Reverse Engineering Algorithmic Mechanism Behind WeChat Red Envelope

Qifan Zhang

47422183
zhangqf@shanghaitech.edu.cn
School of Information Science and Technology
ShanghaiTech University

January 19, 2018

Outline

Outline

Introduction

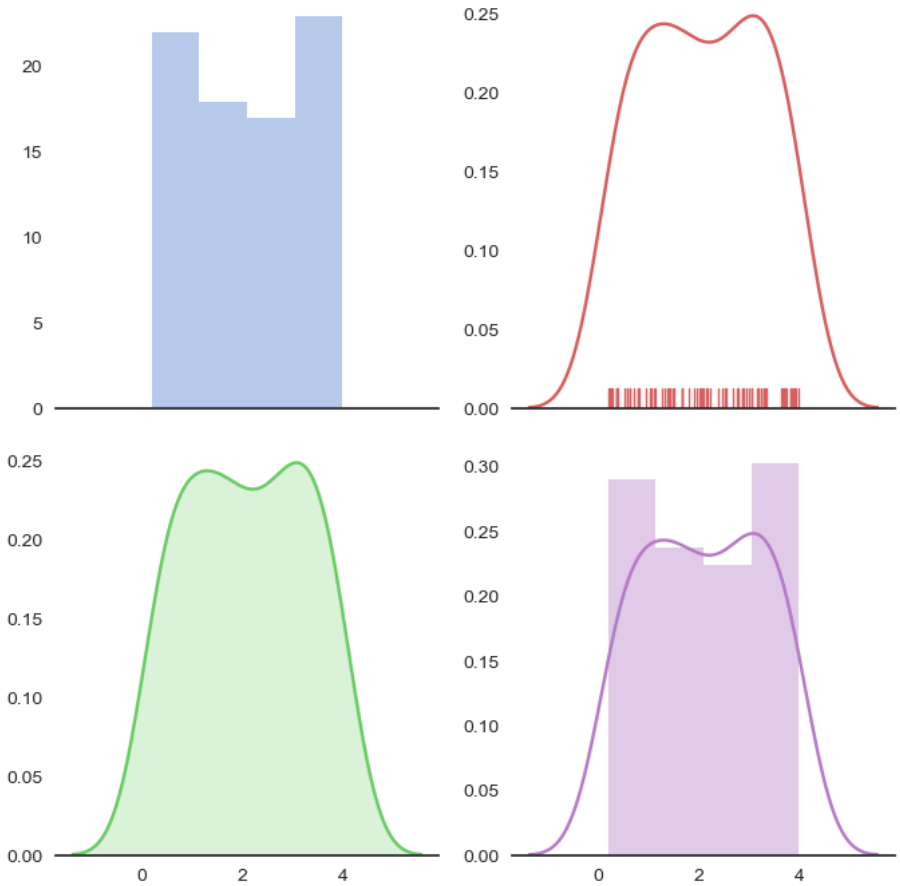
- 10 *yuan* for 5 people each time, 80 trials as prior distribution
- guess and build model over prior distribution
- test my model by 100 posterior trials
- All cell phones used in our experiments are conducted on *WeChat* 6.6.1 on *iOS*.

Outline

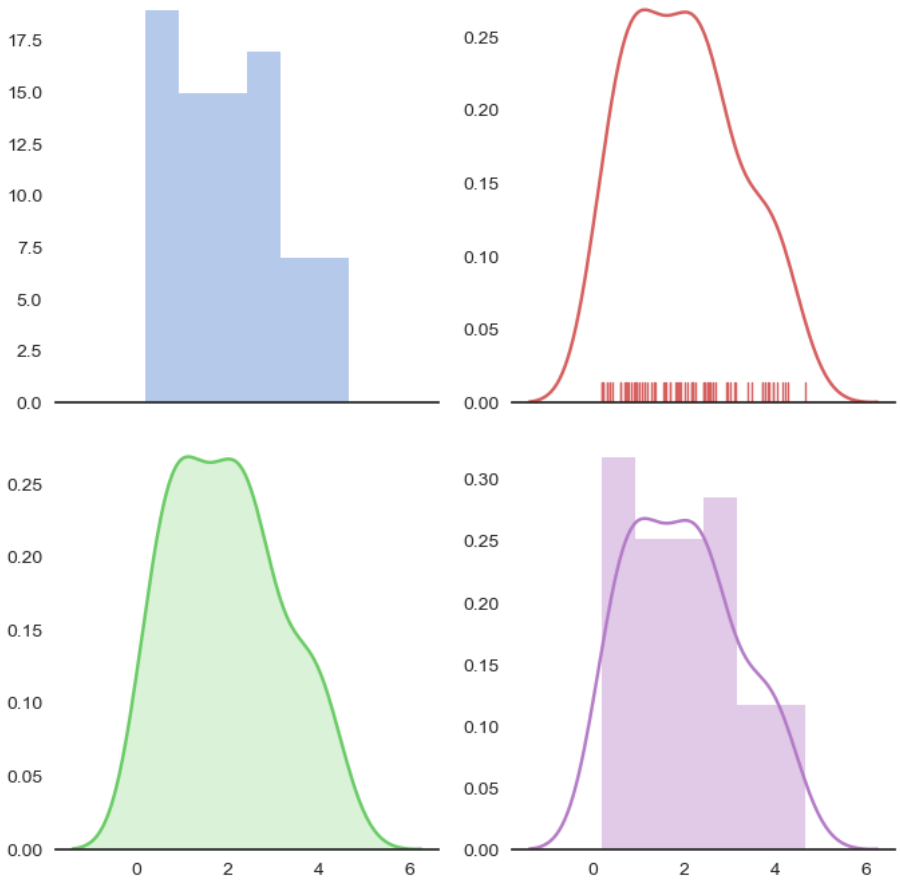
Prior Distribution

- the first participant **will never get more than $\frac{2}{5}$ of the total money (i.e. 4 yuan)**
- **define money gotten by the jth participant is X_j**

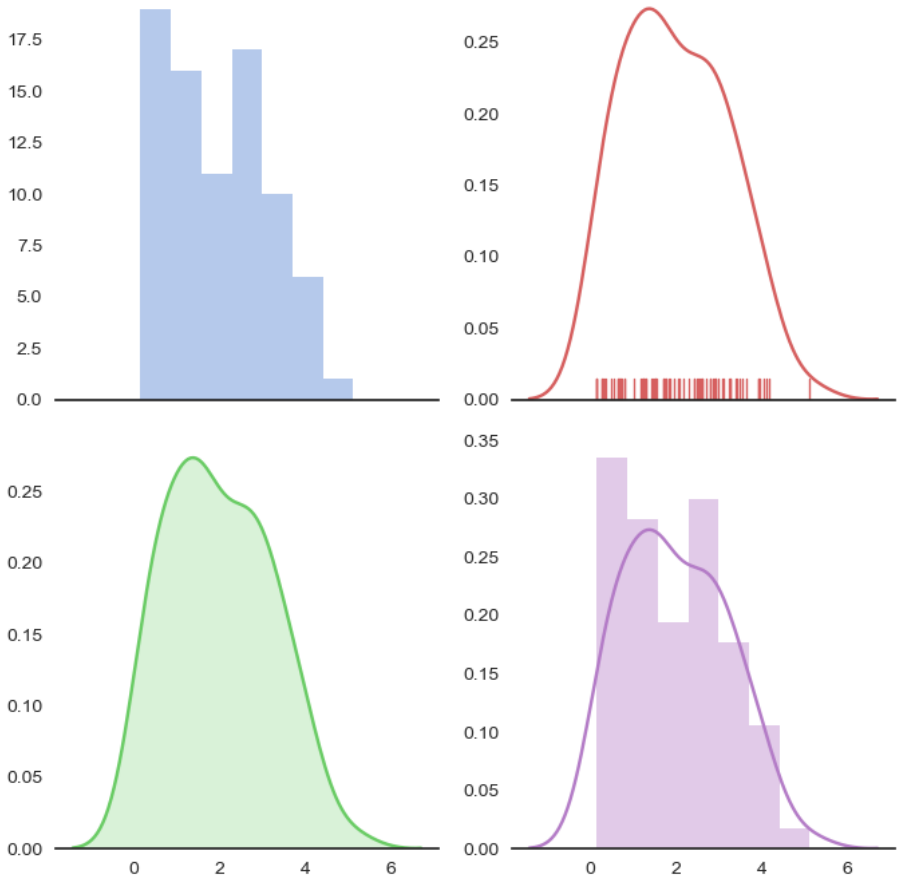
Prior Distribution of X_1



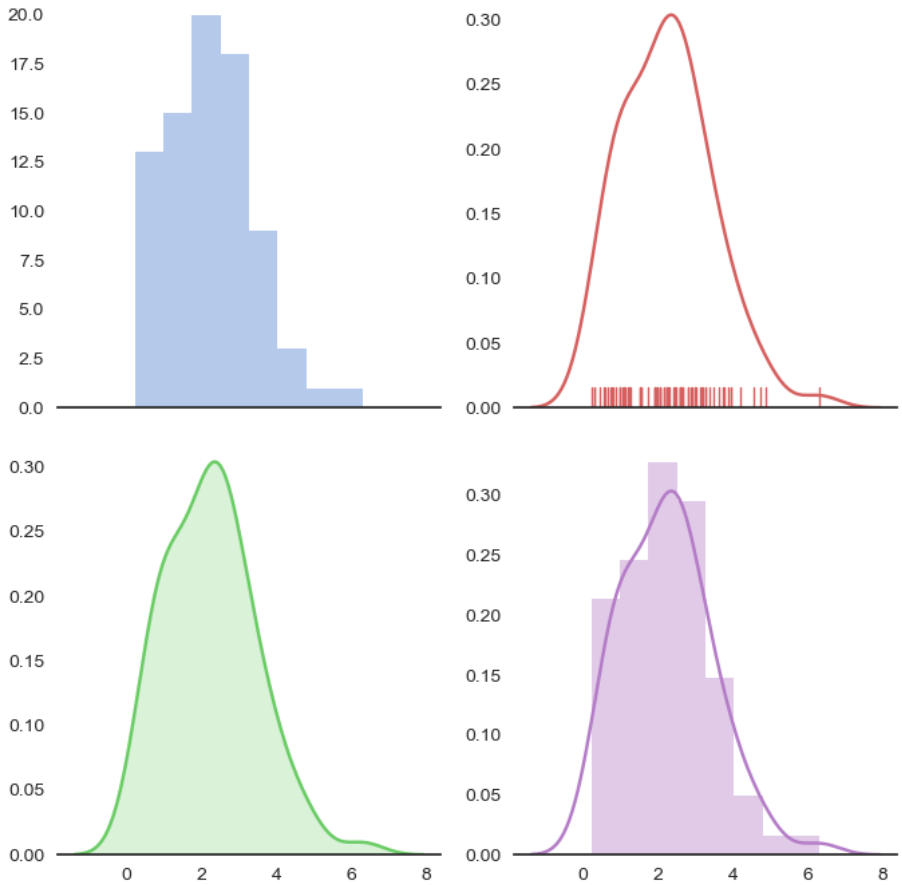
Prior Distribution of X_2



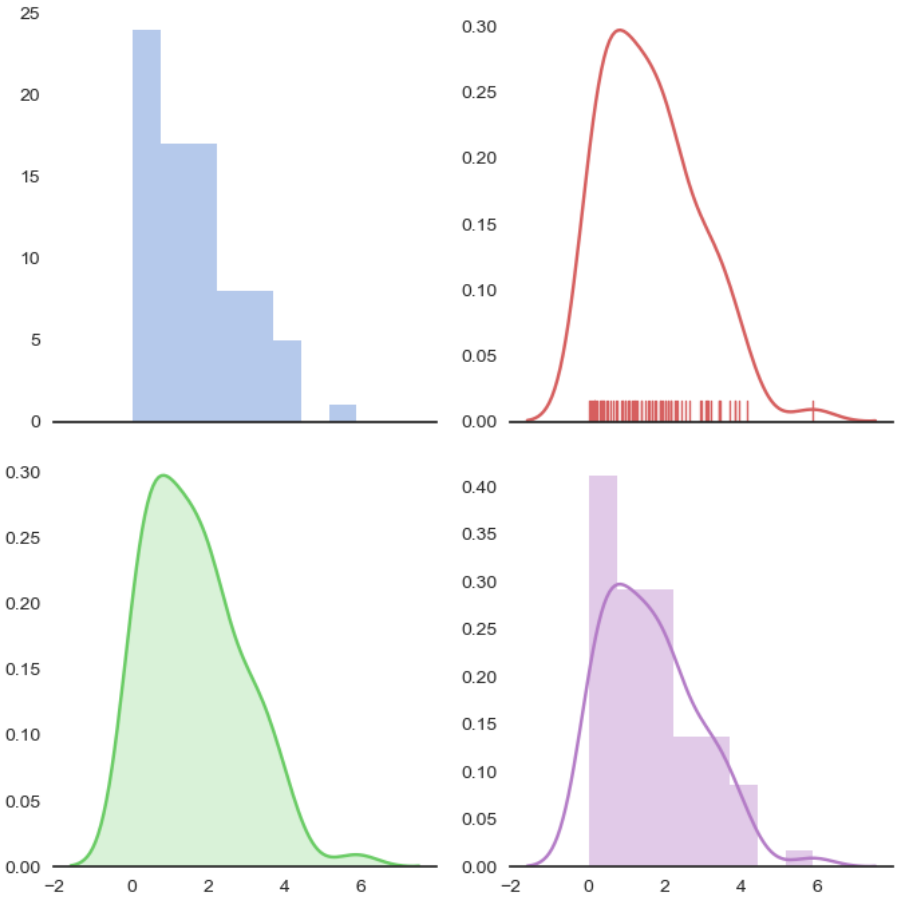
Prior Distribution of X_3



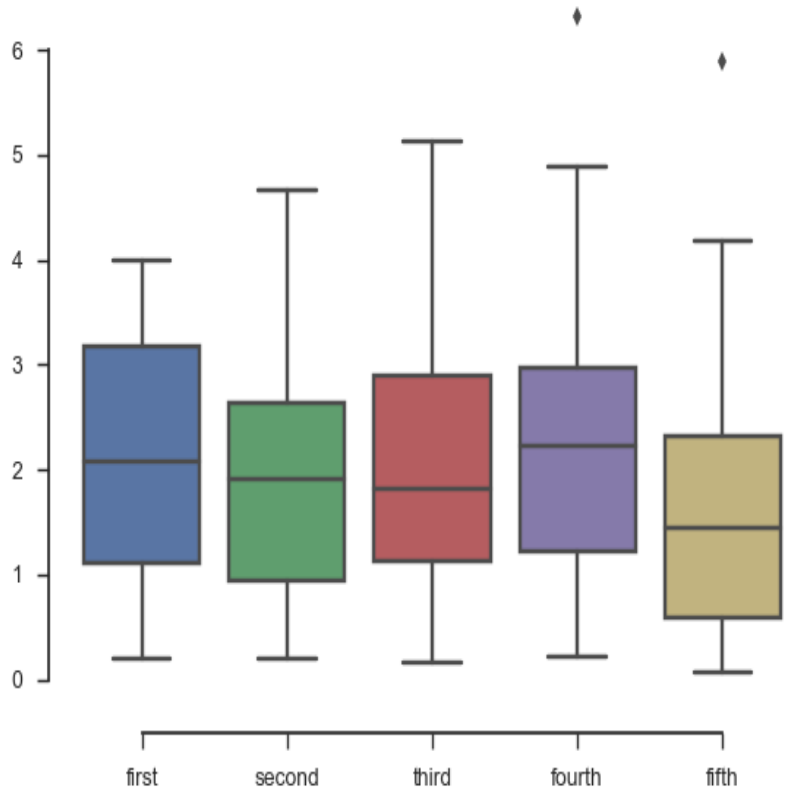
Prior Distribution of X_4



Prior Distribution of X_5



Overall Prior Distribution



Intuitive Results from Prior Distribution

- no red envelop for the *first* participant is larger than *yuan*
- for the *first* participant, the amount of money distributes nearly uniformly from 0 to 4
- I also found that in a 100-*yuan* red envelop, the first participant could get at most 40 *yuan* and in a 1-*yuan* red envelop, the first participant could get at most 0.4 *yuan*.

Distribution for X_1

From *Intuitive Results from Prior Distribution*, I guess that in a 10-*yuan* red envelope for 5 participants:

$$X_1 \sim Unif(0, 4)$$

Distribution for $X_2, X_3, ..., X_5$

Inspired from X_1:

- do $X_2, X_3, ..., X_5$ follow *Uniform Distribution*?
- if so, what is distribution domain for each of them?

Distribution for $X_2, X_3, ..., X_5$

After several times of guessing, trying, verifying and correcting, I found a perfect model:

$$X_j | X_1, X_2, ..., X_{j-1} \sim Unif(0, \frac{2(n - \sum_{i=1}^{4} X_i)}{5 - j}) \quad 2 \leq j \leq 4$$

$$X_5 = n - \sum_{i=1}^{4} X_i$$

Generalized Distribution for $X_1, X_2, ..., X_n$

Then I generalize the model fora *n-yuan* red envelop for m participators:

$$X_1 \sim Unif(0, \frac{2n}{m})$$

$$X_j | X_1, X_2, ..., X_{j-1} \sim Unif(0, \frac{2(n - \sum_{i=1}^{j-1} X_i)}{m + 1 - j})$$

$$2 \leq j \leq m - 1$$

$$X_m = n - \sum_{i=1}^{m} X_i$$

Outline

Stimulation Code

I use *Python* programming language to stimulate the process of giving out red envelopes based on my model and algorithm.

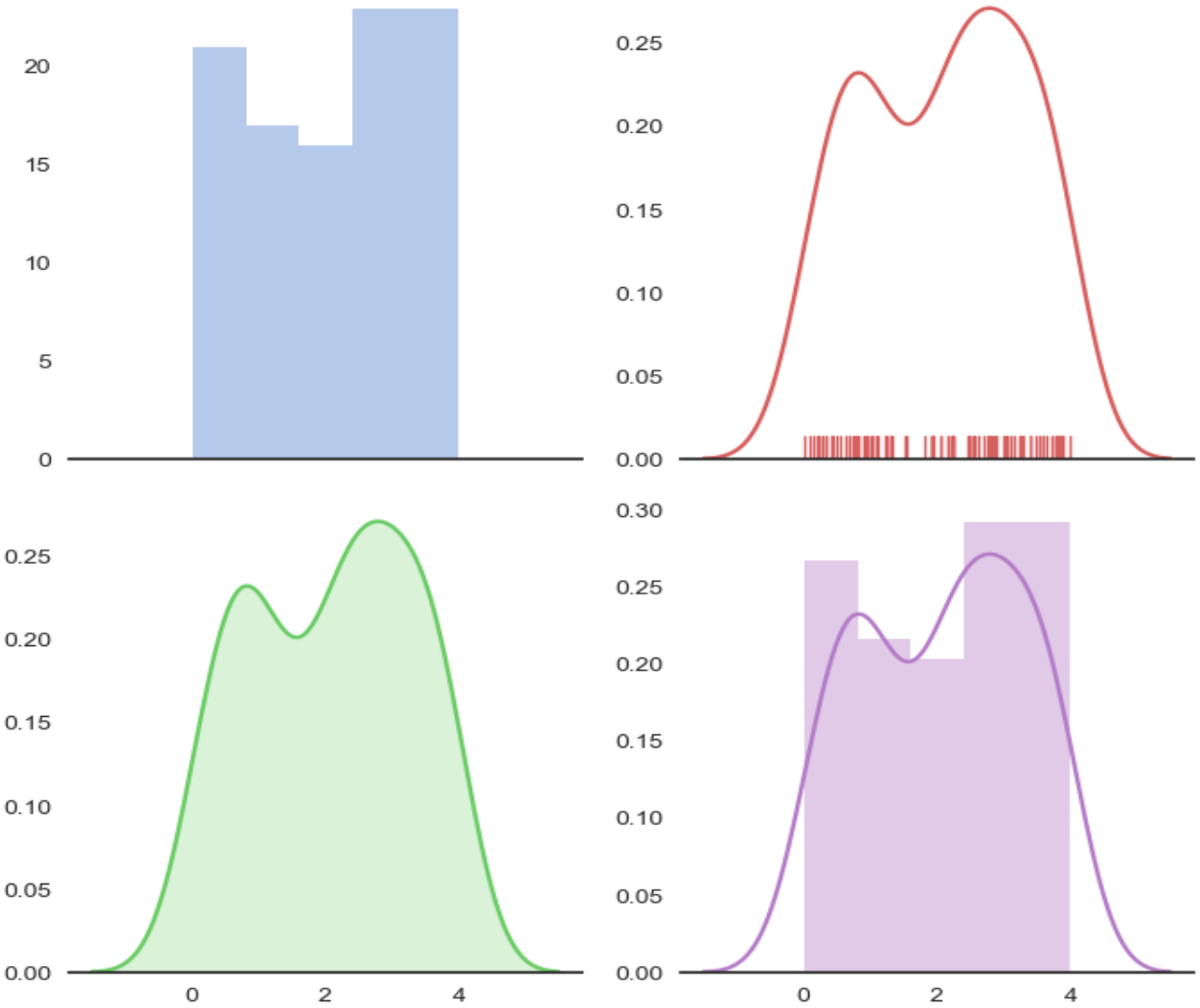
Stimulation Code

```
import random
import sys
f = open("output.csv","w")
def Rp(value,num):
  value_left = value
  result = []
  for i in range(num-1):
    now = random.uniform(0, (2/(num-i))*value_left)
    now = float('%.2f' %now)
    result.append(now)
    value_left = value_left - now
    value_left = float('%.2f' %value_left)
  result.append(float('%.2f' %value_left))
  result_str = ''
  for i in result:
    result_str = result_str + '%.2f,'%i
  result_str = result_str.rstrip(',')
  print(result_str, file = f)
```

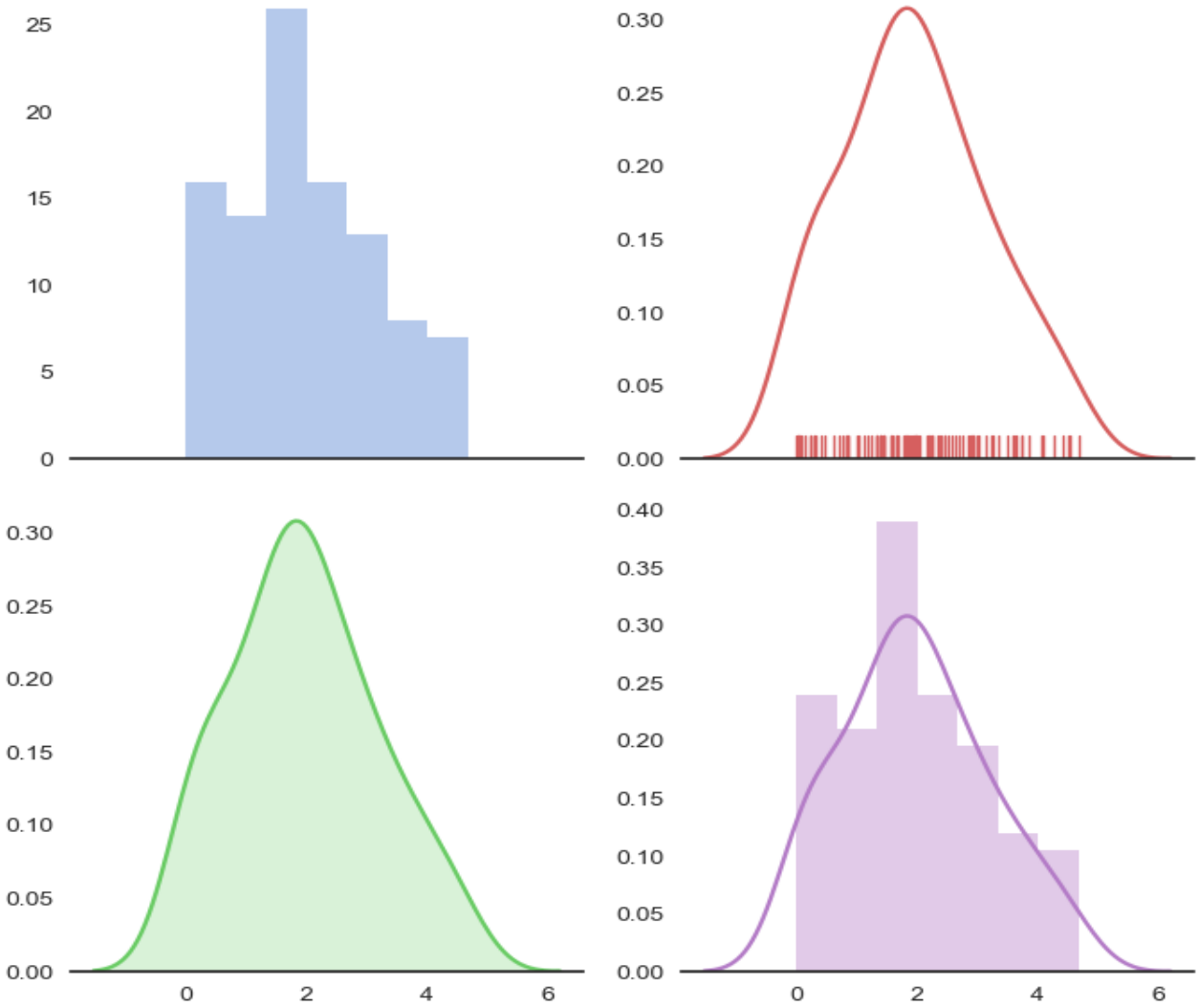
Stimulation Results

I generate 100 cases for a 10-*yuan* red envelop for 5 participators. Results are below:

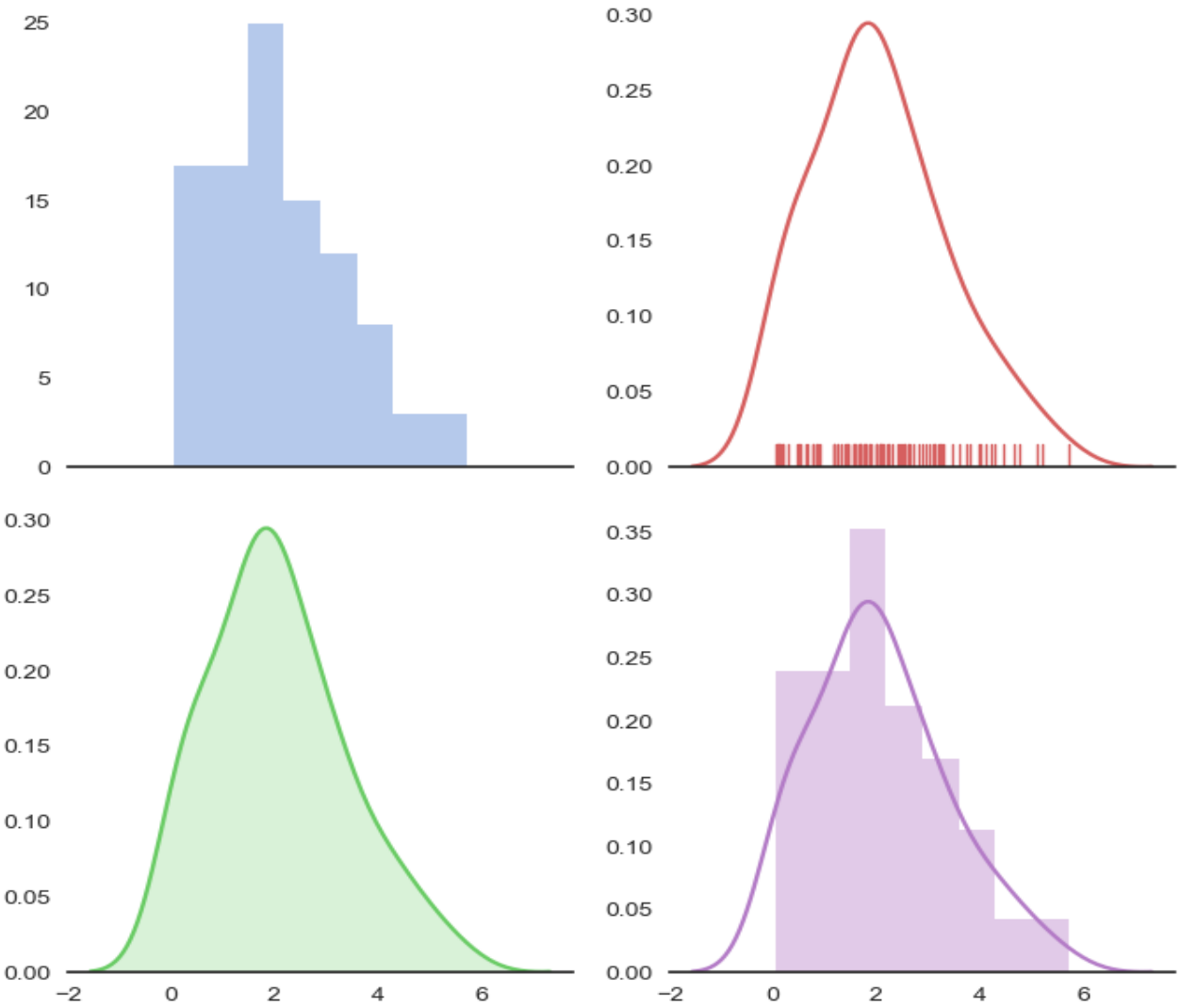
Stimulation Result for X_1



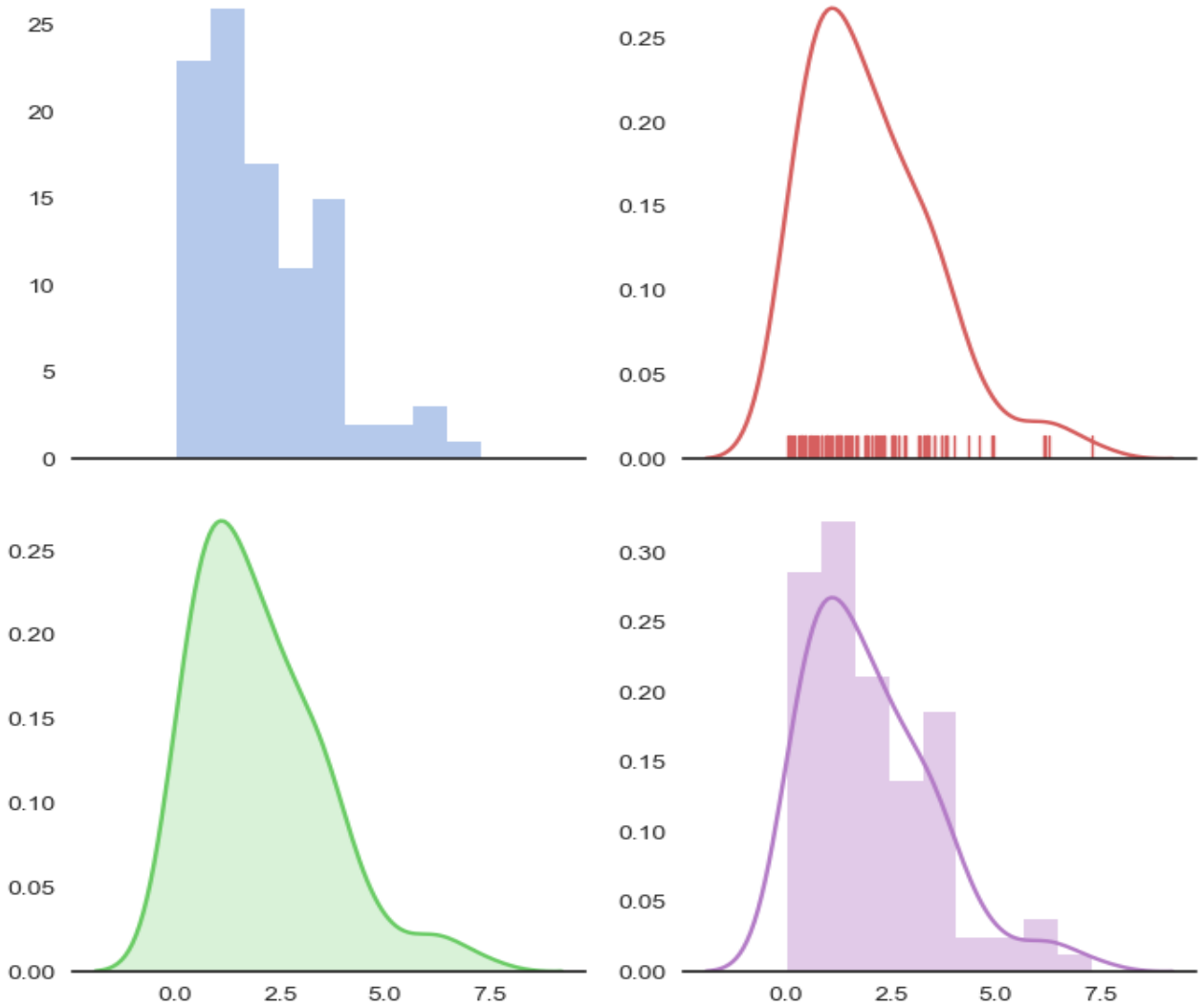
Stimulation Result for X_2



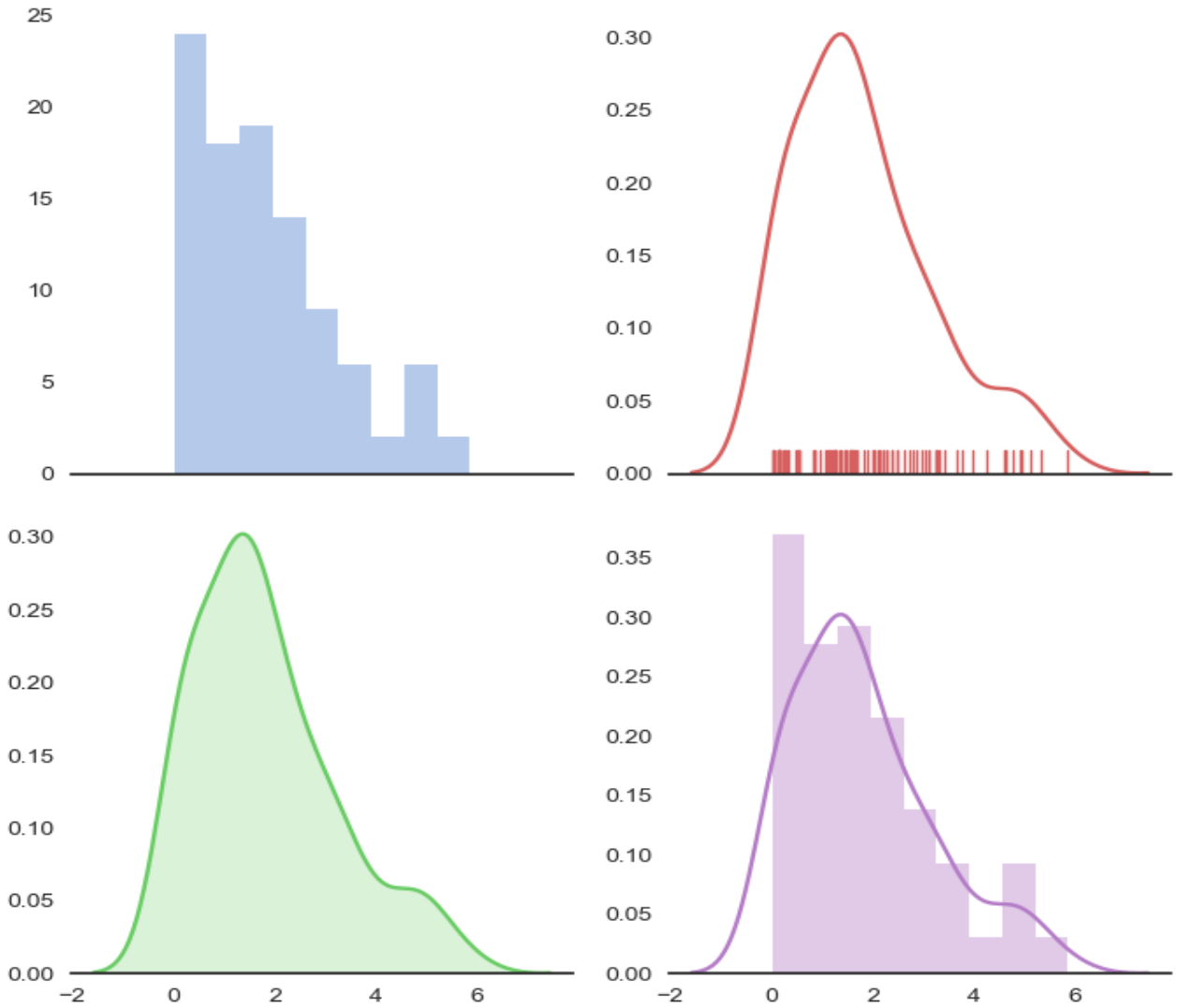
Stimulation Result for X_3



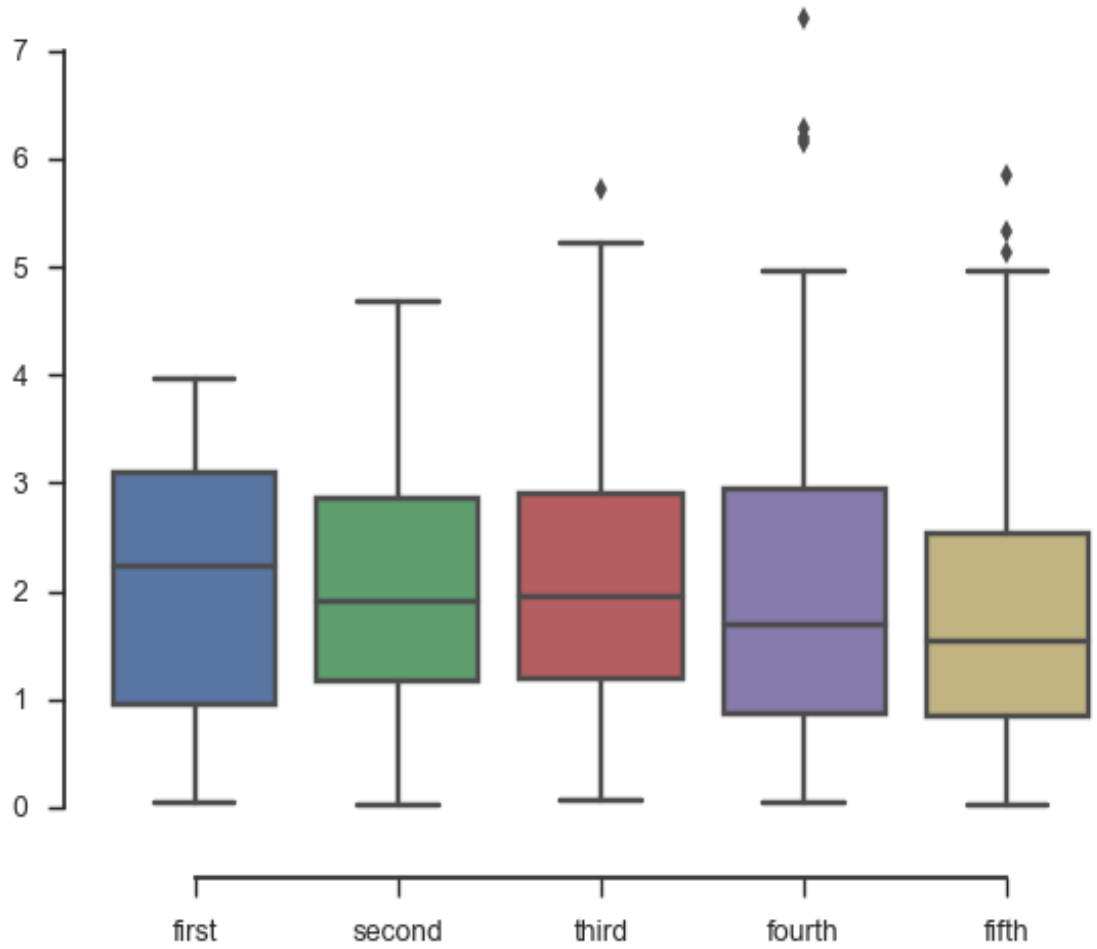
Stimulation Result for X_4



Stimulation Result for X_5



Overall Stimulation Distribution



Stimulation Conclusion

Intuitively, my model satisfies real trials very well.
In detail:

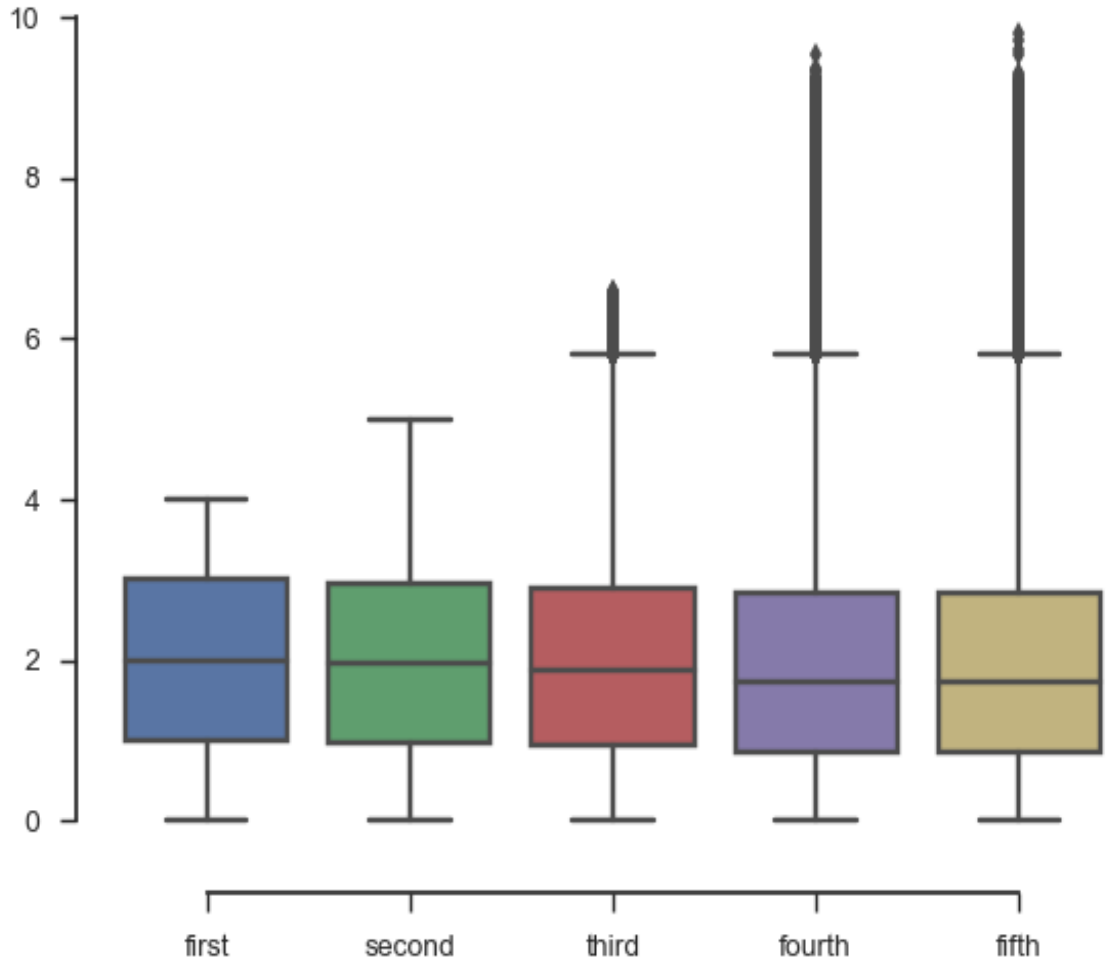
- every participant has a similarly-same mean of money.
- money of latter participants' red envelopes has **larger variance** than former participants'.

Enhanced Stimulation

To prove my intuitive insights:

- **Assuming that my model is correct**, we could only do a large number of trials to gain a nearly perfect mean and variance.
- I do 1 million times of trials and get the result.

Enhanced Stimulation Result



Enhanced Stimulation Conclusion

- Mean of the amount of money is **really the same** (≈ 2)
- Variance is getting larger with rank of the participant getting larger.

The second proved insight explains why we feel the distribution of the *third, fourth and fifth* participants' amount of money *not perfectly* accord with our stimulation results.

Outline

Conclusion- *from Academic Side*

Define X_j as the money the jth participant gets for a *n-yuan* red envelop for m participators.
The distribution is:

$$X_1 \sim Unif(0, \frac{2n}{m})$$

$$X_j | X_1, X_2, ..., X_{j-1} \sim Unif(0, \frac{2(n - \sum_{i=1}^{j-1} X_i)}{m + 1 - j}$$

$$2 \leq j \leq m - 1$$

$$X_m = n - \sum_{i=1}^{m} X_i$$

Conclusion- *from Academic Side*

- Mean of the amount of money is **really the same**.
- Variance is getting larger with rank of the participant getting larger.

Conclusion- *from Intuition Side*

- The faster you participate in the *Red Envelope* game, the more stable the expected amount of money you get is
- If you want to try your luck and get larger amount of money, please participate in this game a little later.

Of course, it is only meaningful when you **have grabbed** a red envelope.

Outline

Acknowledgment

During this project, I collaborated and discussed with my classmates **Cheng'an Wang**, **Huifan Zhang**, and **Letong Wang**.
Data of real trials are collected by **Cheng'an Wang**, **Huifan Zhang** and me.
I got inspiration of *Uniform Distribution* from *Zhihu*[1] and *Zybuluo.com*[2]. I think the idea *Uniform Distribution* is quite excellent, but the author does not choose a right domain for it.
All the resources of this report have been pushed to *my Github*[3].

Outline

Reference

- [1] https://www.zhihu.com/question/22625187/answer/85530416
- [2] "Brief Introduction to Framework Design of WeChat Red Envelope" https://www.zybuluo.com/yulin718/note/93148
- [3] https://github.com/KevinZhang199803/SI140_Final_Project