Homework #4[1]

Due by 11:59pm 4/5/2017

Overview

In this assignment, you will implement a simulation of a (simplified) local fair. You will design an experiment to run the simulation in order find a solution that meets the objectives of the fair organizers: minimize the customers' wait time while maximizing profit. You need to validate your proposed solution by performing statistical analyses on the results of your experiment.

The Local Fair Problem

This local fair runs for 100 days during the summer. Each day, the fair begins at 10am and runs until 6pm. We will make the simplifying assumption that the customer arrival patterns do not vary from day-to-day. In general, a bunch of them arrive at the start of the day, but others will still trickle in throughout the day until nearly closing time. For the purpose of this assignment, we will assume customers come individually rather than in groups. Upon arrival, the customers must first go buy tickets, which they can then use throughout the fair for various attractions. A customer will stay in the fair until they ran out of tickets or until closing time, whichever comes first. The fair currently has 13 employees who are paid $10 an hour. The cashier works from 10am-5pm, but everyone else works from 10am-6pm.

Customer Arrivals

Early Customers: About 200 customers try to come for the fair opening (10am) each day, though some arrive a little earlier and some a little later. Model this as a normal distribution with the mean at 10am, and a standard deviation of 15 minutes. People who arrive before 10am will wait in the tickets line until 10am, when the ticket booth opens.

Late Customers: Additionally, between 10:30am and 4:00pm, people will arrive following the Poisson process at a rate of 15 people per hour.

Getting Tickets

The fair has a booth selling tickets, which customers would need for the rides and games. Customers can choose to either wait in the “cash” line, where they can exchange their money for tickets from a human, or the “express” line, where they swipe their credit card to buy a fixed amount of tickets from a machine.

---

[1] This file and other support files will also be available in a Google Drive directory (after the initial commit to the repository on GitHub, small typos will be corrected here and any late-breaking additional files will be added here).

- Cash line: The customer can buy any arbitrary number of tickets. The cost is $0.25 per ticket. The service time is dependent on the number of tickets sold (because the cashier has to unspool the tickets) at the rate of 1 seconds per 10 tickets bought; plus there is the basic transaction time, which takes about 2 minutes on average; you can model this with an exponential.
- Express line: Always sells 200 tickets ($50) per transaction. The service time is: 15 seconds.

There are no limits for the number of people who can wait in lines for either services. We will also assume that people will wait patiently to buy the tickets once they are committed to the line. However, at 5pm the booth is closed, and all customers who are still waiting in either lines are turned away.

Since the Early Customers probably plan to spend more time in the fair, they have an 80% chance of wanting the the Express line. In contrast, only 10% of the Late Customers will want the Express line a priori. However, if, upon arrival, the customers destined for the Cash line discover it has too long of a wait (more than a wait time of 20 minutes *and* the line is at least twice as long as the Express line), 20% of them will pick Express line instead. Once a line is picked, customers will not switch lines or give up and leave.

The Cash line customers will buy tickets following a normal distribution with a mean of 100 tickets and a standard deviation of 30 tickets.

Main Attractions

We model three attractions:

- Merry-Go-Round:
    - cost to ride: 8 ticket per person per ride
    - service capacity: 20 seats
    - service time:
        - customers saddling up: this can largely be done in parallel. On average people take about 2 minutes to get ready. Model this as an exponential.
        - actual ride time: 4 minutes exactly.
        - customers leaving: also in parallel. On average people take about 30 seconds to gather their stuff and leave. Model this as an exponential.
    - queue model: one queue for all customers who want to ride the merry go round. Assume a queue capacity of “infinity” -- as in, customers won’t be turned away when the queue is too long.
    - operator cost: 1 person, $10 an hour from 10am - 6pm.
- Stalls and games:
    - cost to play: 1 ticket per game
    - service capacity: 50 (10 stalls, each stall can accommodate up to 5 players)
    - service time:

        - On average, people finish a game in 30 seconds. Model this as an exponential.
    - queue: No queue -- if a customer comes and there is no free space, they will just go on to do something else.
    - operator cost: 1 person per stall, $10 an hour from 10am - 6pm.
- Roller coaster:
    - cost: 16 tickets per person per ride
    - service capacity: a train with 2 seats per row for 30 rows.
    - service time:
        - customers getting on: ride operator has to assign seats and make sure that everyone is safely buckled in. Assume the operator works at a rate of 3 seconds per pair of customers, in addition to an average overhead of 1 minute (Modeled as an exponential).
        - actual ride time: 3 minutes
        - customers getting off: can be done in parallel. On average people take about 30 seconds to gather their stuff and leave. Model this as an exponential.
        - operator cost: 1 person to run the ride.
    - queue model: one queue for all customers who want to ride the roller coaster. Assume a queue capacity of “infinity” -- as in, customers won’t be turned away when the queue is too long.

Customer Behavior

We will make an idealized assumption that the customer has only two states: waiting in line or being served; we also won’t worry about simulating getting them from one attraction to another. We further assume that the customers pick attractions randomly without context (their choice of the next activity does not depend on what they’ve already done). An idle customer will pick which attraction to try next following this distribution:

- 40% Roller Coaster
- 35% Stalls and Games
- 25% Merry-Go-Round

If they do not have enough tickets for an attraction, they will pick from the remaining attractions (e.g., if they have just 8 tickets left, there’d be a 58% chance that they’d pick Stalls and Games; 42% chance they’d pick Merry-Go-Round). If they only have enough tickets left for Stalls and Games, but all spaces are full, then this customer is considered to be done for the day (wasting however many tickets they have left).

Specifications for Building the Simulation Model

The main component of your program is to simulate the local fair scenario as described above, and output variables of interest:

- On average, how many people are at the fair (you may have to consider whether the average changes at different times of the day)

- What is the average queue length for the various queues in the scenario?
- What is the average wait time for the various queues in the scenario?
- On average, how many tickets did the customers “waste” (fair ended before they can spend them all)
- On average, how much money does the fair make in a day?
- On average, how busy are the various human employees (the cashier, the merry-go-round operator, the roller-coaster-operator, and the 10 stall operators)?

You may also need to write some additional program(s) to run the experiment (see below) and perhaps to do some analysis on the outputs.

You should prepare a README file (separate from the analysis report) for the grader. It should explain how to run your program and provide any other relevant information that might help the grader evaluate your program.

Experimental Design

Our goal for setting up the simulation is to find ways to improve customer wait times at various lines while maximizing profit. Some alternatives under considerations are:

- improve ticket selling by:
    - adding more cash line(s)
    - adding more express line(s) (a new vending machine will incur a one-time cost of $1000, however)
    - no modification can be made on the amount of tickets solve via the vending machine (i.e., you can’t force customers to buy $1000’s worth of tickets)
- impact the merry-go-round line by:
    - renting a bigger carousel that fits 30 people (additional equipment rental cost of $5000)
    - downgrading to a smaller carousel that fits 15 people (save $2000)
- impact the stalls and games:
    - adding more stall(s) at the additional cost per stall: $500 for the equipment, wages for the human operator
    - getting rid of existing stall(s): pay fewer human operator(s), get $200 for selling the equipment
- no modification can be made on the roller coaster ride (too expensive to rebuild)

Design some experiments to help the fair organizers to figure out their best option. You do not need to experiment with *all* of the above alternatives. Pick and choose a couple that looked the most promising after you’ve simulated the initial condition.

Report

Your report should explain your experimental design in detail. Give justifications for your choices. Describe enough details about your experimental decisions such that another student could replicate your experiment using their own simulation implementation.

After running the experiments, you should present the results in a clear and easy to understand way. Use tables and graphs as necessary. Some issues you should think about and address in the report:

- How good is the default setting? Without changing anything, is the system stable (i.e., no line grows out of hand)? Is the local fair making money?
- Assuming that you cannot change customer arrivals or behaviors, does changing the fair infrastructure (ticket booths, attractions) improve our two objectives?
- What if we *can* advertise to get more customers to come (should we target Early Customers or Late Customers?), or somehow convince them to prefer some attractions over others. How would these modifications change the bottom line?

Based on your experimental analyses, draw conclusions about what the local fair should do.

Grading Guideline

Assignments are graded qualitatively on a non-linear five point scale. Below is a rough guideline:

- 5 (100%): The program works as expected; there is a clear enough README for the grader; the report is complete and thorough -- insightful observations and correct analysis.
- 4 (93%): The program works as expected, but possibly with some minor flaws; there is a clear enough README for the grader; the report is complete with correct analysis.
- 3 (80%): The program works as expected, but possibly with some minor flaws; there is a clear enough README for the grader; the report is complete but cursory, or there are some minor inaccuracies in the report.
- 2 (60%): The program has major problems; there is a clear enough README for the grader; OR: the program is OK but the report is missing/incomplete/incorrect.
- 1 (40%): A serious attempt has been made on the assignment.