Group 7:

- Ersakanti Belva
- Elvindia Puja Hanggrita
- Randy Ansari Nur Hidayat

Cookie Cats Experiment

April 22, 2022

Cookie Cats is a hugely popular mobile puzzle game developed by Tactile Entertainment. It’s a classic “connect three” style puzzle game where the player must connect tiles of the same color in order to clear the board and win the level. It also features singing cats.

As players progress through the game they will encounter gates that force them to wait some time before they can progress or make an in-app purchase. In this project, we will analyze the result of an experiment where the first gate in Cookie Cats was moved from level 30 to level 40. In particular, we will analyze the impact on player retention and game rounds.

Dataset Cookie Cats

Goal/Objective

Analyze the impact on player retention and game rounds after the first gate was moved from level 30 to 40.

General Problem

Players were forced to wait some time in a certain gate before they could progress or make an in-app purchase. We want to ensure that the players keep playing even after encountering the first gate while also wanting to see the impact on the retention rate and game rounds.

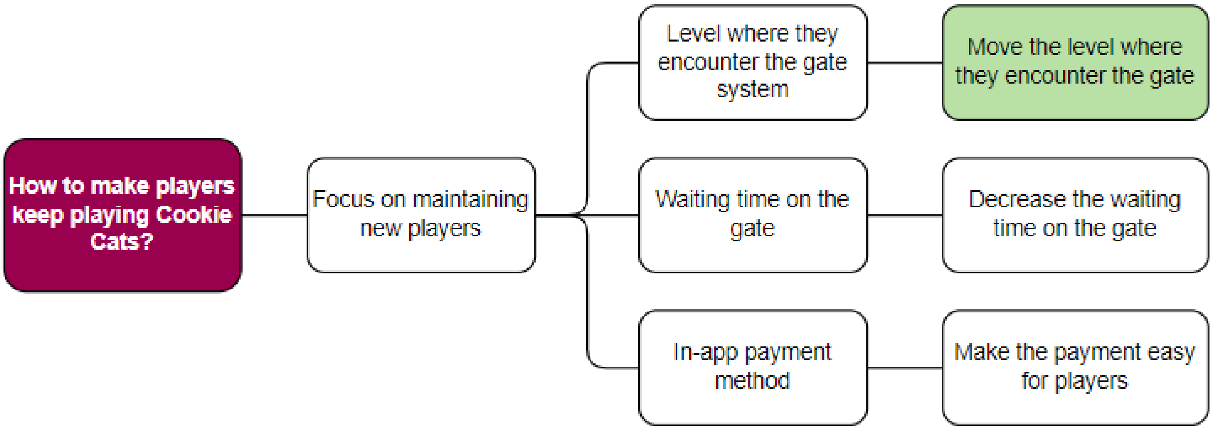
Root Cause Analysis

- Players stop playing after encountering the first gate
- Some players feel that they waited too long to progress to the next level or to make an in-app purchase

Problem Statement

- Players feel that Cookie Cats's gate system is not placed in the right level
- Players feel that the waiting time in certain gate is too long

Proposed Solution



How to make players keep playing Cookie Cats?

Focus on maintaining new players

Level where they encounter the gate system

Move the level where they encounter the gate

Waiting time on the gate

Decrease the waiting time on the gate

In-app payment method

Make the payment easy for players

Key Metrics

The impact on retention rate and game rounds of the players.

Population

New Cookie Cats players.

Business Hypothesis

Moving the first gate from level 30 to level 40 will impact the retention rate and game rounds of the Cookie Cats players.

Hypothesis

Retention 1

H0: There is no statistical significant relationship in retention_1 between gate_30 (the control group) and gate_40 (the Experiment group)

H1: There is a statistical significant relationship in retention_1 between gate_30 (the control group) and gate_40 (the Experiment group)

Retention 7

H0: There is no statistical significant relationship in retention_7 between gate_30 (the control group) and gate_40 (the Experiment group)

H1: There is a statistical significant relationship in retention_7 between gate_30 (the control group) and gate_40 (the Experiment group)

Game Rounds

H0: The median sum_gamerounds between gate_30 (the control group) and the gate_40 (the Experiment group) are similar.

H1: The median sum_gamerounds between gate_30 (the control group) and the gate_40 (the Experiment group) are not similar.

Experiment Groups and Period

Sample	: New Cookie Cats player
Control	: 44,700 players with the first gate on level 30
Experiment	: 45,489 players with the first gate on level 40

Experiment Monitoring

The experiment monitoring was done in Google Data Studio.

https://datastudio.google.com/u/0/reporting/fb461048-e7a3-43e8-9747-2e4827ca171d/page/dd2qC/edit

Analysis

https://colab.research.google.com/drive/1mTFb4QhGUVzTvaDiBmTTxVAoyjkjRdJs?usp=sharing#scrollTo=EOutOcbfO2-k

- **Data Collection**

userid — a unique number of 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).

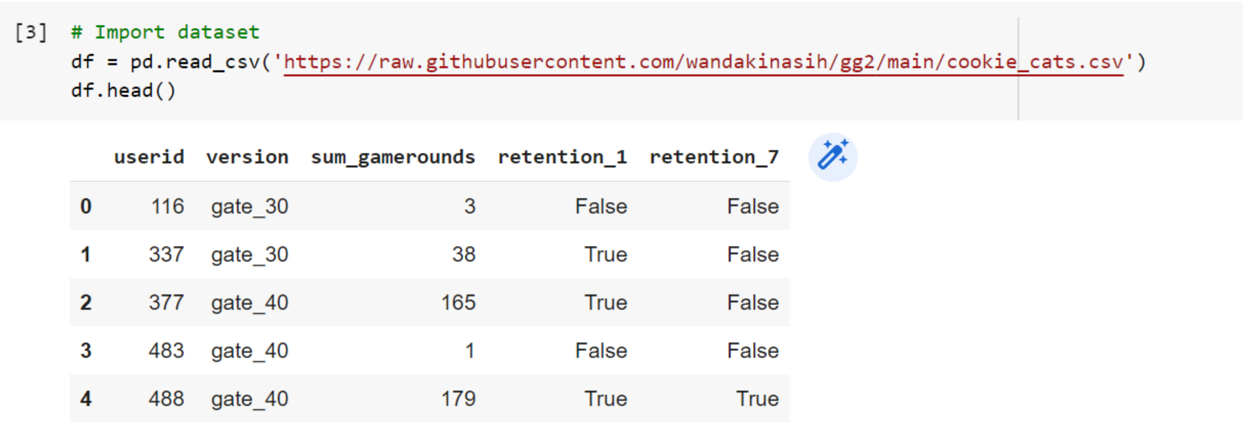
sum_gamerounds — the number of game rounds played by the player during the first week after installation

retention_1 — did the player return, and play 1 day after installation? True or False

retention_7 — did the player return, and play 7 days after installation? True or False

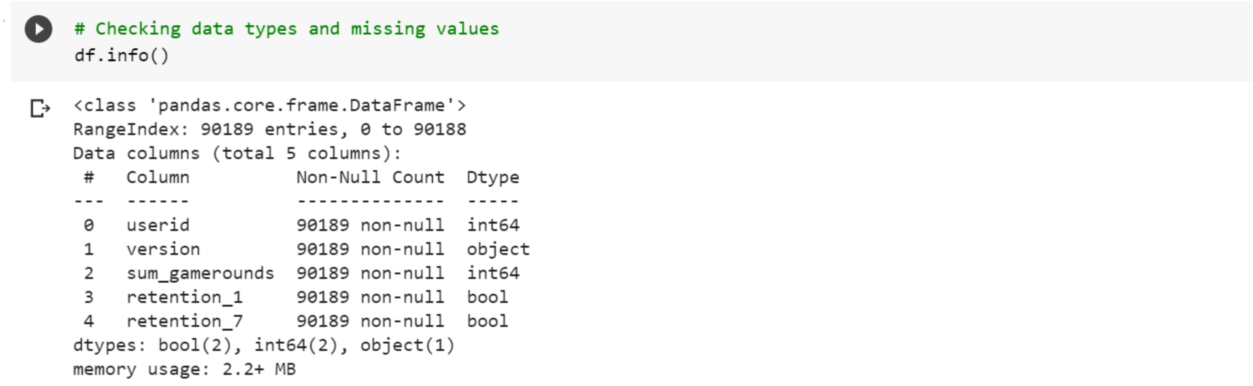
- **Exploratory Data Analysis**

In EDA, we would know is there any missing values in data.



```
[3]  # Import dataset
     df = pd.read_csv('https://raw.githubusercontent.com/wandakinasih/gg2/main/cookie_cats.csv')
     df.head()
```

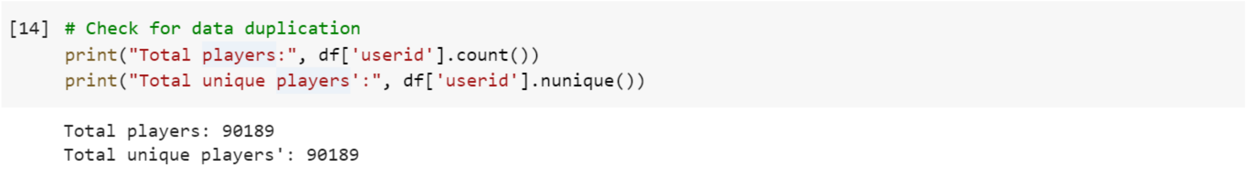
	userid	version	sum_gamerounds	retention_1	retention_7
0	116	gate_30	3	False	False
1	337	gate_30	38	True	False
2	377	gate_40	165	True	False
3	483	gate_40	1	False	False
4	488	gate_40	179	True	True



```
# Checking data types and 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  bool  
 4   retention_7     90189 non-null  bool  
dtypes: bool(2), int64(2), object(1)
memory usage: 2.2+ MB
```

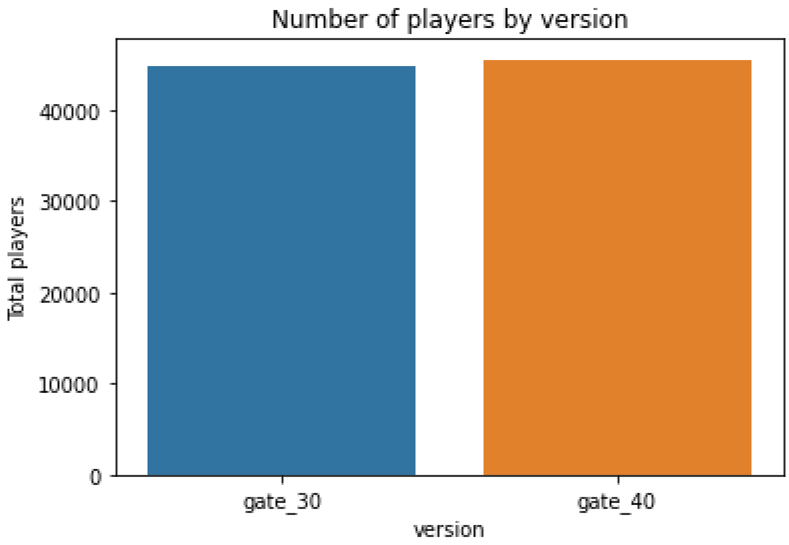
From this table, there are no missing values.



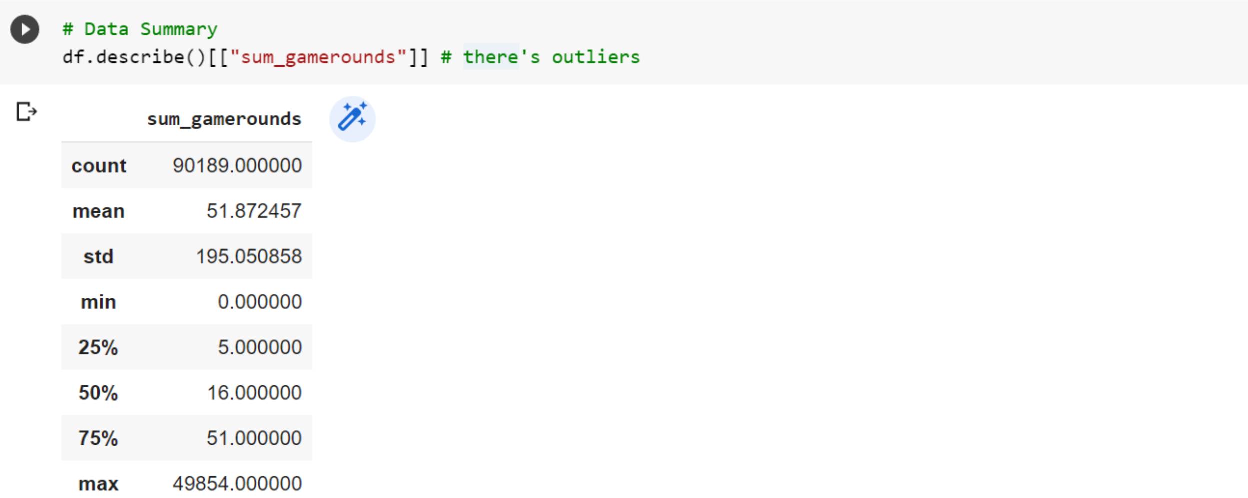
```
[14] # Check for data duplication
     print("Total players:", df['userid'].count())
     print("Total unique players':", df['userid'].nunique())
```

```
Total players: 90189
Total unique players': 90189
```

We also would know is there any duplicated data with nunique() method. Based on the output of the code, there is no duplicated data.

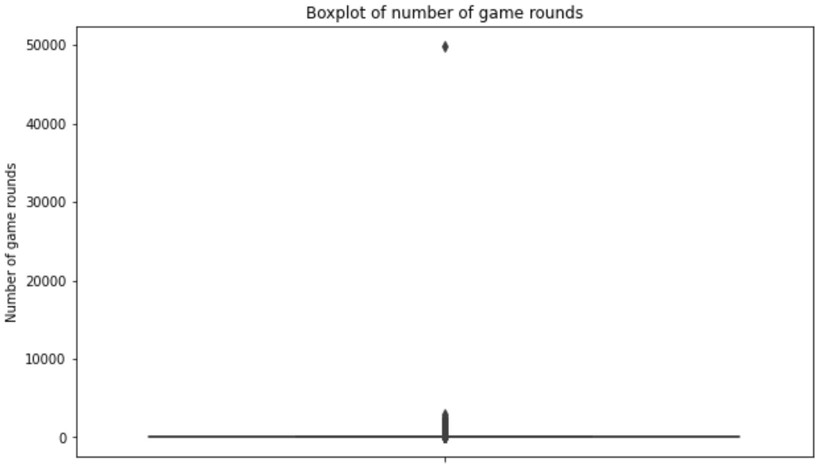


As we can see that the number of players in each version are about the same.

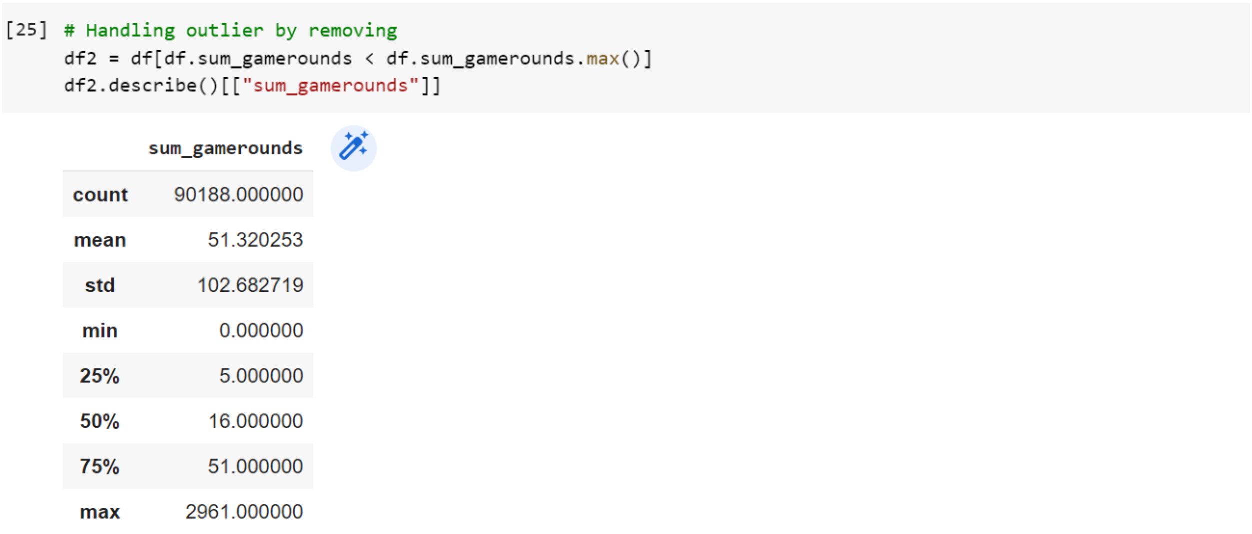


```
# Data Summary
df.describe()[["sum_gamerounds"]] # there's outliers
```

	sum_gamerounds
count	90189.000000
mean	51.872457
std	195.050858
min	0.000000
25%	5.000000
50%	16.000000
75%	51.000000
max	49854.000000



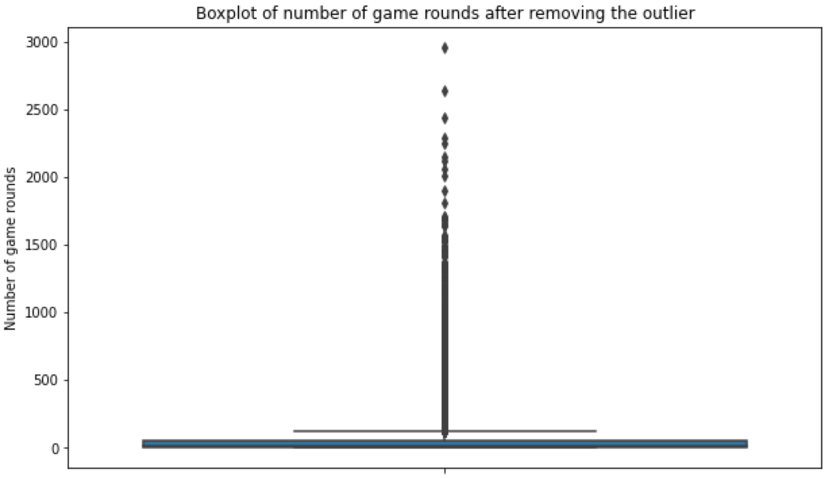
We can see that the maximum values are indicated as outliers. Then, we will handle the outlier by removing it.

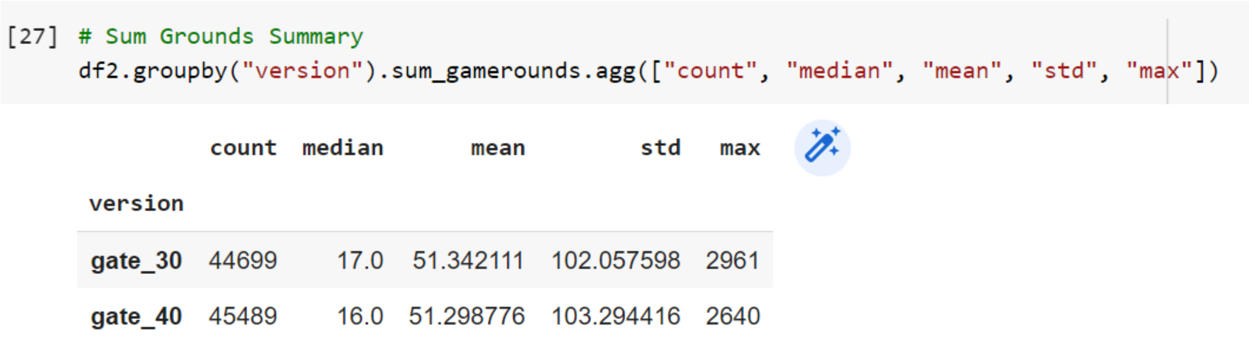


```
[25] # Handling outlier by removing
     df2 = df[df.sum_gamerounds < df.sum_gamerounds.max()]
     df2.describe()[["sum_gamerounds"]]
```

	sum_gamerounds
count	90188.000000
mean	51.320253
std	102.682719
min	0.000000
25%	5.000000
50%	16.000000
75%	51.000000
max	2961.000000

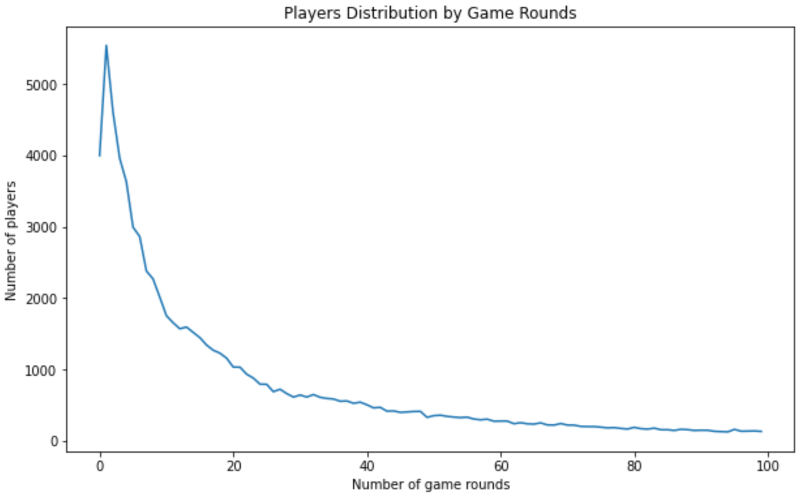
From this table, we can see that the extreme value has been removed and we get this boxplot.



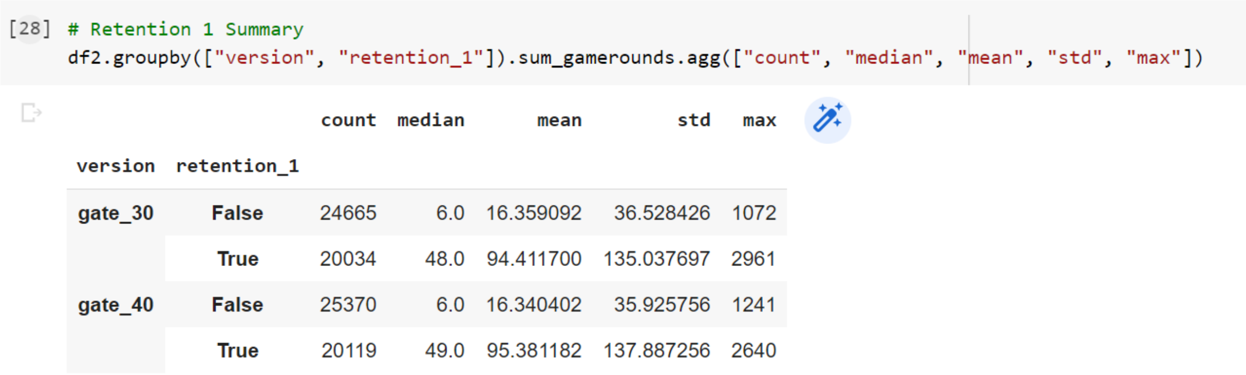


```
[27] # Sum Grounds Summary
     df2.groupby("version").sum_gamerounds.agg(["count", "median", "mean", "std", "max"])
```

	count	median	mean	std	max
version					
gate_30	44699	17.0	51.342111	102.057598	2961
gate_40	45489	16.0	51.298776	103.294416	2640

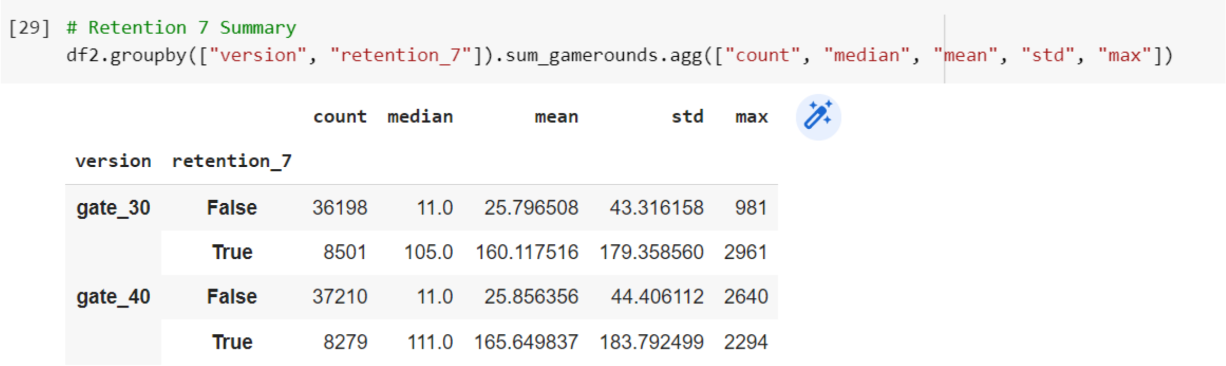


From the distribution chart, we know that many players didn't play the game after install or after a very minimum number of game rounds.



```
[28] # Retention 1 Summary
     df2.groupby(["version", "retention_1"]).sum_gamerounds.agg(["count", "median", "mean", "std", "max"])
```

version	retention_1	count	median	mean	std	max
gate_30	**False**	24665	6.0	16.359092	36.528426	1072
	True	20034	48.0	94.411700	135.037697	2961
gate_40	**False**	25370	6.0	16.340402	35.925756	1241
	True	20119	49.0	95.381182	137.887256	2640



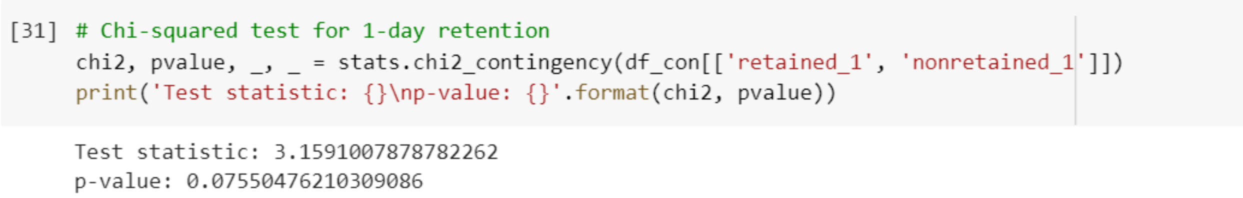
```
[29] # Retention 7 Summary
     df2.groupby(["version", "retention_7"]).sum_gamerounds.agg(["count", "median", "mean", "std", "max"])
```

version	retention_7	count	median	mean	std	max
gate_30	**False**	36198	11.0	25.796508	43.316158	981
	True	8501	105.0	160.117516	179.358560	2961
gate_40	**False**	37210	11.0	25.856356	44.406112	2640
	True	8279	111.0	165.649837	183.792499	2294

We also get the summary in retention_1 and retention_7 data in count.

- A/B Testing

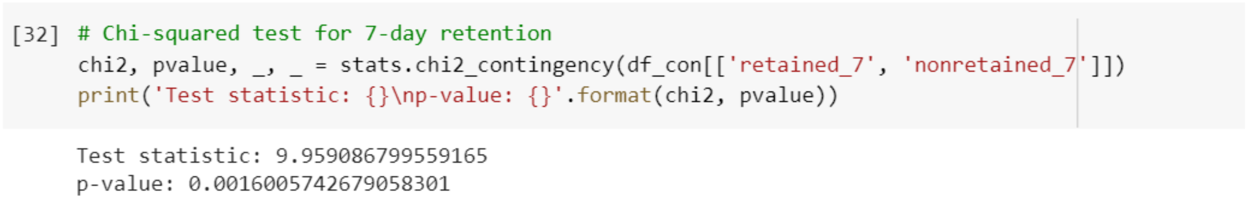
Retention rate test using Chi Square



```
[31] # Chi-squared test for 1-day retention
     chi2, pvalue, _, _ = stats.chi2_contingency(df_con[['retained_1', 'nonretained_1']])
     print('Test statistic: {}\np-value: {}'.format(chi2, pvalue))

     Test statistic: 3.1591007878782262
     p-value: 0.07550476210309086
```

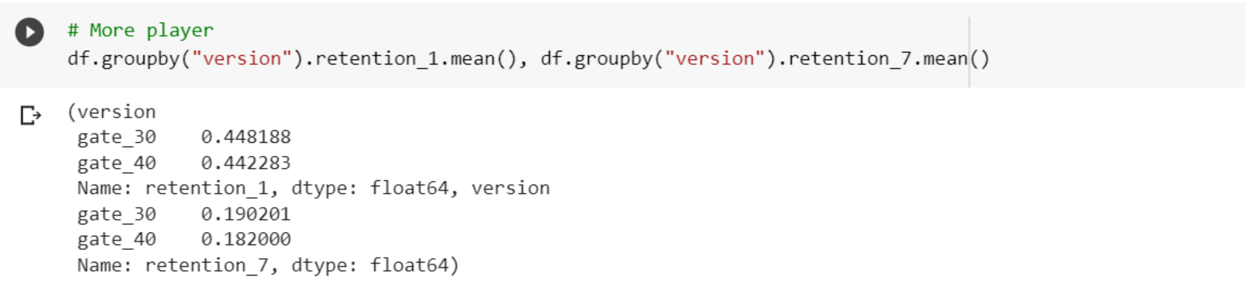
p-value, 0.0755, higher than the 0.05 significance level, it **failed to reject the null hypothesis**, there is no statistical significant relationship in retention_1 between gate_30 (the control group) and gate_40 (the Experiment group)



```
[32] # Chi-squared test for 7-day retention
     chi2, pvalue, _, _ = stats.chi2_contingency(df_con[['retained_7', 'nonretained_7']])
     print('Test statistic: {}\np-value: {}'.format(chi2, pvalue))

     Test statistic: 9.959086799559165
     p-value: 0.0016005742679058301
```

p-value, 0.002, lower than the 0.05 significance level, it **rejects the null hypothesis**, there is a statistical significant relationship in retention_7 between gate_30 (the control group) and gate_40 (the Experiment group).

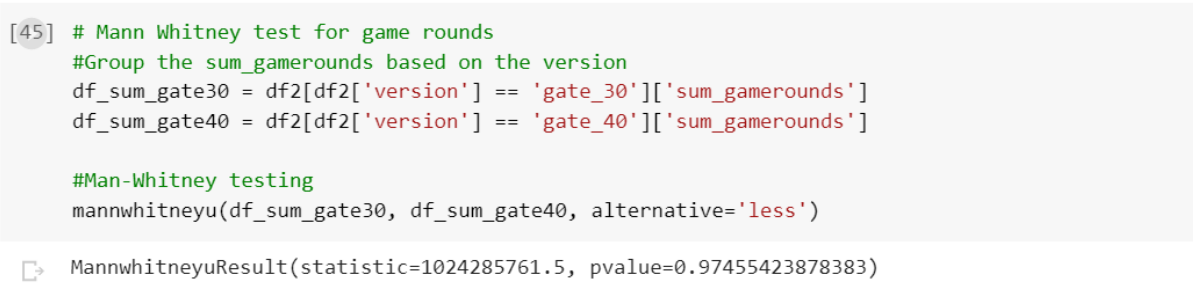


```
# More player
df.groupby("version").retention_1.mean(), df.groupby("version").retention_7.mean()

(version
 gate_30    0.448188
 gate_40    0.442283
 Name: retention_1, dtype: float64, version
 gate_30    0.190201
 gate_40    0.182000
 Name: retention_7, dtype: float64)
```

We can conclude that the 7-day retention rate is different when the gate is at level 30 and between at level 40.

Game Rounds Testing using Mann Whitney



```
[45] # Mann Whitney test for game rounds
     #Group the sum_gamerounds based on the version
     df_sum_gate30 = df2[df2['version'] == 'gate_30']['sum_gamerounds']
     df_sum_gate40 = df2[df2['version'] == 'gate_40']['sum_gamerounds']

     #Man-Whitney testing
     mannwhitneyu(df_sum_gate30, df_sum_gate40, alternative='less')

     MannwhitneyuResult(statistic=1024285761.5, pvalue=0.97455423878383)
```

p-value, 0.974, higher than the 0.05 significance level, it **failed to reject the null hypothesis**, the median sum_gamerounds between gate_30 (the control group) and the gate_40 (the Experiment group) are similar.

Conclusion and Action Plan

From the analysis, we can conclude that there is a difference between control group (players with gate in level 30) and treatment group (players with gate in level 40). It means that there is an impact on player retention and game rounds. We also get that in gate 30, the retention rate is slightly more than the retention rate in group of gate 40. Therefore, we advise that it’s not needed to change the gate level to level 40 to keep the retention rate.

To maintain new players, we can try another method like decreasing the waiting time or make the in-app payment easier.