

In this homework assignment you will use `python` to study a probabilistic model about a grocery store checkout queue.

Clone the file `homework4-exercises.ipynb` into your Jupyter Notebook and solve the exercises directly there.

You must submit two PDF files to gradescope:

- A L<sup>A</sup>T<sub>E</sub>X'ed document in PDF format with your answers and conclusions to the two tasks, referencing the python code and citing all sources used.
- A PDF document from exporting your Jupyter Notebook

## 1 A Discrete Event Simulation of a Grocery Store Checkout

Let's build a simulation of the checkout at a grocery store. Customers like to be able to checkout quickly; it ranks almost as important as low prices on customer satisfaction surveys. After collecting groceries, a customer joins a line waiting for an available cashier to checkout. The customers have different numbers of items, which affects their personal processing time. As is common in this area of study, the cashiers will be referred to as servers and line-ups will be referred to as queues. If you wish to learn more about this rich problem, the topic is called 'queueing theory'.

We will use `python` for implementing the simulation. Discrete event simulations are naturally expressed in an object-orientated framework. `Python` has this capability, but is an advanced feature that we will not use. More readable and efficient code could be obtained in another language like `C++` or `Java`.

As a model assumption, customers arrive at the checkout in a Poisson process with rate  $\gamma$  (i.e. we have assumed that there is a random, exponentially distributed, amount of time between arrivals with mean time between arrivals  $1/\gamma$ ). Each customer has a random number of items, which is assumed to follow a negative binomial distribution. This is an assumption with little merit in reality. A better model would use an empirically measured distribution for the number of items per customer.

We model the time a customer spends at the cashier, called the processing time, as  $\delta + \alpha N$ , where  $\delta$  is a fixed time at the cashier mostly due to payment processing,  $\alpha$  is the time at the cashier per item, and  $N$  is the (random) number of items that the customer is purchasing. Since  $N$  is a random variable, the time at the cashier is a random variable. Employing a cashier is expensive for the store, so the number of cashiers will remain fixed at 5.

The efficiency of checkout depends on the choices by the store manager, as well as the customers. The checkout procedure depends on the design of the line-ups and choices made by the customers. The store can have multiple line-ups in many different configurations leading to the cashiers. The maximum number of items a customer may have in order to enter a line could possibly distinguish the lineups as in the case of 'Express' checkouts. When given a choice of which lineup to join, the customers may choose randomly or pick the shortest line or pick the line with the fewest total number of items.

Two possible procedures are:

- (A) Each cashier maintains a separate line. A customer upon arrival joins the line with the fewest people.
- (B) All customers join a single line in the order they arrive. The person at the front of the line proceeds to an available cashier.

Procedures (A) and (B) have been implemented for you in `DES_grocerystore.m` (DES stands for discrete event simulation). Read through the provided code. To learn how the code works, it might be useful to step through the code with the debugger and watch how the customers table changes. The customer table

stores information on each customer: arrival (at the checkout) time, number of items, processing time, queue number, queue position, and departure time. Check for logic errors and inform me of any as soon as possible. Considerable bonus points will be awarded to anyone identifying an error.

## 2 Assignment

**Task 1.** Compare procedures (A) and (B). Explain any interesting features of the two checkout time histograms. You might want to change the bin width for the histograms.

Other options for procedures are:

- (C) All customers join a single pool and the person with the fewest items always proceeds first to an available cashier. When there is a tie, the person having waited the longest goes first.
- (D) If a customer has fewer than, say, 10 items, he joins the shortest line labeled ‘Express’. Otherwise, he joins the shortest non-express line.
- (E) All customers join a single line after which they join another line for a cashier, which may have up to, say, 3 people in it. The person at the front of the line can wait to make a choice about which line to join.
- (F) All customers join a single pool and are processed in random order.
- (G) Same as (A), except each customer randomly selects a line upon arrival at the checkout.
- (H) Same as (A), except that the last person in any line will switch to a shorter line if available.

There are endless possible procedures. Selecting a procedure depends on picking an objective and there are many possible desirable objectives. We could work to make the mean (or median) checkout time as short as possible. We could set a threshold and minimize the number of checkout times above that threshold. We could minimize the mean checkout time per item. Maybe a low variance of checkout times is important. Perhaps the checkout experience is most affected by the lengths of queues, so it would be wise to minimize the average of the longest length queue. Even though procedure (C) seems unfair, it minimizes the total people-minutes spent waiting in line. The ‘best’ procedure depends on a clear specification of the objective.

**Task 2.** Implement another procedure. It could be one of (D) through (H) above or one of your design. Use the simulation to make recommendations to the store manager. Clearly state the objective, which could be something other than what is listed above. Your recommendation should refer to a plot of the steady checkout time distribution, but primarily be clearly stated in English sentences.

Feel free to share your implementations with others in the class. Carefully check the logic and code of any implementations you receive. It would be interesting to have a comparison of many procedures designed and implemented by many people in the class.

Note that a necessary condition for the checkout process to be stable (queue lengths don’t grow without bound) is that the mean processing time divided by the number of servers is less than the mean time between arrivals. This condition is not sufficient since the checkout procedure may result in idle servers.

There are many generalizations that can be made to this simulation. It is unlikely that customers arrive at a constant rate over an entire day. The arrival distribution could be changed. A natural choice would be an inhomogeneous Poisson process. The fixed time at the cashier  $\delta$  should likely also be a random variable and depend on the payment method (cash, credit, or other) and how the groceries are packed. A better description of how processing time depends on the number of items would likely not be linear. Also, the processing time should depend on the type of items, not just their number. Before implementing too many generalizations, it is worth getting a baseline for the checkout design recommendations, as in this assignment.