

GROUP PROJECT REPORT



ANALYSIS OF GOOGLE PLAY STORE APPS

IE-6200 Engineering Probability & Statistics SEC 02

Prof. Mohammad Dehghani

Group 12:

Bhavya Batra

Janvi Jani

Anish Nair

ANALYSIS OF GOOGLE PLAY STORE APPS

1) **ABSTRACT**

The global mobile application market size was valued at USD 154.05 billion. They are easy to create and also profitable if deployed in the right manner. In this project, we perform a detailed analysis of the Android app market using R Studio. The aim of the project is to collate and compare different attributes of applications on google play store in-order to derive meaningful insights. A good visualization not only helps us accurately interpret data, but also improve our comprehension, communications and decision making. Hence, we have created a number of tables, graphs and plots throughout the course of our project as it helps a developer to understand the trends, patterns and customer demands better and thus them to popularize their product.

2) **OBJECTIVE**

For any project to flourish and be lucrative, we need to have detailed knowledge about the market that we're about to enter. The google app store platform has over a million applications and hence it is imperative for developers to know about the trends on the Android platform before investing all their resources into an application. Through our exhaustive analysis, a developer can gain actionable insights and capture the Android market! It helps one to gain insights about their competitors- what are their strategy, strengths, and weaknesses. It helps developers avoid some major mistakes made in other applications and give them an edge over their competitors.

3) **DATA COLLECTION:**

Data Description

We acquired the dataset from Kaggle for better consistency and accuracy.[1] The Dataset represents the different categories of applications present in the google play store.

The dataset contains a total of 7800 data entries and 11 attributes. These 11 variables are listed as follows:

1. App: Names of the apps present in the Google play store
2. Category: Represents the Category of all the applications
3. Rating: Represents the ratings given to all the apps on a scale of 1-5
4. Reviews: Represents the number of reviews given to the apps
5. Size: Size of all the apps in MB
6. Installs: Represents the total number of installs of each app
7. Type: Type of app (Free or paid)
8. Price: Represents the price of each app in USD
9. Content: Age group the app is targeted at
10. Genres: Same as category
11. Log_Installs: Log of Installs for the ease of calculation

Data Preprocessing

The raw dataset contained 10,000 observations with 13 variables initially. The following preprocessing steps were taken to bring the data in a representable form:

1. Removing all the rows containing NA and missing values
2. Deleted unnecessary columns like Last. Updated, Current. Ver and Android Ver
3. To ease our calculations, we calculated the log of the Installs variable
4. After filtering the data, we had a main dataset with 7724 observations and 11 attributes

4) DATA VISUALISATION

Exploratory Data Analysis

After extracting the essential data from the dataset, we obtained a **Bar plot of the Total Number of Apps in each Category**. Clearly, the Family category has the maximum number of applications, whereas Beauty or Events accounts for the least number of applications. Below is a Bar plot that explains the numbers.

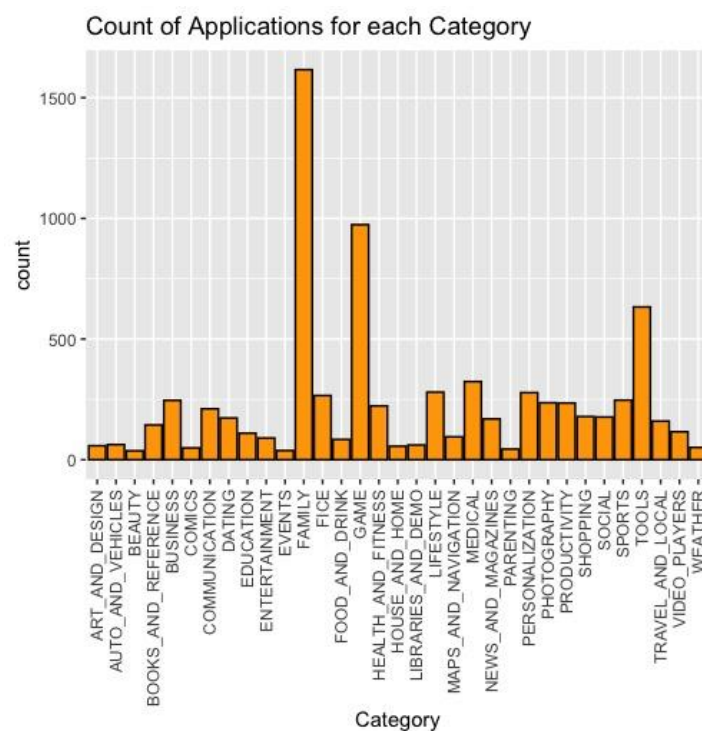


Figure 1

To get a better insight, we created a **Bar Plot to find the Top 15 categories or Genres** of apps that are most popular among users. In this popularity is considered in terms of the number of installs users have made. From the plot below it can be inferred that Tools is the most popular genre among users followed by Entertainment.

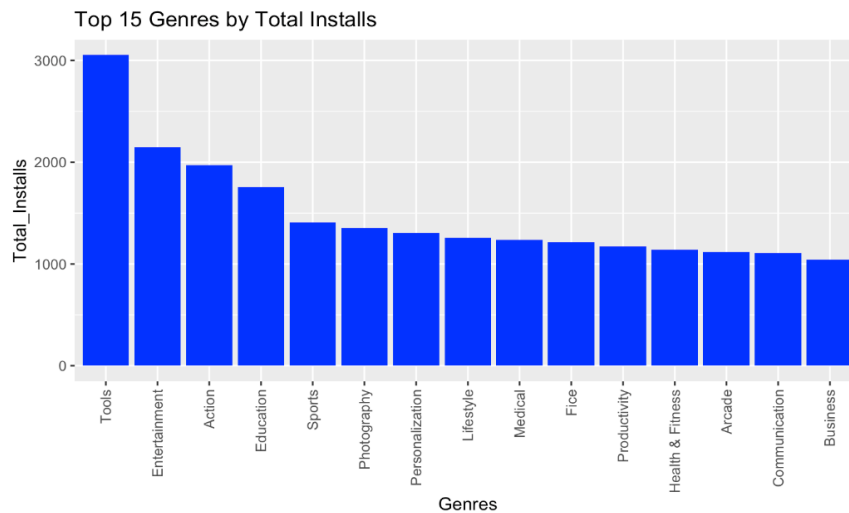


Figure 2

A **Histogram for the Installs and Rating differentiated by Type** was created to show how the number of installs of free or paid apps changes as the rating changes. The plot shows that as Rating increases, the number of installs for both free and paid app increases. People install more apps if rating is more than 3.5. The histogram below is Highly negatively skewed with skewness equal to -1.75.

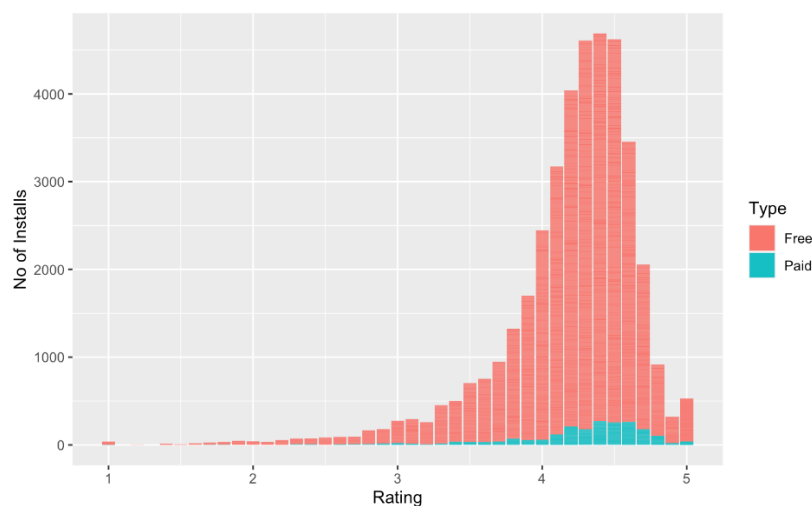


Figure 3

We made a **Pie chart to represent the percentage of the content type**. The chart below shows that 79.9% of apps in the play store are meant for all age groups whereas 0.4% of apps are meant for adults.

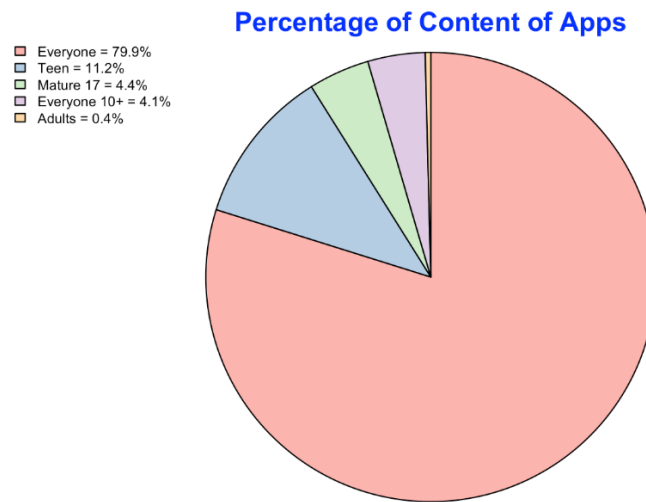


Figure 4

We also implemented a **Boxplot that represents the Total Installs(log) by the Type of Application**. The purpose was to check which type of app is installed more by the people. The plot clearly shows that the quantiles of free apps are more than paid apps. Therefore, we can conclude that people install free apps more than paid apps. Below is the Boxplot and the quantiles calculated for both free and paid apps.

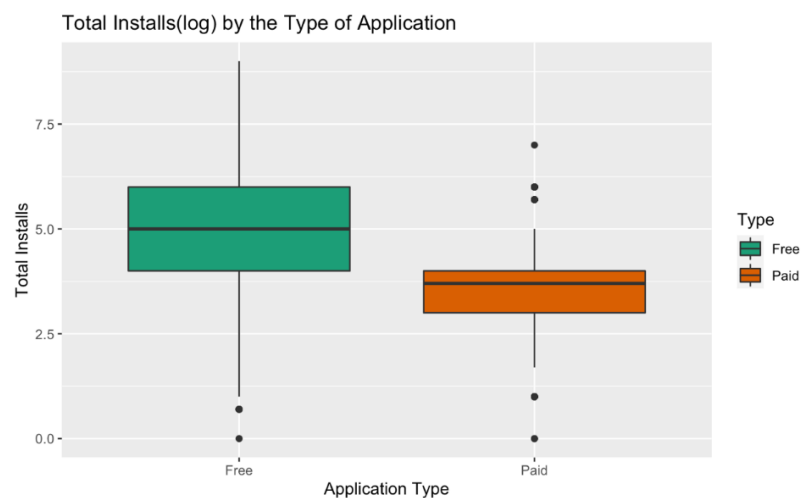


Figure 5

```

Console Terminal x Jobs x
~/R files/ ↗
> #Calculating Quantiles for free and paid apps
> Free_df <- google_apps_df[google_apps_df$Type=="Free",]
> No_Free_Installs <- quantile(Free_df$Log_Installs)
> No_Free_Installs
  0%  25%  50%  75% 100%
  0   4   5   6   9
> Paid_df <- google_apps_df[google_apps_df$Type=="Paid",]
> No_Paid_Installs <- quantile(Paid_df$Log_Installs)
> No_Paid_Installs
  0%  25%  50%  75% 100%
0.00000 3.00000 3.69897 4.00000 7.00000

```

Figure 6

We plotted a **histogram** to showcase the distribution of the **App Ratings**. Our objective was to showcase the distribution of the variable. From the plot below, you can see that the variable is left-skewed.

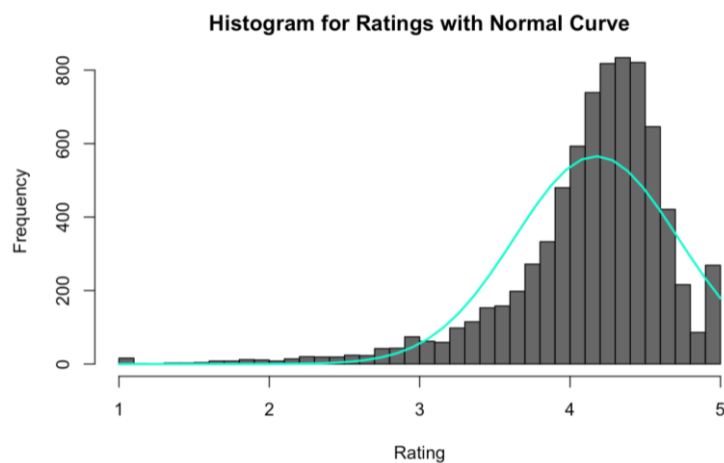


Figure 7

Fitting Distributions:

The graph below showcases the distribution of the continuous variable Size. The graph is right skewed. Using Fitting Distributions, the AIC values obtained for normal distribution and lognormal distribution were **70649.81** and **64755.69** respectively.

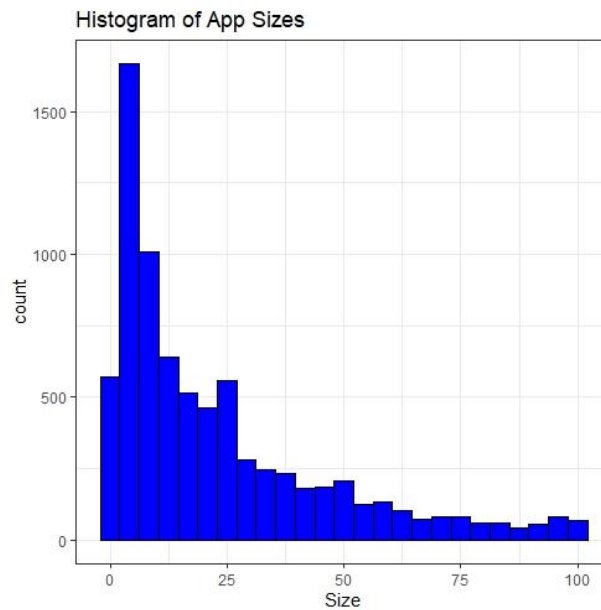


Figure 8

As the AIC value for lognormal distribution is less than the AIC value for normal distribution, it is a more fitting distribution. This is also indicated by the Cullen and Frey Graph:

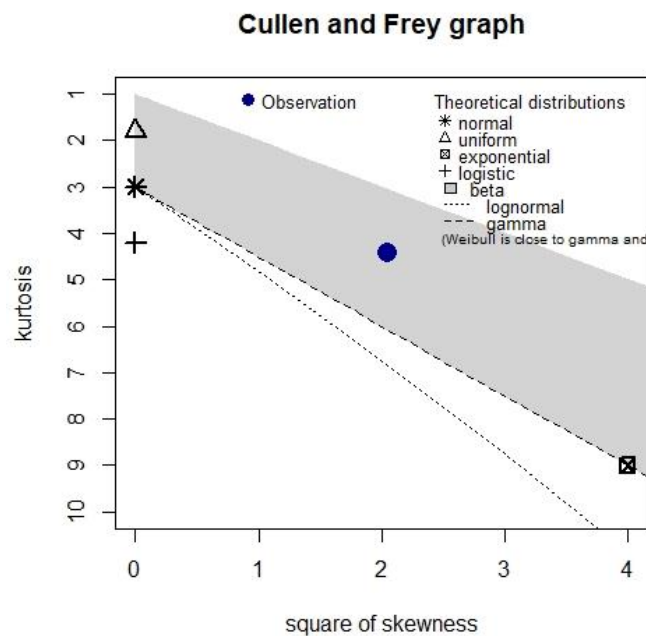


Figure 9

Below are the QQ-plots and PP-Plots for the normal distribution (top) and the lognormal distribution (bottom). For the lognormal distribution, we can see that the lognormal line lies in line with the graph values. Whereas in the case of normal distribution, there is a slight deviation. Therefore, from the evidence shown, we can state that the Size of the App values follow a lognormal distribution. [2]

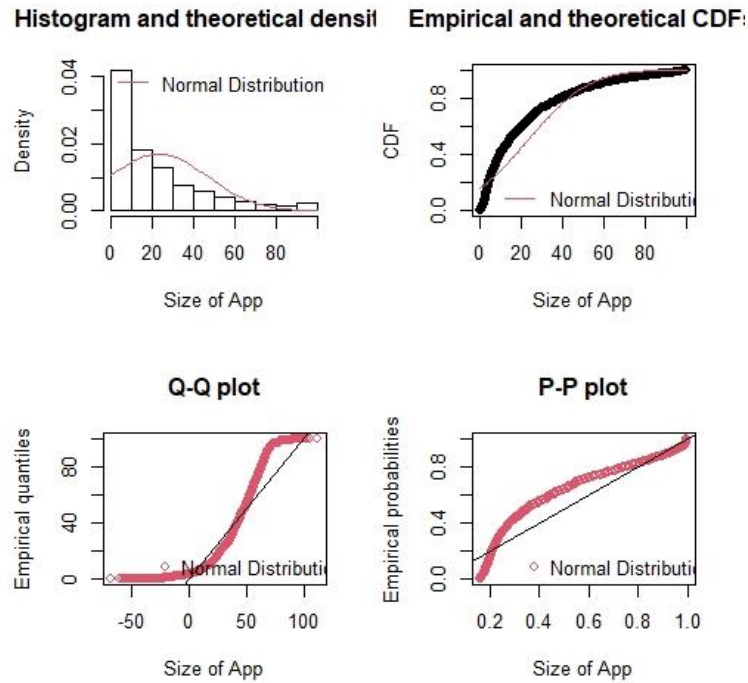


Figure 10

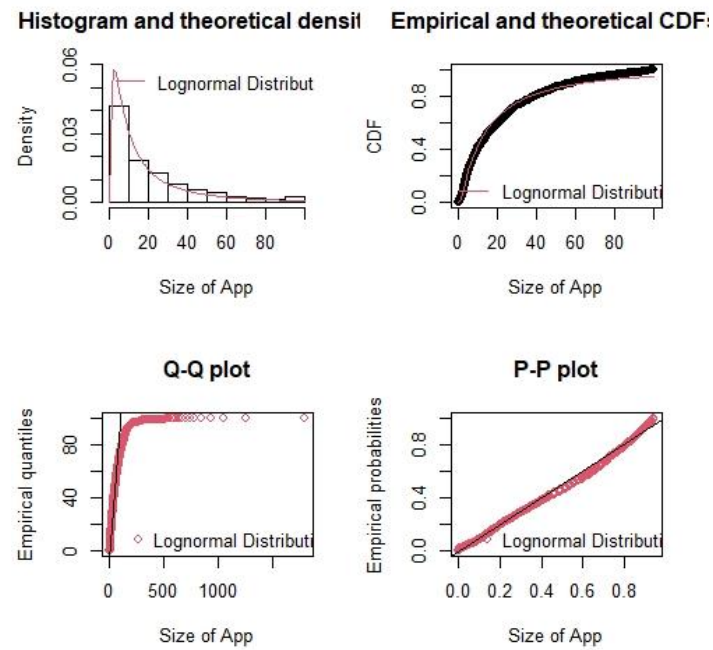


Figure 11

5) STATISTICAL ANALYSIS

Probabilistic Analysis

1) PMF and CMF of ratings v/s the no of installs

Here, we first find out the count of installed applications according to ratings given to each application and put them in rating bins using of (1-2, 2-3, 3-4 or 4-5). After that, we find out the CMF and PDF of each rating bin.

Ratingbins <fctr>	Log_Installs <int>	pmf <dbl>	cdf <dbl>
1-2	66	0.008545902	0.008545902
2-3	286	0.037032241	0.045578143
3-4	1928	0.249643921	0.295222064
4-5	5443	0.704777936	1.000000000

4 rows

Figure 12

Then, we calculate the Expected Value (E) of the ratings given to every app in the app store, with weights provided by the probability distribution table. The weighted mean of app ratings comes up to **4.188995**

Below is a scatter plot graphically representing our findings. As we can see, as the number of app ratings increases, the number of installs also increase. Hence, we can conclude that the higher the rating of the app, the greater is the number of installs.

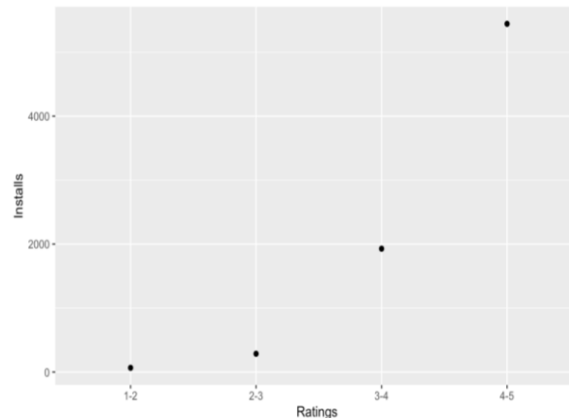


Figure 13

2) PMF and CMF of Size of Apps vs No of Installs

Here, we follow the same procedure as mentioned above, first, we create size bins (we have created the following types of bins for size of application: <0 mb, 1mb to 2mb, 2mb to 3mb, ,90-100mb, >10mb), then we create a new data frame to find the CMF and PDF of applications installed based on size of applications. The table looks like:

Sizebins <fctr>	Log_Installs <int>	pmf <dbl>	cdf <dbl>
0	3247	0.42043247	0.4204325
1	1401	0.18140619	0.6018387
2	992	0.12844749	0.7302862
3	606	0.07846692	0.8087531
4	449	0.05813803	0.8668911
5	325	0.04208209	0.9089732
6	226	0.02926324	0.9382364
7	161	0.02084682	0.9590833
8	117	0.01514955	0.9742328
9	199	0.02576719	1.0000000

1-10 of 10 rows

Figure 14

Below is a scatter plot graphically representing our findings. We can conclude that, as the number of installs of an application decreases as the size of the application increases.

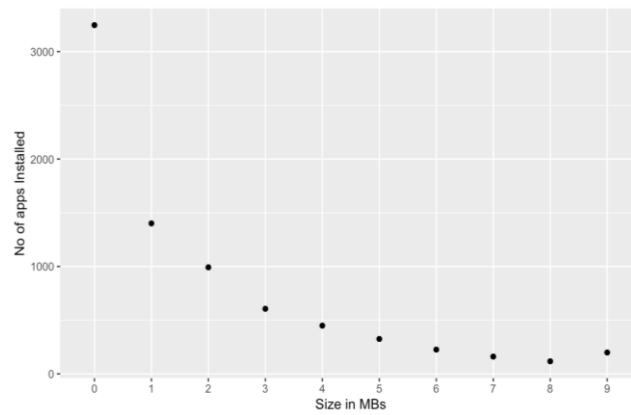


Figure 15

3) Joint Probability

Here, we calculate joint probability of apps installed compared with size of apps and apps installed compared with rating given to the app. We take two vectors (joint_freq and joint_prob) and a function (that itself takes two arguments) and build a matrix by calling the given function for each combination of the elements in the two vectors. Our final joint frequency table looks like:

	1-2	2-3	3-4	4-5
<=10mb	214302	928642	6260216	17673421
10mb-20mb	92466	400686	2701128	7625643
20mb-30mb	65472	283712	1912576	5399456
30mb-40mb	39996	173316	1168368	3298458
40mb-50mb	29634	128414	865672	2443907
50mb-60mb	21450	92950	626600	1768975
60mb-70mb	14916	64636	435728	1230118
60mb-80mb	10626	46046	310408	876323
80mb-90mb	7722	33462	225576	636831
90mb-100mb	13134	56914	383672	1083157

Figure 16

After calculating joint frequency, the next step is to calculate the joint probability distribution table. We do that using the round() function. Here, the argument '3' states that we need the output rounded up to 3 digits. The joint probability distribution table contains value between 0 and 1 and the addition all the rows and columns in this table adds up to 1.

	1-2	2-3	3-4	4-5
<=10mb	0.004	0.016	0.105	0.296
10mb-20mb	0.002	0.007	0.045	0.128
20mb-30mb	0.001	0.005	0.032	0.091
30mb-40mb	0.001	0.003	0.020	0.055
40mb-50mb	0.000	0.002	0.015	0.041
50mb-60mb	0.000	0.002	0.011	0.030
60mb-70mb	0.000	0.001	0.007	0.021
60mb-80mb	0.000	0.001	0.005	0.015
80mb-90mb	0.000	0.001	0.004	0.011
90mb-100mb	0.000	0.001	0.006	0.018

Figure 17

4) Correlation Coefficient

Correlation coefficients are used to measure how strong a relationship is between two variables. Here, we are calculating correlation coefficient between the ratings given to an app and the reviews given to an app. We would like to know if an app given a higher rating also has more comments or vice versa.

We start by calculating the correlations for each field. For which, we need to create and use bins for rating and also reviews. We use the same rating bins that we created earlier and then we create bins to calculate the total number of apps installed for each rating bin and then add the column to our data frame. We then calculate the correlation coefficient using the cor() functions and then passing our calculated coefficients as arguments. Our correlation coefficient comes up to **-0.501764**.

From this, we can conclude that since the value of correlation coefficient is in-between 0 and -1 the points are scattered, we can say that there exists no relationship between the number of reviews that an app receives and the rating that the app receives. [3]

Hypothesis Tests

1) Checking whether there is a significant difference between the mean rating of free and paid apps.

We have used a two-tailed t-test with a confidence interval of 95% to test the above hypothesis. The null and alternative hypothesis are defined below:

H_0 - True mean rating between paid and free apps are the same

H_1 - True mean rating between paid and free apps aren't the same

On performing the two-tailed t-test, we get a p-value of **0.0004144**, which is less than the α value of 0.05. Therefore, we can reject the null hypothesis and state that there is a significant difference between the mean rating of free and paid apps.

2) Whether the mean Size of Apps in the Family Category is the same as the Population Mean.

To test the above hypothesis, we have used a one sample t-test with the confidence interval of 95%. The null and alternative hypothesis are defined as:

H_0 - Mean of size of family apps is equal to 30.16281 MB

H_1 - Mean of size of family apps is not equal to 30.16281 MB

On performing the one sample t-test, we get a p value of **2.2e-16** which is less than the α value of 0.05. Therefore, we can reject the null hypothesis and state that there is a significant difference between the mean size of apps in the family category and the population as a whole.

3) Whether there is a significant difference between the mean of Rating from a random sample of 1000 apps and the mean of Rating for the entire population'

For the above hypothesis problem, we have used a one sample z-test with a confidence interval of 95%. Assuming the entire dataset as the population, we want to test whether the mean rating of the sample we are taking is equivalent to the mean rating of the population. The null and alternative hypothesis are defined as:

H_0 - Mean of sample Rating is equal to 4.155

H_1 - Mean of sample Rating is not equal to 4.155

On performing the one sample z-test, we get a z-value of **0.8863162**. The z-value lies within the range $[-1.96, 1.96]$. Thus, we fail to reject the null hypothesis and conclude that there is no significant difference between sample mean Rating and population mean Rating

6) ADVANCED ANALYTICS:

We used Multiple Linear Regression Model to predict the number of installs of apps using three independent variables (Ratings, Size and Reviews). Before starting with the training phase, the dataset was divided into 2 groups:

1. Training set 70% of the dataset
2. Testing set 30% of the dataset

Once the dataset is trained, the testing set is sent in the predict function for testing the trained model. The two plots represent the predicted value of the training and testing model

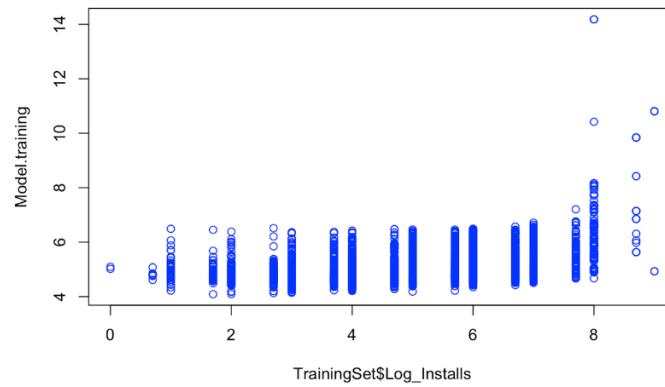


Figure 18

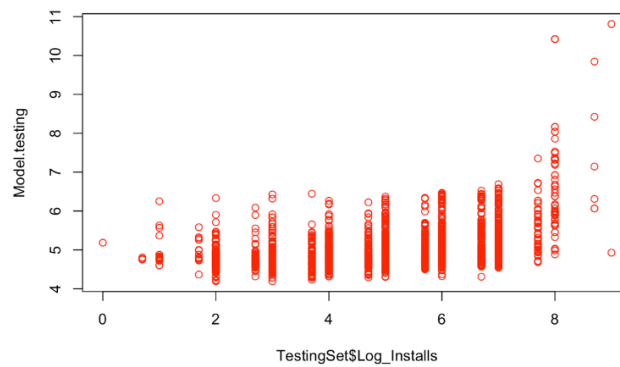


Figure 19

Below is the summary of all the values obtained from the regression model.

```

Console Terminal Jobs
~/R files/
Call:
lm(formula = .outcome ~ ., data = dat)

Residuals:
    Min       1Q   Median       3Q      Max
-6.1833 -0.9794  0.1866  1.1489  4.0702

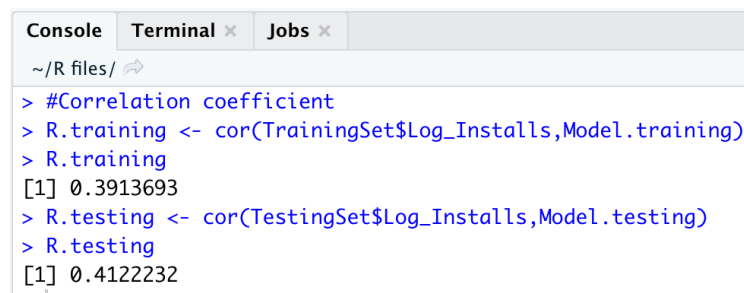
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  5.06404    0.02009  252.100 < 2e-16 ***
Rating       0.09233    0.02019   4.572 4.94e-06 ***
Size         0.43256    0.02071  20.889 < 2e-16 ***
Reviews      0.34333    0.02070  16.585 < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.477 on 5404 degrees of freedom
Multiple R-squared:  0.1532,    Adjusted R-squared:  0.1527
F-statistic: 325.8 on 3 and 5404 DF,  p-value: < 2.2e-16

```

Figure 20

We calculated the correlation coefficient as 0.4 for the predicted value. The below output shows that the three variables used in the model to predict the number of installs is somewhat linearly related with the Installs attribute.



```
Console Terminal x Jobs x  
~/R files/ ↗  
> #Correlation coefficient  
> R.training <- cor(TrainingSet$Log_Installs,Model.training)  
> R.training  
[1] 0.3913693  
> R.testing <- cor(TestingSet$Log_Installs,Model.testing)  
> R.testing  
[1] 0.4122232
```

Figure 21

7) **CONCLUSION**

Through this project, we have been able to apply all what we learnt in the labs on a real-world data set containing information of apps in the Play Store. We were able to experience the steps involved in the approach to solve any real word problem. From the Exploratory Data Analysis where you get an intuitive idea and feel about the data you are working with to the implementation of statistical analysis like hypothesis tests and advanced analytics like linear regression.

8) **REFERENCES:**

[1] <https://www.kaggle.com/lava18/google-play-store-apps>

[2] <https://www.kaggle.com/ismailsefa/google-play-store-apps-data-analysis-eda>

[3] S Shashank1, Brahma Naidu2, Google Play Store Apps- Data Analysis and Ratings Prediction, International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056, Volume: 07 Issue: 12 | Dec 2020.