

Mini-project 3: NYC Bike Share

CX 4230, Spring 2023¹

Spring 2023: Due Fri Apr 28 at 11:59 pm

¹ Last updated: Thu Apr 13 11:35:29 EDT 2023.

In this mini-project, you will implement a discrete-event simulation to study Citi Bike, which is New York City's bike-sharing service. The goal of this mini-project is to get hands-on experience building a discrete-event simulation.

Here are the basic ground rules:

- You may work individually or in pairs. See section 4.
- You can implement your solution in any programming language.
- You will submit a write-up (PDF file) *and* your source code. See section 5.
- The assignment is due on Friday, April 28, at 11:59 pm. See section 5 for the late policy.

1 Scenario: The NYC mayor needs your help!

Path frequencies
("infrequent" paths may be omitted to reduce clutter)



Figure 1: The NYC Citi Bike service places rental bikes all around the city at *stations*. The available data indicate what trips took place: when and where a rider picked up a bike and returned that same bike (possibly different locations). This sample visualization shows stations (orange dots) and trip edges (blue edges between starting and ending locations). The darkness and thickness of a trip edge indicates how many such trips took place, with edges omitted when they fall beneath some threshold.

New York City runs a bike-sharing service, called Citi Bike.² There are bike stations all around the city where riders can pick up a bike, ride around, and return the bike (possibly at a *different* location from the pick-up). This information is illustrated in fig. 1.

² See: <https://citibikenyc.com/>

City planners have determined that, roughly speaking, the cost to run this service is proportional to the total number of bikes in service. The mayor needs your help figuring out how to allocate a fixed number of bikes to all stations in a way that can meet demand.

The city planning office has been collecting data about current usage of the Citi Bike service.³ We preprocessed some of this data from June 2022 to help you build your simulation. These data are described in section 3.

³ See: <https://citibikenyc.com/system-data>

2 Instructions

2.1 Task 1: Create a simulator

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

- You wish to model one logical-day (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 σ .⁴

⁴ See: https://en.wikipedia.org/wiki/Log-normal_distribution

- After returning her bike, the rider leaves the system.

Explain how you *verified* your simulation code.

2.2 Task 2: A baseline experiment

Suppose the number of available bikes at *every* station is fixed at 10 bikes per station. Suppose the number of riders $n = 3,500$ 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.)
- The p_i values are given in the data file, `start_station_probs.csv` (see section 3).
- To derive $q_{i,j}$ values, use the raw data of `trip_stats.csv` (see section 3).

2.3 (PAIRS ONLY) Task 3: An “idealized” experiment

Suppose there is no limit on the number of bikes at each station. Use your simulator to determine, for each station, the minimum number of bikes that could be made available to meet demand fully.

3 Data file schemas

We are providing two data files for you to use to get the parameters p_i (starting-station probabilities) and $q_{i,j}$ (trip destination probabilities) per section 2. These data are available on the GT GitHub at <https://github.gatech.edu/cx4230/sp23mp3-citibike> in the data/ subfolder.

The first data file is named `start_station_probs.csv`. It contains data for 81 bike stations. The format is comma-separated values. The first line is a header line. The remaining lines are (station name,

probability) pairs. A sample appears in table 1. From it, you can see that a rider who “arrives” into the simulation will choose the Grove St PATH station with probability 0.043504.

start_station_name	probability
South Waterfront Walkway - Sinatra Dr & 1 St	0.044679
Grove St PATH	0.043504
Hoboken Terminal - Hudson St & Hudson Pl	0.033629
Hoboken Terminal - River St & Hudson Pl	0.029832
Newport Pkwy	0.027035

Table 1: A sample of rows from the `start_station_probs.csv` file, which you should use for the p_i parameter in your simulations.

The other data file is `trip_stats.csv`, which you should use to derive values for $q_{i,j}$. This file contains a little over 5,100 start-end trip pairs. There are five columns, a sample of which appears in table 2. Each row indicates a starting station for the ride (the start column), an observed destination station for the ride (end), and the number of times a ride was observed between this pair of stations. Therefore, you should use these values to determine a suitable value for $q_{i,j}$ in your simulation. (You may ignore the mean and std columns.)

start	end	count	mean	std
11 St & Washington St	11 St & Washington St	142	25.929108	39.186350
11 St & Washington St	12 St & Sinatra Dr N	44	56.655303	149.709313
11 St & Washington St	14 St Ferry - 14 St & Shipyard Ln	48	12.481597	16.335279
11 St & Washington St	4 St & Grand St	47	7.348582	2.465807

Table 2: A sample of rows from `trip_stats.csv`, which you should use to determine the $q_{i,j}$ parameters of your simulation.

4 Teaming

You may work individually or in pairs. Teams of two have additional requirements for the assignment, as noted above.

To “declare” your team, do the following:

- Visit the Canvas “People” page for Mini-project 3 (“MP3”).⁵
- If you are working individually: Pick an unused team number and add yourself.
- If you are working in a team of two: Both of you should pick an unused team number and add yourself to the same unused team number.

⁵ Link: <https://gatech.instructure.com/courses/310452/groups#tab-35753>

5 How to Submit

Create a *private* git code repository at [github.gatech.edu](https://github.com/gatech) (not github.com!)

For teams, only one person needs to create the repository. Place both

your code and a PDF report summarizing your results in this repo. You'll submit the URL to this repository on the page for this assignment in Canvas. In the repository itself, create a `README.md` file that tells us a) your individual or team number and member(s) of the team; b) which file is your PDF report; and c) what the organization of your code is (so we can inspect and evaluate it).

Important! Since your repository will be private, please be sure to add *everyone* from the teaching staff to your repo: the two TAs, Albert Chen (achen355) and Varun Lakshmanan (vlakshmanan30), and Prof. Vuduc (rvuduc3). Otherwise, we will not be able to see and grade your submission and you will get a zero.

Late submissions. You may submit the assignment *without penalty* until Sunday, April 30, at 11:59 pm. There is a 25% penalty if you submit on Monday, May 1. No assignments will be accepted after that time.

References