Computer Sciences - 2019/2020

Google App Store - Most Reviews

Brandon Solo & Marcos Sánchez Bajo

INTRODUCTION AND GOALS

This study will focus on the possible relation between the size, price and category of the app with the number of reviews. Why number of reviews and not ratings or installs? Because there will be many apps with just a few people who have rated the app 5 stars, and these people could be those who have a conflict of interest.

We aren't going to trust in the number of installs too much either because we don't have access to a continuous sample (the apps are categorized with having +1000, +10000, … installs) that allow for an accurate prediction of the real number of installs an app could have, as well as the fact that maybe a lot of people installed the app, tried it for a minute and then deleted it because they didn't like it.

However, whenever a customer decides to take the time to write the review, it is because they believe that it will make a difference, and therefore most become a user since they are at least a little bit invested into wanting this app to succeed. They also provide feedback which the developer can use to improve the app.

If a relation is found, it could prove useful to a developer to adjust their app to more closely resemble apps that prove to statistically receive more reviews. We will proceed to obtain conclusions about the apps population, finding how many of the apps achieve a certain level of success measured in number of reviews, and studying which characteristics an app should have in order to achieve this success.

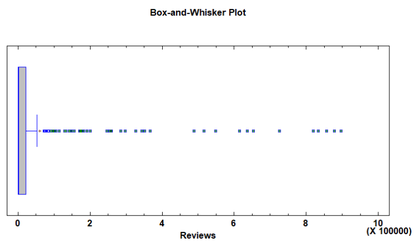
We will use a random sample of 300 apps belonging the Google Play Store from this source: https://www.kaggle.com/lava18/google-play-store-apps#googleplaystore.csv

DESCRIPTIVE ANALYSIS

Before anything, let's first take a look at how many reviews do apps receive in general. Doing a box plot of the reviews, we get the following:

Summary Statistics for Reviews

Count	300
Average	60567.6
Median	1466.5
Mode	2.0
Standard deviation	158039.
Coeff. of variation	260.93%
Minimum	1.0
Maximum	896118.
Range	896117.
Lower quartile	64.5
Upper quartile	21636.5
Interquartile range	21572.0
Stnd. skewness	25.7394
Stnd. kurtosis	48.5064

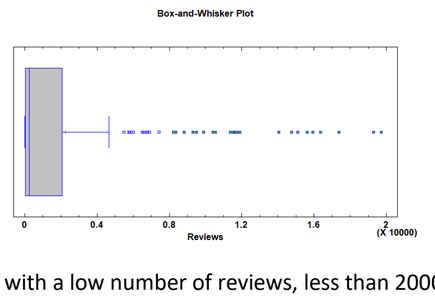


Here we can see that ¾ of apps have less than 20000 reviews, and of course given the size of the data, we have quite a lot of outliers. In order to simplify our data just a bit more, we will only consider apps that have less than 20000 reviews. The reason for choosing this number is that it simultaneously gets rid of apps that can probably already be considered successful since they have over 20000 reviews, but also maintain ¾ of our sample.

After removing these outliers, we get the following box plot:

Summary Statistics for Reviews

Count	220
Average	2266.44
Median	258.5
Mode	2.0
Standard deviation	4126.99
Coeff. of variation	182.092%
Minimum	1.0
Maximum	19727.0
Range	19726.0
Lower quartile	29.0
Upper quartile	2070.0
Interquartile range	2041.0
Stnd. skewness	13.9457
Stnd. kurtosis	14.3586



We seem to still have a great number of apps with a low number of reviews, less than 2000. This isn't surprising of course, since it's much easier to get just 1 review than it is to get 10000. This will also give us a good idea of what to expect when we try to fit this data to a distribution model.

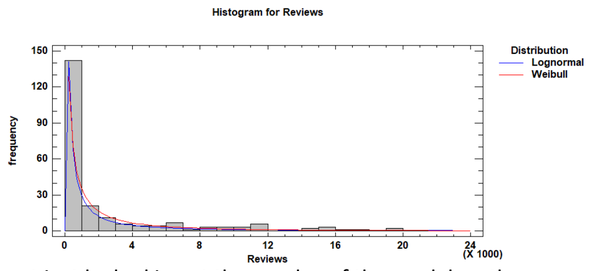
DISTRIBUTION FITTING

It would be very interesting to know what is the likelihood of a random app to have more than a certain number of reviews, not considering any other factor. A distribution model will be very helpful for this.

As stated in the previous section, our data seems to be very skewed towards the right, and therefore the first models to come to mind are Lognormal and Weibull models. We use the Chi-Squared test and obtain the following results:

Goodness-of-Fit Tests for Reviews
Chi-Square Test

	Lognormal	*Weibull*
Chi-Square	63.8	59.0
D.f.	30	30
P-Value	0.000311699	0.00121466



It's hard to tell how well a model fits a dataset just by looking at the overlay of the model on the histogram, so we use the Chi-Square Test to determine this for us. We see that low p-values indicate that neither model is good enough. However, looking at the alternative distribution models, there seems to be a good fit:

Comparison of Alternative Distributions

Distribution	*Est. Parameters*	*Chi-Square P*
Birnbaum-Saunders	2	0.517597
Weibull	2	0.00121466
Lognormal	2	0.000311699
Gamma	2	0.00000108518
Loglogistic	2	2.61823E-7
Exponential	1	0.0
Largest Extreme Value	2	0.0
Normal	2	0.0
Laplace	2	0.0
Logistic	2	0.0
Pareto	1	0.0
Uniform	2	0.0
Inverse Gaussian	2	0.0
Smallest Extreme Value	2	0.0

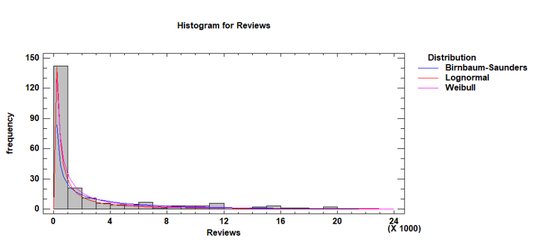
We try the Birnbaum-Saunders model and obtain the following result:

Fitted Distributions

Birnbaum-Saunders	*Lognormal*	*Weibull*
shape = 4.39096	mean = 7076.42	shape = 0.435175
scale = 203.417	standard deviation = 203545.	scale = 880.084
	Log scale: mean = 5.5048	
	Log scale: std. dev. = 2.59219	

Goodness-of-Fit Tests for Reviews
Chi-Square Test

	Birnbaum-Saunders	*Lognormal*	*Weibull*
Chi-Square	29.0	63.8	59.0
D.f.	30	30	30
P-Value	0.517597	0.000311699	0.00121466

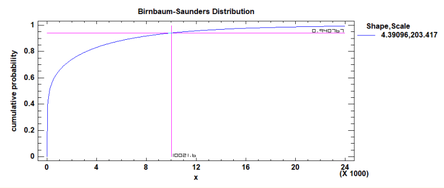


Looking at the overlay of the model on the histogram, it seems to fall short when describing lower number of reviews, but this is only because we are seeing a simplified version of the data. The Chi-Squared test can consider the entire sample, and it has deduced that Birnbaum-Saunders is the best model. With this data, we could now know what is the probability of a random app having more than a certain number of reviews.

There is no definitive metric for how many users makes an app successful, but we can probably agree that receiving over 10000 reviews for your app is a nice milestone to cross. However, what is the probability that your app will surpass this limit? According to our data, around 6%.

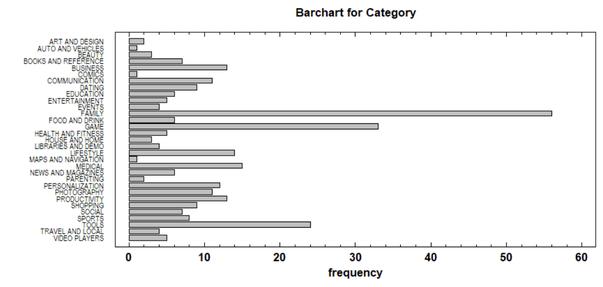
Upper Tail Area (>)

Variable	*Dist. 1*	*Dist. 2*	*Dist. 3*	*Dist. 4*	*Dist. 5*
10000	0.0588727				



CONFIDENCE INTERVALS AND HYPOTHESIS TESTING

Now we can start including parameters to more precisely determine how many reviews an app will get. Starting with the category of the app, we noticed that, when we make a barchart of apps by category, the most frequent category was Family, followed by Game:



Perhaps apps developers already know which are the most downloaded and reviewed apps in the market and they are creating apps in these categories in order to attract more people. We want to know if this is the case of Family apps, the most popular category, and see if there exists a significant difference between the number of reviews received in this category in comparison to others.

We can figure this out using a hypothesis test for the mean. As both categories have n>30, we know by the central limit theorem that the sample mean will be a random variable whose expected value will be the category average reviews, and the variance of the sample will be the variance of the category too:

Summary Statistics

	FamilyReviews	*nofamilyReviews*
Count	56	244
Average	95228,6	66708,1
Standard deviation	132638,	169222,
Coeff. of variation	139,283%	253,676%
Minimum	3,0	1,0
Maximum	587523,	896118,
Range	587520,	896117,
Stnd. skewness	5,50896	22,03
Stnd. kurtosis	4,94398	38,1885

t test to compare means
Null hypothesis: mean1 = mean2
Alt. hypothesis: mean1 > mean2
not assuming equal variances: t = 1,37296 P-value = 0,086409
Reject the null hypothesis for alpha = 0,1.

As we reject the null hypothesis, we can conclude that Family apps receive on average a higher number of reviews compared to the rest of categories, considered a 90% confidence level. Although it would have been better to have 95% or 99% confidence, we are sure that we lack a lot of data to make statements with such high confidence.

It would be interesting to make a Paid vs Free comparison too, but unfortunately our sample has only 22 apps in the Paid group. However, we can strongly suspect that it will play some role, since it's easier to get a free app than a paid one, obviously.

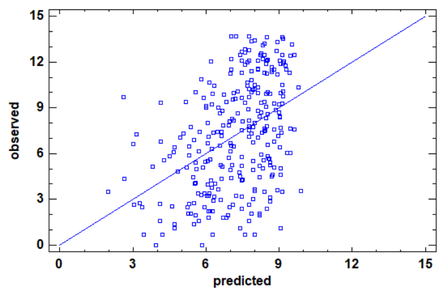
MULTIPLE REGRESSION

Once we know which is the best category for an app based on the of number of reviews, we will proceed to find a relation between reviews and some of the characteristics which a developer has power over when creating the app, including size and price of the app and two cualtitative variables which will distinguish between a free or paid app, and if the app belongs to the family category or not. We will use dummy variables:

	PriceCategory=1
Free app	1
Paid app	0

	CategoryCodes=1
Family category	0
Other category	1

We created our linear regression model, and applied transformations over the Reviews and Size variables or else we would not meet the homocedasticity condition. We chose to perform a logaritmic tranformation, as we know that the Data is right skewed. Also, an exponential with c<1 did not offer good results. We obtain the following model:



Parameter	*Estimate*	*Standard Error*	*T Statistic*	*P-Value*
CONSTANT	1,88012	0,9687	1,94087	0,0533
PriceCategory=1	3,04667	0,896129	3,39981	0,0008
CategoryCodes=1	-1,09066	0,497669	-2,19154	0,0292
LOG(Size)	1,10984	0,178918	6,20306	0,0000
Price	0,0287658	0,119806	0,240103	0,8104

Considering a hypothesis test where H_0:βi=0, H_1:βi≠0, we fail to reject H_0 for the variable Price, therefore being a non-significant regressor, as p-value>α=0.1. After its elimination, we obtain the following model:

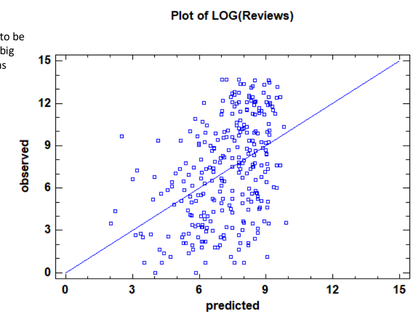
Parameter	*Estimate*	*Standard Error*	*T Statistic*	*P-Value*
CONSTANT	2,02827	0,74551	2,72065	0,0069
PriceCategory=1	2,90244	0,663907	4,37176	0,0000
CategoryCodes=1	-1,09013	0,49684	-2,19413	0,0290
LOG(Size)	1,10826	0,178501	6,20869	0,0000

Note: Free apps seem to be significant and have a big impact in this model, as previously suspected

Analysis of Variance

Source	*Sum of Squares*	*Df*	*Mean Square*	*F-Ratio*	*P-Value*
Model	695,641	3	231,88	21,73	0,0000
Residual	3040,73	285	10,6692		
Total (Corr.)	3736,37	288			

R-squared = 18,6181 percent
R-squared (adjusted for d.f.) = 17,7614 percent
Standard Error of Est. = 3,26638
Mean absolute error = 2,747
Durbin-Watson statistic = 1,98333 (P=0,4438)
Lag 1 residual autocorrelation = 0,00447927



Although all our variables are significant for the model now, we should now check if the model is valid. We will first see if the residuals fit in a normal distribution:

Tests for Normality for RESIDUALS

Test	*Statistic*	*P-Value*
Chi-Square	54,4983	0,014312

Considering a hypothesis test where H_0:model fits in a normal distribution, H_1:model does not fit, we reject H0 as p-value<α=0.1. Therefore the residuals do not fit in a normal distribution. In addition to this, by looking at the value of R-squared, we can see that our model only covers 18.6181% of the population. This means that our dataset lacks some of the data needed to accurately predict values of Reviews for a given set of independent variables. This conclusion was expected, as app reviews depend not only on the characteristics of the app, but also on lots of external factors including luck or trends, which we are not able to predtict with our dataset.

CONCLUSIONS

To conclude, we will collect all the data we have obtained from our study, and discuss how it can be used to answer our initial question. As it was expected, we found a big level of right skewness on the population when analysing the number of reviews, meaning that the majority of the apps receive none to a few reviews. However, the box and whisker plot revealed a reasonably high amount of outliers, indicating the presence of famous or trending apps. Although achieving this amount of reviews is the goal of every app developer, this is quite unlikely to happen, so we calculated the probability of achieving a more feasible amount of reviews: 10000. Our conclusions are not good news for the developers, resulting that only around a 6% of the apps surpass this number.

When analysing different app categories, we realised that some categories appeared more often than others. We suspected this was due to their increased popularity, so we proceeded to test if this was the case of Family, the category with the highest frequency. A mean test proved that this type of apps receive more reviews on average, therefore this category will provide better results if a developer chooses it when creating a new app. Also, as seen in the "Multiple Regression" Section, Free apps (priceCategory=1) seem to have a strong impact on the number of reviews.

By studying the possible correlation between other characteristics of an app and reviews, we noticed a regression model could not fit. This is probably due to lack of significant data and the unpredictability of the market, where lots of external factors determine the success of an app. However, we can deduce from the slopes of the variables in the model that creating a Family, free, big sized app will provide somewhat higher chances to be into the 6% of apps that achieve a decent amount of reviews.