Algorithm Schematic Behind WeChat Red Envelope

Shang Zhang School of Information Science and Technology ShanghaiTech University

***Abstract*—Snatching red envelope is a good way to entertainment, and is very popular in people's daily life. What the algorithm Schematic Behind WeChat Red Envelope? According to my record, I have several suppositions.**

I. Introduction

Firstly, I need confirm the variables that associated with the amount of money. It could be the order, WeChat account, or just randomly distributed to get sum equal to the total money. How it works? If there's something with the order that people's snatch, should earlier snatching would get more money? or later one. Not only should we analyze data carefully, but speculate before data analysis would be more helpful to modeling.

.Speculation

I put 1 yuan into each red envelope, with 10 people to distribute it. And I repeated 10 times. What got my attention is that the last person seems would get more money than the other person. With Fig1 presents it, where the red podetium means the price that the first one get, and the orange podetium means the price that the last person get. From the Fig1, it appears that orange podetium have more proportion. And it is necessary to calculate the average and the median.

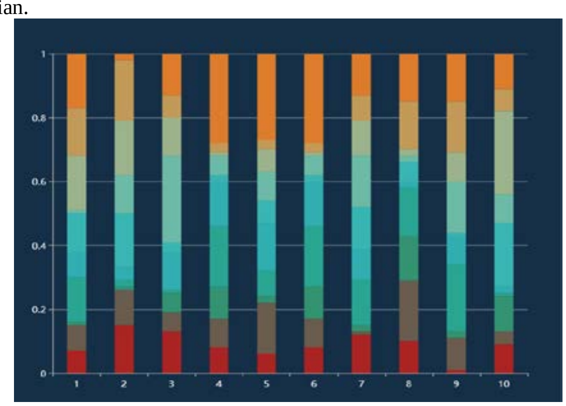

Fig1.10 times of hand out with each podetium

With different colors to present the order, from the bottom to top. Although it gives some information about how the order affect the distribution, it also seems inconspicuous to observe it.

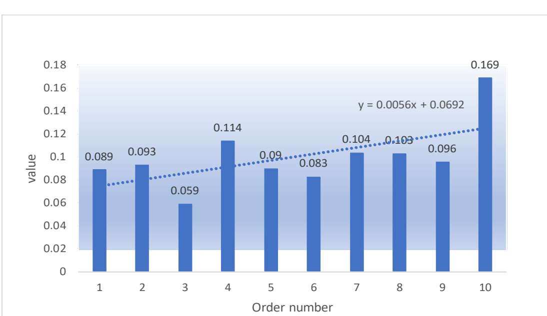

Fig2.Average value of each order

The last person has the highest average value price. It should be reasonable to consider that the value in the last hand out may works differently.

If each hand out is distributed as *Uniform* in some interval, then it can't be *Unif(0,1)*, since there is no more than some value that greater than 0.5, even the highest value is under 0.3 according to my record. So, in the view of randomness, I may consider each hand out is under some proportion of the surplus, or in the interval around the mean, maybe both and take the lesser one. But, there is unusual phenomenon, that is the 0.01 value is almost appear in every hand out, which isn't fellow the 'random'.

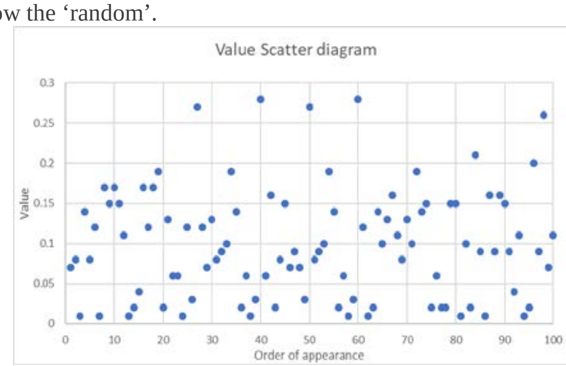

Fig3. '0.01' appears exactly ten times in my data

That inspires me maybe the value would get the minus region, and then take the least value 0.01. Also inspires me that the proportion interval probably covers the minus region around the average of the surplus.

II. Model and Algorithms

It's good start to use Matlab to model the data. Normal function is cognitive for me to model.

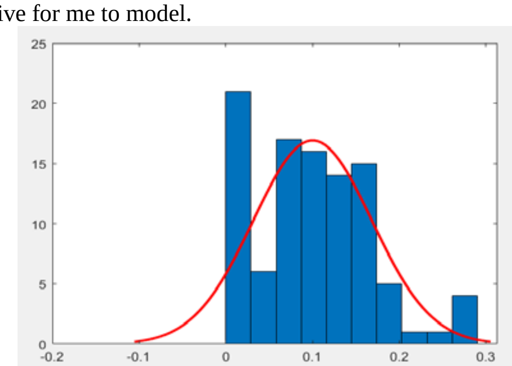

Fig4. Normal curve fitting

A. Model

From the Fig4, we can see that the confidence interval of Normal function indeed cover the minus region, which accord with previous suppositions. We assume that the value of each hand out is correlate with the surplus of the red envelope. And distributed randomly in the interval around the mean. Since it could reach the minus region, the stretch must greater than 1/10 of the surplus. (Like the test I made: the first hand out have surplus of total value of the envelope, with mean 0.1 of each hand out, the interval is around 0.1).

According the data, Since the 0.01 appears 10 times, with probability 1/10, the highest region the value got, and by symmetric (mean is central of the interval), the formula of calculate the interval should be *(mean-5/4mean, mean+5/4mean).*

B. Algorithms

The Algorithms based on the model that previous made. Using python to present the algorithms as follow:

```
import random
def last_hand_out(bag,surplus):
    mean  = surplus/2
    value  = random.uniform(-1/4*mean,2*mean)
    if value < 0:
        value = 0.01
    bag.append(value)
    bag.append(surplus-value)

def hand_out(bag,surplus,n):
    mean  = surplus/(n-len(bag))
    value  = random.uniform(-1/4*mean,9/4*mean)
    if value < 0:
        value = 0.01
    bag.append(value)

def weixin_red_envelope(money,n):
    bag    = []
    surplus= money
    while len(bag) < n-2:
        hand_out(bag,surplus,n)
        surplus=surplus-bag[len(bag)-1]
    last_hand_out(bag,surplus)
    print(bag)
```

Fig5.python model

Simple python program which execute the algorithms, here I used random module to create the value that lie in the interval. Notice that I believe that the distribution in the interval is *Uniform*, which means that the probability of values that around the average are same. And here I used different function to model the last hand out, the interval cannot exceed 1, the last hand out would give all the rest money, so it should consider differently.

III. Simulation Results

Simulation works well compared with previous data. The three groups of data are presents in graph:

Graph

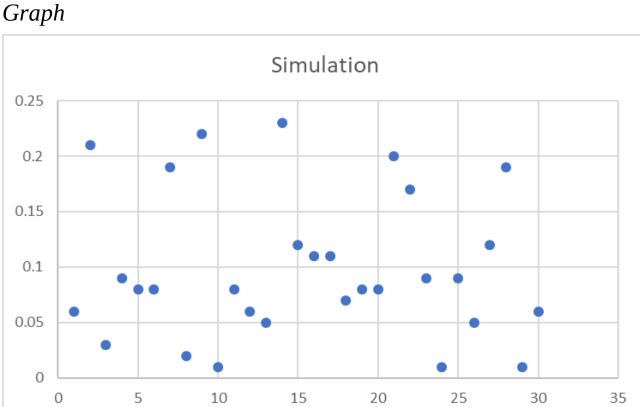

Fig6.Simulation scatter diagram

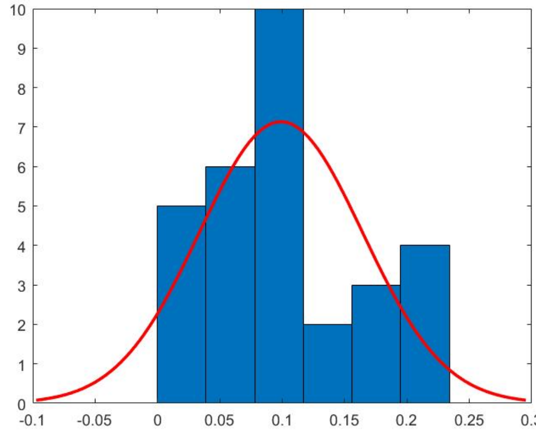

Fig7.Normal curve fitting

As we can see, the 0.01 appears three times as expected. And from the algorithms, the last hand out function is distinguish with the hand out function, which provides some interpretation of the problem that discussed earlier.

IV. Conclusion

From the view of previous point, I recognized that the value of each person get from the red envelope is fair on the whole picture. For the first people, is totally fair. And from the experiment and simulation results to see, there indeed have some fluctuate with the order, since the algorithm I used is based on the residue theory, the last hand out is different with other hand out, because the confidence interval of Normal function would cover the minus region, thus we consider the last two hand out independently, the goal is to complete the total value. Maybe the group of WeChat just want some odd mechanism to get more entertainment, which we have to consider the minus region. With high risk of losing money in the last snatching, and more repaid. Some data from the internet also proved that, and I won't discuss them.

V. Acknowledgment

During this project, I read part contents from Zhihu

References

[1] Joseph K. Blitzstein, Jessica Hwang-Introduction to Probability (Chapman & HallCRC Texts in Statistical Science)-Chapman and HallCRC (2014)