# CTA Simulation

CSC 521 – Final Project

Dhruv Dobariya  
Daniel Kwan  
  
  
Under the guidance of Dr. John McDonald  
  
DePaul University  
Winter 2023

## Table of Contents

[CTA Simulation 1](#_Toc129716931)

[Table of Contents 2](#_Toc129716932)

[Executive Summary 3](#_Toc129716933)

[Domain Knowledge 3](#_Toc129716934)

[Technical Summary 4](#_Toc129716935)

[Code Description 5](#_Toc129716936)

[Results 6](#_Toc129716937)

## Executive Summary

Many people living in Chicago take the Chicago Transit Authority (CTA) Train for commuting and for leisure. However, there have been a steep decrease in the frequency of CTA trains after Covid. We have developed a simulation model for the Blue Line of CTA using the Monte Carlo Method. The Monte Carlo Method is a simulation technique which helps calculate outcomes using input parameters and random numbers. The aim of this project is to calculate the number of passengers that are able to board the train at each station of the Blue Line using a set of input parameters such as the number of trains, day of week, weather conditions, and big events.

With our simulation, we were able to determine that the current CTA schedule is adequate for most situations. However, in significant outlier events with good weather and a big event, the typical number of trains can’t cover 95% of all cases. As Chicago population grows and ridership rebounds from COVID-19, we can expect these circumstances to occur more frequently, perhaps necessitating additional resources for the Blue Line. In such circumstances, we would be able to adapt our simulator to predict that future demand.

## Domain Knowledge

Domain knowledge, as in every Monte Carlo simulator, plays a key role in this project. The Chicago Transit Authority (CTA) handles public transportation in Chicago. The CTA runs daily trains; only the red and blue lines operate for the full 24 hours. There are several other routes such as Brown, Pink, Green, Yellow, Purple and Orange. We are simulating the Blue Line which runs from O'Hare Airport to Forest Park. It has 33 stations in total which includes stations in Downtown Chicago. The Blue Line has 200 trains that run in each direction each day. A single Blue Line train has 8 coaches in total with a peak coach capacity of 80, for a total of 640 possible passengers per train.The CTA publishes daily ridership data going back to 2001. We used this data to inform our ridership distributions.

## Technical Summary

The distributions we use in our Monte Carlo Simulation are built from daily ridership data collected by the CTA from 2001-2023. For the sake of consistency, we limited our analysis to the years 2015-2018, which avoids excessive ridership inflation due to population growth as well as passenger impacts due to the COVID-19 pandemic. These distributions initially appeared to be bimodal in nature (*Figure 1*); however, further examination revealed that there are two different distributions separated by time: One distribution covers weekdays with commuter traffic, while the other is for weekends and holidays with significantly less traffic (*Figure 2*).

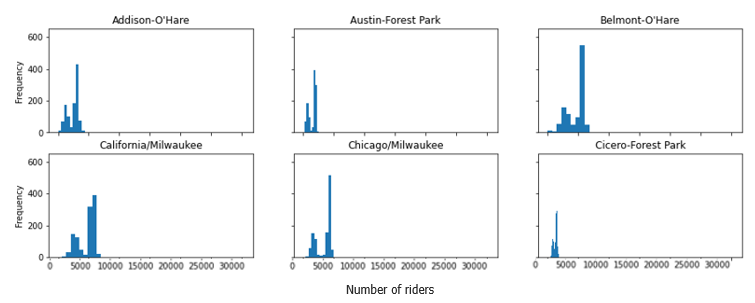


Figure 1: Histogram of the frequency of ridership numbers at a subset of Blue Line Stations.

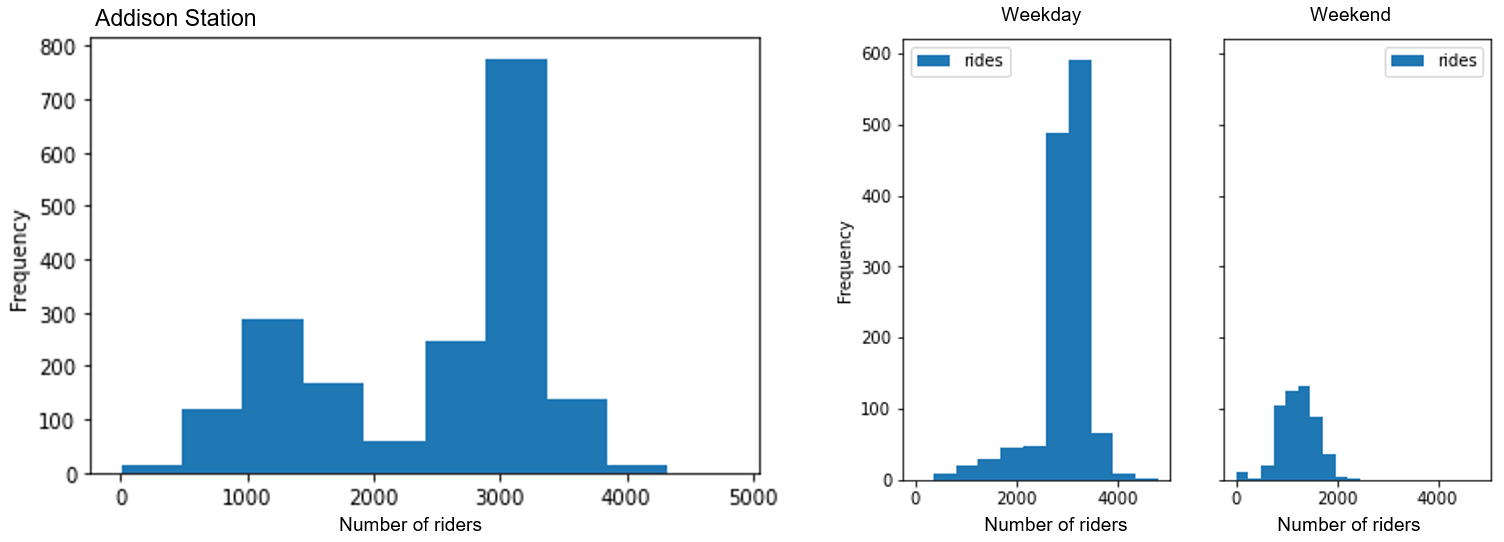


Figure 2: Histogram of Addison station broken down by weekday vs weekend.

### Code Description

The core of our simulation uses a for loop to iterate through the 33 stations on the Blue Line, and an accumulator to track the number of passengers on the train. At each station, we use the calculated means and standard deviations to generate a random integer that represents the number of passengers boarding the train at that station. We log riders disembarking and embarking and determine the train’s capacity, returning excess riders to the platform if necessary. Finally, we use a dictionary to track how many people on the train want to deboard at each future station, and evenly distribute the new passengers across valid stations when they board.

Since the above loop only covers one direction on the line, we wrote an additional method to simulate the opposite direction. This method is largely identical, but for realism’s sake does iterate through the stations in the reverse order.

Our simulation parameters cover three conditions: Work week commuter traffic, weather conditions, and the presence of large crowd events such as sports games. For commuter traffic, we simply supply the initial Monte Carlo object with a different distribution. Weather conditions and big events are covered by generating the set of random passenger numbers and applying a multiplier or penalty as appropriate.

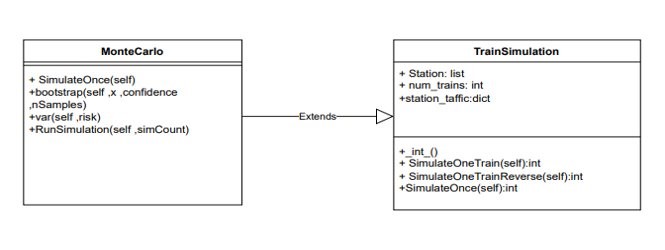


Figure 3: UML diagram of the simulator code.

### Results

In our analysis we simulated a variety of conditions to contrast our parameters. First, we ran different simulations covering commuter traffic and weather conditions. In Figure 4 (left), we can see that the number of trains required increases as circumstances lead to more passengers. Intuitively, we see that commute timing is a much more significant factor for train ridership than weather—since many commuters must go to work regardless of the conditions outside. In Figure 4 (right), we examine extenuating circumstances: What is the worst-case scenario we can expect in 95% of all trials? On a sunny day with a big event, Chicago rail might expect to require over 250 trains per day in order to not leave any riders stranded on a platform. This is an increase of 25% over the typical number of trains scheduled, and with five major sports teams and the nation’s third largest population, we might expect these circumstances to occur multiple times per year.

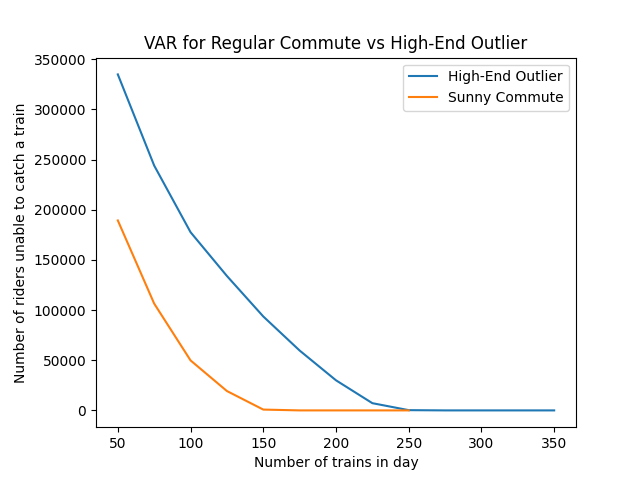
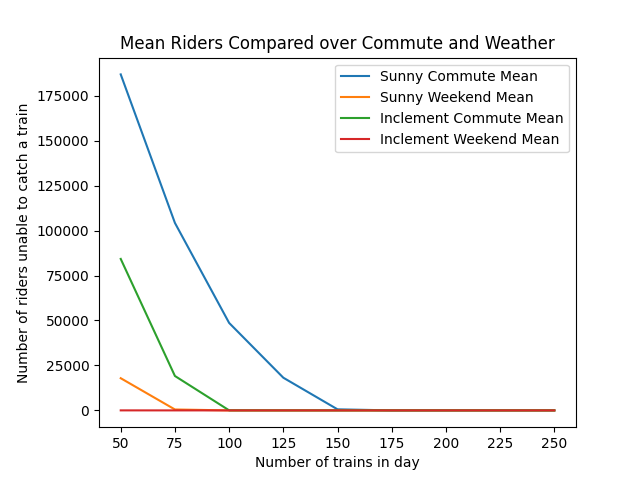
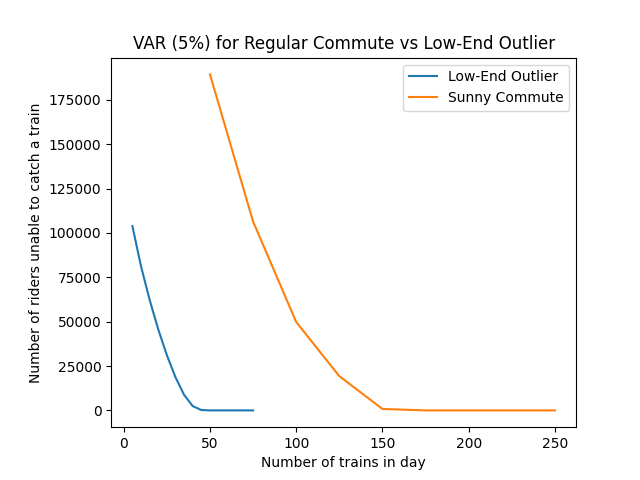


Figure 4: Commute and Weather Patterns (left), High-End Outlier Event (right)



Finally, we examined the fewest number of trains required—the 5% VAR for the lowest-traffic time period, an inclement weather weekend. Here we can see that even on Chicago’s busiest train line, we’d need fewer than 50 trains over the course of a day to satisfy ridership demand.

Figure 5: Low-End Outlier Event

## Individual Report: Dhruv Dobariya

**Milestone 1**

From day 1 of milestone 1 itself I started searching for hot topics for our project. I went to the web to catch ideas if I got any and tried my best to come up with a topic. I travel in CTA every day to commute either to work or class. Once there was a construction going on at Belmont Station during rush hours. I realized that it should not be happening during rush hours as a lot of people get affected by trains getting late due to construction. I got my idea of simulating CTA train to improve frequency from that incident.

**Milestone 2**

We started meeting as a group initialized strategies to work on. I went online and started gathering information on ridership of CTA trains. The website of CTA and Wikipedia were the best resources I found to understand length and breadth of the CTA “L” system. It helped me to gain domain knowledge needed to build this simulation. Apart from that I used to talk to Station Handlers at a few stations like Fullerton to get basic information on ridership based on real life parameters. From all the resources, I learned that CTA trains run at specific times. There are several complications in running trains such as, the track of a single route has specific DC voltage. Therefore, there is a limit on the number of trains that can run on a single route. I presented the idea in class. I wrote notes and feedback given by professor on to which we had meeting as group. After the first meeting we decided that I will start a base simulator. So, I started working on a base simulator by creating a list of 33 stations of CTA Blue Line. I went on the web to set basic integers for NumPy random numbers generator.

**Milestone 3**

In the next group meeting we discussed our simulator more in-depth and started outlying plans for base simulator. With notes and discussion in group meeting and feedback from professors, I extended the simulation to work in the Monte Carlo base class and built out initial boardings and departures of passengers at each station of Blue Line. The simulator is simulating ridership from O’Hare to Forest Park with random ridership generated at each station. In this simulator I changed the random input values where I considered that each coach has a maximum capacity of 80 passengers. I also printed total footfall at each station which gave me basic idea of where the simulator is heading.

**Milestone 4**

After having the clear basic idea of how the simulation works. I discussed the simulator with professor in class and asked for feedback for the next directions. Since the discussion I have started asking questions to the simulator so that more complexities can be added. Our group had a meeting after that, and we discussed our paths for milestone 6.

**Milestone 6**

I have initiated the presentation for milestone 6 and added descriptions as asked by the professor. I created a UML Diagram of our code which outlines the explanation of different methods and parameters. After finishing the presentation as a group, we had a meeting, and we decided our paths for the final code. We discussed some ways to add random events and weather forecasts for generating random numbers. I started working on that and did some research through some websites (citation provided) and came up with percentages to multiply it with random numbers generated. After that I initiated final project report and wrote some of the technical and non-technical summary.

## Individual Report: Daniel Kwan

**Milestone 1**

## Works Referenced

1. https://www.transitchicago.com/assets/1/6/Monthly\_Ridership\_2022-10(Final).pdf
2. https://data.cityofchicago.org/Transportation/CTA-Ridership-L-Station-Entries-Daily-Totals/5neh-572f
3. https://en.wikipedia.org/wiki/Chicago\_Transit\_Authority
4. https://en.wikipedia.org/wiki/Blue\_Line\_(CTA)
5. https://www.transitchicago.com/
6. https://numpy.org/
7. https://www.mlb.com/cubs/schedule.
8. https://www.weather.gov/lot/.
9. https://en.wikipedia.org/wiki/List\_of\_Lollapalooza\_lineups\_by\_year