**Google Play Store Report and Analysis**

1. **Introduction**

This report will include data and information regarding the Google Play Store and will be used to answer a user story. The data utilized for answering the user story consists of 10,841 rows and 13 columns all related to Google Play Store statistics and metrics. These ranged from the category of the app, the overall rating, installs, current version, price, and other similar factors. Meaning there are 10,841 apps and relevant information contained within the data.

The question we are tasked with is, “User would like to know which category of apps have the highest rating per number of installs”. In this report I will be discussing more in-depthly in the following sections about the data, methods used, analysis performed, and the overall results. The hypothesis I will be trying to test in this report will be that the category of apps will not have a large enough variance to determine the top categories by the metric of rating per number of installs.

1. **Body**

**Data:**

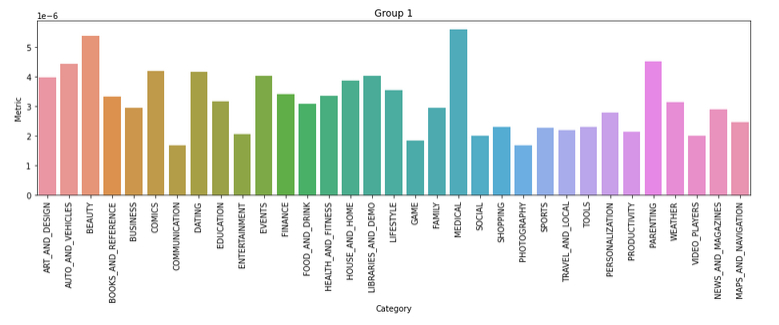
To start off this analysis I initially gathered all of the data into a Data Frame for inspection of rows which will be relevant to the user story and the hypothesis which will be tested. After the initial inspection I noticed numerous columns would be unneeded for this analysis as well as many missing values in the data. I decided I will be needing to keep 3 of the 13 rows in the data, which are category, Rating, and Installs. Between the Rating and Installs column there were 1,474 rows which I decided to remove as there were missing values. While I could have used another method such as using the mean or median for the certain category, I determined this may lead to further inaccuracies in the data. Although apps in the same category may be similar the purpose of the apps can vary immensely.

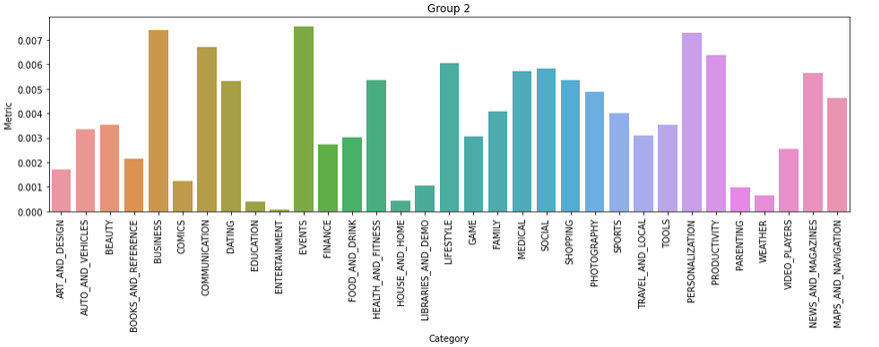
The next step taken was to start cleaning the data and preparing it for later analysis. One of the first things I noticed was an issue in the Installs column. This is a numerical column and would be needed for mathematical operations, however, the data was categorical and had symbols (i.e. 1,000,000+). After this column was changed to an integer data type as well as removing symbols, it was now available to be used for mathematical operations later on. By inspecting the category and Installs columns further, by looking at unique values I found some which were irrelevant to the column (such as 1.9 as a Category, or Free as a Install count). Additionally, in the Installs column I noticed an extremely large range of values which would cause further error in later analysis and decided I could do a better job if I was able to split this data into two separate data frames, one for larger apps and one for smaller apps. Using the .describe() method in Python I was able to easily find the median value of the dataset for a cutoff marker as well as notice further areas for data cleaning. I found that there were less than 1% of the data which had an Installs value of less than or equal to 50 as well as more than or equal to 500,000,000. I decided to remove these additional rows that way the analysis performed later will be able to yield stronger results.

I now had two separate dataframes, one which was Apps with Installs more than or equal to 500,000 (the median) and the second with Installs less than 500,000. Now, I will discuss in the section below my method for analysis.

**Method:**

Thinking back to the user story, I knew I needed to create a new column in the dataframes for a metric. I decided to be most accurate to the user story I needed to divide the Rating column by the Installs column. I will be referring to this new column as the Metric. After this Metric was created I knew I no longer needed the Rating and Installs column from each data frame as they each already served their purpose. This left the data frames with only category and metric columns. Since there is still a fairly large variance between Installs even after splitting the data into two groups, I realized this data needs to be normalized. The reason the normalization was not performed earlier, is if it was done prior to splitting the data frame into two separate data frames some of the problem I am trying to avoid of a large variance error would come into play. I decided to use min-max scaling for normalization as I wanted the values to easily be used in division with a column (Rating) that ranged from values 0-5. The min-max method scales all values between 0 and 1 meaning this process becomes much easier. Below I will show visualizations of the average metric value for each category, in both groups.



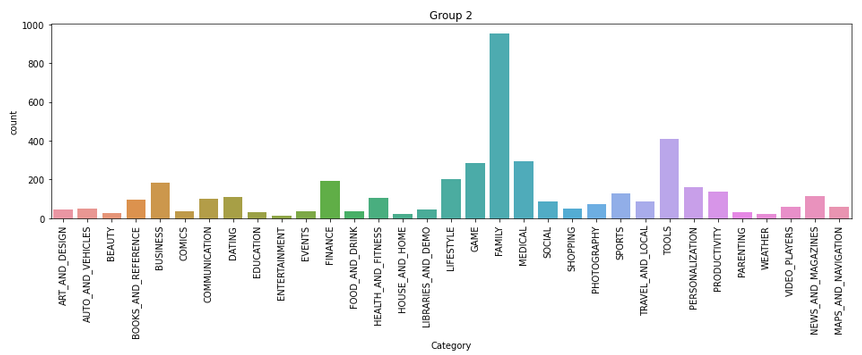
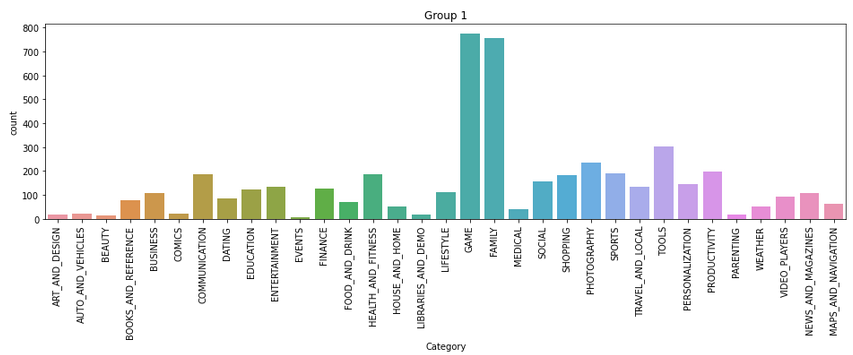


Next, was time to start preparing the data that had been created for analysis and to determine which form of analysis would be best to test my hypothesis. To remind you, the hypothesis I came up with is, “the category of apps will not have a large enough variance to determine the top categories by the metric of rating per number of installs.”. After researching many statistical tests, I opted for a one way ANOVA test. In order to perform this analysis I needed to separate all of the separate categories from each of the two data frames as their own variable. The category column consists of 33 unique values, meaning the need to create 66 variables in order to account for both data frames and each category.

**Analysis:**

Now, after separating every unique category as its own variable for each data frame (greater than or equal to 500,000 which will be referred to as group 1, and less than 500,000 which will be referred to as group 2), it is finally time to test the hypothesis and see if the variance is large enough between each category and it’s metric to determine which category has the highest rating per install. For the one way ANOVA test I utilized the f\_oneway method from scipy.stats library.

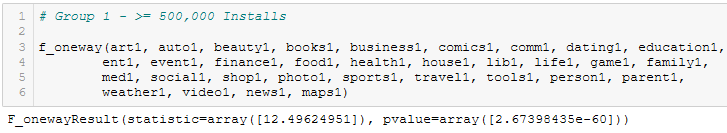
A limitation to this analysis however, is the difference in the amount of apps per category. While most categories have relatively similar counts, there are two app categories (Game and Family) however, the biggest difference is in group 2 with the Family app, which is much larger than any other categories count. This means when more data is provided the results of this analysis may change in the future as other app categories grow. I will show two visualizations which represent the amount of apps in each category, for both groups.



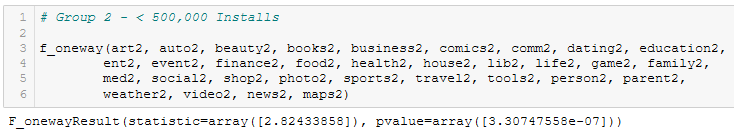
Now, it is time to discuss the results of the analysis.

**Results:**

From the results of group 1 the P-value of the ANOVA test was extremely low, meaning it had a value much less than 0.05 and allows for rejecting the null hypothesis. This means in group 1 the category of apps do in fact have a large enough variance of ratings per install to determine the top category of apps. I will show the equation used for this below, along with its result:



From the results of group 2 the P-value of the ANOVA test was very low, but not as significant as group 1. However, it was still lower than 0.05 which allows for rejecting the null hypothesis in this scenario as well. I will show the equation used for this below, along with its result:



So, we can now determine in both groups the top categories of apps by the ratings per install metric requested. To determine this I utilized a groupby statement along with the mean values for the metric column and then sorted the values to find the top categories for both group 1 and group 2. For group 1, the top 3 categories consist of Medical, Beauty, and Parenting respectively. However, to be true to the user story which asks for the top app category, and with installs of over 500,000 are Medical apps. For group 2, the top 3 categories consist of Events, Business, and Personalization respectively. To answer the user story, the top category of apps with installs of under 500,000 are Event apps.

**Conclusions:**

To conclude this report for apps which gain popularity, and have install counts of over 500,000 the top category by metric of ratings per install are Medical Apps. While apps which do not gain as much popularity and have install counts of under 500,000, the top category of the metric ratings per install are Event apps. If I was to continue further analyzing the data my next steps would be to find a stronger metric than ratings per install, and analyze the total data set rather than splitting into two groups. The reason this was not done was like previously mentioned, but I wanted to stay true to the user story and directly use the metric which was asked for (rating per install). I did briefly try using the entire dataset utilizing this metric, however, it rewarded apps much higher with say 50 installs and a high rating over a decently high rating but over 10,000,000 installs. The reason this choice was made in the end is when an app gains high popularity it becomes much harder to please an entire user base that large rather than dealing with 50 users which is much more manageable. However, with more time I believe a proper model could be created for this particular use and could assign weights based upon certain levels of installation values.