A/B Testing on Mobile Game Marketing and Design

Marketing Analytics II

Anshika Ahuja, Apoorva Jasti, Karan Palsani

Under the guidance of Prof Duan

The University of Texas at Austin
Texas McCombs
MS Business Analytics
McCombs School of Business

Table Of Contents:

1. Background and Problem Statement

We ventured into the **impact of game design and marketing on user retention rates** or bounce rates.

Cookie Cats is a hugely popular mobile puzzle game developed by Tactile Entertainment. It is a classic "connect three"-style puzzle game where the player must connect tiles of the same color to clear the board and win the level. It also features singing cats. You can check out a short demo on the following YouTube link: https://youtu.be/GaP5f0jVTWE

As players progress through the levels of the game, they will **occasionally encounter "gates" that force them to wait a non-trivial amount of time or make an in-app purchase to progress**.

In addition to driving in-app purchases, these gates serve the important purpose of giving players an enforced break from playing the game, hopefully resulting in that the player's enjoyment of the game being increased and prolonged.

The question that we are trying to answer over the course of this project is: Where should the gates be placed? In the original version of the game, the first gate was placed at level 30. **In this project, we are performing an AB-test to decide what would be the most optimal placement of the first gate in Cookie Cats. More specifically, we will look at the impact of this decision on player retention.**

The data for this has been taken from a Kaggle dataset. The data has been collected in this dataset by the game developers over the two versions of the game described above. The exact process is via game data stored on user's phones in the form of cookies created when the game is installed. The exact data collected was:

- The number of levels played over a certain number of days by the user
- The number of days this user continued playing the game

Research Impact:

It probably will not surprise you to hear that **online gaming** is a money-making **industry**. According to GamesIndustry.biz, a $134.9 **billion industry** to be exact, which is 10.9% higher than last year (2018). Even the current pandemic phase has only furthered the online gaming industry's dedicated user base.

This industry has a knack for digital marketing. Digital marketing is in fact a requisite in this gaming industry. This industry produces digital products, so it is easy to use different digital marketing techniques in the gaming industry. The gaming industry is undergoing some swift changes, and the marketing industry needs to keep up by constantly fine-tuning their marketing tactics.

Currently, the app stores either lack a means of A/B testing (like Apple App Store) or offer limited functionality (as with Google Play Experiments), despite the fact this strategy is paramount to identifying what really increases game or application engagements.

The benefits of A/B testing for game developers are nearly endless, but some of the highlights include:

- Higher conversion rates
- Analysis of engagement metrics
- Decision making based on data and statistics
- Better use of resources
- Insights into customer behavior

A/B testing is the perfect tool as it allows all developers to identify room for growth in their apps' landing pages, make significant improvements based on the results, and of course, understand their users' behavior.

With the emergence of analytics in the marketing space, user behavior mapping and A/B testing have become inseparable. As soon as you combine them, you can bring your A-game to the party. Most game design questions would be easily answered if you could just test your players' reactions to different solutions. Our desire is to test the significance of one such gaming scenario via this project to understand the value added by this technique to this niche sector.

Initially the control group has a gate placed on level 30, but what if we place the gate at level 40. There are chances the user might be interested to play more i.e. retention might increase for each player and might further increase user traffic for the mobile game but to be confident enough we should back up our conversion rates with valid explanation and statistical analysis. To check whether placing the gate is leading to more retention we perform A/B testing.

The unit of diversion is a user id through which the user (experimental units) are randomly split into two different groups i.e. Control and Experiment. Evaluation metric or response variable chosen here is the retention of a player. Our interest via this AB test lies in comparing means of both the control and experimental group

2. Data Overview

The data is from 90,189 players that installed the game while the AB-test was running. There are five major variables used for this project and the AB-test.

The variables used are:

- userid - a unique number that identifies each player.
- version - whether the player was put in the control group (gate_30 - a gate at level 30) or the test group (gate_40 - a gate at level 40).
- sum_gamerounds - the number of game rounds played by the player during the first week after installation
- retention_1 - did the player come back and play 1 day after installing?
- retention_7 - did the player come back and play 7 days after installing?

When a player installed the game, he or she was randomly assigned to either gate_30 or gate_40.

The dataset is an experimental dataset with a primary data source. It was collected by the game developer, Tactile Entertainment and has been hosted on Kaggle.

3. Exploratory Data Analysis

To start with we looked at a rough summary of the dataset:

```
#Check for missing values
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 90189 entries, 0 to 90188
Data columns (total 5 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   userid          90189 non-null  int64
 1   version         90189 non-null  object
 2   sum_gamerounds  90189 non-null  int64
 3   retention_1     90189 non-null  int64
 4   retention_7     90189 non-null  int64
dtypes: int64(4), object(1)
memory usage: 3.4+ MB
```

From the above output, it can be seen that:

1. The values in the dataset do not have any null values.
2. The userid contains all unique ids.
3. The sum_gamerounds variable is an integer value while retention_1 and retention_7 are boolean variables (1 or 0; True or False).
4. version is a categorical variable.

The next important check for any AB test is to verify the split between test and control groups:

```
#Looking at the split between test and control groups
df.groupby('version').count()
```

version	userid	sum_gamerounds	retention_1	retention_7
gate_30	44700	44700	44700	44700
gate_40	45489	45489	45489	45489

Here we can see an even split between the two groups and thus can confirm that we have avoided one aspect of selection bias here.

The next check if for any outliers in our dataset, one example is shown below with the sum_gamerounds variable:

```
#Analyzing player level history
df['sum_gamerounds'].plot.box(figsize=(5,10))
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f2a04aadf60>
```

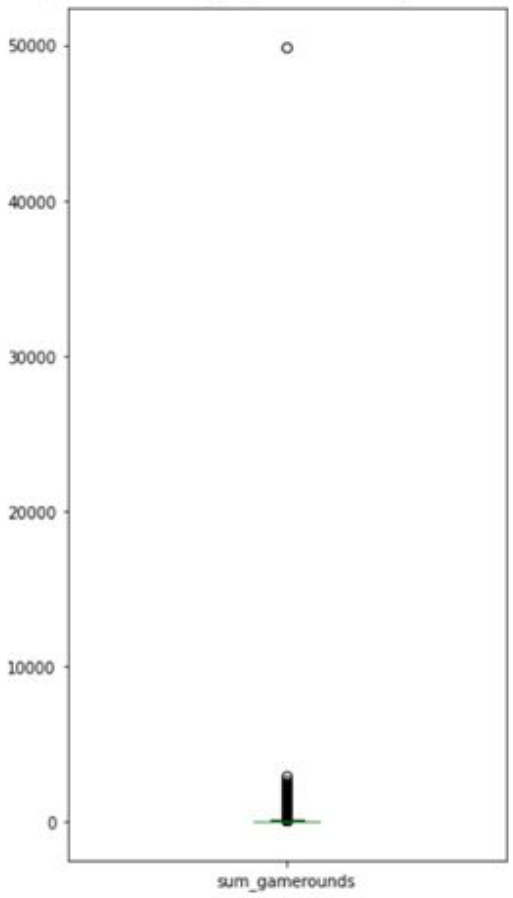

Fig 1: Player level history

Here, we can see one extreme outlier which shows almost 50,000 gamerounds within 14 days of the game's release. This outlier was removed from our dataset immediately.

```
#Removing the outlier
df['sum_gamerounds'].describe()
df = df[df.sum_gamerounds != 49854]
```

We next checked for the player distribution in terms of the numbers of rounds played and, as expected, we could observe the classic long tail plot here. Showing a few overachievers followed by the rest of the herd:

```
#Player level distribution
plot_df = df.groupby('sum_gamerounds')['userid'].count()
ax = plot_df[:].plot(figsize=(10,6))
ax.set_title("The number of players that played each # of rounds during the first week")
ax.set_ylabel("Number of Players")
ax.set_xlabel('# Game rounds')
```

```
Text(0.5, 0, '# Game rounds')
```

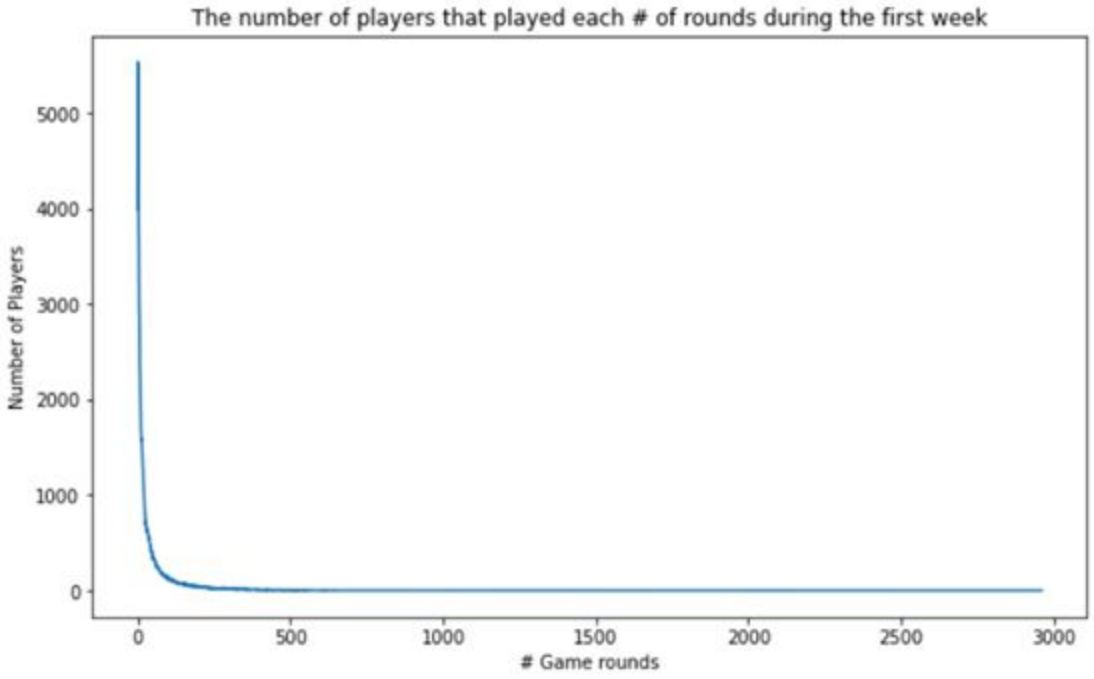

Fig 2: No of rounds played during the first week vs the no of players

Finally, we verified if the characteristics of sum_gamerounds has any major differences in the two groups. This information would be useful to understand the validity of the A/B test. Therefore, we had to compute the below summary statistics for the same:

	version	Total_Games_Played	Total_Users	Average	Min_value	Max_value
0	gate_30	2294941	44699	51.342111	0	2961
1	gate_40	2333530	45489	51.298776	0	2640

This table tells gives us:

1. Total_Games_played: The total games played in each variant.
2. Total_users: Total number of users who installed the game.
3. The average number of games played by a user in each variant is nearly the same for the experiment and control group.
4. The minimum and maximum numbers of games played by users.

The main aim of this analysis is to understand when the user engagement is higher and if there were any major differences straight off the bat.

4. Methodology

4.1 Hypothesis Testing using Bootstrapping

4.1.1 Overview

Bootstrapping is a resampling technique used to make an inference about an estimate. It resamples data from a single dataset to create multiple samples. We can then perform hypothesis testing late standard errors, construct confidence intervals, and perform hypothesis testing for numerous types of sample statistics. This method is an alternative approach to traditional hypothesis testing and is notable for being easier to understand and valid for more conditions.

4.1.2 Choosing the metric

From the figure 2 above, we can see that more than 50% of the players install the game but never play it, some play a couple of games in the first week and some get hooked. Our aim is to make the players like the game and get hooked to it.

For the metric, we wanted a variable which can gauge how engaging the game is and if the users find the game to be fun. A common metric used in the gaming industry is retention which basically depicts the number of players that are still hooked to the game, Here we decided to use 2 metrics namely - 1 day retention and 7-day retention. Our hypothesis is that higher the retention values, the easier it will be to retain them for longer periods of time and build a large player base.

4.1.3 1-day Retention

Overall 1-day retention

First we calculated the overall 1-day retention for both the groups

```
df['retention_1'].mean()
```

```
0.4452144409455803
```

Only a little Less than 50% of the players come back to play the game after installation. Next, we looked at the difference in 1-day retention between the two AB groups.

1-day retention for both AB groups

```
[ ]  #Proportion of users converted in at 1-day for each AB group
     print (df[df.version == 'gate_30']['retention_1'].mean())
     print (df[df.version == 'gate_40']['retention_1'].mean())
```

```
0.4481979462627799
0.44228274967574577
```

From the above results, we observed that there was a slight decrease in 1-day retention when the gate was moved to level 40 (44.2%) compared to the control when it was at level 30 (44.8%).

The change is really small, but since our metric is retention even a small change can have a huge impact on the game's popularity.
From this calculation we were confident that there is a difference between the two groups, but we wanted to figure out how certain we should be that a gate at level 40 will be worse than a gate at level 30 even in the future.

How confident are we in the difference in retention numbers?

To assess the certainty of these results, we decided to use Bootstrapping for Hypothesis Testing. Here we repeatedly resampled our data and calculated the retention values. Then we checked the variation in the retention values to get an idea of the uncertainty of the numbers.

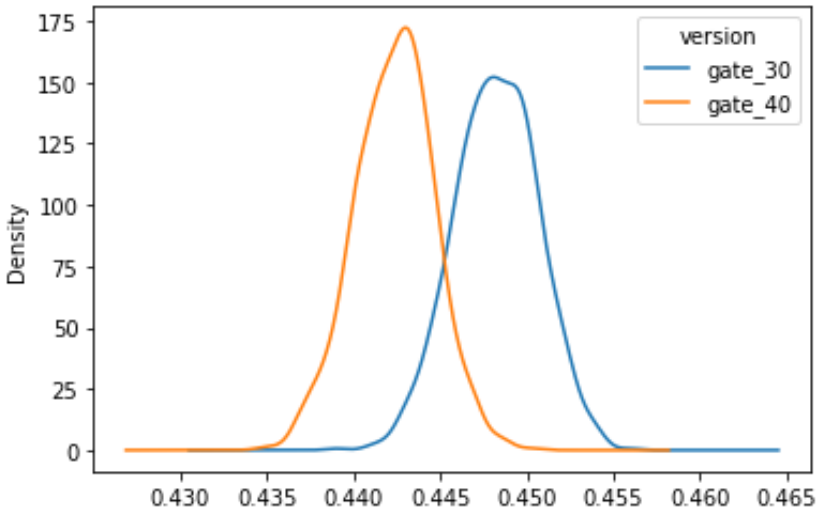

Fig 3: Variation in the retention values for both AB groups

The above figure showed us the uncertainty in the distributions of 1-day retention values for both the AB groups. It was clear that there was some evidence of a difference. To get a more clear picture, we decided to plot the % difference between the two groups.

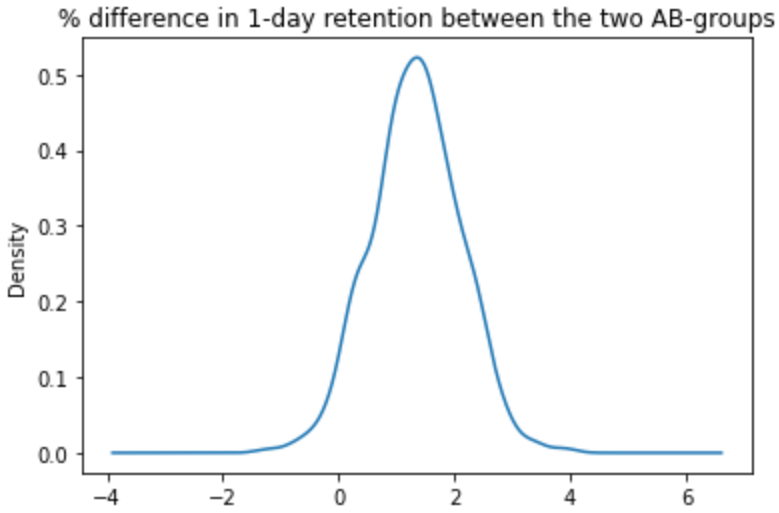

Fig 4: % difference in 1 day retention between two AB groups

From figure 4 above, it is evident that the % difference with the highest density lies between 1%-2%. We also concluded that around 96% of the distribution is above 0% and is in favour of a gate at level 30.

4.1.4 7-day Retention

The above bootstrap analysis told us that there is a high probability of 1 day retention being more when the gate is at level 30. However, the important point to consider here is that since players have only been playing the game for one day, it is possible that most of the players haven't even reached level 30. That is there will be no effect of the gate whether it is at level 30 or at level 40. Therefore, to get a better picture, we decided to analyze the 7-day retention numbers.

Overall 7 day retention

Similarly we calculated the overall 7-day retention for both the groups

```
[57] df['retention_7'].mean()
```

```
0.1860557945624695
```

The 7-day retention values showed a totally story. Only 18.6% of the players come back to play the game 7 days after installation.

7-day retention for both AB groups

```
[58] #Proportion of users converted in at 7-day for each AB group
     print (df[df.version == 'gate_30']['retention_7'].mean())
     print (df[df.version == 'gate_40']['retention_7'].mean())
```

```
0.19018322557551623
0.18200004396667327
```

- Similar to 1-day retention, 7-day retention is also slightly lower when the gate is at level 40 (18.2%) than when the gate is at level 30 (19.01%)
- The difference in7-day retention (0.81%) is higher than that of 1-day retention (0.6%). This might be because more players would have reached the gate.

- The overall 7-day retention is lower than the overall 1-day retention. This means that fewer people play a game a week after installing than a day after installing which makes sense.

But as we did for 1-day retention, we will use bootstrapping to figure out how certain are our findings for 7-day retention.

How confident are we in the difference in retention numbers?

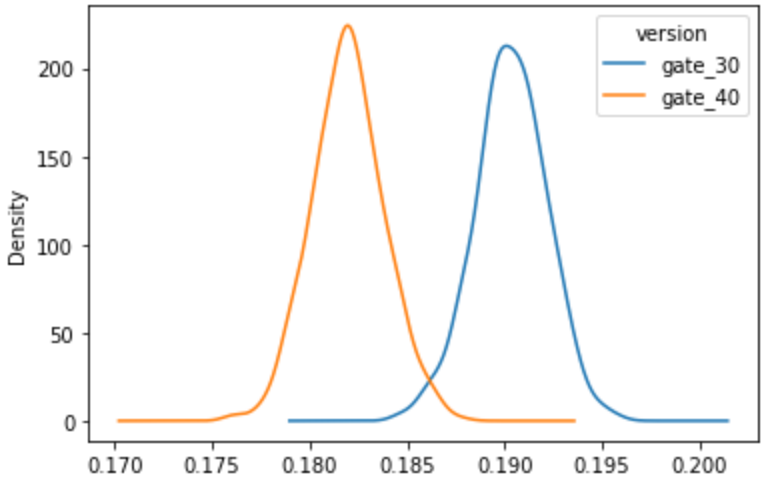

Fig 5: Variation in the retention values for both AB groups

The above figure showed us the uncertainty in the distributions of 7-day retention values for both the AB groups. It was clear that there was some evidence of a difference. To get a more clear picture, we decided to plot the % difference between the two groups.

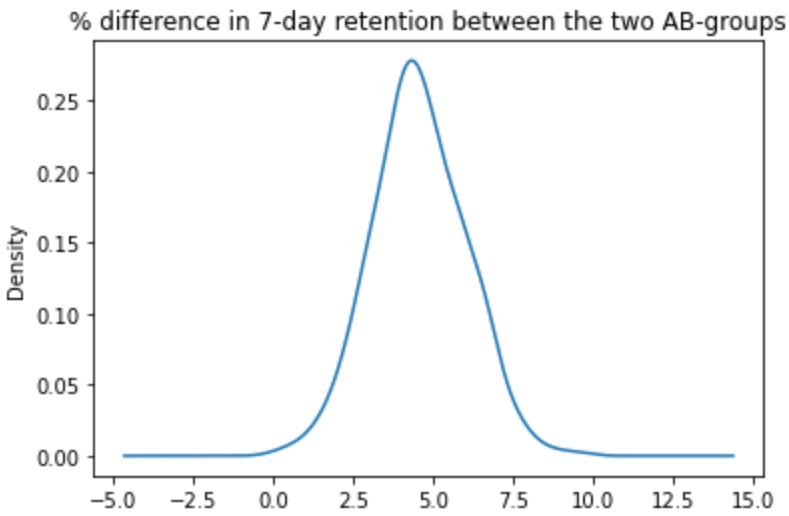

Fig 6: % difference in 7- day retention between two AB groups

The above result indicates that there is strong evidence of 7-day retention being higher when the gate is at level 30 than when the gate is at level 40. From the graph, it is evident that the % difference with the highest density lies between 1%-2%. We also concluded that around 96% of the distribution is above 0% and is in favour of a gate at level 30.

4.1.5 Key Findings and Conclusion

Our conclusion from the above results is that if we want to keep the retention numbers high - for both 1-day retention period and 7-day retention period , we should not move the gate from level to level 40

4.2 Hypothesis Testing using T-test

4.2.1 Overview

We used an independent sample t-test to compare the difference between means of two different groups drawn from the same population. We ensured that the 2 conditions needed to perform t-test are satisfied :

- The samples drawn should be random - Given in dataset
- The data should follow a normal distribution - Checked in EDA

Then we calculated the mean of each group i.e. Control and Experiment to look at which group has a `higher rate of retention". When calculating it, the variability among users and for the whole group was expected. To account for this variability, we computed the margin error of the difference between means. The margin of error was calculated using the pooled standard deviation of the parameter and a Z-score multiplied to it.

```
retention_day_1 = df.groupby('version')['retention_1'].sum()
user_table_day1 = df.groupby('version')['userid'].count()
retention_gate_30_day1=round((retention_day_1['gate_30']/user_table_day1['gate_30']),4)
retention_gate_40_day1=round((retention_day_1['gate_40']/user_table_day1['gate_40']),4)
```

```
[22]
p_pool_day1 = (retention_day_1['gate_30'] + retention_day_1['gate_40'])/(user_table_day1['gate_30']+user_table_day1['gate_40'])
p_pool_day1
```

```
0.4452144409455803
```

```
[23] se_pool_day1=round(mt.sqrt(p_pool_day1*(1-p_pool_day1)*(1/user_table_day1['gate_30']+ 1/user_table_day1['gate_40'])),4)
se_pool_day1
```

```
0.0033
```

4.2.2 Confidence Interval

We also looked at the confidence interval which gives us the range of all the plausible values that our result can have in this population. We decided to look at the 95% confidence interval. As per this experiment, the Control group is better than the Experiment group. We did the similar calculation for day 7 as well.

```
[24] # For 95% confidence interval the value of Z is 1.96 either we can use Z score table or we can use scipy package to calculate it
     alpha=0.05
     z=round(norm.ppf(1-alpha/2),2)
     #Marginal Error calculation
     Marginal_Error=round((z*se_pool_day1),4)
     Marginal_Error
```

```
0.0065
```

```
[25] #The mean difference in the samples
     p_difference = round((retention_gate_40_day1-retention_gate_30_day1),4)
     p_difference
```

```
-0.0059
```

```
[26] print ("The confidence interval is (%s, %s)" %(p_difference-Marginal_Error,p_difference+Marginal_Error))
```

```
The confidence interval is (-0.0124, 0.0005999999999999998)
```

```
[27] t_statistic=p_difference-Marginal_Error
     t_statistic
```

```
-0.0124
```

```
[ ] if t_statistic>0.05:
        print("Experiment performed better than control. It is statistically and practically significant")
    elif t_statistic>0:
        print("Experiment performed better than control.It is statistically significant but not practically")
    elif t_statistic<0:
        print("Control group is better than the Experiment group. Hence, No need to make changes. ")
```

4.2.3 Estimating Effect Size

Lastly we decided to calculate the effect size to understand the impact of the change. Higher the sample size, higher are the chances of rejecting any null hypothesis even when there is a weak relationship in the population. Hence, we should calculate the effect size which tells us about the impact of the change. Effect size also complements the statistical significance

We computed the effect size using Cohen's d in the Independent sample (denoted by d).

- If the value of effect size is between 0.20 and 0.50, the impact is weak.
- If the value of effect size is between 0.51 and 0.80, the impact is moderate.
- If the value of effect size is greater than 0.80, the impact is strong.

```
[29] #cohen's d
     d=round((p_difference/se_pool_day1),2)
     d

     -1.79
```

4.2.4 Key Findings and Conclusion

The experiment shows a negative impact of the launch of the Experiment group or introducing the gate at level 40. Hence, we concluded that we shouldn't be launching the new version.

5. Marketing Strategy Recommendations and Future Scope

Final Recommendations

From both our analysis- Bootstrapping and T-test we concluded that there would be a negative impact from the launch of the new version and there is significant evidence that moving the gate from level 30 to level 40 would actually decrease the retention numbers.

Hence, our final recommendations to the Cookie Cats game developers would be to not move the gate from level 30 to level 40 and continue with the current version of the game.

Future Scope

We would like to look at the effect of changing the gate by other metrics such as geography, in-game purchases etc. However, retention still remains to be one of the most important metrics as the money spent in-game would not matter if we aren't able to increase our retention numbers and get more players hooked to the game.

So, why is retention higher when the gate is positioned earlier? One could expect the opposite: The later the obstacle, the longer people are going to engage with the game. But this is not what the data tells us. Hence, we need more information to understand if customers are facing different obstacles in earlier levels leading to this.

The theory of hedonic adaptation can give one explanation for this. In short, hedonic adaptation is the tendency for people to get less and less enjoyment out of a fun activity over time if that activity is undertaken continuously. By forcing players to take a break when they reach a gate, their enjoyment of the game is prolonged. But when the gate is moved to level 40, fewer players make it far enough, and they are more likely to quit the game because they simply got bored of it.

6. Appendix

6.1 Data Dictionary

Column Name	Data Type	Entry Example	Description
userid	int	116	A unique number that identifies each player
version	string	3	Whether the player was put in the control group (gate_30 - a gate at level 30) or the group with the moved gate
sum_gamerounds	int	‘gate_30’	The number of game rounds played by the player during the first 14 days after install.
retention_1	binary	TRUE	Did the player come back and play 1 day after installing?
retention_7	binary	FALSE	Did the player come back and play 7 days after installing?

6.2 References

- https://www.kaggle.com/yufengsui/mobile-games-ab-testing
- https://classroom.udacity.com/courses/ud257
- https://towardsdatascience.com/a-summary-of-udacity-a-b-testing-course-9ecc32dedbb1