Gacha System in Mobile Games

Lai On Ping

Dec 11, 2023

1 Introduction

Gacha is a Japanese word for toy vending machine, where entering some money will return some random toys. The mechanic was introduced to mobile games to obtain in-game items since early 2010s in Japan. We call each draw a "roll". The probability of obtaining your target item simply follows a geometric distribution. However, even there was no scheme and each roll is completely random, there is a very low chance that some players were unable to get the item even they have spent thousands dollars. Gacha games were often criticized as gambling. Therefore, most of the newer Gacha games will have a mechanism called "pity roll" which will guarantee players to obtain target item after certain amount of rolls. Moreover, government regulations require the game companies to state clear the probabilities for each item in the pool.

This project is to investigate/examine one of the Gacha game, Genshin Impact. Here are some basic information provided by the game:

1. Players are guaranteed to obtain the highest rank items at their 90th rolls.
2. The base probability to obtain the highest rank items is 0.006 for each roll.
3. The consolidated probability to obtain the highest rank items is 0.016 (include guarantee).

From 1 and 2, let $Y \sim$ geom(p), and the random variable, N , the number of rolls to obtain the the highest rank items should follow:

$$N = \begin{cases} Y \sim geom(p) & i < 90 \\ 1 & i = 90 \end{cases} \tag{1}$$

However, this is very different with the real data from a fan-made website that the majority of N falls between 74-80 and very very unlikely get 90. To explain this phenomon, a github user, OneBST, performed a data analysis and proposed a "linear" model [1]:

$$p_i = \begin{cases} 0.006 & i \leq 73 \\ 0.006 + a * (i - 73) & 74 \leq i \leq 89 \\ 1 & i = 90 \end{cases}$$

Where a is a constant coefficient. My project is to simulate this model to estimate a and the true probability of obtaining my target item.

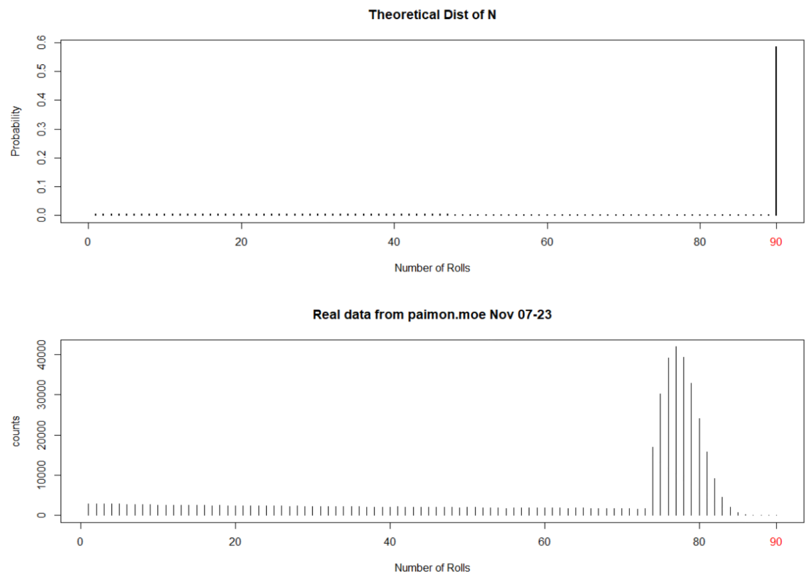

Figure 1: Theoretical Distribution based on rule 1 and 2 vs player data[2]

2 Simulation Implementation

Variables	**Definitions**
r	number of rolls
k	the kth win (Dummy variable)
$N = (n_1, n_2, \dots)$	number rolls for each "win", $n_k = 1, \dots, 90$
p	probability of success
a	coefficient for $74 \leq N \leq 89$

Table 1: Variables

Algorithm 1 Simulation Algorithm

```
nsim ← 100000                                   ▷ Number of wins to simulate
r ← 1                                           ▷ First roll
k ← 0                                           ▷ Number of wins
Initialize N as a numeric vector
a ← 0.06                                        ▷ Proposed coefficient
for i in 1 to nsim do
    while r < 90 do
        p ← if (r < 74) then p = 0.006 else p = 0.006 + a · (r − 73)
        X ← rbinom(n = 1, size = 1, prob = p)
        if X = 0 then
            r ← r + 1                           ▷ if lose keep rolling
        else
            N[k + 1] ← r                        ▷ record the rolls in vector N
            k ← k + 1                           ▷ Number of wins + 1
            r ← 1                               ▷ reset r after winning
            break
        end if
    end while
end for
```

3 Result

3.1 Compare with proposed probability

a	Consolidated probability $= nsim/\sum n_k$	mean $\bar{N}$
0.05	0.01596372	62.64205
0.052	**0.01598353**	**62.56441**
0.053	**0.01601271**	**62.45041**
0.054	0.01601497	62.44157
0.056	0.01602452	62.40437
0.06	0.01604035	62.34278

Table 2: Consolidated probability and mean N for different value of a

If the proposed probability 0.016 is true, then the true a is between 0.052 and 0.053.

a	Consolidated probability $= nsim/\sum n_k$	mean $\bar{N}$
0.0523	0.01601081	62.45782
0.0525	0.01600209	62.49184
0.0527	0.01601774	62.43079

Table 3: Consolidated probability and mean N for different value of a (Cont.)

3.2 Optimization

Use the algorithm 1 to create a function "roll" such that the output is the consolidated probability . Applying the optimize() function 100 times would obtain $\bar{a} = 0.05452396$ with standard error 0.001965305 and the mean consolidated probability is 0.01601984. Therefore, the approximated 95% confidence interval for a is between 0.05067196 and 0.05837596.

3.3 Compare with player data

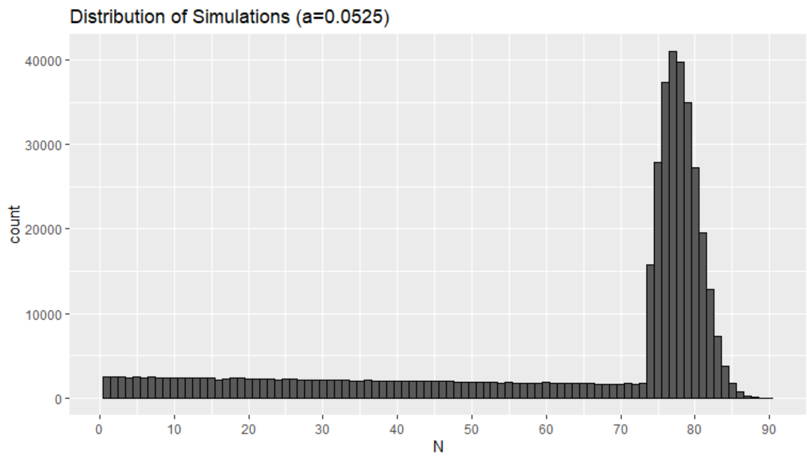

Figure 2: Simulation results (a=0.0525)

The distribution is very similar to the figure 1.2. In order to compare the distribution, apply the two-sample Kolmogorov-Smirnov (KS) test with different simulation results using seed 789687:

a	0.052	0.0525	0.053	0.056	0.058	0.06
p-value	0.1148	0.115	0.116	0.1378	0.09631	0.09585

Table 4: Consolidated probability and mean N for different value of a (Cont.)

None of the simulations have enough evidence that they are different with the real data.

4 Discussion

Section 3.1 suggested that a should be between 0.52 and 0.053 such that the consolidated probability (proportion of number of wins to total number of rolls) aligned to the proposed probability 0.016 by the game company; while section 3.2 suggested $a = 0.0545$ However, section 3.3 suggests that $a = 0.056$ will be closer to the player data, where the true probability maybe slightly higher than the proposed probability. This is possible since the game company may want to avoid any risk of lawsuits.

However, even the proposed probability is true, there are other factors that caused the difference between sections. First, the data was obtained by players' volunteer submissions. Players are more willing to share their data when won via low number of rolls, which may cause inflation to number of counts for low Ns. Besides, some submissions are from the game servers and some are from manual inputs. The proportions are unknown. Measurement errors are not under controlled. Second, the two-sample KS test where designed for continuous data, but the distribution is discrete. This may reduce the validity of the result.

Overall, there are no evidence to suggest the game company has a lower probability than they had proposed. An interesting founding is that consolidated probability and mean are the same with geometric(p=0.006), but the distribution was adjusted such that players were guaranteed to obtain what they purchased.

5 Appendix

The code is in another submission file.

References

[1] OneBST. GI GACHA DATASET. https://github.com/OneBST/GI_gacha_dataset, 2022.

[2] Paimon.moe. Global Wish Stats. https://paimon.moe/wish/tally?id=300056, 2023.