ORIE 4580 Final Report

Philip Ayoub (pja66), Natalie Saviola (nms237),
Jeremy Markus (jem476), and Nicole Saboya (nbs57)

Fall 2021

1 Executive Summary

Helicopters are essential when an emergency patient needs immediate attention and transfer to a medical facility but are also a valuable commodity, so it is crucial to optimize their utilization across Upstate New York. In this report, we explore the different factors contributing to a helicopter's response time and effectiveness in facilitating the treatment of patients. Our simulation model accounts for many external variables, including whether conditions are safe for helicopter flights, severity of injury, and time for flight delay and preparation. Ultimately, we provide our recommendations for the number of helicopters to employ and at which locations across the region they should be placed based on our measures of average response time, response fraction, and utilization.

When all of the hospital bases are accessible, we believe only 10 helicopters should be used in total. This is because there is not a statistically significant difference in average response time between 10, 11, and 12 helicopters and minimizing the number of helicopters in use also reduces operating costs. These 10 helicopters should be distributed as follows: 1 in Albany, 2 in Syracuse, 1 in Rochester, 1 in Elmira, 1 in Ithaca, 1 in Buffalo, 1 in Utica, and 2 in Watertown.

Additionally, to avoid excessive costs, we also investigated where helicopters should be placed when just five bases are available. We used a clustering approach to perform a cursory overview of helicopter coverage range. This allows us to visualize where helicopters stationed at various bases are able to fly to, and we can quickly identify the bases that provide the most total coverage of Upstate New York. The best locations for these five bases are in Buffalo, Rochester, Ithaca, Watertown, and Albany. Again, we believe only using 10 helicopters is the optimal solution because of negligible difference in our metrics of interest as the number increases from there. To be concrete, we believe the helicopters should be distributed as follows: 3 in Albany, 2 in Rochester, 1 in Watertown, 3 in Ithaca, and 1 in Buffalo.

2 Problem Description

Currently, there are twelve helicopters in operation that transport emergency patients from the scene of an incident to a hospital in the upstate New York area. Our goal is to optimize this current operation by determining how many helicopters are needed and where they should be based to minimize the time it takes to arrive at the location of a scene after the emergency call was received. We will do this by simulating the process from when a call is placed to the helicopter dispatch (HD) to when the helicopter arrives at the receiving facility (hospital) with the patient. To do this, we have considered three factors: if it is safe for the helicopter to fly from when the call was received, if there are helicopters available from when the call was received, and if the flight will be cancelled when the helicopter is en-route. Additionally, since ambulance crews want to live in areas with reasonably high populations, helicopter bases are being considered in Buffalo, Rochester, Elmira, Ithaca, Sayre PA, Watertown, Syracuse, Binghamton, Utica, and Albany. We will perform the simulation under two different scenarios:

1. All hospitals have helicopter bases

2. Only five hospitals will have helicopter base capabilities

When performing these two analyses, we will be tracking the percentage of calls that were deemed safe to fly and, thus, dispatched; the average time taken to arrive at the scene of an emergency from when the call was received by the HD (response time); the time taken to arrive at a receiving facility from when a call was received by the HD (time to definitive care); the fraction of calls received at the HD where a helicopter arrives at scene (response fraction); and the utilization of the helicopters.

3 Modeling Approach and Assumptions

3.1 Simulation Model

Simulation is a powerful tool that can be a great aid for developing policy but only if executed correctly. When building our model our team had two overarching areas of focus: *Quality of Randomness* and *Quality of Emulation*.

3.1.1 Quality of Randomness

Allows the assumption that there exists a perfect model that generates answers to all the questions of interest and emulates the environment to a 'T'. If the designer is not careful with the data they use and how they use it, then this model will not only be worthless, it will be dangerous. Assuming answers that have been formulated on 'faulty' assumptions can lead to various issues for both the model and the client. As a team we went to great lengths to ensure that this was not the case for our model. We used a subset of data provided and leveraged its power by finding the key variable distributions and generation parameters. All this may seem rather technical but to us it means we can be sure that we are generating the independent, correct numbers with no correlation to incorrect assumptions. We formulated numbers for 5 main facets of the model:

- Call rates
- Call time
- Safe to fly
- Cancel delay
- Time on Scene

Each of these numbers were generated by using the data given paired with a number of data science techniques to derive key statistics that would let us simulate random numbers. These steps ensure that we are not only generating numbers that match the data, but are representative of the larger set of calls that may occur, both in the past and in the future.

3.1.2 Quality of Emulation

”Quality of Emulation” represents a more concrete idea: It describes the overarching model, and the various steps it takes to represent the ”real world” scenario in question. Allow me to walk you through the different stages of our model. Before the model begins, the user must input 4 parameters:

1. How many helicopters are there?
2. Which hospital locations have helicopter bases?
3. How many days should the model simulate?
4. How many different helicopter location permutations should the model try?

This allows the same code base to simulate multiple scenarios in terms of helicopter availability and base location. Once given these input parameters, our model begins to simulate how the helicopters interact with the environment and records important statistics. The following 'decision points' ingrained in the model were learned through comprehensive research interviews with helicopter pilots and base technicians.

Step 0: Initializing a time variable The model works by keeping track of the current 'time' of the simulation. This value allows it to know — at all times — when helicopters are not available and when they should be placed back into rotation.

Step 1: Helicopter Locations There are 12^{13} possible helicopter permutations that could exist. Simulating all of these is an unrealistic request. Thus, in order for us to cut down this massive number, the model randomly generates helicopter permutations to test. Since we are simulating trials in the hundreds, our team feels comfortable that a proper solution will shine through.

Step 2: How many calls per hour? The unit of time used in the simulation is hours. Thus, based on the information derived for the average amount of calls per hour, we randomly generate how many calls will happen each hour of the day.

Hour	Rate	Hour	Rate	Hour	Rate
1	0.723	9	1.70	17	3.56
2	0.618	10	2.75	18	3.20
3	0.374	11	3.19	19	3.05
4	0.319	12	3.72	20	2.28
5	0.266	13	3.92	21	1.40
6	0.277	14	3.67	22	1.17
7	0.489	15	3.81	23	0.881
8	0.777	16	3.68	24	0.813

Step 3: When was the call received? Once we know how many calls will happen per hour, we can simulate at what time during the hour a call will take place.

Step 4: Location of Patient In order to process a call, the *longitude* and *latitude* of the patient must be determined. This is executed by randomly selecting an instance from the given data set and using the *longitude* and *latitude* provided.

Step 5: Safe to Fly? This was a key step that was illustrated to us through our market research. If it is not safe to fly, the call is logged and we move on to the next instance; else, the simulation proceeds.

Step 6: Helicopter Available? This was a another key step that was illustrated to us through our market research. If there is no helicopter available due to either distance (180 km) or availability constraints, the call is logged and we move on to the next instance. Otherwise, the closest available helicopter is selected and sent on route to the scene.

Step 7: Cancel Delay This is the final stage in which a helicopter can be canceled. This decision is again simulated through the process described above. If a helicopter is canceled, then it is sent back to its base of origin; else, it proceeds to the scene.

Step 8: Scene Time Once the helicopter is on scene, the model simulates how long it spends on scene tending to the patient.

Step 9: Hospital After the helicopter and medics finish up on the scene, the model determines which hospital to take the patient to. There are 2 possible scenarios:

1. A 'major' hospital is closest: Sayre, Albany, Syracuse, Rochester, for which it goes directly there.
2. A 'normal' hospital is closest for which the model makes a decision on whether to send it to the closest hospital or travel the extra distance to the closest 'major' hospital if one is in range. This decision simulates some patients needing specialized care only available at 'major' hospitals.

Step 10: Hospital Delay The transfer of a patient is not simultaneous, and thus the model generates how long the helicopter waits at the hospital until returning to base.

Step 11: Helicopter Return Once the patient is dropped off at the hospital, the model determines the time back to its base of origin (distance/speed) and once past, puts the helicopter back into circulation.

After this last step, successive calls are simulated until the number of days simulated reaches its set value. Then a new helicopter permutation is simulated. Each call is being processed as if all the results are known the instance the call is received. Thus, this allows us to simulate calls that overlap and take place while other helicopters are out on call. The model then picks the best permutation of helicopter distributions based on which minimizes 'Average Response Time.' Overall, this is the high-level view of our model.

3.2 K-Means Add-On

The above describes our core-simulation engine. However, in order to tackle the second question, Which '5 hospitals' should be used as hospital bases, we had to get creative. This is due to the fact the simulation space for this question is 252 times larger, 10 choose 5 possible permutations, then the original question of where the helicopters should be located given all 10 locations. Thus, we used this thoroughly documented data science technique to allow us to eliminate the meaningless answers (ie. No helicopters at major locations) and intelligently predict which hospitals would be most beneficial to Upstate New York if they contained bases. More detail on this technique will be described below.

4 Data Analysis

4.1 Overview

To calculate the input data distributions for the simulation model, the data of the existing operation was utilized. The provided data set allowed for the calculation of unknown input distributions while some distributions were given. The parameters of the distributions that were not given were calculated using the observed data and methods of data analysis. In the simulation model, random numbers from those distributions were generated.

4.2 Call Rate and Time

For call simulation, the interarrival times of calls were randomly generated after calculating the call rate of each hour of the day as shown in the table in section 3.1.2.

4.3 Call Location

To generate a call location, an observed call location was randomly selected from the provided data set. A key aspect we added to introduce more 'randomness' into the system is a location jitter. For every randomly selected call, the probabilistic distribution we used allowed for a 50% chance of a shift in the call location. The location jitter has the potential to shift the call location by a maximum of 1 km in one of the following directions with equal probability: north, east, south, west, northeast, northwest, southeast, or southwest. The decision to bootstrap from the observed call locations was driven by the intuition of potential caller locations. In any state, the population density varies based on town or city or general region. In this context, assuming every point in the geographical region of analysis has equal probability of being the origin point of a call would be naive due to the natural variation observed in population density maps. By bootstrapping from the observed call locations we can run a simulation that more accurately considers the demand of different regions for helicopter assistance based on how densely it is populated. Regions with a higher density of residences will have more observations in the data set that the call locations are randomly selected from and the inverse is true for regions with low population density. The decision to add the location jitter was to create more randomization in the simulation to mimic the randomness observed in real life while still maintaining the observed regional differences in population density.

4.4 Helicopter Times

When a call is received, assuming that there is an available helicopter, there is a given distribution of how long it takes for flight safety to be determined as well as a probabilistic distribution of whether the 'safe flight' conditions are met. If it is safe to fly, there is a given distribution of how long it takes the crew to prepare to dispatch. Once the helicopter is in the air, the travel speed is 160 km/hr and a helicopter can respond to calls within a 180 km radius of its base. The travel time to the scene varies on the distance between the helicopter base and call location, but the speed of flight is a constant rate. While in the air there is the potential for the helicopter dispatch to be cancelled. The risk of call cancellation is dependent on whether the simulated cancellation time is less than the travel time for the helicopter to arrive at the scene. The time when the helicopter is cancelled has an exponential distribution with a constant rate. If there is no cancellation, then once the helicopter arrives on scene, there is a distribution of how long the crew is on site. After being on site, the helicopter crew must transport the patient to a hospital. The potential hospitals a patient can be transported to are in Buffalo, Rochester, Elmira, Ithaca, Sayre PA, Watertown, Syracuse, Binghamton, Utica and Albany. These are also the locations of the potential helicopter bases.

4.5 Hospital Selection

If the closest medical facility is one of the major trauma centers (Rochester, Syracuse, Albany, or Sayre, PA), then the patient is transported to that hospital. Otherwise, the patient is transported to the closest medical center with probability 0.807 or the patient is transported to the closest major trauma center with probability 0.193. Transporting a patient has a travel time based on the distance between the call location and the medical center the patient is taken to. At the hospital there is a distribution of how long the helicopter crew is at the hospital before leaving and returning to the helicopter's base location.

5 Model Verification

5.1 Case 1: 1 Base, 1 Helicopter

To ensure the validity of our model and verify the reliability of its results, we ran our simulation under the simplified case where there is one helicopter available and one possible base. Under this condition, we further analyzed our response time. To better understand if the response time made sense, we calculated the average distance between that base and the call locations that were reachable by that base and not closer to any other bases. Under the assumption that the helicopter travels at 160 km/hr, we found the average time it took for the helicopter to fly to the scene of the accident, assuming that the helicopter was able to fly to all of the bases. However, it is not the case that the helicopter was able to fly to every scene due to a helicopter not being available or because of unsafe conditions. To account for this, we further added to this example case by simulating if it was safe or unsafe for the helicopter to fly by using a Bernoulli distribution and adding a simulation clock to the test case. We also verified that the helicopter is available or not at that current time. To do this, we added the time it took for the helicopter to arrive at the hospital and back to the base location, changing the status of the helicopter from unavailable to available.

We then checked that the rest of our metrics were correct by further adding on to the simplified model. In the end, we had a model that showed us how one helicopter at one base would perform in a 24 hour period which we used to compare the values of our original model.

5.2 Case 2: 2 Bases, 1 Helicopter

We then ran our simulation under the assumption that there were two bases but one helicopter. Ideally, our simulation would pick the base that lowers the average response time. To check that our simulation picked the correct base for the helicopter to be located at, we simulated the response time of the helicopter for each of the bases and ensured that the simulation picked the base that gave the lowest response time in our simplified model. To ensure that this was accurate, we ran the original model in the one helicopter and one base case twice using the two different bases and checked that the average response time for the base that the simulation picked under the one helicopter, two base model was smaller than the average response time of the base that the simulation did not pick. By running the model under these assumptions and creating a more simplified simulation in which there is one helicopter and one base available and checking its results with our core simulation, we were able to ensure that the model picked the bases that would produce the lowest response times and that the model calculated the required metrics correctly.

5.3 Case 3: 1 Base in Rochester, 1 Base in Watertown, 1 Helicopter

Finally, we simulated a model with only two helicopter bases (Rochester, Watertown, as a somewhat extreme "edge case") and one helicopter. We then decided to minimize the model based on "percent dispatched." The model returned that the one helicopter should be located in Rochester, which makes sense with our assumptions. Since Rochester is more central to the other cities and has a larger cluster of cases, the model should be expected to select it over Watertown as the ideal location for the single helicopter.

Overall, these "tests" gave us confidence that our model was making the correct decisions when simulating the environment at hand.

6 Model Analysis

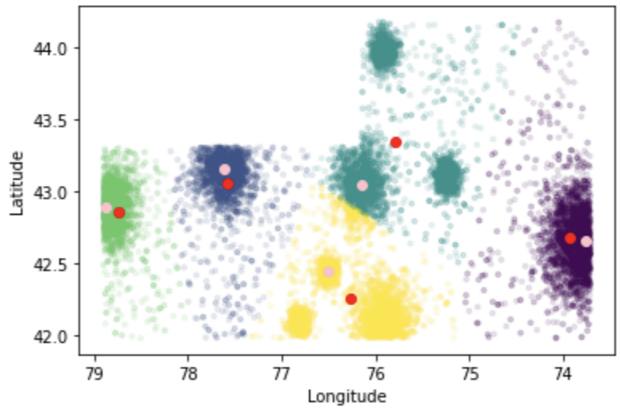
To make recommendations on what bases to distribute the maximum of 12 helicopters, the simulation model was designed to produce the following metrics: the percentage of calls dispatched, average response time, time to definitive care, response fraction, and utilization of helicopters. To optimize the base location and number of helicopters at each base, the simulation model was run with varying numbers of helicopters (1 to 12) to calculate the previously stated metrics. The decision to use the model in this manner was driven by the goal to see how each potential additional helicopter in the system affected the metrics of interest. The main goal of the analysis was to minimize the average response time it takes for a helicopter to arrive at a call scene. With more helicopters in the system, the assumption is that the average response time would decrease due to an increase in helicopter availability. Additionally, the optimal location of helicopter bases needs to be determined based on the simulation, where the distribution of helicopters to bases minimized the average response time. We constructed our simulation model by randomizing the placement of helicopters in the system via either a moderately grouped initialization, i.e. fewer bases with more helicopters, or a more dispersed distribution of helicopters. Each option has its own benefits and drawbacks. Having fewer bases with more helicopters at each could minimize the cost since the resources needed by a helicopter need to be distributed to fewer locations; however, there could be an increased risk of call locations being outside the maximum travel range of 180 km. The option to have more base locations with fewer helicopters at each would allow for more coverage of potential call locations. Naturally, it is logical to have more helicopter bases so that the majority — ideally all — calls will have a helicopter within range regardless of the call scene location.

Based on the simulation results, once there are 10 or more helicopters in the system, the average time to definite care is approximately less than one hour, which is an imposed comparison driven by the range of severity that a medical emergency can have and thus the time sensitivity of the case. The metric of primary interest is average response time, and the simulation with the minimum value is when there are 12 helicopters in the system. This result was expected. However, to it was necessary to determine if the difference in average response time between having 10 helicopters versus 12 helicopters was statistically significant. The difference between the average response time when there are 10 helicopters and when there are 12 is about 2.46 minutes with a 95% confidence interval of [-0.3444, 5.2644] minutes. Similarly, the difference between average time to definitive care with 10 helicopters versus 12 helicopters is about 2.16 minutes, with a 95% confidence interval of [-1.3029, 5.6229] minutes. Since both the confidence interval for the difference in average response time and average time to definitive care contain the value 0, there is not statistical evidence to claim that 12 helicopters in the system significantly reduces the average response time and average time to definitive care when compared to 10 helicopters. To minimize average response time, the recommendation is to use 10 helicopters and distribute them among the bases as follows: Albany - 1, Syracuse - 2, Rochester - 1, Elmira - 1, Ithaca - 1, Buffalo - 1, Utica - 1, Watertown - 2.

The central metric of average response time is reasonable to base a recommendation for the helicopter distribution on, but it is somewhat flawed. The general average response time is not weighted based on the utilization of helicopters by base. If the average was calculated with a weight corresponding to either **a)** the proportion of calls to which a particular helicopter was dispatched, or **b)** the proportion of calls to which each helicopter base dispatched a helicopter, then the metric would be improved and give an estimate that accounted for the utilization of each helicopter or each base location.

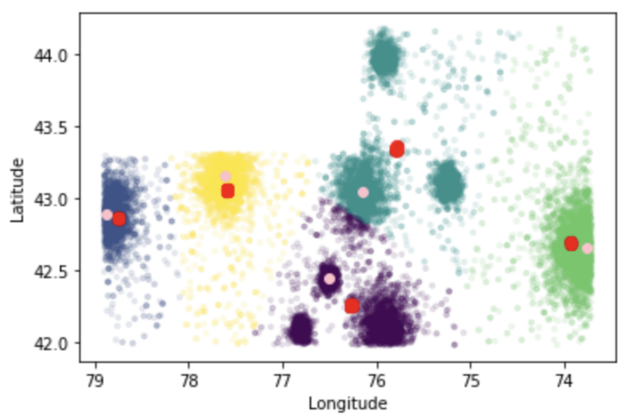
6.1 K-Means Add-On Overview

In order to determine which locations would optimize helicopter response performance when limited to just five bases, we initially performed k-means clustering. The goal of the clustering was to divide upstate New York into five distinct groups and gain a better understanding of where helicopters would need to be placed in order to be able to reach the most people with the limited base locations. First, we drew a random sample of 16,994 latitude and longitude pairs — the original number of observations — from the data (with replacement) and ran a k-means clustering analysis with five clusters. The results are shown below, with each cluster labeled by color and red dots indicating the cluster means and the pink dots representing the nearest hospital locations to each of the cluster centers. The nearest hospitals are located in Rochester, Buffalo, Ithaca, Syracuse, and Albany.



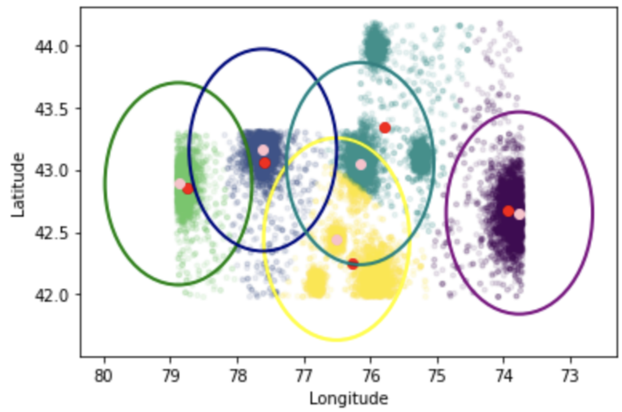
6.2 Step 0: Bootstrapping

We wanted to know how variable these cluster centers were based on the data, so we ran a 1,000-sample empirical bootstrap analysis of the k-means clustering. For each bootstrap sample, we again drew 16,994 latitude and longitude pairs from the original data with replacement. The results are shown below, with the red dot clusters representing the 1,000 different centers. As can be seen, the centers are fairly consistent. Given the context of the problem, we felt confident that the closest hospitals to our single random cluster center (above) are the same as those nearest the bootstrap clusters.



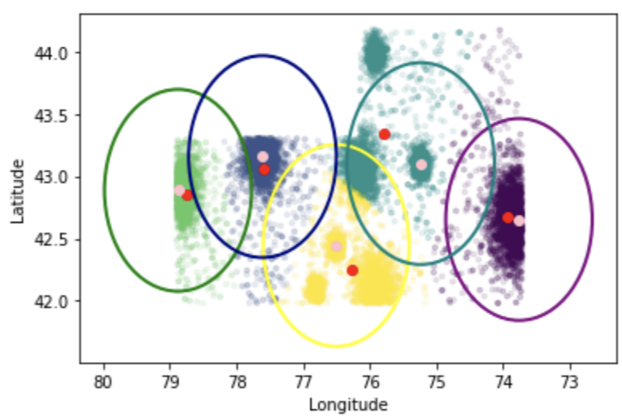
6.3 Step 1: Distance Constraint

The next step was to ensure that these hospital base locations would be able to reach the most amount of people across upstate New York. To visualize this, we plotted circles with radii of 180 kilometers (the maximum distance the helicopters can fly to respond to a call). The coverage zones appear to be elliptical due to the different axis spacing and the curvature of the Earth, but the width and height are 180 kilometers for all five zones.

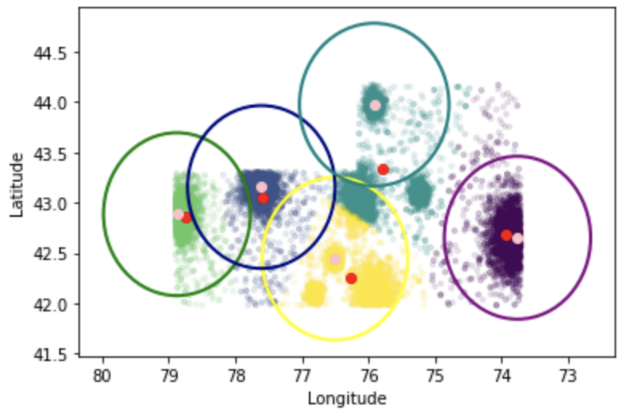


6.4 Step 2: Watertown Vs. Utica

Our initial observations from the above graph were that Watertown is almost entirely excluded from helicopter coverage and there is redundant coverage in Syracuse and Ithaca. To hopefully improve performance and utilization metrics, we moved the base from Syracuse to Utica to see if that makes a difference in the total coverage.



With this switch, there is now more coverage in the eastern part of upstate New York, but Watertown remains outside of the helicopters' maximum range. We moved the Utica base to Watertown.



Unfortunately, in this case, Utica is excluded. We then ran the model simulations with a base in Utica and with a base in Watertown and compared the results to determine the optimal base locations between the two cities. We kept the other four bases (Buffalo, Rochester, Ithaca, and Albany) the same because the coverage ranges look sufficient from the clustering analysis and plotting.

6.5 Step 3: Final Suggestion

Based on the results from the two simulation setups, we recommend that the helicopters should be distributed among Buffalo, Rochester, Ithaca, Watertown, and Albany. The numbers at each location vary and depend on how many total helicopters are available, but priority should be given to larger cities. The overall results are displayed in the tables below (Table 4).

Our decision to place helicopters in Watertown instead of Utica was primarily driven by the fact that the model tended to avoid placing any helicopters in Utica when optimizing the results. This tells us that removing helicopters from the larger cities in favor of adding them to Utica does not improve response time. However, the simulations show that placing helicopters in Watertown not only lower response times, but also generally tends to improve both response fraction and utilization metrics. Furthermore, average response time for the simulation with Utica converges to around a third of an hour as the number of total helicopters increases to twelve. When we utilize Watertown instead, the average response time is closer to a quarter of an hour as the total number of helicopters increases to twelve.

Both simulations indicate that, when there are very few helicopters available, they should be placed in Buffalo. This makes sense, since Buffalo has the largest population among the cities in upstate New York. As the number of available helicopters increases, however, the two simulations begin to converge. When Utica is an option, it appears that distributing the helicopters fairly evenly among Buffalo, Rochester, Albany, and Ithaca is the best course of action. But when Watertown is among the five bases instead, there should be an approximately equivalent number of helicopters stationed at each base.

7 Results

Table 1: **Table Key**

Statistic Name	ID	Units
Num. of helicopters	NH	Integer
Helicopters Distribution	HD	Array
Percent Dispatched	PD	Percentage
Average Response Time	RT	Hours
Average Definitive Care Time	DC	Hours
Response Fraction	RF	Percentage
Utilization	U	—

7.1 Question 1: All Hospitals have Bases

Table 2: **10 Location Index**

Sayre	Albany	Cuse	Roch	Elmira	Bing	Ithaca	Buffalo	Utica	Watertown
0	1	2	3	4	5	6	7	8	9

Table 3: **10 Location Data**

NH	HD	PD	RT	DC	RF	U
1	[0, 0, 0, 0, 0, 0, 0, 1, 0, 0]	.113	.339	1.12	.102	5.35
2	[0, 1, 0, 0, 0, 0, 0, 1, 0, 0]	.243	.407	1.55	.232	5.97
3	[0, 2, 0, 0, 0, 0, 1, 0, 0, 0]	.360	.406	1.15	.325	5.63
4	[0, 1, 0, 0, 0, 0, 0, 3, 0, 0]	.337	.400	1.15	.312	3.96
5	[0, 0, 1, 1, 1, 0, 1, 1, 0, 0]	.590	.359	1.09	.538	5.31
6	[0, 0, 2, 1, 0, 0, 1, 1, 0, 1]	.633	.353	1.11	.590	4.95
7	[1, 1, 2, 1, 1, 0, 0, 1, 0, 0]	.737	.341	1.09	.684	4.92
8	[0, 2, 1, 1, 0, 1, 1, 0, 1, 1]	.751	.317	1.06	.693	4.60
9	[0, 1, 1, 1, 2, 1, 0, 1, 1, 1]	.818	.283	1.02	.775	4.37
10	[0, 1, 2, 1, 1, 0, 1, 1, 1, 2]	.855	.254	.996	.815	4.14
11	[1, 2, 2, 1, 0, 1, 1, 1, 1, 1]	.842	.229	.977	.806	3.74
12	[1, 2, 2, 1, 0, 2, 0, 1, 1, 2]	.851	.213	.960	.812	3.37

7.2 Question 2: Five Hospitals have Bases

Table 4: **5 Location Index**

Albany	Rochester	Watertown	Ithaca	Buffalo
0	1	2	3	4

Table 5: **5 Location Data**

NH	HD	PD	RT	DC	RF	U
1	[0, 0, 0, 0, 1]	.109	.350	1.098	.097	5.00
2	[0, 0, 0, 0, 2]	.192	.399	1.180	.175	4.62
3	[1, 0, 0, 0, 2]	.198	.397	1.155	.185	3.18
4	[1, 0, 0, 1, 2]	.469	.408	1.170	.426	5.40
5	[1, 1, 1, 1, 1]	.559	.382	1.121	.509	5.33
6	[1, 2, 1, 1, 1]	.626	.383	1.136	.571	4.93
7	[0, 1, 2, 2, 2]	.629	.339	1.080	.584	4.25
8	[1, 2, 2, 2, 1]	.685	.321	1.054	.636	4.02
9	[3, 2, 1, 2, 1]	.732	.297	1.048	.696	3.94
10	[3, 2, 1, 3, 1]	.731	.275	1.019	.693	3.48
11	[2, 3, 2, 2, 2]	.745	.276	1.013	.700	3.24
12	[3, 2, 2, 2, 3]	.734	.267	1.021	.693	2.94

8 Conclusion

If we are not limited in the number of bases, we recommend there be 10 helicopters: 1 in Albany, 2 in Syracuse, 1 in Rochester, 1 in Elmira, 1 in Ithaca, 1 in Buffalo, 1 in Utica, and 2 in Watertown. As the number of helicopters increases up to 10 helicopters, we have an increasing response fraction, which means that with 10 helicopters available, it is more likely that a helicopter will arrive at the scene. Additionally, our response time decreases as more helicopters are available. However, after 10 helicopters, we do not see a significant increase in the response fraction or a significant decrease in the response time. The results of our model converge at 10 helicopters, telling us that the best option would be to only use 10 at the bases mentioned above so as to decrease response time and increase response fraction.

If we are limited to 5 bases, we recommend there be 10 helicopters: 3 in Albany, 2 in Rochester, 1 in Watertown, 3 in Ithaca, 1 in Buffalo. Similarly, we recommend these 10 helicopters because the difference in the response time after 10 helicopters becomes rather insignificant ($< 3\%$).

9 Appendix

9.1 Final Variables

Listed below are the final static variables of our model:

- **speed**: 160
- **max_distance**: 180
- **hosp_names**: Array of all hospitals
- **hosp_loc**: Array of tuples representing the Latitude and Longitude of the corresponding hospital.
- **lambCancelDelay**: The parameter to be used when generating the "Cancel Delay"

9.2 Input Distributions

9.2.1 Call Times

It is provided knowledge that the distribution of call interarrival times is a nonhomogeneous Poisson process, so it is exponentially distributed with a rate function that varies by time t, the hour of the day. To calculate the rate function, the data was subset by each hour of the day (00 to 23 in military time) and the average number of calls per hour was calculated over the whole year of data. The rate function lambda is represented by the dictionary 'call rates'.

9.2.2 Safe Flight

The distribution of how long it takes to conclude if there are safe flight conditions has the provided distribution of Triangular with a minimum of 5 minutes, an average of 7 minutes, and a maximum of 10 minutes. The probability distribution of whether the conclusion will be that the flight conditions are safe is Bernoulli with the probability of it being safe to fly, p, calculated from the observed data as the proportion of the total number of calls where it was safe to fly, $p = 0.899$. After flight approval, it is provided that the amount of time it takes the helicopter crew to prep and then depart has a Triangular distribution with a minimum of 5 minutes and a maximum of 10 minutes.

9.2.3 Cancel Delay

The rate of call cancellation for the exponential distribution was calculated from the observed data. However the observed data was censored which complicated the calculation. First, the data was subset to contain only the flights that had safe flight conditions and a helicopter available. Then the number of observations with cancel times was calculated and the number of observations without cancel times was calculated. The rate lambda is equal to the number of observed canceled flights over the sum of the number of observed cancelled flight with a cancellation time and half the number of flights where a cancellation was not observed, $lambda = 0.205$. In the code final, this calculation was completed in the 'Final Variables' section and it is 'lambCancelDelay'.

9.2.4 Scene Time

The distribution for the time the helicopter team was at the call scene was calculated by fitting various distributions to the empirical distribution of scene times, or the histogram of scene time, using the scipy package function 'dist.fit'. When the fitted distributions were overlaid on the histogram, the four distributions that visually fit the data well were the Beta, Gamma, Lognormal, and Weibull. Then QQ Plots were created for each of the four potential distributions, comparing it to the observed distribution. From the QQ Plot it showed that either the Beta or the Gamma distribution fit the date best. The chosen distribution was Beta with the parameters, based on the parameter inputs used in the 'scipy.stats' package, $a = 2.95$, $b = 11072$, and $scale = 1302.6$.

9.2.5 Hospital Time

For finding the distribution of the time the helicopter stayed at a hospital after arriving with a patient, the same method of analysis that was used for finding the scene time distribution was utilized. The same four candidate distributions were found when fitting models with the 'dist.fit' function from the scipy package to the empirical distribution of time at the hospital. From the QQ Plots created, the conclusion was that either the fitted Beta or Gamma distribution was optimal. The chosen distribution was Beta with the parameters, based on the parameter inputs used in the 'scipy.stats' package, $a = 2.90$, $b = 812$, and $scale = 141$.

9.2.6 Hospital Choice

The distribution of what hospital patients were taken to was interesting because of the given constraint that if the closest medical center to a call location is a major trauma center, the patient will be transported to said major trauma center. To calculate the proportion, p, of the patient being transported to the closest medical center (not a major trauma center) for the Bernoulli distribution, the data was subset to contain calls where the closest medical center to the call location was **not** a major trauma center. Then the proportion of calls where the patient was transported to the closest major trauma center instead of the closest medical center was calculated as *1-p*. This analysis was and the resulting value was $p = 0.807$.

9.3 Helper Functions

9.3.1 randomHeliAmounts

Input: This function takes in the number of total helicopters in the system, the location([lat,lon]) of the bases in the form of an array of tuples and names of the bases in the form of an array.

Body: The function has 2 paths it can take: The first is to bunch up the amount of helicopters in each base, it takes this branch with probability .50. The other option is to more evenly distribute the number of helicopters in the various bases. It takes this branch with .50. For each of these case once finished running, the number of helicopters in each cell of the array sum to the input parameter defining the total number of helicopters in the system.

Returns: An array of length *len(hospitals with bases)*.

9.3.2 haversine_fun

Input: This function takes in 4 parameters: Latitude of location 1, Longitude of location 1, Latitude of location 2, and Longitude of location 2.

Returns: The *Haversine* distance in kilometers (km).

9.3.3 minHD

Input: The longitude and latitude of the scene, and the available helicopter array

Body: Using *haversine_fun* this function find the closes base that has a helicopter available, and is within 180 Km. If one exists it returns its index and the distance from the base to the scene. If one doesn't exist it returns -1 for index and 5000 Km for distance.

Returns: The index of the base, and the total distance from that base to the scene.

9.3.4 minHospital

Input: The longitude and latitude of the scene, and the available helicopter array

Body: This function finds the closes hospital to the scene of the call. However, if the closest hospital is not a 'major' hospital it goes to the closest 'major' hospital with probability, '.193' as opposed to just the closest hospital.

Returns: The index of the hospital, and the total distance from the scene to the hosiptial.

9.3.5 hosptitalToBaseDist

Input: base index and hosptial index

Body: This function uses *haversine_fun* to generate the distance from the given hospital to the given base.

Returns: distance from hospital to origin base

9.3.6 jitter

Motivation: Because we decided to bootstrap call location samples directly from the original data, we wanted to incorporate additional randomness into the model. Our "jitter" helper function accomplished this by shifting the call latitude and longitudes from the initial locations.

Input: For each call, the original latitude and longitudes are randomly selected from the existing data. From there, we pass both values (as a pair) through our jitter function. The first step in this function is to generate a uniform random variable between 0 and 1 using np.random.uniform(). If this value is greater than 0.5, the call location is returned as is. If the value of the uniform random variable is less than or equal to 0.5, however, then we proceed with the function.

Body: A second uniform random variable between 0 and 1 is then generated using np.random. uniform(). This represents the distance the call is to be shifted from its initial location, measured in kilometers. We then use the inverse haversine function (from the haversine package) to calculate the new latitude and longitude points. The inputs to the inverse haversine function are the original location, the distance from the original point, and a direction. The direction is chosen by randomly selecting an entry in our *directions* vector, which contains values for North, East, South, West, Northeast, Northwest, Southeast, and Southwest. The output of this function is a new pair of latitude and longitude values that is slightly different from the original point obtained from the data.

9.3.7 call_rate

Input: An hour rate, that represents on average how many calls are placed in that hour.

Body: Uses an exponential distribution to generate inter arrival times of the calls and add them to an array.

Returns: An array of length n (n is the number of calls that hour) with cells filled with integers between 0 and 60 that represent the minute in which the call was placed in that particular hour. The array is sorted in ascending order.

9.4 Core Simulation

Input: The number of simulations, the number of helicopters it will perform the simulation with, an array of the names of the locations of where the hospitals are located, and an array of the location of those hospital bases with each base corresponding to a specific latitude and longitude.

Body: The model first initializes the empty vectors to record the relevant statistics for call time, response time, time to care, and response fraction, as well as the system variables for simulation time and helicopter bases, locations, and return time. To run through the number of simulations and number of days for each simulation, we wrote a series of nested *for* loops. The outer loop iterates through the number of days for each simulation and the inner loop iterates through the number of hours for each day (24 total).

Next, we calculate the call rate based on the given hour of simulation and generate the number and time of calls. These are added to the call time vector and the simulation clock to advance it for each call. From there, we iterate through a *for* loop based on the number of calls generated for the given hour. For each call, we randomly sample a location from the original data. Next, we generate a random Bernoulli trial with a probability of 0.89925 (the probability that it is safe to fly as determined from the original data). If it is 1, we proceed with the rest of the simulation. If it is 0 instead, we proceed to the next call and append zeros to the simulation statistic vectors because there is no data.

If the simulation establishes that it is safe to fly, we then apply the jitter function (as detailed above) to the call location. If the nearest helicopter base is less than or equal to 180 kilometers from the call location and has at least one available helicopter, we then continue with the simulation. Additionally, we determine if the case needs to be taken to a trauma center or a standard medical facility (emulating injury severity in real life) using a random binomial instance with a probability of 0.807 in favor of a standard medical facility, as determined through previous analysis. If the nearest hospital or trauma center is beyond the range of the call location, we proceed to the next call and append 0 to the simulation statistic vectors. When the hospital is close enough, we next generate a delay time and a prep time using np.random.triangular with lower and upper bounds and modes determined from previous analysis. We then subtract 1 from the number of helicopters available at the nearest base and append 1 to the calls dispatched vector, indicating that a helicopter was sent from a base to respond to the call.

Next, we randomly generate a cancellation time using np.random.exponential with the rate found though previous analysis. As long as the time it takes the helicopter to be dispatched and arrive at the scene is shorter than the cancellation time, we can proceed with the rest of the simulation. Of course, if it is not, we proceed to the next call and append 0 to the simulation statistic vectors. When a call is not cancelled, we then append 1 to the response fraction vector, indicating that a helicopter has responded to this particular call. We generate a random value for time on scene using sc.stats.beta.rvs and a parameters found through previous analysis. We append the time it takes the helicopter to arrive at the scene to the response time vector. The time to care value is calculated by summing the delay time, prep time, time to scene, time on scene, and time from scene to the hospital and is appended to the time to care response vector.

Lastly, we "return" a helicopter to its base by calculating the time it takes for it to travel from its present hospital location to its original base (which may be 0) and adding that to the time at the hospital (generated using sc.stats.beta.rvs and previously determined parameters). We add 1 back to the respective count of helicopters at that base.

Returns: An array where each helicopter should be located with each index of the array corresponding to a specific base location. The values at each index correspond to how many helicopters should be stationed at each base. The core simulation also returns the total percent of calls dispatched, the average response time, the average definitive care time, the response fraction, the utilization, the standard deviation of the response times and definitive care times, and the number of response times and definitive care times we analyzed to produce the average response and definitive care times.

9.5 Main Function Call

This section of the code base calls the core simulation function. It defines the variables that represent the number of trails that should be run and how many days each trial should simulate. It also states the hospital names and locations of those with helicopter bases. Besides the variable definitions, this code chuck mainly consists of a for loop that goes from 1 - 12 representing how many helicopters are in the system. In addition, the returned statistics are "Pretty Printed" to the console. This feature is largely redundant and could have been minimized but was implemented to increase the read ability of the code base.

9.6 K-Means Add-On

Motivation: With so many possible permutations for distributing the helicopters among the five base locations, we wanted to reduce the possible number of simulations to minimize computation time and energy. The best method we decided on was to use k-means clustering to divide Upstate New York into five regions centered around select hospitals.

Clustering Background: K-means clustering is an unsupervised learning method that partitions a data space into k groups. Each group is centered around a mean, with data points being assigned to clusters based on the smallest distance from the point to one of the established group mean centers. Initial means are set at the first step (which may or may not be close to the optimal means) and with each successive iteration of the algorithm, points are reassigned to the nearest cluster mean and then the means are shifted to represent the new means of the assigned groups. Over a series of iterations, these k groups should converge to close to ideal clusters.

Procedure: We used the KMeans function from the sklearn package in Python to compute the clusters. The function takes inputs for the number of clusters, random state, maximum iterations, and tolerance (among other values). We experimented with a few different values for these inputs, but decided to proceed with the defaults as given by the sklearn documentation. Of course, we set the number of clusters to 5.

From there, we selected a random sample of n = 16994 points from the original data set, with replacement, and fit the k-means algorithm based on this sample. The output allowed us to view the cluster centers and the clustering assignments for each data point (labelled 0 through 4).

We then used the haversine function to determine which hospital locations were closest to each of the cluster centers.

Graphing: We used the matplotlib.pyplot function to plot the data points and colored each point based on its cluster assignment label to visualize how the clusters were formed across Upstate New

York. Helicopter range circles were calculated using the inverse haversine function to measure 180 kilometers from the hospital base in both the vertical and horizontal directions. Colors were matched between the cluster labels and range circles to improve the aesthetic quality of the graphs and make them easier to interpret.

Bootstrapping: We wanted to ensure that the first random sample of data points was not an outlier and that the hospital bases that we had selected were indeed closest to a confidence interval of cluster means. This was accomplished through empirical bootstrapping. We repeated the random sampling with replacement 1,000 times, calculated the k-means clustering on each sample, and recorded the five cluster centers. Then, we plotted the 1,000 estimations for each of the five cluster means. As is seen in the graph included in the report, the confidence ranges are fairly narrow, so we believe that the nearest hospitals selected in the first step are indeed the ideal hospitals to station helicopters at based on the k-means clustering algorithm.