

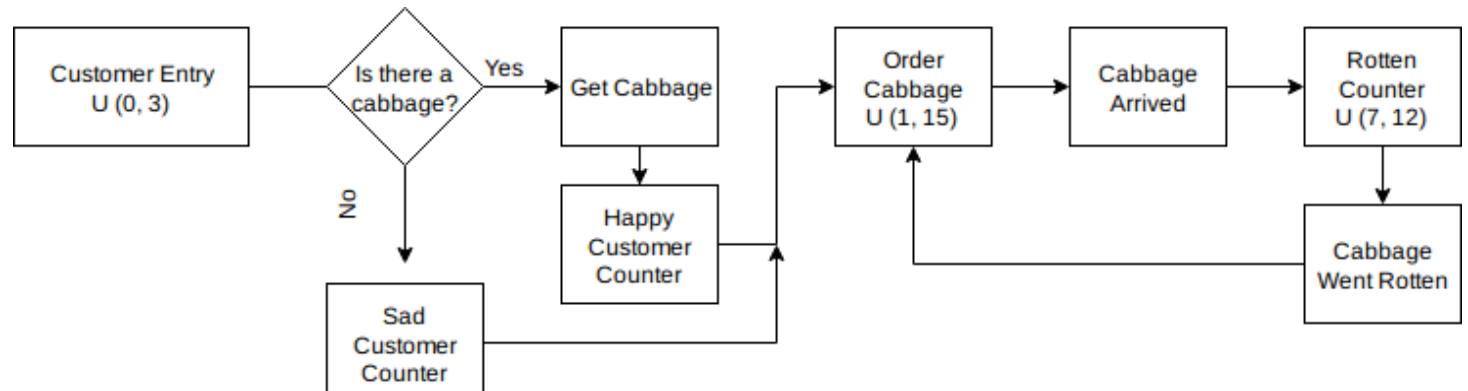


# Produce Restock Simulation

Vanita Thompson

# Goals and Flow Diagram

- I assume cabbage rots in 7-12 days.
- In this simulation they are delivered in 1-15 days.
- Customers purchase cabbage every 0-3
- My goal is to limit the amount of rotting cabbage.



# Uniform Probability Distribution


- - Customer arrival  $U(0, 3)$  days.
- - Order arrival  $U(1, 15)$  days.
- - Rotten time for arrived cabbage  $U(7, 12)$  days.

```
import matplotlib.pyplot as plt
import random
import numpy as np
import sys
import time

def order_cabbage():
    order_time = uniform(1, 15) # Order arrival time for coming cabbages
    return order_time

0
11 def order_arrived(available_cabbages):
12     available_cabbages += 1
13     rotten_time = uniform(7, 12) # Rotten time for arrived cabbages
14     return available_cabbages, rotten_time
15
16
17 def cus_entry(available_cabbages, happy_cus_left, sad_cus_left):
18     cus_arr_time = uniform(0, 3) # Customer arrival time after one customer
19
20     if (available_cabbages > 0):
21         available_cabbages -= 1
22         happy_cus_left += 1
23         return available_cabbages, happy_cus_left, sad_cus_left, cus_arr_time
24     else:
25         sad_cus_left += 1
26         return available_cabbages, happy_cus_left, sad_cus_left, cus_arr_time

def main(stock, terminators, verbose_fel=False, test_mode=False):
    assignments
    # ...
    # stock
```



# Simulation Conditions

- When one customer left, new one's entry will be added to event list.
- One customer can only buy one cabbage.
- When a customer is arrived, one cabbage will be ordered.
- If there is no available cabbage, customer will be sad.
- If there are available cabbage(s), customer will be happy.
- When a cabbage went rotten, new one will be ordered.
- If one cabbage was bought, delete earliest future rotten cabbage event.
- Serve earliest expiry date cabbage to customer to reduce rotten cabbages and improve efficiency.
- Maximum 0.003 sad customer left rate.
- Minimize number of rotten cabbages.

# Conclusion

- With lower stock size, most customer left the market as sad, because stock size is not enough for continuous customers.
- Increasing the number of stock size results in rotting much more cabbages, and it leads to waste.
- Therefore, we need to use a larger number of trials and Monte Carlo method to calculate average necessary values of each simulation such as stock size, sad customer rate, and total number of rotten cabbages.

