Patient Flow and Staff Scheduling for Medical-Surgical Unit

Computer Integrated Surgery II, Spring 2018

Evelyn Yeh and Sara Cronin

Background

Due to rising healthcare costs, there is a substantial amount of pressure on hospitals to deliver the most cost-efficient care possible. Currently, many hospital expenditures have been shown to be unnecessary, whether due to inefficient usage of resources, poor communication, or other failures of the system[1]. To reduce this budgetary waste, hospitals must improve their resource management. In turn, hospital resource management depends heavily on being able to predict and improve patient flow, which is a main determinant of patient safety, satisfaction, and quality of care. Optimal patient flow has been shown to maximize effective treatment, increase efficient use of hospital resources, and minimize complications that are linked to prolonged hospitalization[2].

Focusing on patient flow brings staff scheduling, which is a significant contributor to hospital expenses, into the equation. Patient flow has many sources of uncertainty that make staffing floors difficult. All patients have different attributes and needs for their care, and hospital procedures can take shorter or longer than expected, contributing to variability in patient movement through medical units. Because patient flow is highly variable, determining the number of nurses and staff that should be available for each unit is burdensome. In addition to uncertainty in patient flow, there are general staffing restrictions as well as guidelines based on the patients' needs. For instance, excessive handoffs between nurses are highly discouraged, as they can lead to loss of patient information, miscommunication, and overall poorer patient outcomes. Depending on the hospital unit, nurses are also allowed a maximum number of patients to avoid overburdening them. All of these are just some of the sources of uncertainty that complicate the staff scheduling dilemma.

Currently, in order to determine staffing for the day, hospital staff employ a method named real-time demand capacity management (RTDC), in which they congregate in a room prior to the workday beginning and manually predict patient influx and discharges to determine the number of nurses who should be working and other staff schedules[3]. If the floor is understaffed, nurses will be overwhelmed with a high patient census, leading to worse patient outcomes and longer wait times[4]. If the floor is overstaffed, there will be budgetary waste with idle nurses, leading to higher healthcare costs for patients. However, there are limitations to this process of assigning staff. RTDC requires a daily unit evaluation that can be eliminated through automation, and it is highly subjective and variable since it depends on the clinicians' opinions of their unit's patients.

To rectify these shortcomings, RTDC could be potentially automated by analyzing and employing historical data to create a patient census model. This improvement would save time on the staff's part and ideally be more accurate than their predictions. For our project, our main objective is to develop this census model using a simulation of patient flow in the medical unit that we are working with, the 4-Pavilion unit of Howard County General Hospital. With this census model, we will be able to output an hourly patient census that can be used to create a nurse schedule that adequately staffs the unit while minimizing extraneous costs.

Approach

The first step to addressing this problem was to get an understanding of the temporal trends in patient movement through the 4P unit. This was done by consulting a nurse manager for the unit, Ms. Anita Ben. She informed us that in her experience, the season and day of week were significant determinants of the busyness of the unit, in terms of both how many patients entered and how long each patient stayed. The time of day was also determined to be important, so we also decided to split each day into four time ranges. These were based on the times during which the nurse manager on 4P consults the census to determine the staffing for the next shift. The resulting time ranges were 7am - 3pm, 3pm - 7pm, 7pm - 11pm, and 11pm - 7am. The temporal trends suggested by Ms. Ben were supported by the data as well.

In the resulting temporal organization, the patients who were in 4P during the same season, day of the week, and time range were grouped together. Distributions of patient attributes given the set of temporal parameters were obtained from this group of patients. Here, consistent with the discussion above, a set of temporal parameters includes a season, day of week, and time range - for example: Winter, Monday, 7am-3pm. Once the temporal organization to be used in the simulation was determined, the historical data was examined and used to create the necessary distributions. The first aspect to be determined was the census distribution for each set of temporal parameters. The patient admission, transfer, discharge data were converted to a Python pandas dataframe and the relevant information was converted into a usable form, which included creating a variable that stored the time spent on the given unit. First, dictionaries were created which linked an ID unique to each visit, the time of entry into the 4P unit, and the length of stay in 4P. This information was then integrated into a SimPy simulation that determined the historical hourly census based on the admission, transfer, and discharge data given. For each set of temporal parameters a distribution of census values was created and stored to be sampled from in the patient movement simulation.

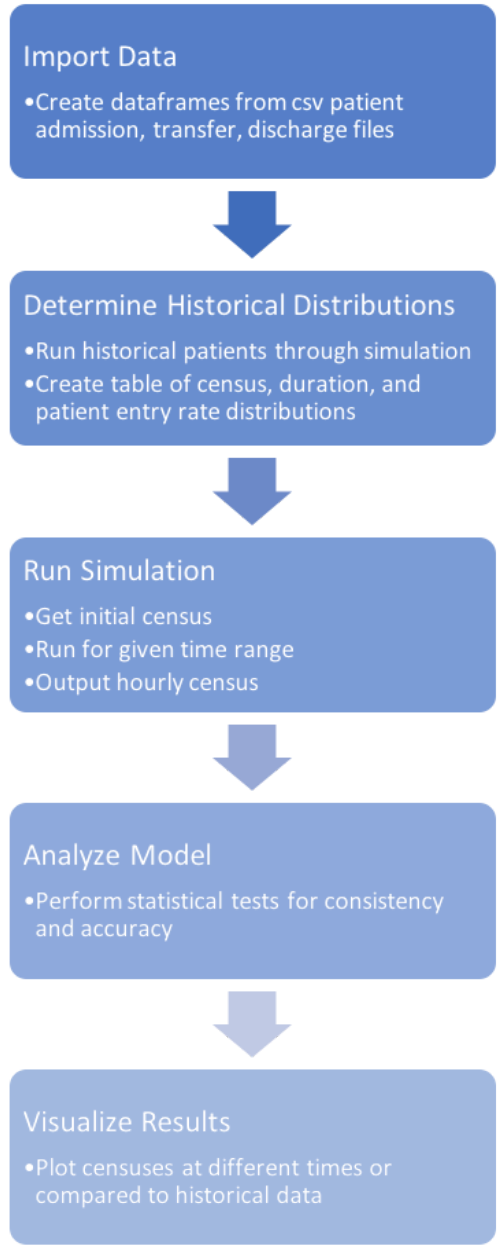


Figure: Project workflow

In addition to census distributions based on time, durations and patient entry rate also had to be determined based on historical data. These were obtained by segmenting the historical patient admission, transfer, discharge data by the temporal parameter sets. For each parameter set a list of the length of stay durations for each patient whose entry time met the given parameters was created. In addition to a length of stay list, each temporal parameter set contained a probability distribution of the patient entry rate. In general the possible values ranged from 0 to 4 patients per hour. The duration lists and entry rate probability distributions were stored to be sampled from by the simulation.

The first step in operating the simulation is to obtain input from the user. The user is allowed to input the starting and ending date and time for the simulation. For instance, the user can input “05/09/18 05:00” for the start time and “05/12/18 15:00” for the end time. Then, the simulation will use historical data to simulate the patient census from 5am on a Wednesday during the season Spring to 3pm on a Saturday in Spring - a total time of 3 days and 10 hours. We designed it this way so that the simulation length can be altered by the user depending on their needs. Some users may want to look at short-term census predictions for immediate staff scheduling purposes, while others may want to observe long-term for potential staff recruitment.

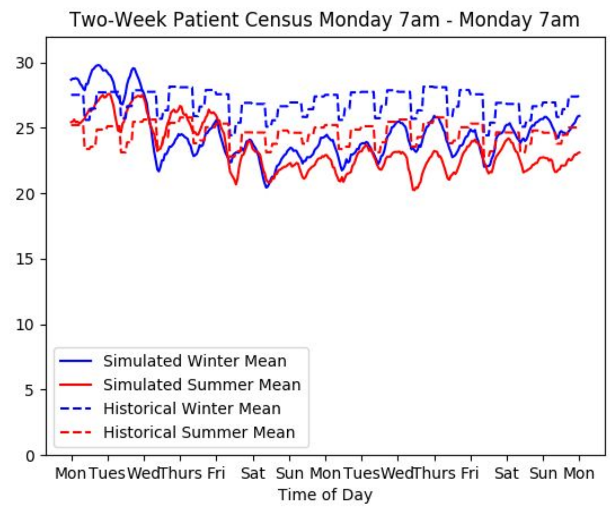
After obtaining user input, the simulation begins by getting an expected current census at the start time that is indicated, using the generated census distributions that have been described previously. To clarify the process, the simulation would use historical data to determine a starting census of 24 patients at 5am on 05/09/18 in the example above. Then, the simulation will assign lengths of stay for these starting patients. After a patient’s length of stay has elapsed in SimPy, the patient will be discharged and the census will be automatically updated. For the rest of the simulation time, the census will be checked hourly by default. The frequency of checking the census can be easily modified in our code to either check more frequently and gain more detail about the unit or to check less frequently and operate faster over a long period of time. Every hour, the program samples from the census distributions to update the census. If the census is supposed to increase from the current census, then a certain number of new patients with calculated lengths of stay will be admitted to the unit. Using the example above, the simulation would perform its first check at 6am on 05/09/18. It would sample from the patient entry rate per hour probability distribution to determine how many patients should enter. Each new patient would be given a length of stay sampled from the distribution corresponding to the temporal parameter set. The census will be incremented to reflect the new admissions. Ms. Ben, the nurse manager for 4P, informed us that the maximum bed capacity of 4P is 30 beds. So when updating the census, the program will limit the census to 30 patients and keep track of any patients that have to be rejected because the unit is over capacity. As mentioned before, the census decrements automatically when each patient has completed their length of stay.

This process will proceed until the simulation reaches the user’s given end time. While the simulation is running, it will keep a list of each hour’s census to show how the census has changed over time. The program also keeps a list of how many patients were rejected from admission to 4P each hour because the unit was over capacity. Finally, the program concurrently keeps a list of how many nurses would be appropriate to staff the unit during each hour. From our conversation with Ms. Ben, we learned that the ideal number of patients for one nurse is five patients. So the appropriate number of nurses is calculated for the census at each hour and tracked throughout the simulation as well. At the end of the simulation, the full census history is outputted.

The final components of the project produce plots and run statistical tests on simulated census outputs. The results of these are given below.

Results

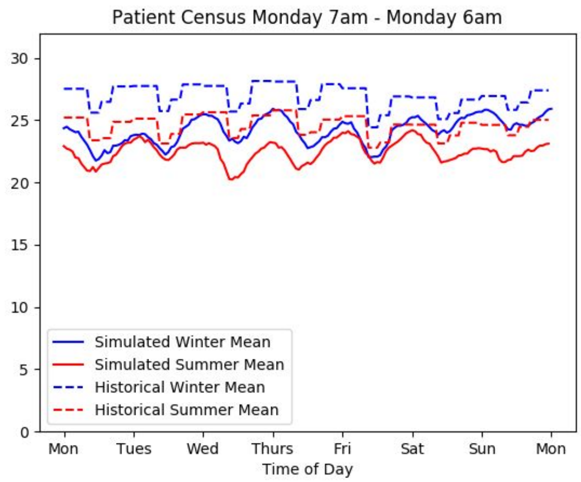
Various plots were created to visualize the simulated census and provide an indication of the simulation's performance.

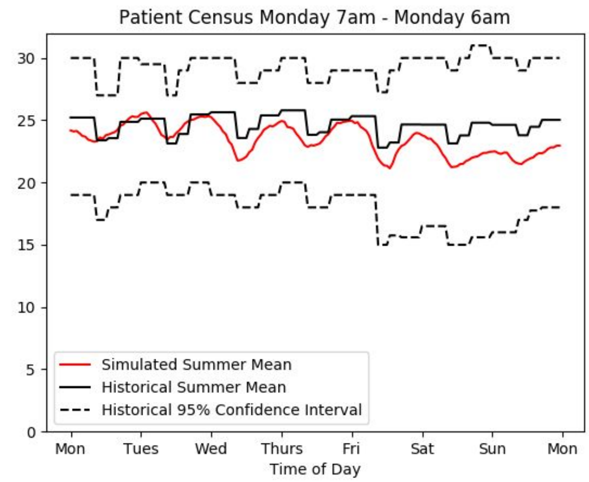


First a series of 25 trials of a two week period was run for both the summer and winter seasons. The mean of these results is plotted along with the mean of the historical census data for corresponding time frames. The historical means show the seasonal trend of winter census values being higher than summer ones. In general this is also shown in the simulated censuses.

The largest discrepancy between the simulated and historical censuses is seen in the first 2-4 days of simulation time. This can be attributed to the lengths of stays assigned to the initial patients entered into the simulation. These lengths of stay are sampled from the full length of stay distribution, which does not account for the fact at the initial simulation state patients would have already spent some time the unit and would therefore have shorter remaining lengths of stay. When the length of stay value sampled from this distribution was lessened for the initial simulated patients the temporal peaks evident in the original distribution were lost leading to poor results in the first few simulated days. Additionally, this decrease in initial patient lengths of stay led to an average census below the historical average once an equilibrium was reached. Therefore the decision to keep the full length of stay distribution for the initial patients was made.

Additional plots were created to analyze the predictions, and they are shown below.





The plot on the left gives the results of 25 simulation iterations of a one week period for both summer and winter were plotted and compared to the historical means for the given time ranges. The plot on the right displays the results of 50 iterations of a one week period in the summer. The historical mean and 95% confidence interval is also shown. The simulated summer mean is shown to fall within the confidence interval of the historical mean.

A series of statistical tests were performed to quantify the graphical observations and verify that the simulation census output was consistent with historical data. First a t-test for the equivalence of the means of the simulated and historical censuses was performed. This indicated whether the means of the distributions differed. Next a Pearson's correlation coefficient was computed to indicate the degree of linear correlation between the historical census and simulated census over time. A table summarizing the results is given below.

Simulation	t-test p-value	Pearson correlation coefficient
Winter- 25 weeks	0.0169	0.393
Summer- 25 weeks	0.0653	0.438
Winter- 50 weeks	0.051	0.578
Summer- 50 weeks	0.137	0.561

As indicated in the table each trial failed to reject the null hypothesis that the means of the distributions were equal at the .01 level. With increasing trials the p-value also increased, indicating that the mean of the simulated census was closer to the historical when averaged over more weeks. The correlation coefficient indicated a positive correlation between the simulated census and the historical census data. This too improved with an increase in number of simulated weeks. Each correlation coefficient had a corresponding p-value less than .001.

The simulation tended to perform better for summer time frames compared to winter. This could be due to the increased variability seen in the winter months.

Significance

We created a simulation that captures the temporal trends and variability of the unit in question. The program outputs an hourly census that could be compared to the true state of the unit or used to predict future patient census. Our work automatically combines historical trends and the current state of the unit to output census predictions which saves time for the nurse manager in charge of scheduling and allows nurses schedules to be altered to match the predicted census well before the start of the shift. This reduces costs involved in having an overstaffed unit or needing to call unscheduled nurses in to work.

Management Summary

Both of us participated in meetings with our mentor, traveled to HCGH to speak with the nurse manager of 4P, and documented the code that we respectively developed. Evelyn developed the simulation structure that could output an hourly census, implemented a method for user input, and implemented nurse scheduling that is updated according to the census. Sara performed the analyses of the historical data, generated distributions of patient movement attributes, and analyzed and created visualizations of the simulation results.

Originally, we had a different idea of what the deliverables of our project would be. Early on, we had hoped to take on the challenge of developing an optimization algorithm to be solved for creating staff schedules as our maximum deliverable. After further meetings with our mentor and beginning to work on the project, we realized that we did not take into account many factors that affect the simulation aspect. Preprocessing the data and creating a reliable simulation would require much more effort than we expected, so we modified our project timeline and deliverables to be more feasible to complete. We established our expected deliverable, the bulk of the project, to be creating a patient census model that could generate an hourly census of 4P. Rather than attempting an optimization problem, we altered our maximum deliverable to consist of simpler nurse scheduling. Our new goal was to determine how the census would affect the number of nurses needed for the unit and to implement nurse matching based on our findings.

Because we took a lot of initiative in fleshing out and beginning to code for the project rather than receiving assignments from our mentor, we initially developed a simulation that predicted patient flow throughout the entire hospital, not just 4P, since we were given access to patient data from the entire hospital. The focus of this original simulation, which is reflected by this simulation structure, was to monitor the movement of patients through the hospital and determine how their movement affected their stay in 4P. After consulting with our mentor and gaining a clearer vision of the project's objectives, we modified the simulation to focus only on patient census in 4P, since that is more relevant for determining how to schedule nurses for the unit. In terms of determining the patients' length of stay, we focused on temporal attributes, such as what time, day, and season the patients were admitted to 4P, rather than the preceding department. This modification to the project simplified the work and narrowed the focus of our project, so that our results would be more useful to Ms. Ben, the nurse manager of 4P.

In the end, we were able to complete all of our minimum and expected deliverables, but not our maximum deliverables. We have been reworking our patient census model to make it more accurate and efficient, since we found that our initial implementation consistently underestimated the historical data and took a significant amount of time to run. Our new implementation generates a census that is more efficient and consistent with the data. We also implemented nurse matching to occur concurrently with the census updates.

Since we have been focusing on improving the census model, we have not created a program that can analyze the effects of different nursing levels on the census and management of the unit, which was our maximum deliverable. This could be one of the next steps of our project. Additionally, our current implementation of nurse matching is quite basic and based solely on the patient census. To improve the nurse matching, we can also take patient attributes into account, such as their age, gender, reason for visit, and other characteristics. These attributes can be used to determine which patients may require a higher level of care from nurses and to ensure that nurses are not overburdened with too many demanding patients or idle with a few straightforward patients. Lastly, to make our project more accessible to nurse managers who may want to implement our model, a program should be written to directly output an optimal nursing schedule in a format that is easy for them to use.

Through this project, we have learned that we must maintain constant communication with our end users to determine what is best for them. When beginning projects, we should develop a more detailed and clear plan with our mentor to ensure that the objectives of the project are clear to everyone involved. We also learned to place a larger emphasis on our users' needs and to develop our project in a way that will have the most effective results for them.

Technical Appendices

- Code File List
    - FullDriver.py
    - importdata.py
    - gethistoricalinfo.py
    - getdictionaries.py
    - HistoricalSimulation.py
    - createlookup.py
    - segment.py
    - processdata.py
    - createsimulation.py
    - getstartingcensus.py
    - gettimeleft.py
    - getduration.py
    - getplots.py
    - gethistorical.py
    - Stats_Tests.py
- Documentation
    - Provided on website

Acknowledgements
We would like to thank our mentor Dr. Siddiqui as well as our collaborators Eric Hamrock and Scott Levin. Thanks to Anita Ben and the staff of the 4P unit of HCGH for providing clinical insight and allowing us to examine their data and workflow.

References

1. Reid PP, Compton WD, Grossman JH, et al., eds. Building a Better Delivery System: A New Engineering/Health Care Partnership. Washington, DC: National Academies Press; 2006.
2. Levin S, Dittus R, Aronsky D, et al. Evaluating the effects of increasing surgical volume on emergency department patient access. BMJ Qual Saf. 2011;20:146–152.
3. Resar R, Nolan K, Kaczynski D, et al. Using real-time demand capacity management to improve hospital-wide patient flow. Jt Comm J Qual Patient Saf. 2011;37(5):217–227.
4. Cure, L., Zayas-Castro, J., and Fabri, P. (2014) Challenges and opportunities in the analysis of risk in healthcare. IIE Transactions on Healthcare Systems Engineering, 4(2), 88–104.