CX 4230 - Mini Project 3

Eduardo Amorim

April 2023

1 Introduction

Our goal in this project is to create a simulator for the Citi bike rental in NYC.

2 DES Model

First, create a discrete-event simulation model with the following characteristics.

- 24 logical-hours of simulation time.
- There are some number of stations, m. Each station is modeled, conceptually, as an elementary queue.
- On a given day, suppose there are n riders in total.
- The n riders arrive randomly. Their interarrival times constitute an autonomous, stationary, and independent stochastic process, distributed exponentially with mean rate λ.
- When a rider arrives, she selects a bike station randomly. The probability of picking station i is p_i.
- The rider goes to station i. If a bike is available, she takes it. Otherwise, she must wait until a bike is available there. Allow for the possibility of an unlimited number of bikes at the station, which will be needed by one of the experiments below.
- The rider chooses their destination randomly. The probability of selecting station j, given that the rider is at station i, is given by $q_{i,j}$. It is possible that $i = j$, that is, the rider uses the bike but ends up returning it to the same station.
- The rider uses the bike for some amount of time before returning it. The amount of time she uses the bike is drawn from a log-normal distribution with mean μ and standard deviation
- After returning her bike, the rider leaves the system.

Please see the code for the implementation of the simulation below:

```
class Simulation:
  def __init__(self, env):
    self.env = env
    self.total = 0
    self.success_rental = 0
    self.wait_time = 0

  def bike_station(self, env, station):
    yield self.env.timeout(random.expovariate(LAMBDA))
    start_time = self.env.now
    self.total += 1
    station_index = random.choices(range(STATIONS_COUNT), STATIONS_PROBABILITIES)[0]
    while bikes[station_index] == 0:
      yield self.env.timeout(1)
    end_time = self.env.now
    bikes[station_index] -= 1
    self.success_rental += 1
    self.wait_time += (end_time - start_time)
    end_index = random.choices(range(STATIONS_COUNT), END_PROBABILITIES[station_index])[0]
    self.env.process(self.return_bike(station_index, end_index))

  def return_bike(self, station_id, end_index):
    trip_time = random.lognormvariate(MEAN_TRIP, STD_DEV_TRIP)
    yield self.env.timeout(trip_time)
    bikes[end_index] += 1
```

2.1 Verification

To verify the simulation code, we can create an analytical. The simplest, but effective way we can do this is to model a single station with a single bike and two riders. Our parameters will then be:

- Station capacity: 1
- Riders count: 2
- Arrival rate: 2 riders/hour (yields a $\lambda = 0.0333$ riders per minute.
- Ride time: 10 minutes ($\mu = 2.3026,\ \sigma = 0$)
- Start station probability: 1 (only one available)
- End station probability: 1 (only one available)

Now, we need to verify that the simulation correctly handles the following scenarios:

1. A rider arrives and finds the bike station empty, then waits for the bike to become available and then takes the bike. After 10 minutes, returns the bike to the same station and leaves the system.

2. A rider arrives and finds the bike station available, takes the bike. After 10 minutes, returns the bike to the same station and leave the system.

We can now calculate the probability of a successful rental and the average waiting time analytically as follows:

1. Probability of a successful rental:
   Since there is only one bike and one station, the probability that the rental is successful is just the probability the bike is available when a rider arrives.
   Since we have a rate of two riders per hour, the expected time between two arrivals is 30 minutes.
   We can then calculate the probability for a successful rental by the following expression:
   Note: Consider NS as unsuccessful and S as successful.
   $$P(NS) = 1 - e^{\frac{-\mu}{2}} = 1 - e^{\frac{-2.3026}{2}} = 0.393$$
   Therefore,
   $$P(S) = 1 - P(NS) = 0.607$$

2. Average waiting time:
   The average waiting time for a successful rental is equal to the expected time between a rider's arrival and the availability of the bike at the station. Since there is only one bike and one station, the waiting time is equal to the ride time of the previous rider.
   Recall that the ride time is log-normally distributed with mean 2.3026 and standard deviation of 0. Then, to compute the expected value, we can use the following expression:
   $$E[X] = e^{\mu + \frac{\sigma^2}{2}}$$
   Note: Consider RT the ride time.
   Therefore,
   $$E[RT] = e^{2.3026+0} = 10$$

Now that we computed analytically the solutions, let's run the simulation and check if the values are contained within a confidence interval.

1. Probability of successful rental:
   Simulation output: 0.6006 where the 90% confidence interval is (0.5801, 06211).

2. Average waiting time:
   Simulation output: 10 where the 90% confidence interval is (10, 10).

Since our simulation yielded values consistent with the analytical solutions, we have an indication that our model is accurate.

3 A baseline experiment

Suppose the number of available bikes at every station is fixed at 10 bikes per station. Suppose the number of riders $n = 3500$ and use your simulator to estimate the following:

1. the "probability of a successful rental," that is, the fraction of riders who are able to get a bike;
2. a rider's average waiting-time for a bike, considering only riders who successfully got a bike.

Compute a 90% confidence interval for your estimates.

For the simulation parameters, use the following:

- For the system-wide interarrival times, which are exponentially distributed, use $\lambda = 2.38$ riders per minute.
- For the ride times, which are log-normally distributed, use $\mu = 2.78$ and $\sigma = 0.619$. (This value corresponds with an average ride-time of $e^{\mu} \approx 16$ minutes.)

Please see below the python implementation of the experiment:

```
SIMULATION_TIME = 1440 # minutes in a day
STATIONS_COUNT = 81
INIT_BIKES_PER_STATION = 10
RIDERS_COUNT = 3500
LAMBDA = 2.38
MEAN_TRIP = 2.78
STD_DEV_TRIP = 0.619

start_station_probs_df = pd.read_csv("start_station_probs.csv")
trip_stats_df = pd.read_csv("trip_stats.csv")
start_station_totals = trip_stats_df.groupby('start')['count'].sum()
trip_stats_df.reset_index()
averages = []
for index, row in trip_stats_df.iterrows():
  s = row['start']
  x = row['count'] / start_station_totals.loc[s]
  averages.append(x)
trip_stats_df['average'] = averages
trip_stats_df = trip_stats_df.drop(columns=['mean', 'std'])
trip_stats_df

p_i_j = trip_stats_df['average'].to_numpy()
stations = start_station_probs_df.iloc[:,0].to_numpy()
q_i_j = np.zeros((81,81))
for i in range(q_i_j.shape[0]):
```

```
  for j in range(q_i_j.shape[1]):
    try:
      q_i_j[i][j] = trip_stats_df.loc[(trip_stats_df['start'] == stations[i]) &
      (trip_stats_df['end'] == stations[j])]['average'].iloc[0]
    except:
      q_i_j[i][j] = 0
END_PROBABILITIES = q_i_j.tolist()
STATIONS_PROBABILITIES = start_station_probs_df.iloc[:, 1].to_numpy().tolist()

success_rentals_list = []
average_wait_time_list = []
for i in range(100):
  env = simpy.Environment()
  sim = Simulation(env)

  stations = [simpy.Resource(env) for _ in range(STATIONS_COUNT)]
  bikes = [INIT_BIKES_PER_STATION for _ in range(STATIONS_COUNT)]

  for i in range(RIDERS_COUNT):
    env.process(sim.bike_station(env, i))

  env.run(until=SIMULATION_TIME)
  success_rentals_list.append(sim.success_rental)
  average_wait_time_list.append(sim.wait_time / sim.success_rental)

success_rentals_conf_interval = st.t.interval(confidence=0.9,
        df=len(success_rentals_df)-1, loc=success_rentals_df.mean(),
        scale=st.sem(success_rentals_df))

average_wait_time_conf_interval = st.t.interval(confidence=0.9,
        df=len(average_wait_time_df)-1, loc=average_wait_time_df.mean(),
        scale=st.sem(average_wait_time_df))
```

With this parameter setup, our simulation returned:

1. Probability of successful rental: 0.983014, with a 90% confidence interval of (0.9818892013022361, 0.9841393701263353).

2. Average waiting time: 34.719549, with a 90% confidence interval of (34.566191830681085, 34.87290634009903).