertisement for example) or we could not find the exact ways to implement the data to our answer.

4 Comparative Analysis of Paid and Free Applications

In the previous section we have discovered that users are willing to pay for paid applications if they find them more convenient and include more interesting features than free applications. But, is that truly the case? Do paid applications make users more comfortable when using them? In this section, we made a complete comparative analysis of the two types of application in order to find out whether users rate paid apps better overall based on their additional features.

4.1 Sentiment and Polarity Comparison

The clearest way to see if users are more pleased and comfortable with paid apps or not is through the review they made. Figure 5 displays the sentiment percentage of review between Free and Paid applications. As we can see here, figure 5 shows us a significant more positive reviews than paid

applications received. In particular, free applications only have 60% of all reviews are positive ones, around 20% are neutral and over 22% are negative ones. On the other hand, paid applications have around 80% positive, 20% more than free applications, with 5% neutral and 15% negative. From this graph, we can see that definitely paid apps have larger positive polls of reviews than free apps. However, only the quantity of reviews is not enough. We need to know the exact quality of reviews as well. Figure 6 shows the distribution of polarity in all reviews between Free and Paid applications. In other words, figure 5 shows us the size of positive, neutral, and negative review pools of each type, while figure 6 shows us the quality of those pools. We can see that not only Paid applications have larger poll of positive reviews, but also their polarity is also higher. It is even clearer when we focus on the polarity range from 0.25 to 0.75. In every point inside those range, paid applications have more positive reviews rate than free ones. Moreover, that goes the same with negative reviews. While free applications' negative reviews ranges from -0.0 to -1.0, paid applications' negative reviews ranges from -0.0 to -0.5, and paid apps' negative pools are overall smaller than those pools of free apps are.

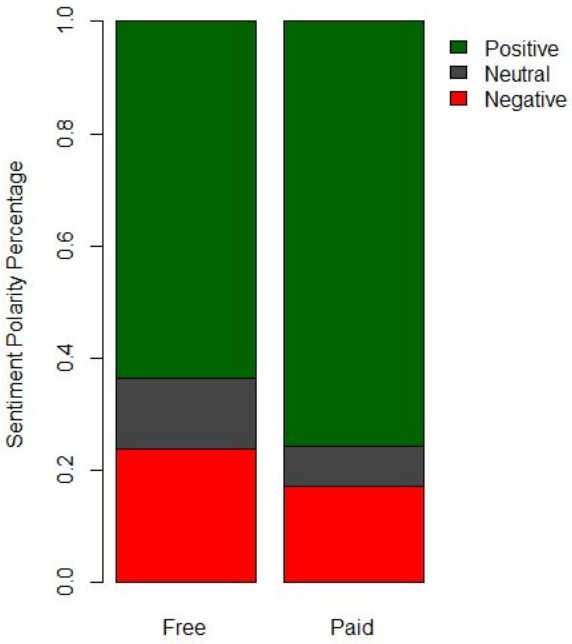

Figure 5: Sentiment Percentage of review between Type

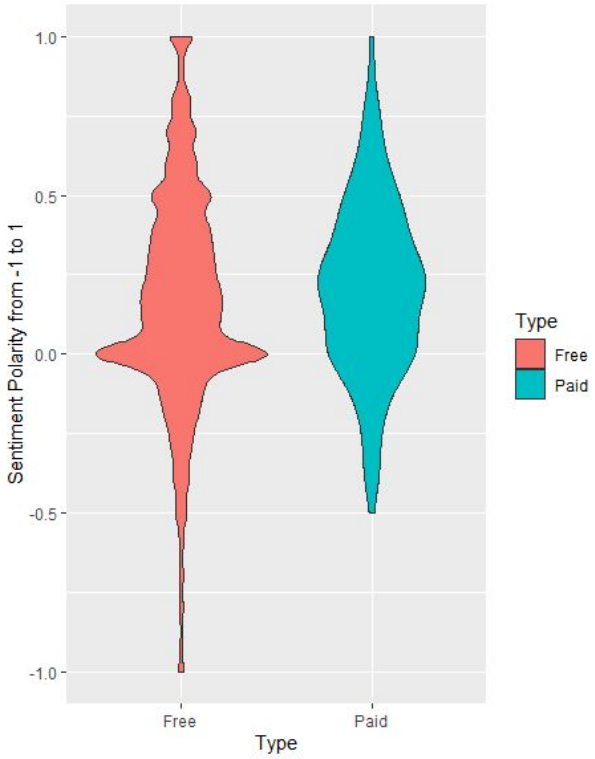

Figure 6: Polarity of review between Type

4.2 Number of Reviews per Rating between Type

In figure 2, we already knew that free applications have much larger number of installs than paid applications do. This is understandable since one doesn't cost users anything, and one costs users at least "something". The actual price of the applications may be small, but it could have a big effect on users' preference to install. This problem occurs because users not only need to pay for that price, but they also need to have credit cards and link them to their Google Accounts. With scandals that have happened

recently about leaking personal information, especially credit card information, some may just prefer using free applications to ensure that they will not be scammed. But is that fact mean number of Review of paid applications lower than that number of free applications? And what is the relationship between number of Reviews and the actual rating between those two types? We can find out the answers to those 2 questions in figure 7.

Figure 7 displays the distribution of Rating to number of Reviews between Paid and Free applications. As we can see, the number of reviews of Free applications are much larger than the number of reviews of Paid ones. This is reasonable since free applications have much more install attempts, compared with paid applications, which are much less common. The graphs also reveals that for most of the categories, the higher rating that applications have, the more numbers of reviews they receive. This is a very interesting finding, and it also confirms the information that Figure 4 and 5 gave us: Most of the reviews are positive, so that if an application receives more reviews, the likelihood that it has a good rating goes higher.

One more interesting finding is that even though free applications and paid applications' number of reviews between 3.0 rating and 4.0/5.0 rating have a huge gap, that number between 4.0 and 5.0 rating does not have a big gap like that. Especially for paid applications, the difference in number of reviews is significantly small, for same categories it is almost 0. This may be because according to Figure 3, the install attempts of 4.0 and 5.0 apps don't have much difference. As a result, the numbers of reviews don't have that much difference too, as users want to express opinions about the apps.

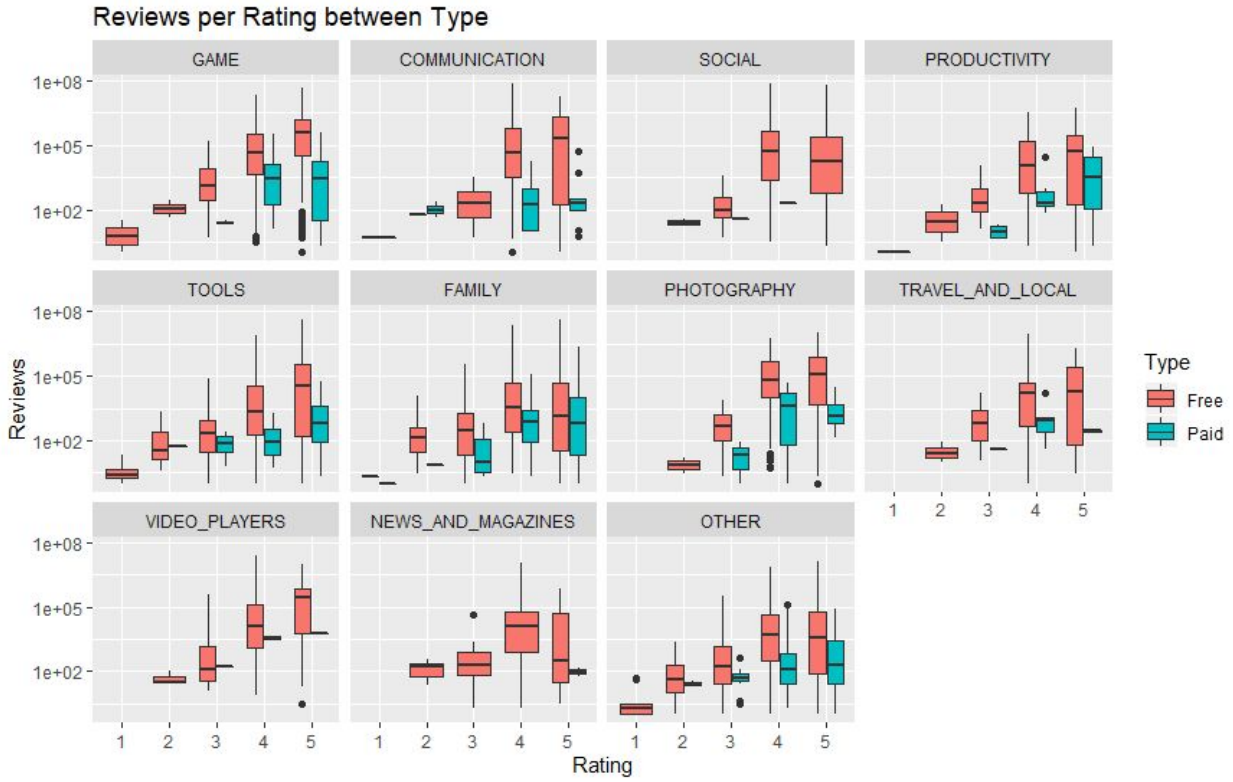

Figure 7: Number of Reviews per Rating between Type

4.3 Further dead end questions

When we first decided to do this section, we want to answer many questions about paid and free apps: “Why do developers choose to price their apps, instead of making their apps free?”, “What benefit do they have when pricing their apps? What is the drawback?”, “How much money or benefit can free applications make if their makers didn’t price them?”. We couldn’t answer these questions mainly because we didn’t have enough data about the advertisement cost of each of those app, as well as how much benefit their makers would receive.

5 Conclusion

After performing the data cleaning and rigorous analysis, we found many interesting facts from the dataset. First, we know that Game and Communication are on top popularity of all applications, with over 42% of all number of application installs. Social, Productivity and Tools respectively follow the lead, with around 5-6% of all number of application installs each. Second, we found out that $1.5 to around $5 is the ideal price range if developers want to price their applications instead of making them free. Free applications still have the highest number of installs, but $5 apps have that high number as well. Third, we discovered that the number of installs does not always in direct proportion with rating of the app. The rating rises as the number of install increases until it reaches its peak at 4.0 rating. After reaching its peak, as the number of install further increases, the rating decreases. This implies that as an app increases its popular through a certain point, with the metrics of number of installs, its rating will start to go down since it is not possible to please everyone with every feature in the app. Fourth, we found out that most of the reviews are positive, and most of the negative reviews are in low negative form. This benefits developers the most as they can use those reviews to improve their apps without the fear of losing existing users.

Move to the free and paid apps comparison, we also had many interesting findings. First, we found out that paid apps had much more positive reviews percentage than free apps. Also, the positive reviews that paid apps received are more “positive” than free apps, and the negative reviews paid apps received are less “negative” as well. Second, it was revealed to us that free applications have more number of reviews than paid applications on every category, since they are more popular. The number of rating on 4.0 and 5.0 rating applications are close with each other and far higher than 3.0 and below rating. This confirmed our findings above, that 4.0 rating is the point where the most popular applications would get to.

Our conclusion is very interesting and led us to think of new questions that we would need to look into to expand and give more weight to our findings. Our findings suggest that if developers want to create a popular app, they should make a paid app with the price around $5 in Game category. This way they will have the highest likelihood to have a popular app with a decent rating. But our findings cannot cover the other side. What about creating free applications and use their advantage in install attempts to make money from ads? What are the pros and cons of free and paid apps when it comes to income outside of application purchase? To answer these questions, we need more data about the advertisement cost and the income outside of application purchase for every application, as well as a survey to see if users truly want a free application with ads or a paid application without ads and premium features. Moreover, we

would need more data on the review, its sentiment and polarity/subjectivity to subjectively and correctly analyze users' reviews. These questions and new data would create new discussions and interesting findings with further analysis.

A Code for Data Cleaning

```
# Data Cleaning

# Load Libraries

library(tidyverse)
library(stringr)
library(xts)
library(ggplot2)

# Load main dataset
data <- read.csv("googleplaystore.csv")

# Load subdata dataset
subdata <- read.csv("googleplaystore_user_reviews.csv")

# Create clean dataset
data.clean <- data %>%
  mutate(
    # Eliminate some characters to transform Installs to numeric
    Installs = gsub("\\+", "", as.character(Installs)),
    Installs = as.numeric(gsub(",", "", Installs)),
    # Transform reviews to numeric
    Reviews = as.numeric(as.character(Reviews)),
    # Remove currency symbol from Price, change it to numeric
    Price = as.numeric(gsub("\\$", "", as.character(Price)))
  ) %>%
  filter(
    # Two apps had type as 0 or NA, they will be removed
    Type %in% c("Free", "Paid")
  )

# Create subdata clean dataset
subdata.clean <- subdata %>%
  filter(
    Sentiment %in% c("Negative", "Neutral", "Positive")
  )

# Rounds reviews from tenths to whole numbers
index_1 <- which(data.clean$Rating %in% c("1.0", "1.1", "1.2", "1.3", "1.4"))
index_2 <- which(data.clean$Rating %in% c("1.5", "1.6", "1.7", "1.8", "1.9", "2.0", "2.1", "2.2", "2.3",
  "2.4"))
```

```
index_3 <- which(data.clean$Rating %in% c("2.5", "2.6", "2.7", "2.8", "2.9", "3.0", "3.1", "3.2", "3.3",
  "3.4"))
index_4 <- which(data.clean$Rating %in% c("3.5", "3.6", "3.7", "3.8", "3.9", "4.0", "4.1", "4.2", "4.3",
  "4.4"))
index_5 <- which(data.clean$Rating %in% c("4.5", "4.6", "4.7", "4.8", "4.9", "5.0"))
data.clean$Rating[index_1] <- "1"
data.clean$Rating[index_2] <- "2"
data.clean$Rating[index_3] <- "3"
data.clean$Rating[index_4] <- "4"
data.clean$Rating[index_5] <- "5"
data.clean$Rating <- factor(data.clean$Rating)

# Set categories not in the top 10 to ‘other’ in data.clean
index_nottop10 <- which(data.clean$Category %in% c("ART_AND_DESIGN",
  "AUTO_AND_VEHICLES", "BEAUTY", "BOOKS_AND_REFERENCE", "BUSINESS", "COMICS",
  "DATING", "EDUCATION", "ENTERTAINMENT", "EVENTS", "FINANCE",
  "FOOD_AND_DRINK", "HEALTH_AND_FITNESS", "HOUSE_AND_HOME",
  "LIBRARIES_AND_DEMO", "LIFESTYLE", "MAPS_AND_NAVIGATION", "MEDICAL",
  "PARENTING", "PERSONALIZATION", "SHOPPING", "SPORTS", "WEATHER"))
data.clean$Category[index_nottop10] <- c("OTHER")
data.clean$Category <- factor(data.clean$Category, levels = levels(addNA(data.clean$Category)),
  labels = c(levels(data.clean$Category), "OTHER"), exclude = NULL)

# Order category factor by highest number of installs to lowest with ‘other’ at the end in data.clean
data.clean$Category <- factor(data.clean$Category, levels = c("GAME", "COMMUNICATION",
  "SOCIAL", "PRODUCTIVITY", "TOOLS", "FAMILY", "PHOTOGRAPHY",
  "TRAVEL_AND_LOCAL", "VIDEO_PLAYERS", "NEWS_AND_MAGAZINES", "OTHER"))

# Merge the data set onto the subdata set by app, giving the subdata set information about each app
mergedata.clean <- merge(data.clean, subdata.clean, by = "App")

# Removes duplicate entries from data.clean
data.clean <- data.clean %>%
  distinct()

# Removes duplicate entries from mergedata.clean
mergedata.clean <- mergedata.clean %>%
  distinct()

# Remove unused levels from Sentiment
mergedata.clean$Sentiment <- factor(mergedata.clean$Sentiment,
  levels(droplevels(mergedata.clean$Sentiment)))
```

```
#Remove unused levels from Type
mergedata.clean$Type <- factor(mergedata.clean$Type, levels(droplevels(mergedata.clean$Type)))
```

B Properties of Popular Applications

```
# Figure 1: Installs/Category
# Find sum of all installs per category
temp <- data.clean %>%
  group_by(Category) %>%
  summarize(totalInstalls = sum(Installs))
ggplot(data = temp, aes(x = Category, y = totalInstalls, fill = Category)) + geom_bar(stat = "identity") +
  xlab("Category") + ylab("Installs") + theme(axis.text.x=element_text(angle=45, hjust=1))

# Figure 2: Installs/Price/Category
ggplot(data = data.clean, aes(x = Price, y = Installs, fill = Category)) + geom_jitter(position =
  position_jitter(w=0.1, h=0.1)) + geom_smooth() + ggtitle("Installs vs Price") +
  scale_y_continuous(name = "Installs", trans = 'log10', limits = c(-0.001, 1000000000)) +
  scale_x_sqrt(limits = c(0,32)) + facet_wrap(data.clean$Category,)

# Figure 3: Installs/Rating/Category
ggplot(data = subset(data.clean, Rating != "NaN"), aes(x = Rating, y = Installs, fill = Category)) +
  geom_boxplot() + ggtitle("Installs per Rating") +
  scale_y_continuous(name = "Installs", trans = 'log10') + facet_wrap(subset(data.clean, Rating !=
  "NaN")$Category, )


# Figure 4: Category/Sentiment
ggplot(data = mergedata.clean, aes(x = Category, y = Sentiment_Polarity, fill = Category)) +
  geom_violin() + theme(axis.text.x=element_text(angle=45, hjust=1)) + ylab("Polarity from -1 to 1")
```

C Comparative Analysis of Paid and Free Applications

```
# Figure 5: Sentiment Polarity ratio (Left)
par(mar = c(4, 4, 4, 7))
barplot(prop.table(table(mergedata.clean$Sentiment, mergedata.clean$Type), 2), col = c("Red", "gray27",
  "DarkGreen"), ylab = "Sentiment Polarity Percentage", args.legend = list(x = 3.5, y = 1, bty = "n"),
  legend.text = TRUE)

#Figure 6: Sentiment Polarity per review (Right)
ggplot(data = mergedata.clean, aes(x = Type, y = Sentiment_Polarity, fill = Type)) + geom_violin() +
```

```
  ylab("Sentiment Polarity from -1 to 1")

#Figure 7: Review/Type/Category
ggplot(data = subset(data.clean, Rating != "NaN"), aes(x = Rating, y = Reviews, fill = Type)) +
  geom_boxplot() + ggtitle("Reviews per Rating between Type") + scale_y_continuous(name =
  "Reviews", trans = 'log10') + facet_wrap(subset(data.clean, Rating != "NaN")$Category, )
```