UHS COVID Testing Simulation Final Report

Team: William Wijaya, David Shawpindo, Reynard Sebastian.

Problem Description and Motivation

Our project will focus primarily on simulating COVID testing environments run by the University Health Services at UC Berkeley. We are simulating two different COVID testing environments: Surveillance test in Rec Sport Facility (RSF), and Symptomatic and Exposure test in Durant side. The goals of the project are to: (1) better understand the design of the current COVID testing environment, (2) identify and address any inefficiencies in the system that contribute to longer wait times in the physical testing environments and/or the process needed from the point of the test to getting the result to the client, and (3) optimize the systems for minimal risk.

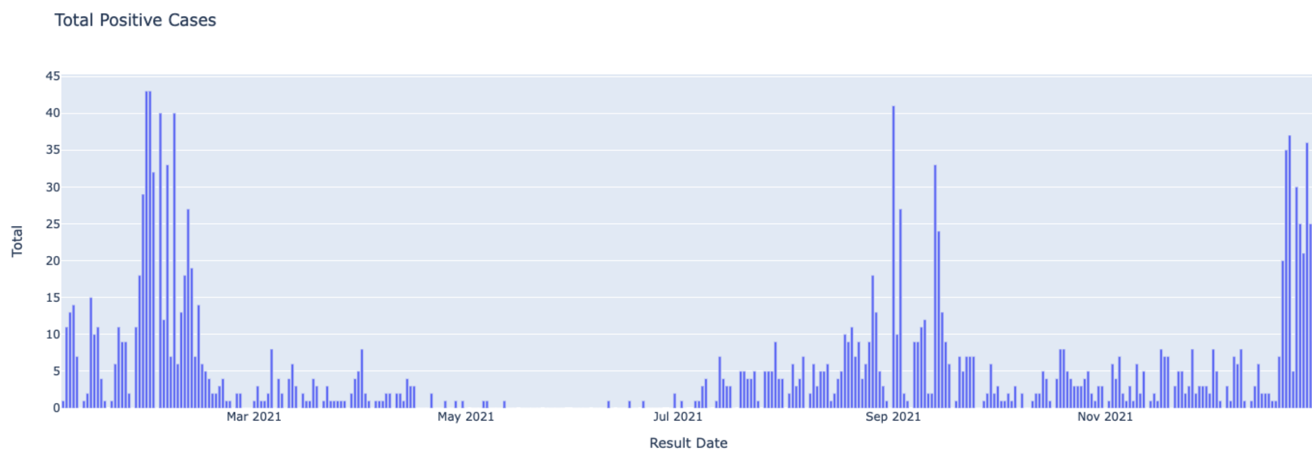
The goal of the above is achieved if we can reduce the total time the person in the system is less than 15 minutes. We limited the staff in RSF and Durant side to be 15 and 5 respectively. This is to minimize the risk of COVID exposure for both students and staff.

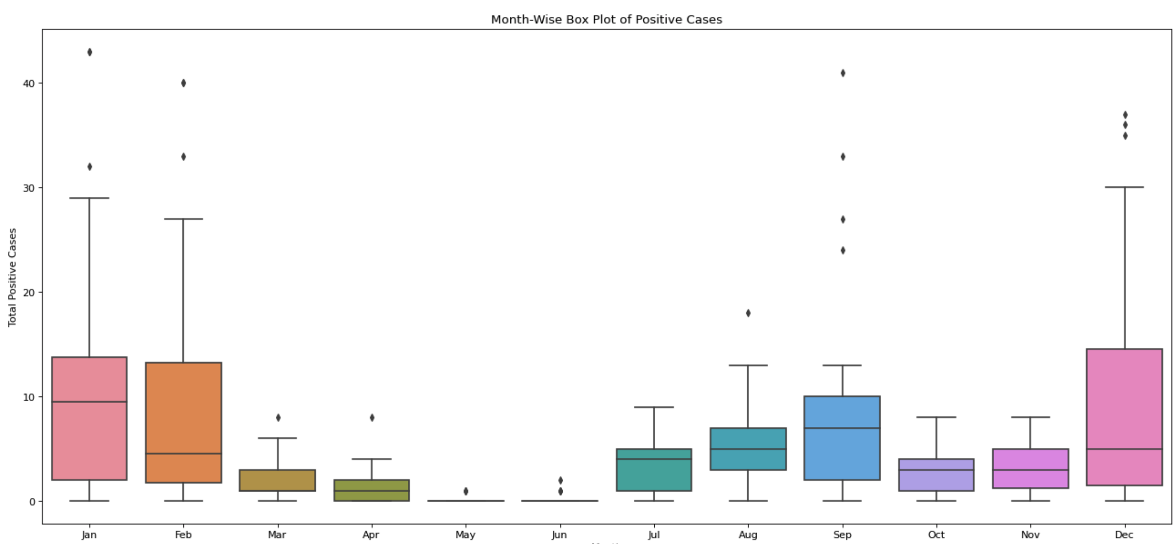
We will perform sensitivity analysis on the amount of staff to better understand how the total time the person in the system changes with the amount of staff.

Data Construction

The first step to achieve the project goal is to obtain data related to the COVID testing requirement in both locations. We asked for data through emailing the UHS that handles data dashboarding. The data contains the amount of COVID testing and average days to result (in days) for each patient type (Faculty/Staff, Graduate Student, and Undergraduate Student) and for each type of testing (Surveillance, Symptoms, and Exposure). The data also records how many positive cases for each date from each patient type and each test type. The timeframe of the data is from January 1, 2021 until October 20, 2021. Also, the surveillance test starts from January 1, 2021, and the Symptoms and Exposure test starts from January 5, 2021. Because of that, we decided to cut our time frame from January 5, 2021 to October 20, 2021. It is important to note that Surveillance testing closes on Saturday and Sunday, while Exposure and Symptoms tests open on all days.

In our simulation in Python, we decided to split our simulation into two parts: day on weekdays and day on weekend. The reason for this is Surveillance tests are not available on weekends and there are less people taking symptoms and exposure tests on weekends. According to the data and numbers provided, we used time series analysis to find the predicted number of covid cases from October, 21 2021 until December, 31 2021 using all the past data. The graph below shows the bar plot and boxplot of the total number of positive covid cases from January 5, 2021 up to December 31, 2021. Note that in the time series model analysis, we implement a breakout model where the positive covid case can increase dramatically at a certain time, in this case our model decides that this happens at the end of the December.





Then, from the 1 year data of the number of COVID cases in UC Berkeley, we can predict the number of people who take the test for each test type on each date. For this section, we only used the data after fall semester class began. This is because we want to model the future (October 21,2021 until December 31,2021) using only present data (data after fall semester class began). Present data and past data are different because:

- For past data, Berkeley conducted the class remotely. Hence, we assume that only a few Berkeley students are in the Berkeley area.
- The way that they conduct the test on Saturday and Sunday after the fall semester class began is different compared to the before fall semester class began.

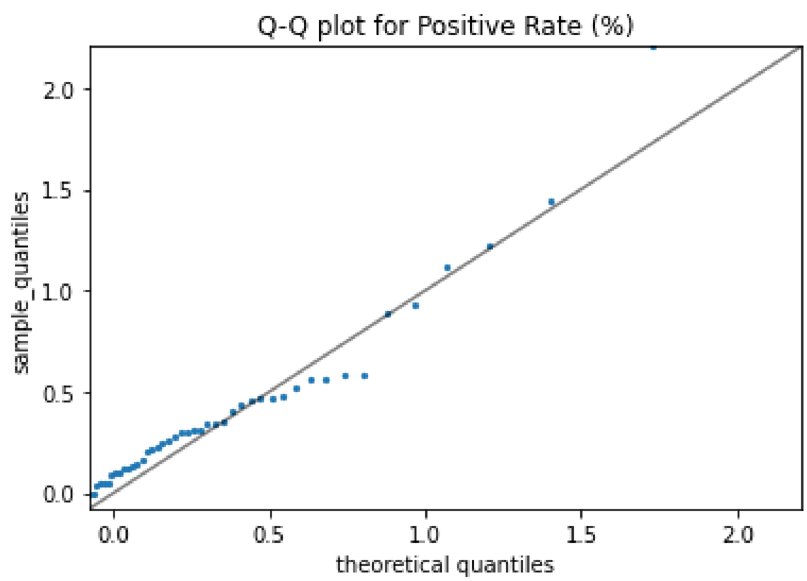
To predict the number of people who take the test for each future date, we used this simple formula:

Tested Positive = # Number of test * Positive rate (%)

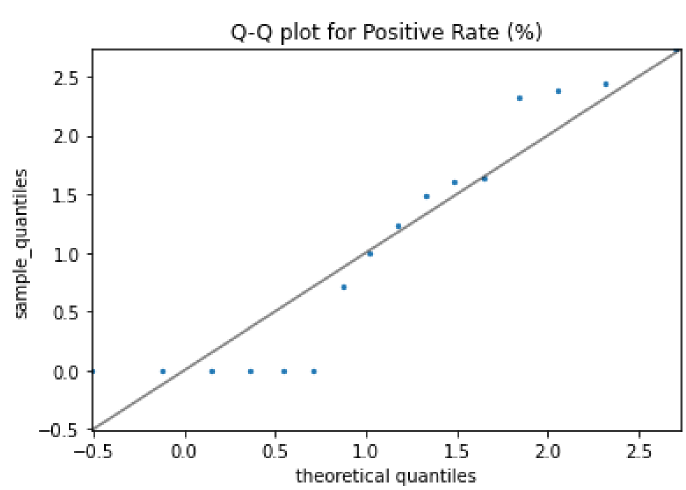
Number of test future = # Tested Positive future / Positive rate future (%)

In this case, we already know the # Tested Positive Future based on the time series analysis above. We want to predict the distribution of the Positive rate using the Q-Q plot for weekdays and weekends. Then, we can find # Number of test future.

The distribution of positive rate in weekdays is exponential with mean = mean of entire positive rate weekdays data - 0.5 and standard deviation = std of entire positive rate weekdays data + 0.05. The Q-Q plot is below:

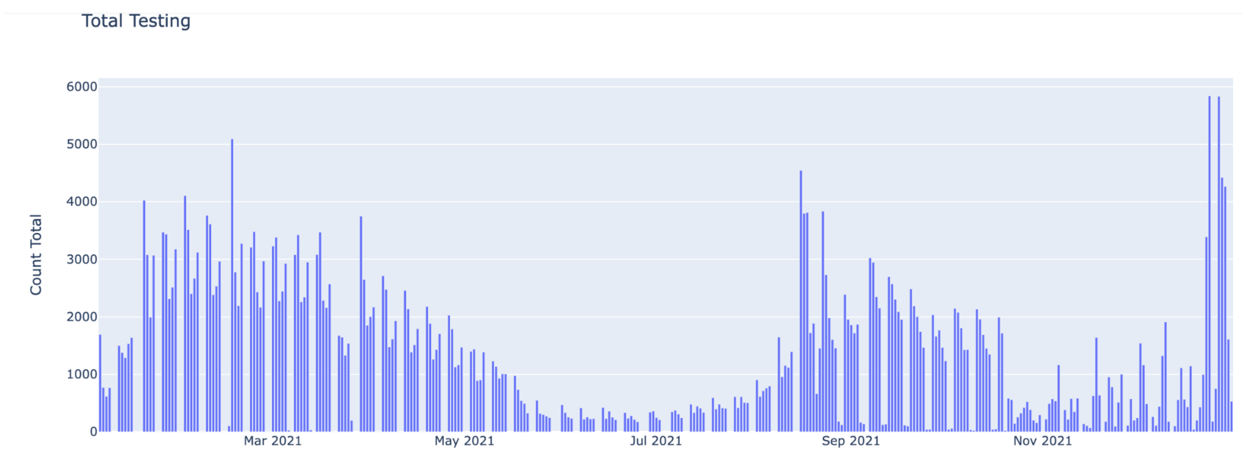


The distribution of positive rate on weekends is normally distributed with mean = mean of entire positive rate weekend data and standard deviation = std of entire positive rate weekend data. We don't have enough data to deduce much from the Q-Q plot, so we assume that the distribution is normal. The Q-Q plot is below:



Note that on both of the distributions above,we restrict the positive rate to be minimal 0 and maximal to be maximum of the entire positive rate weekend/weekdays data.

After that from the statistical analysis (using Q-Q plot), for weekdays, we decided to implement our model that 90% go to the surveillance test, 7% go to the symptoms test, and 3 % go to exposure test. Also, From the statistical analysis, for the weekend, we decided to implement our model that 0% go to the surveillance test, 72% go to the symptoms test, and 28 % go to exposure test. After finding the number of people getting tested at each location, we plotted the total number of people getting tested from January 5, 2021 until December, 31 2021:



To conclude, we have the data containing the amount of people having surveillance tests, symptoms tests, and exposure tests for each day from January 1, 2021 until December 31, 2021. We are simulating two different COVID testing environments: Surveillance test in Rec Sport Facility (RSF), and Symptomatic and Exposure test in Durant side. Hence, on RSF layout, we will use the data on the amount of people having surveillance tests. On the Durant side layout, we will use the summation of the amount of people having symptoms test and exposure test. We will simulate using only the future data (from October 21, 2021 until December 31, 2021) and suggest the amount of staff in each layout that can be implemented in the future.

Model Structure & Design Decisions

We decided to use Simio to answer our project objective that we previously discussed in our problem statement. The model in this section will be designed as close as to the real life situation. The COVID testing layout will follow the following process:

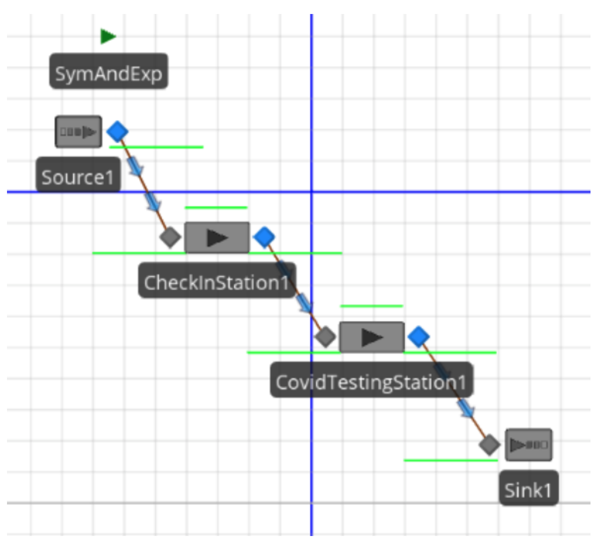
1. A person queues in the check in station queue.
2. Next, the person enters the check in station and is checked in.

3. After the check in, the person lines up at the COVID testing station and waits to be called.
4. The person get tested on the COVID testing station
5. The person leave the system

Both of the layouts will open at 8:00 am and close at 5:00 PM. Note that there is a lunch break at 12:00 am until 1:00 PM. Hence, assume that the system will operate 8 hours / day (from 8:00 am to 4:00 pm) and the reason will be explained soon.

Note that we don't have inter-arrival data nor service time data. The UHS that handles data dashboarding does not want to share row-level data due to privacy reasons. For the inter-arrival data, we will use the rate table, where the table starts at the first day at 12 midnight, the interval size will be 8 in hours and the number of intervals is 216 (72 days of simulation * 24 hour / day * 1/8hours = 216). The reason why we choose the simulation to run from 8:00 am to 4:00 pm is because each day in the rate table contains 24 hours, 24 is divisible by 8, therefore the interval size can be set into 8 hours. Then, the rate (events per hour) outside the working hour is zero (12:00 am until 8:00 am and 04:00 pm until 12:00 pm), and the rate (event per hour) inside the working hour is the amount of people that come to the site per hour based on the Data Construction section. Hence, the data based on the Data Construction section is assumed to be the average amount of people that come to the site. Since each row of the data contains the amount of people per day, we divided the amount of people by 8 and then input it into the rate table since the rate table is expressed as events per hour. Note that, we can only change the rate table based on the interval size and number of intervals only (we cannot change the time manually), that is why we make this model simplification.

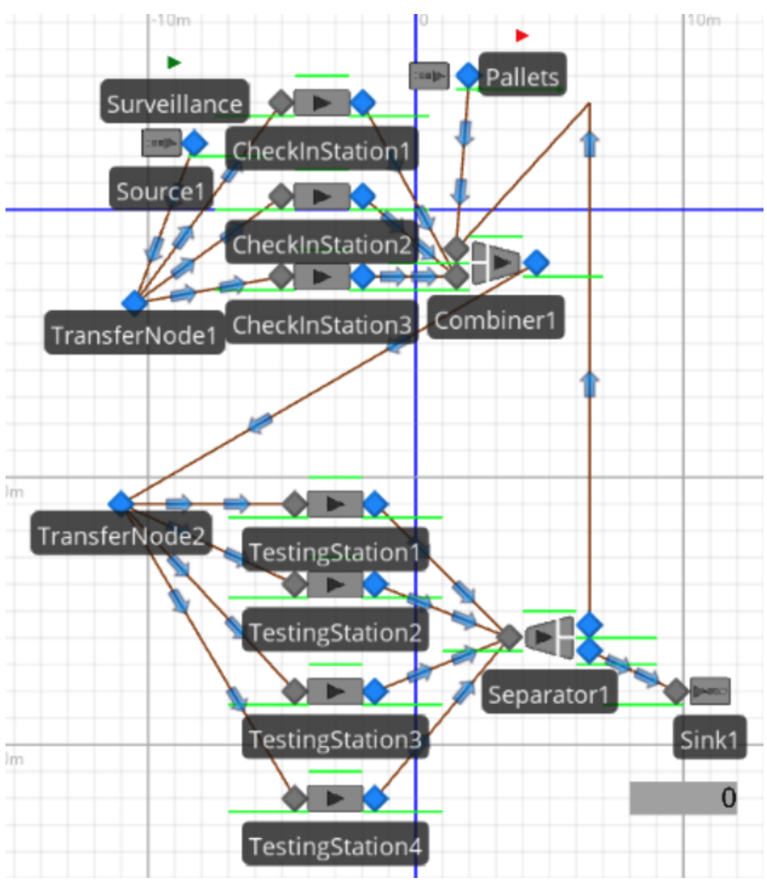
Next, here is a screenshot the layout in Durant Side in Simio:



Let the staff be the server. There are two types of server: check-in station and COVID testing station. For the service time in Durant side, check-in station service distribution is assumed to follow a triangular distribution with min = 30 seconds, mode = 45 seconds, and max = 60 seconds. COVID testing station is assumed to follow triangular distribution with min = 2 minutes, mode = 3 minutes, max = 4 minutes. Both assumptions are based on our observation in the testing site.

The initial capacity (number of stations) of both servers are set to be an integer property. This means we have two control variables in our experiment. Remember that the constraint is that the number of staff must be less than 5. This constraint will be incorporated into the Simio. The length of the queue in the COVID testing station is set to be equal to the initial capacity of the COVID testing station (this is close to the real life situation in Durant Side). Let the walking time from check in station to covid testing station be 5 seconds and the walking time between the covid testing station to sink to be 5 seconds (based on our observation in the Durant Side).

This is the screenshot for the layout in RSF in Simio:



For the service time in RSF, check-in station service distribution is assumed to follow a triangular distribution with min = 30 seconds, mode = 45 seconds, and max = 60 seconds. COVID testing station is assumed to follow triangular distribution with min = 3 minutes, mode = 4 minutes, max = 5 minutes. Both assumptions are based on our observation in the testing site. In addition, here is the process of how person goes through the system:

1. Person queue on the line before check in.
2. A person chooses a check-in station based on the length of the check-in station queue (choose a shorter queue). Each check-in station can hold up to 3 people in a queue.
3. From the queue, one person enters the check-in station and is served.
4. After check-in, the first person will choose the queue on the COVID testing station if the queue for that counter is vacant. Each station can queue up to 5 persons. If a person already line up in the queue of certain station, then the following 4 persons must queue in the same station (i.e., the 5 person queue must be filled before proceeding to the next counter). Also, the first person will line up in a station that has the minimum waiting time.
5. 5 people in the queue enter the COVID testing station and are tested. All 5 people finish at the same time.
6. After COVID test, the person leaves the system.

Note that one station is operated by one staff member. We do not have any property variable in this model. We will change the number of check in stations and COVID testing stations manually, and we will conduct the sensitivity analysis manually. This is because we do not find a way to make the number of stations be the property variable in this unique scenario. So, we will have “N” separate experiments in Simio, where in each experiment the number of check in stations and number of testing stations is changed manually. The photo above is a case where we have 3 check in stations and 4 testing stations.

Let the walking time between check-in station from the entrance to the check-in station be 5 seconds, the walking time between check-in stations to the testing station be 10 seconds, and the walking time between testing stations to the sink be 15 seconds.

Process number 4 and 5 follows a batch queuing system. We found a way to simulate a batch queuing system. First, we combine 5 persons with a combiner after they leave the check-in station (it is as if they form a group of people and the first person will have to wait for the other 4). We combine the group of people using a source called “pallets”. And, we send those groups to stations on a first-come-first-serve basis. Since they will leave the system at the same time, we separate the pallet and the group of 5 people with a separator after they are finished testing at stations.

The difference on the COVID testing station between the first layout and second layout is due to the difference on how COVID testing is conducted. In the first layout, the staff guides the person how to conduct COVID testing individually. Hence, the staff does not need to have direct contact with the person. In the second layout, the staff is the one that helps the person conduct COVID testing, therefore the staff have direct contact with the person. Also, note that the amount of people tested in RSF is significantly higher than that of in Durant side. The layout in RSF is designed such that it can fit with the higher number of people that come to the RSF.

Challenges

As is with the case of building most models, there were some challenges that we encountered and had to solve through making assumptions and simplifying real world situations. The first challenge that we face is that we need to input the data of the rate table with duration of 3 months manually. In other words, we input 512 (216 * 2) rows of data from python into the Simio rate table manually. This is time inefficient, but we do not have any way to import a rate table into Simio. Secondly, we don't have any data on the inter-arrival data, and service time data. So, we have to make a lot of assumptions to fill in the data. Our assumption is based on our observation on the testing site. This means that our model is built based on assumptions, and our model will not be as accurate as the real life situation.

Experimentation & Analysis

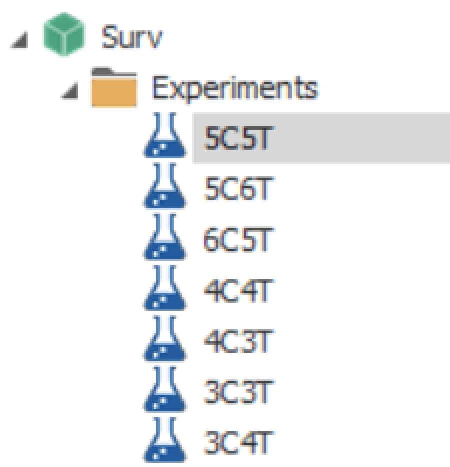
In Simio, we will conduct sensitivity analysis on both layouts by running a series of Simio experiments incorporating the constraints. Remember that we are planning to minimize the total time the person is in the system while satisfying the constraints. The picture below shows the sensitivity analysis for Durant side:

Scenario			Replications		Controls		Responses
	Name	Status	Required	Completed	CheckInCapacity	TestingCapacity	TimeInSystem ...
✓	Scenario1	Canceled	50	50 of 50	1	2	30.748
✓	Scenario2	Canceled	50	50 of 50	1	3	5.65911
✓	Scenario3	Canceled	50	50 of 50	1	4	4.30126
✓	Scenario4	Comple...	50	50 of 50	2	1	165.927
✓	Scenario5	Comple...	50	50 of 50	2	2	30.4589
✓	Scenario6	Comple...	50	50 of 50	2	3	5.59975

The sensitivity analysis above accounts for all the possible CheckInCapacity and TestingCapacity, with the total constraint of 5. The response variable is the average total time each entity in the system in minutes. If the response label is red, it means that the total time is more than 15 minutes. Hence, for red labeled response variables, the combination of CheckInCapacity and TestingCapacity is not possible because it violates the constraint.

As can be seen in the table above, the response variable changes significantly if the TestingCapacity increases (i.e., the number of testing stations increases). This means that one way to attack the bottleneck in Durant Side is to increase the number of testing stations, taking account of the constraint. First, increasing the CheckInCapacity from 1 to 2 while the TestingCapacity is 2 seems to decrease the response variable a little, which means that increasing the CheckInCapacity in this scenario is not worth it. However, when the CheckInCapacity is fixed to one and TestingCapacity increases from 3 to 4 cause a significant decrease in the response variable. Hence, we decide to choose the number of check in stations to be 1 and the number of testing stations to be 4 in Durant Side.

Now, we want to perform the sensitivity analysis in RSF. For each combination, we do the experiment separately and do the sensitivity analysis manually (i.e., removing and adding the stations manually):



We have the following combination of number of check in station and number of testing stations that we are interested to test:

- 5 Check in stations and 5 Testing stations:

Scenario			Replications		Responses
✓	Name	Status	Required	Completed	TimeInSystem ...
✓	Scenario1	Comple...	10	10 of 10	12.4994

- 5 Check in stations and 6 Testing stations:

Scenario			Replications		Responses
☑	Name	Status	Required	Completed	TimeInSystem ...
☑	Scenario1	Comple...	10	10 of 10	12.4994

- 6 Check in stations and 5 Testing stations:

Scenario			Replications		Responses
☑	Name	Status	Required	Completed	TimeInSystem ...
☑	Scenario1	Comple...	10	10 of 10	12.4994

- 4 Check in stations and 4 Testing stations:

☑	Name	Status	Required	Completed	TimeInSystem ...
☑	Scenario1	Comple...	10	10 of 10	12.3497

- 4 Check in stations and 3 Testing stations:

Scenario			Replications		Responses
☑	Name	Status	Required	Completed	TimeInSystem ...
☑	Scenario1	Comple...	10	10 of 10	13.0212

- 3 Check in stations and 3 Testing stations:

Scenario			Replications		Responses
☑	Name	Status	Required	Completed	TimeInSystem ...
☑	Scenario1	Comple...	10	10 of 10	12.9572

- 3 Check in stations and 4 Testing stations:

Scenario			Replications		Responses
☑	Name	Status	Required	Completed	AvTimeInSystem (Minutes)
☑	Scenario1	Comple...	10	10 of 10	12.4387

The reason why we only use 10 replications is because the experiment runtime is long. The response variable represents the average total time the person in the system. From 5 check in stations and 5 testing stations, if we increase the check in station or testing station by 1 , it does not affect the response variable. The reason why this is true is because there is no bottleneck that occurs when the system has 5 check in stations and 5 testing stations (we checked this by running the simulation and seeing the animation); hence, increasing any of the stations is not useful. When the system has 4 check in stations and 4 testing stations, there is still no bottleneck occurring on the system, so the response variable is almost the same compared to 5 check in stations and 5 testing stations. The decrease in time is only due to stochasticity and randomization. From 4 check in stations and 4 testing stations, when we decrease the check in station or the testing station by one, the response variable increases; hence, decreasing either of them at this point is not beneficial. When the system has 3 check in stations and 3 testing stations, the response variable increases. This is not beneficial.

In our sensitivity analysis process, the recommended amount of check in station and testing station in RSF is 4 and 4 respectively.

Conclusion

From our analysis, to minimize the total time the person in the system, while not violating the constraints, we recommend the following stations allocation to be implemented:

- Number of check in stations to be 1 and the number of testing stations to be 4 in Durant Side.
- Number of check in stations to be 4 and number of testing stations to be 4 in RSF.

Do these results seem reasonable? (Are they physically realistic? Do they match your intuition?)

The average total time people spend in Durant is approximately 4.3 minutes, while the average total time people spend in RSF is 12.35 minutes using the station allocation above. This does make sense because based on our observations, usually people will spend approximately 5 minutes in Durant and 15 minutes in RSF. Other than that, based on our observation, currently there are one check in station and 4 testing stations in Durant side. And, there are 4 check in stations and 4 testing stations that are usually active in RSF (there are more stations in RSF but only some of them are active all time). Hence, our result is approximately in accordance with the real life situation.

What did you learn through implementing the project? Any major takeaways?

We definitely learn a lot through doing this project and trying to figure out real world solutions to real world problems. One of the major takeaways from doing this project is that turning the problem into a model and actually simulating it with different factors and finding out the best and most efficient solution to it. Also, being able to make rough measurements on demand as assumptions to be put into the model and also finding out how much difference it makes. I think it is important to have the skill sets taught in this class in the future as it may help us analyze data faster.

What are "next steps" in the project (if you were to continue working on it)?

The "next steps" in the project will definitely range in a multitude of ways. One of the things that our group talked about is not only in terms of the covid testing site, but also work on UC Berkeley's vaccination site. As we know right now, UC Berkeley's offering covid vaccinations and flu shots also through its UHS services. If we were to mention the "next steps" to this project, we would start from finding the most effective way to give out vaccines or flu shots to the students while keeping the danger low. Then it can range to various different industries that relate to these types of lines.

How does your project fit into the large context of the class and the IEOR field?

Our project fits into the large context of the class because it is simulating a system with lines and also designing it to find out the most efficient way to implement staff or workers without changing a lot in terms of the waiting time. In our case, we only analyzed a small issue, which is the covid testing sites in UC berkeley. However, since our project is focused mainly on covid testing sites in Berkeley, it's impossible to say what can be done next using our analysis from this project. With our analysis, we can make more efficient decisions in implementing staff or workers in any sites like ICUs, concert check ins, airport traffic and all other lines in multiple various industries.

Video

Here is the link to the video:
https://drive.google.com/file/d/1SRKK07y1ySWj6jtKzbt5fA2y8BXymwkD/view?usp=sharing