Mobile Video Game Analysis

RA Data Exercise

January 2nd 2022

Aryan Mishra

Part 1: Exploratory Data Analysis (EDA)

For this section, our objective is to get a good understanding of the data so we can infer some valuable insights that might be useful for the reader. Before performing any operations, preliminary data-prepossessing was performed which involved checking and accounting for missing data. There are indeed multiple ways we can deal with missing data, such as imputing the missing values with the mean or median. However, in our case, we simply dropped all the rows which contained missing values. This simplified our data set and didn't come at a great cost since we still had plenty of data to work with. We then continued ahead with the Exploratory Data Analysis.

For EDA, we performed the following operations:

1. Calculating summary statistics.
2. Visualizing the distribution of the received log scores.
3. Calculating game popularity and average player performance.
4. User-level analysis
   (a) Calculating play frequency within a day.
   (b) Calculating play frequency across different days.

1. Calculating Summary Statistics

The following table shows the summary statistics that were calculated.

	u_id	retry_count	score	stars	stage_id	obs_index	log_score	year
count	3.621210e+05	362121.000000	3.621210e+05	362121.000000	362121.000000	3.621210e+05	362121.000000	362121.000000
mean	1.015695e+08	0.031619	2.062362e+06	0.688897	87.329462	4.870494e+07	8.826077	2015.808953
std	8.706389e+05	0.270048	2.589924e+07	0.934186	303.475852	2.795856e+07	6.000948	0.976860
min	1.000001e+08	0.000000	0.000000e+00	0.000000	1.000000	1.081400e+04	0.000000	2015.000000
25%	1.008297e+08	0.000000	0.000000e+00	0.000000	8.000000	2.351664e+07	0.000000	2015.000000
50%	1.016286e+08	0.000000	1.104050e+05	0.000000	22.000000	5.028673e+07	11.611920	2016.000000
75%	1.022716e+08	0.000000	5.691000e+05	1.000000	60.000000	7.328008e+07	13.251813	2016.000000
max	1.032131e+08	18.000000	2.147484e+09	3.000000	2192.000000	9.671235e+07	21.487563	2019.000000

Figure 1: Summary statistics.

As seen from the table above, the mean score is approximately 2,062,362, with the highest score being approximately 2,147,484,000 and the lowest being zero. We also note that the mean number of stars awarded to players is approximately 0.68.

2. Visualizing the Distribution of the Received Log Scores

The following figure shows the distribution of the received log scores.

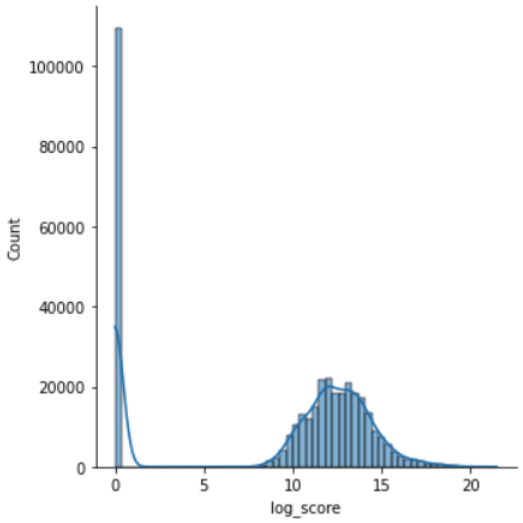

Figure 2: Distribution of log-scores.

As we can see, we have a very high count of zero (log) scores, indicating that a lot of players attempted to play the game but failed to achieve a non-zero score. Scores from 5 to 20 follow a bell-shaped curve, with the peak being somewhere between 10 and 15.

3. Calculating Game Popularity and Average Player Performance.

Looking at the data set, we could infer the level of popularity of game, and how the popularity varies across time. There are several ways we could

quantify this. One metric could be to calculate the number of unique players who played the game across a certain unit of time (e.g. day, year). We could also measure player performance by calculating the mean scores of all those (unique) players who played on a given day. For example, if three players played the game on a given day, with scores of 1,5, and 9, the mean score of would be 5 for that day.

To calculate how many players played the game on a given day and their average scores, we first dropped all the rows with duplicate user id's, and kept the highest score. This way, we can avoid double-counting users who played the game multiple times on a given day. Ultimately, we will end up with only unique user id's.

We then plotted the number of unique players on a given day every 10 months so we can visualize the trends in data. The following figure shows the resulting bar plot.

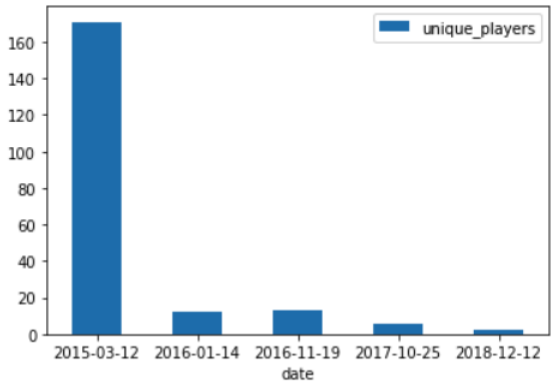

Figure 3: Number of unique players on a given day every 10 months.

The bar plot represents the number of unique players on a given day every 10 months. As we can see, in March 2015, we have the highest number of unique players who played the game, with a count greater than 160. However, this number starts to go down as we progress further in time. This is not surprising since game popularity for most games usually starts to saturate over time after hitting a peak. The next figure shows the mean score over a 10-month interval.

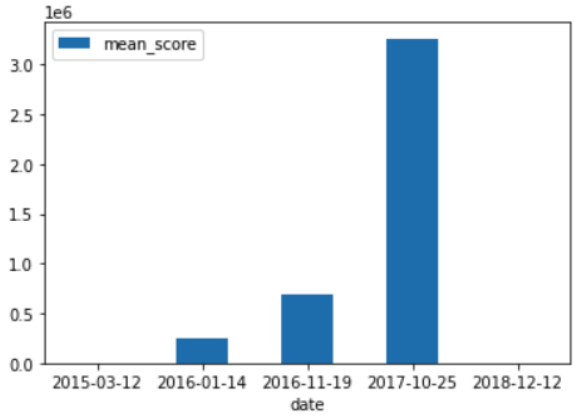

Figure 4: Average scores of players every 10 months.

Looking at the mean score over time, we can infer some insights. Interestingly, the mean score, and thus the average player performance, increases consistently as we move across time (with the exception of December 2018 where the mean score was zero), while the game popularity (generally) decreases over time. This implies that in the beginning, we have a large influx of players who initially try out the game (casual gamers), but as time progresses and the game matures, only the highest performing players end up still playing the game (competitive gamers). Based on my own personal experience, I can support this hypothesis. For example, when the game of Fortnite first started, the game popularity was huge, but the average quality of the players who played the game was pretty low, since most of the players who played the game were casual gamers.

However, if one plays Fortnite now, one can noticeably see the difference in the average quality of players, as more competitive gamers are playing the game compared to casual gamers. Nevertheless, **further research and data will be needed to see if the difference in player quality across time is indeed statistically significant!** Moreover, the trends we are seeing are largely dependent on the time scale we are using. Thus we verify if this general trend holds true using year as the unit of time.

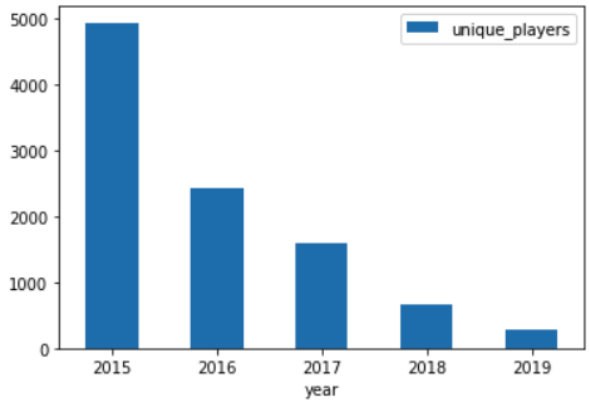

Figure 5: Number of unique players in a given year.

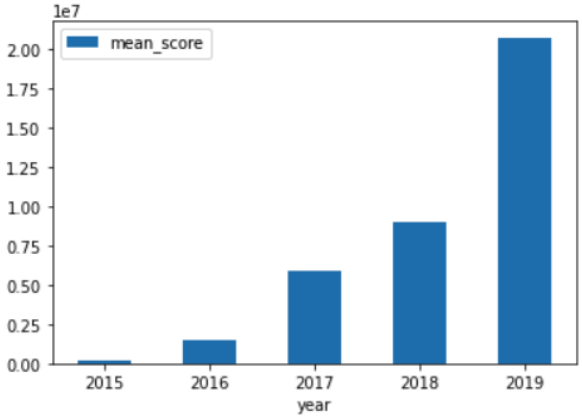

Figure 6: Average scores of players every year.

Indeed, the inverse relationship between the number of unique players and the mean score holds true and is more consistent across the yearly level, with 2015 having the highest number of unique players who played the game and also the lowest mean score, and 2019 having the lowest number of players but the highest mean score.

On an additional note, this is basically the Law of Large numbers in action! When we visualized the trends on a daily level every 10 months, we noticed a general pattern - that game popularity generally increases over time

and is inversely proportional to average player performance, which generally decreases over time. However, there were some minor fluctuations and inconsistencies. Namely 2016-11-19 had more players playing the game than 2016-01-14, and 2018-12-12 had a mean score of zero, much lower compared to 2017-10-25 which had a mean score of greater than 3. This is an example of how when we have a small population (like the number of players in a given day), we generally see more outliers and inconsistent trends, whereas when we have a large population (number of players in a given year), we see trends that are more consistent and aligned with our expectations.

4a) User Level Analysis - Calculating Play Frequency Within a Day.

Given the data set, we can also identify the players who have played the game the most number of times within a given day. To do this, we first checked how many times each player played the game on a given day and then selected the player with the highest count. The following figure shows the number of times each user played the game within a day. For example, user **100000101** played the game **8** times on 2015-03-12, whereas user **100001288** played the game **25** times on the same day.

		count
date	u_id	
2015-03-12	100000101	8
	100001281	3
	100001288	25
	100001327	3
	100001686	3
...	...	...
2019-06-10	103171305	1
2019-06-11	102197628	2
	103208765	3
2019-06-12	103148651	3
	103211454	9

52826 rows × 1 columns

Figure 7: Play frequency within a day.

After performing the required data manipulations, we identified user **102925590** was the player who played the game the highest number of times on a given day. Specifically, this user played the game **137** times on **2017-12-11**!

We can not only identify the players who played the game the most number of times on a given day, but we can also calculate the average number of times players play the game within a day in general, which turns out to be approximately 7 times a day. We can also perform the same calculations but this time aggregate the data by year, so we can find the user who played the game the highest number of times in a given year. It turns out that user **102650635** was the player with the highest number of plays in a given year. Specifically, this user played the game **1335** times throughout **2017**!

4b) User Level Analysis - Calculating Play Frequency Across Different Days.

This time, instead of calculating the number of times users played the game within a day, we can calculate the number of days each user played the game at least once.

	u_id	Number of different days the game was played
0	100000101	3
1	100001281	1
2	100001288	5
3	100001327	4
4	100001686	1
...	...	...
9909	103212247	3
9910	103212318	1
9911	103212744	1
9912	103212776	1
9913	103213122	1

9914 rows × 2 columns

Figure 8: Play frequency across different days.

The figure above shows the number of days each user played the game at-least one time. For example, user **100000101** played the game at-least once across **3** different days, whereas user **100001288** played the game at-least once on **5** different days.

After performing the relevant data manipulations, we found that user **101397994** was the player who played the game at-least once the most number of times across different days. Specifically, this user played the game at-least once on **346** different days - almost everyday throughout a year!

Part 2: Examining the Distribution of Scores Conditional on the Stage of the Game and Stars Received

We will now examine the distribution of scores conditional on the stage of the game and stars received. First, let's plot the kernel density estimation of the log score conditional on the stars, where we plot separate distributions of the log score for each star level.

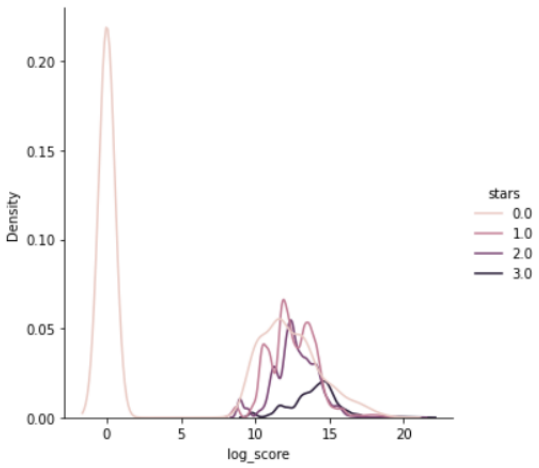

Figure 9: Distribution of log score conditional on star level

As we can see, the majority of the zero scores correspond to a star level of 0. This indicates that many players who played the game failed to achieve any

stars and ended up with a score of zero. However, as we move further along, we can notice some patterns. Firstly, the (non-zero) scores corresponding to zero stars are more positively skewed compared to non-zero stars, indicating most players with zero stars had lower scores than players with non-zero stars. The scores corresponding to 2 stars are slightly less positively skewed compared to scores corresponding to one star, indicating that there are slightly more 2-star players with higher scores compared to 1-star players.The 3-star scores are noticeably negatively skewed, hinting that 3-star players generally score significantly higher than 2-star and 1-star (and of course zero star) players.

In order to further plot the distribution of the scores conditional on the number of stars and stage id, we can first plot the distribution of the stage id to see if there are any significant patterns we need to be aware about. This will also provide us with insights as to how the stages are structured.

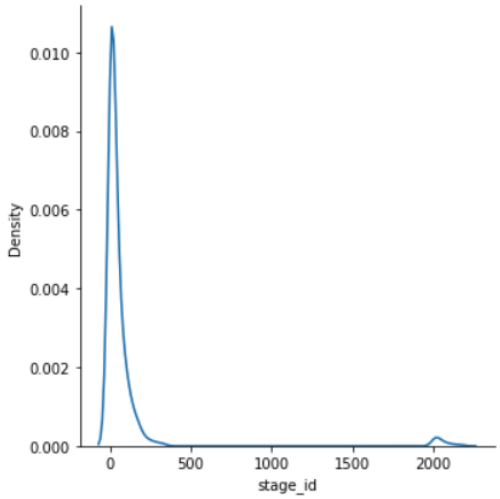

Figure 10: Distribution of stage level

As we can see, we have two peaks when it comes to the distribution of the stage ids. Firstly, most of the plays correspond to stages below stage id 500, with a few plays corresponding to stage id 2000. This means we can divide our analysis into two separate sections, one corresponding to early stage data (stage id $<$ 500), and the other corresponding to late stage data (stage id $>=$ 2000).

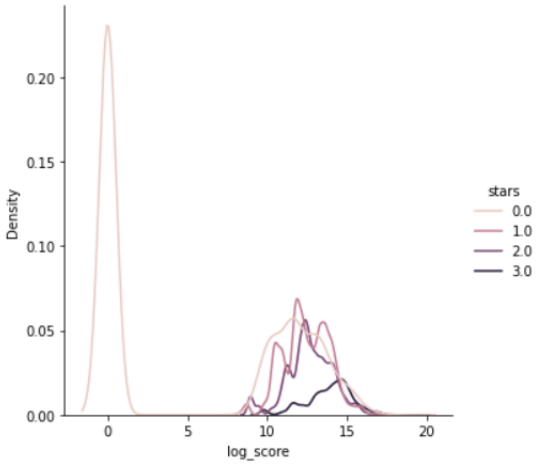

Figure 11: Distribution of log score conditional on star level **and** early stage level

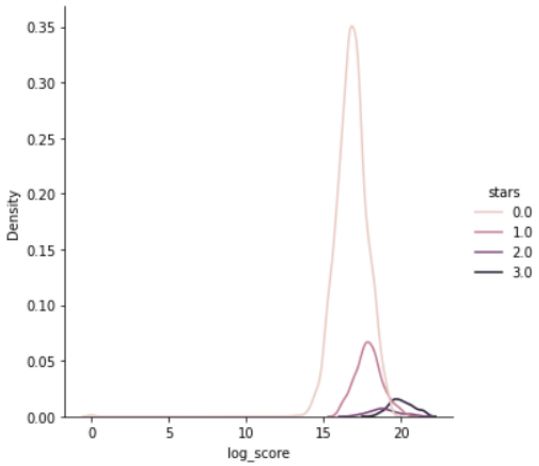

Figure 12: Distribution of log score conditional on star level **and** late stage level

Looking at the two figures above, we can see a noticeable difference between the distribution of the log score conditional on early stage level and late stage level. Firstly, we notice that the early stage level distribution is very similar to the overall distribution of log scores regardless of stage level (figure 9),

with a high density of zero scoring individuals who have zero stars. This is unsurprising since most of the data in the data set corresponds to early stage levels (stage id <= 500). However, looking at the late stage distribution, we notice some significant changes. For example, we notice that no player regardless of the star level scored zero. We also notice that the distribution of the log score conditional on star level and late stage level is more negatively skewed than the distribution of the log score conditional on star level and early stage level. This indicates that scores are generally higher as players progress through later stages of the game. Nevertheless, this doesn't mean that just because a player is in a late stage, they will achieve a star, as seen by a high density of zero-star scores at a late stage level.

Therefore, to summarize, we can infer that stars are only awarded to players with scores higher than zero. A general trend we see is that individuals with more stars generally have higher scores. There might also be a slight positive correlation between stage level and number of stars, since we can see that the distribution of scores is more negatively skewed in late stage level data compared to early stage level data. However, being in a late stage level doesn't necessarily guarantee a star, as seen by the high density of zero stars in figure 12.

Part 3: Proposal of Causal Inference

In this section, we will propose a way to causally test the effect of achieving the goal (reaching a particular star level) on the play decisions of the users. Specifically, we propose applying **Logistic Regression** to test the effect of a star level on the play decision of the users. A very high level overview of how this method can be applied is explained in this section.

Let the play decision of the user be a binary (two level) variable that can be encoded by a dummy variable as follows:

$$Y = \begin{cases} 1 & \text{if player makes a certain decision } D \\ 0 & \text{otherwise} \end{cases}$$

Our goal will be to model the relationship between $p(X) = \Pr(Y = 1|X)$ and X, where X is our independent variable. Specifically, our independent

variable X will be the number of stars. Therefore, we are predicting $Y = 1$ (player makes a decision D) using the number of stars. In a logistic regression model, we use the logistic function,

$$p(X) = \frac{e^{\beta_o + \beta_1 X}}{1 + e^{\beta_o + \beta_1 X}}$$

to model $p(X)$. After manipulating the equation above and taking the logarithm, we end up with the following equation:

$$\log(\frac{p(X)}{1 - p(X)}) = \beta_o + \beta_1 X$$

The equation above represents the logistic regression model. In this model, increasing X (the number of stars) by one unit changes the log odds by β_1, or equivalently, it multiplies the odds by e^{β_1}. Since the coefficients β_0 and β_1 are unknown, we use available training data to estimate them. Specifically, in the case of logistic regression, we use the maximum likelihood approach to estimate the coefficients. Once the coefficients have been estimated, we simply input the coefficients into the equation $p(X) = \frac{e^{\beta_o + \beta_1 X}}{1 + e^{\beta_o + \beta_1 X}}$ to calculate the probability of a player making a certain decision D.