Anna Paulsen, Mujtaba Pal, Teresia Sami
MATH 282-02

Case Study 4: Bus Line

SIMULATION GOALS

The goal of our simulation was to create an optimal bus schedule for customers and the bus line. We hope to optimize this schedule by:

Reducing average customer wait time

By keeping track of the wait time per customer from their arrival at the bus stop/bus station, we would be able to know when the waiting time had become longer than we wanted and to send in a bus for the customers to board. This is helpful for the customers since they do not have to wait at the bus stop for too long as they wait for the bus while the bus line does not send in buses unnecessarily.

Reduce bus delays

By ensuring that the buses coming in are contingent on the customers' wait time and the number of customers at the bus stop/ bus station, this would reduce bus delays for customers. This is because the bus arrival becomes a dependent variable on the customers hence curbing delays.

Optimize profits for bus line

By only sending in passengers when the number of passengers at the bus stop/station is greater than half of the bus capacity for the bus station and three quarters of the bus stop capacity, our simulation would optimize profits for the bus line. This is because the buses would be sent in with the guarantee of customers at the bus station ensuring constant flow of clientele and guaranteed profits from the passengers' bus fare.

Increase overall efficiency

Through this simulation, customer needs are met by reducing their wait time by factoring that into the conditions under which a new bus is sent into the bus station. This would ultimately increase customer satisfaction with the bus line that implements our simulation. By ensuring that buses are sent in only when necessary, the bus line optimizes its profits and its resources. These two factors cement the goal of increased overall efficiency of the bus line and for the customers.

SIMULATION SET UP

For our simulation set up, we had several inputs and expected outputs:

INPUTS

For the inputs, they were categorized in fixed and non-fixed variables depending on various factors.

Fixed Variables

Time Interval

The time intervals in our simulations were pre-determined as 7-9am as the rush hour and 9am-12pm as the non-rush hour. This was suggested in the prompt but also corresponds with real life where bus lines usually have more clients in the early mornings majorly because of commuting to work. Our simulation maintained this interval in all our four scenarios.

Bus capacity

For our simulation, we used a bus capacity of about 20 people because we were assuming that the bus was partially filled when it stopped at the bus station/bus stop. This was also constant in all our simulation variables in order to adequately keep track of the passengers' wait time and the number of passengers needed. While it is possible to use different bus capacities, a fixed capacity was best for our code in order to get consistent results and compare them more efficiently.

Non-fixed variables

Bus stop/bus station capacity.

While we chose a bus station capacity of 300 and a bus stop capacity of 10, these numbers can be varied depending on the size of the station. These numbers were approximates based on research on a local bus line. The bus stops are a lot smaller than bus stations hence the difference in the chosen values. While the outputs may change, they would be adjusted relative to each other so the ratio would be constant.

Number of passengers

Our simulation was modeled after the Pennsylvania bus station which is mid-sized. Relative to other bus lines that operate in other cities like New York, the number of passengers would increase dramatically because of the city's population. However, just like the bus station/stop capacity, the outputs would also change relative to each other.

EXPECTED OUTPUTS

The expected outputs from our simulation were:

Number of buses

From running the simulation, we were looking to get the total number of buses that would be sent to the bus station for the different time intervals (rush hour versus non-rush hour) and at the two different locations (bus stop versus bus station). Thus would give us an idea of how many buses the bus line would need to service its clients. Additionally, this could help the bus line project their profits.

Average wait time.

Another expected output was the average wait time per customer. This was essential in helping determine when another bus would be sent in. It was calculated from the arrival of the passenger at the bus station until the bus arrived.Since different passengers arrived at different times, the average wait time was the mean of the total wait time of all the passengers divided by the total number of passengers.

Bus arrival times

Apart from the total number of buses that were sent in, the time the buses arrived was also recorded in comparison to our T=120 minutes for rush hour and T=180 minutes for non-rush hour. The bus arrival times would give a general idea of the length of time between the arrival of the buses which would help approximate the bus line's overall efficiency.

ASSUMPTIONS MADE

Some of the assumptions made during our simulation include:

- In our simulation we assumed that rush-hour was from 7-9am and non-rush hour from 9am-12pm. This was a key assumption to make in order to account for the fact that the bus line would receive different number of passengers at different times of the day due to factors like commuting
- Another assumption that we made was that outside factors like weather and customer preference patterns would not be factored into the simulation. Recent concerns like customers' views on COVID protocols, we assumed would not affect the bus business. The ever-constant changing nature of these factors is the prime reason why we could not account for them in the simulation.
- Additionally, we assumed that the bus line's functionality was constant. This means that we would not consider any disparity that would arise should it be a weekend instead of a weekend. We modeled the simulation in a constant non-changing form. We also ignored possibilities of administerial influence from the bus line officials in our simulation.
- Finally, our simulation inputs, even though modeled after a local bus line, were numbers that we arrived at by assumptions. These were: Bus station - 300
Bus stop - 10
Bus capacity - 20

Explaining the Code:

We had 4 different scenarios we looked at and thus four different codes. The major crux of our code and our logic remained the same but we changed conditions depending on our scenario to try to come up with optimal schedules for the bus line. We did some research and found the number of passengers we expected to arrive for our scenarios. We then made a random number generating function to assign our passengers random arrival times. The interval of the random number generating function varied depending on the time interval of our scenario (120 minutes for rush hour and 180 minutes for non-rush hour) and we generated random numbers equal to our number of passengers. The random arrival times were sorted in ascending order and stored in a function called *passArrival.* We had counters for our total number of buses (*buses)* and passengers who had arrived to board the bus *(passengers)* set equal to zero outside the loop. We also made matrices to keep track of the arrival times for the buses and the waiting times for the individual passengers. We also wanted to keep track of the maximum waiting time (*maxWait)* for any specific passenger which we defined to be zero outside the loop. We made a *waitStart* variable to store the arrival time of the first passenger.

Inside our bigger for loop, we first added one to our passenger counter every time a passenger arrived. We wanted to check for the max wait time for any passenger. To do this, we first checked if *waitStart* was equal to zero. If it was that meant no one was waiting so we assigned the arrival time of the passenger who had just arrived to be equal to *waitStart*. Otherwise *maxWait* became equal to the difference between the arrival time of the passenger who had just arrived and the longest time that a passenger before had already been waiting. Once we had taken care of storing the max waiting time, we looked at the conditions under which we would send buses. For our rush hours at the bus station and the bus stop, we sent a bus everytime half of the bus capacity (*busCapacity)* had been reached and *maxWait* was greater than 15 minutes. For the non-rush hour at the bus station and the bus stop, we varied our condition and sent a bus everytime three fourth of the bus capacity (*busCapacity)* had been reached and *maxWait* was greater than 25 minutes. If these conditions were not met for their respective scenarios, we looked at whether the number of passengers had exceeded the bus capacity (*busCapacity)* and sent a bus if it had. In the case that even this condition was not met, we looked at whether the number of passengers had exceeded the bus stop capacity (*busStopCapacity)* and sent a bus if it had. Whenever any of our conditions for sending a bus were met, we added to our counter of total buses and noted the arrival time of the buses in our *busArrival* matrix by assigning them the arrival time of the passenger who had just arrived (*passArrival)*. Whenever a bus had arrived, we also set *maxWait, waitStart* and *passengers* equal to zero again. We also had an inner loop inside which we found the wait times for the passengers by finding the difference between the bus arrival times and the passenger arrival times for each passenger.

At the end of our code, we calculated the average waiting time *(averageWait)* for our passengers and also looked at the max wait time.

Results:

We found different results for our four different scenarios. For the rush hour at the bus station, we expected wait times for passengers to be the lowest and bus arrival times to be the least spread out of all of our scenarios. Our results matched our expectations and we found that the average wait times for passengers were about 5 minutes as is visible in Fig 5 while buses arrived approximately every 12 minutes as can be seen in Fig 5. These times were pretty fast and made sense given that it was a bus station and was rush hour. A total of 9 buses arrived over the two hour period. Wait times for individual passengers were also pretty low as shown in Fig 1.

For the non-rush hour at the bus station, we expected wait times for passengers to be higher than rush hour and bus arrival times to be more spread out. Our results again matched our expectations as we found that the average wait time for the passengers was about 11 minutes as is visible in Fig 6 while buses arrived approximately every 25 minutes as can be seen in Fig 6. A total of 7 buses arrived now within the three hour period. Wait times for individual passengers went up more than the rush hour as can be seen in Fig 2.

For the rush hour at the bus stop, we expected wait times for passengers to be higher than that at the bus terminal for rush hour and bus arrival times to be more spread out as well. Our results found something similar. Average wait time for the passengers was found to be about 11 minutes as is visible in Fig 7 while buses arrived approximately every 27 minutes as can be seen in Fig 7. A total of 4 buses arrived now within the two hour period. Wait times for individual passengers went up more than those for the rush hour at the bus station as shown in Fig 3.

Finally, for the non-rush hour at the bus stop, we expected wait times for passengers to be the highest and bus arrival times to be the most spread out. We found that the average wait time for the passengers was now about 19 minutes as is visible in Fig 8 while buses arrived approximately every 50 minutes as can be seen in Fig 8. A total of 2 buses arrived now within the three hour period. Wait times for individual passengers also went up much more than previous scenarios as shown in Fig 4.

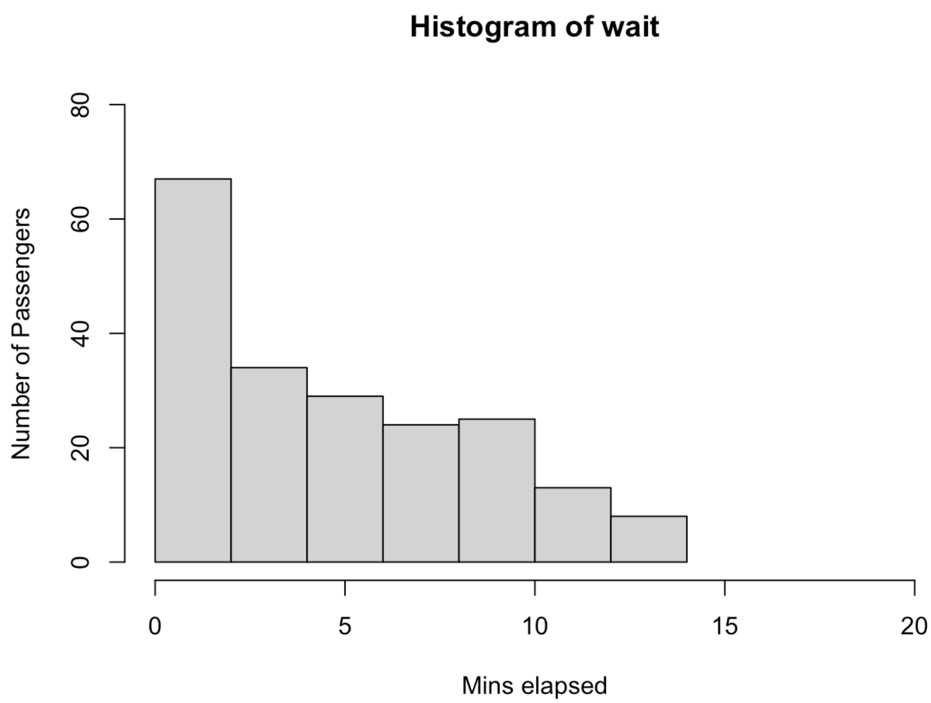

Figure 1. Histogram of a sample of wait times for the Bus Station during rush hour. The interval between the minutes elapsed for the different bars represent the wait times for the passengers.

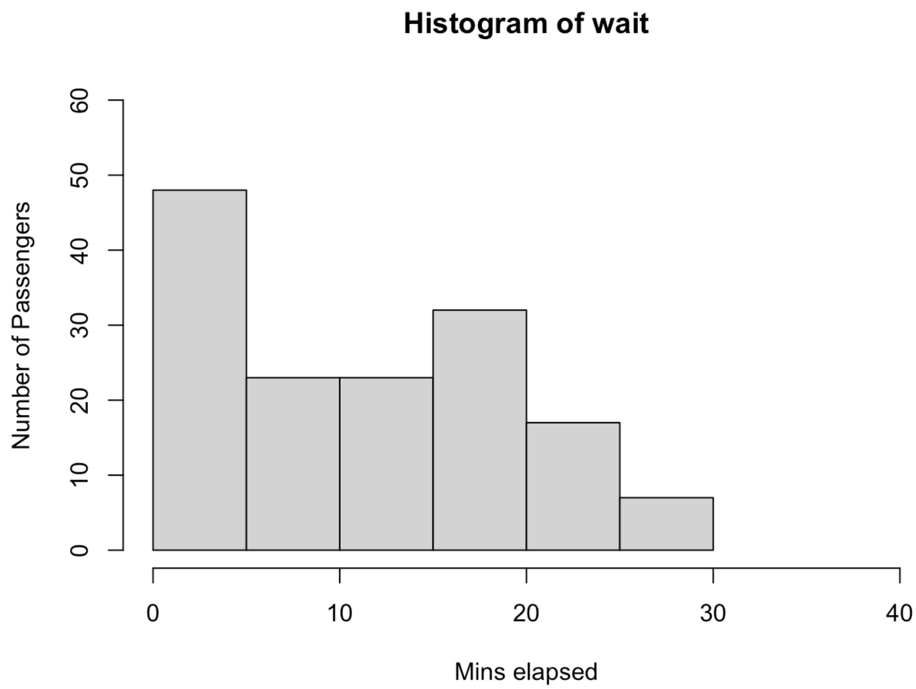

Figure 2. Histogram of a sample of wait times for the Bus Station during non-rush hour. The interval between the minutes elapsed for the different bars represent the wait times for the passengers.

Figure 3. Histogram of a sample of wait times for the Bus Stop during rush hour. The interval between the minutes elapsed for the different bars represent the wait times for the passengers.

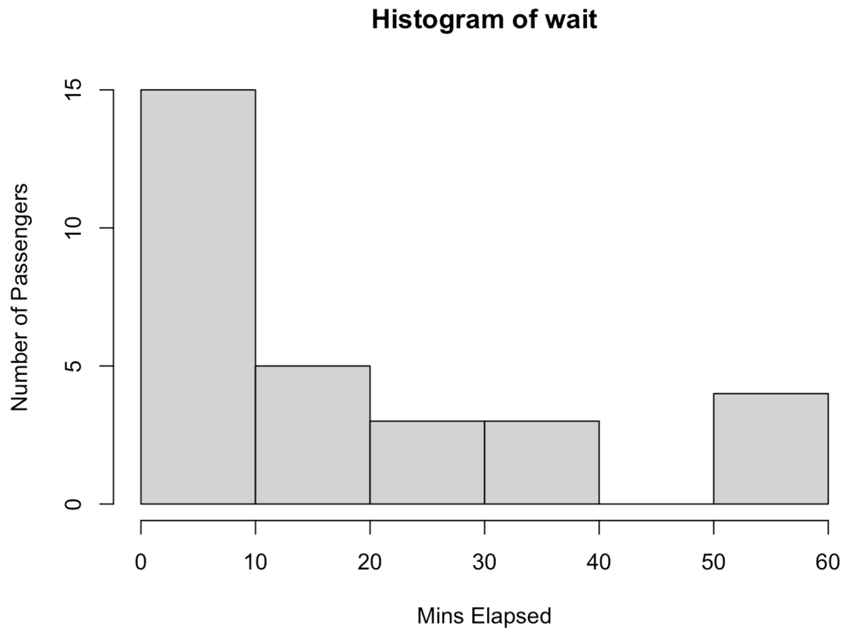

Figure 4. Histogram of a sample of wait times for the Bus Stop during non-rush hour. The interval between the minutes elapsed for the different bars represent the wait times for the passengers.

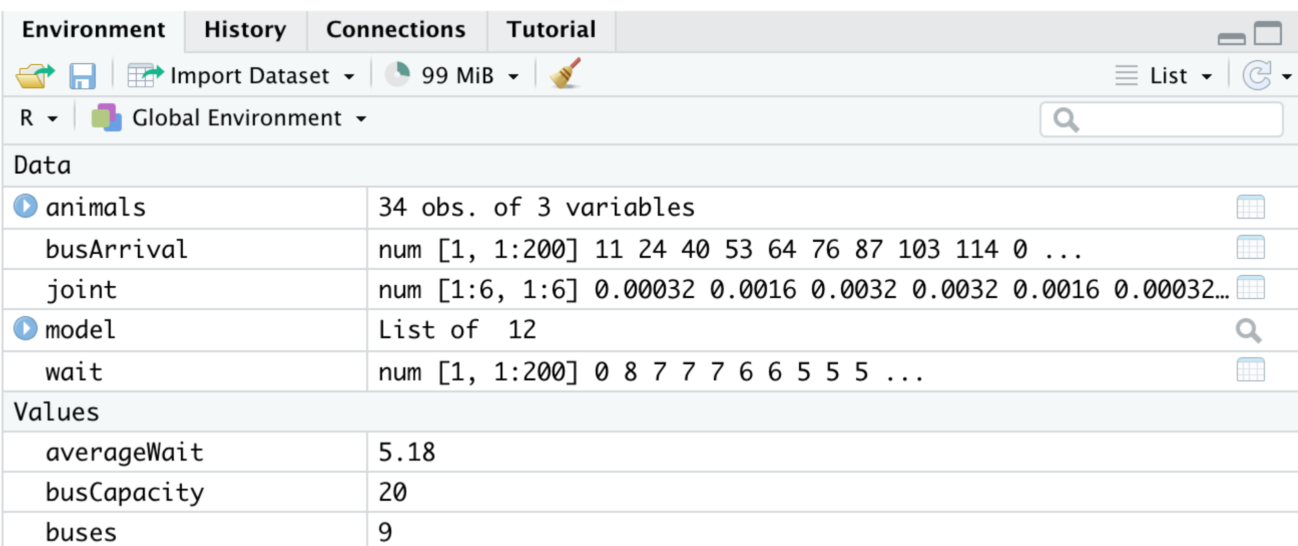

Figure 5. Results showing bus arrival times for the Bus Station during rush hour

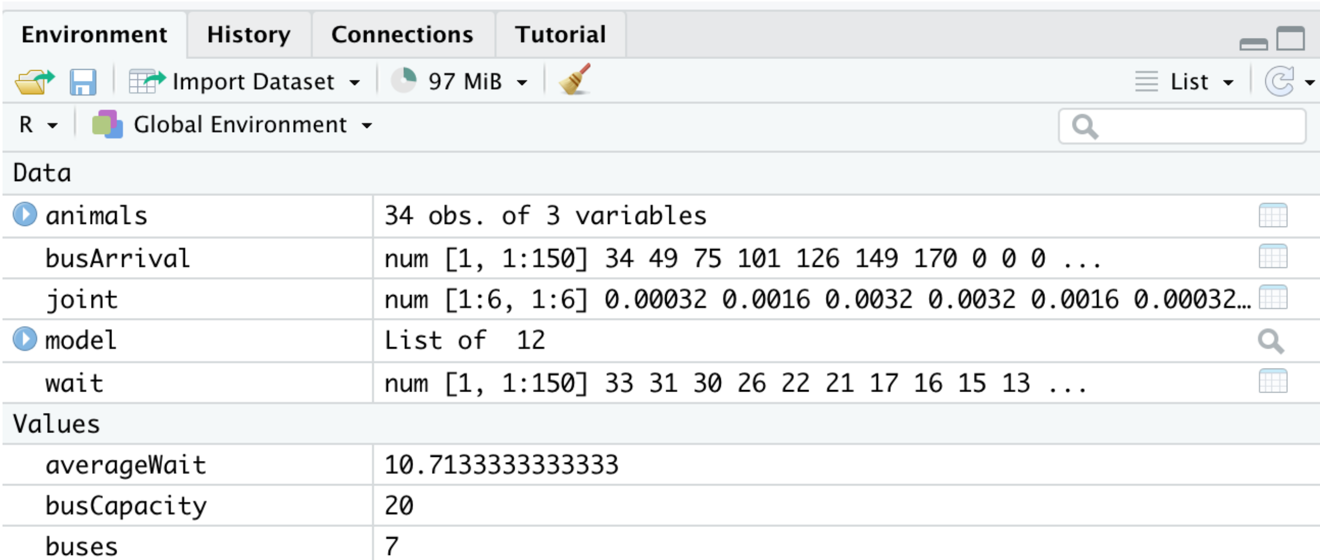

Figure 6. Results showing bus arrival times for the Bus Station during non-rush hour

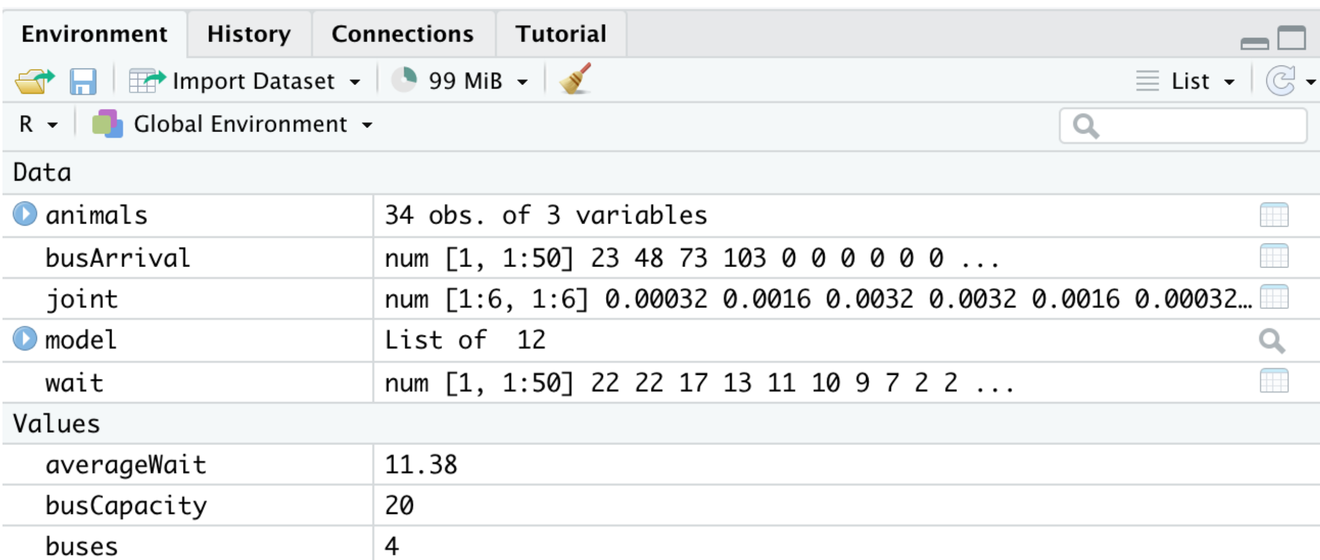

Figure 7. Results showing bus arrival times for the Bus Stop during rush hour

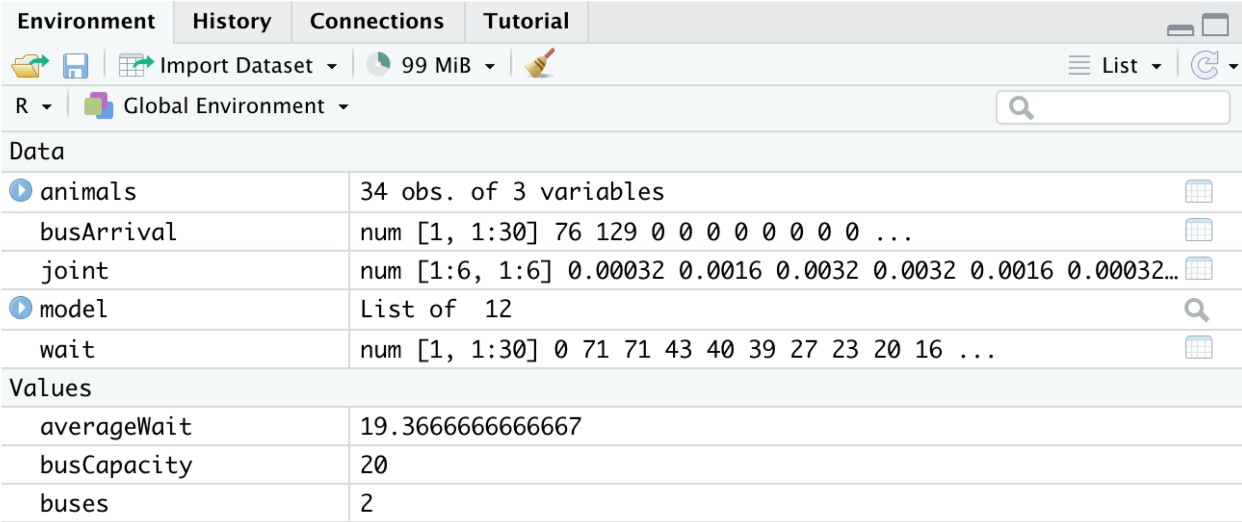

Figure 8. Results showing bus arrival times for the Bus Stop during non-rush hour

Recommendations:

Keeping our results in consideration, we would like to present different schedules to the bus line for all of our scenarios. For the rush hour at the bus station, we feel like the bus line should send in buses every 10 minutes. For the non-rush hour at the bus station, we feel it is optimal for the bus line to send in buses every 20 minutes. For the rush hour at the bus stop, we feel like the bus line would gain most by sending in buses every 30 minutes. Lastly, for the non-rush hour at the bus stop, we feel like the bus line should send in buses every 60 minutes. We feel like these schedules are optimal for the bus line because they take into account maximizing customer satisfaction while also ensuring maximum profits can be earned for the bus company.

Reflection: Anna

Overall, this case study enhanced my understanding of R programming and also challenged me to think creatively about a solution to a real world problem which was really cool. I also really enjoyed working with my peers, as collaborating allowed me to get a new perspective on the problem and also just get to know each other. One takeaway I have from this project is that collaboration with respect to problem solving definitely makes the task easier. Personally, I feel that this project was also a nice conclusion to the course as it allowed me to apply all the programming skills we have learned throughout the semester along with learning how to explain real world scenarios mathematically. This has been a great experience and will definitely help me in future classes that require coding.

Reflection: Mujtaba

I think Math Modeling was fun and I learnt a lot and a huge amount of credit for that goes to the Professor for making me interested in the class and also making it a fun experience. The course did wonders for my programming skills especially in R because this was the first time I did a project of this scale in R. I have not been the biggest fan of coding and was not excited to see it become such a huge part of this class but I have to say I enjoyed this case study the most. It gave me a chance to explore myself and while it challenged me, it also made me realize that I had the ability to take this challenge and grow from it which gave me a lot of confidence. Being able to successfully *(hopefully)* come up with an idea for this project and code went a long way in building my confidence and allowing me to trust my abilities. It was very helpful to have learnt R like this and I am sure it will be a very valuable skill to have in my arsenal for future courses and projects in my life and as I look towards getting a job. I think the case studies were a great chance to apply the concepts we learnt over the semester.

I will say that the most helpful component of this course was the luxury of having a professor who acted as a mentor and was always willing and available to help. The thought of not being afraid to ask any sort of questions and even crack jokes with the professor has made this one of my favorite classes on campus and also led me to learn more than I ever thought I would. I could not have gone through this class without the countless hours I spent in her office trying to sometimes ask the most basic of questions while simultaneously trying to understand the most complex of concepts. The homeworks was not always a joy to do every week but kept me in check and forced me to be on top of the material which I really appreciate. All in all, this has probably been one of the best classes I have taken at this college which has expanded my skills, built my interest into similar courses to come and kept me awake and alert at a time I may not usually like being awake at.

Reflection: Sami

Personally, the project illustrated the usability of Math Modeling in daily life and how these techniques can be utilized to make life easier. Using R to collectively think about how we could come up with a bus simulation to avoid overcrowding and increase overall efficiency showed me the relevance of this class in general. While getting started was very challenging, having groupmated to work with and office hours was super helpful.

My biggest lesson from this project and class in general is that I do not have to have it figured out but. Sometimes that meant requesting an extra session on Friday afternoon or maybe just listening to fellow students. Asking questions and not being afraid to do so has undoubtedly been my biggest saving grace for this class. Additionally, helping other students where they were stuck was fun and I got to closely interact with other classmates this way.

All in all, taking this class has been a great experience for me. Apart from separable equations and nullclines, I have learnt to take it a step at a time. To just start. For me, this is what Math is about, the willingness to try. Mathematically, I would like to think I am more adept in techniques of mathematical modeling now more than I was at the beginning of the semester and I hope to utilize some of those skills in my data analytics internship this summer and for future classes as well.