Monte Carlo Methods

Geneva Porter **816408535**

20 November 2018

San Diego State University
Professor J Mahaffy, Math 636

1 Operating Room and Recovery Room Usage

Consider the classic hospital expansion problem put forth by Schmitz ans Kwak in their 1971 paper "Monte Carlo Simulation of Operating-Room and Recovery-Room Usage." In this paper, they outline a simulation based on the number of operating rooms and recovery rooms planned for a new hospital wing ICU, with the goal of answering the following questions:

(1) How many more surgeries are performed because of the increased bed capacity?

(2) How much operating room time and space will the surgeries require?

(3) How much recovery room time and space will the surgeries require?

We will use Schmitz and Kwak's results for the first question to answer the second and third questions. They were able to gather enough data to create a rubric for the percentage of increased surgeries as well as the likelihood of any particular surgery that is needed. Their simulation was structured using the following criteria: Incoming patients will each need surgery, and the type of surgery is simulated using a probability distribution. Operating rooms open at 7.5 hours (0700), and the length of each surgery varies between 0.5 and 4 hours. Once a patient is out of surgery, the OR takes 0.25 hours to clean and prep for the next patient. The recovery room opens at 0800. Patients that use a recovery bed will take 0.08 hours to travel from the operation room to the recovery room, and stay there for the appropriate length of time depending on their surgical procedure. Once a patient leaves the recovery room, it takes 0.25 hours to prepare the bed for the next patient. Beds are filled ordinally as they become available. If there are no beds available when a patient leaves surgery, a new bed will be opened.

1a. Simulations

First we will perform two simulations (both using the same set of random numbers) for a hospital expansion that assumes 32 patients come to the ICU per day. We will examine 2 scenarios in which there are either 4 or 5 operating rooms available. Tables **i** and **ii** in Appendix A hold results of these simulations.

The simulation in Table **i** assumes that there are 4 operating rooms available. The last OR closes at 1975 hours, and each one is used an average of 1.2578 hours. The recovery room closes at 2283 hours, and 10 beds are used for recovery.

For a simulation with 5 operating rooms, the last one closes at 1775 hours, and each one is used an average of 1.2578 hours. The recovery room closes at 2083 hours, and 12 beds are used for recovery.

If we were to run this simulation 100 times, we would get a better sense of expected values. In a simulation with 4 operating rooms, the average OR closure is at 19.2125 hours with a standard deviation of 1.5992. Each OR is used an average of 1.1251 hours, with a standard deviation of 0.1518. The average recovery room closure is at 22.2625 hours, with a standard deviation of 1.6407. An average of 9.9100 beds are used for recovery each day, with a standard deviation of 1.1984.

For 5 operating rooms, the average OR closure is at 17.2200 hours with a standard deviation of 1.2600. Each OR is used an average of 1.1251 hours, with a standard deviation of 0.1472. The average recovery room closure is at 20.2900 hours, with a standard deviation of 1.2711. An average of 11.4800 beds are used for recovery each day, with a standard deviation of 1.3370.

1b. Staffing

One issue that Schmitz and Kwak did not cover is staffing. Obviously, a bigger hospital means more employee hours. If we examine the staff of nurses needed to maintain this expansion, we can assign hours based on the number of patients present in the recovery room. We assume that the recovery room must always have at least 2 nurses, each nurse can supervise at most 3 patients, and time assignments will be multiples of whole hours. We can use the results for each simulation to create some simple tables illustrating the number of nurses needed for each hour that the recovery room is open. Table 1 below show theses findings.

In both cases (4 and 5 operating rooms) The maximum number of nurses needed during peak hours is 4. If there are 4 operating rooms, 38 staffing hours are needed. For 5 operating rooms, 36 staffing hours are needed.

If we examine 100 simulations and take the average of those results, we see slightly smaller values. With 4 operating rooms, an average of 36.0600 staffing hours are needed, with a standard deviation of 3.4868. An average of 3.4800 nurses are needed during peak hours, with a standard deviation of 0.5021. It takes an average of 34.6900 staffing hours for 5 operating rooms, with a standard deviation of 2.6273. An average of 4.0100 nurses are needed during peak hours, with a standard deviation of 0.5024.

Time Block	4 Operating Rooms Used	5 Operating Rooms Used
0800 - 0900	2	2
0900 - 1000	2	2
1000 - 1100	2	2
1100 - 1200	2	3
1200 - 1300	3	3
1300 - 1400	2	4
1400 - 1500	3	4
1500 - 1600	2	4
1600 - 1700	4	3
1700 - 1800	3	3
1800 - 1900	4	2
1900 - 2000	3	2
2000 - 2100	2	2
2100 - 2200	2	closed
2200 - 2300	2	closed

Table 1 Staffing Needs with 4 or 5 Operating Rooms

1c. Discussion

When comparing the single simulations between a hospital with 4 or 5 ORs, we see that fewer rooms means longer staffing hours yet fewer recovery beds used. This is because patients will be moved to the recovery rooms at slower pace, creating more opportunity for recovery beds to be prepped. Repeating these simulations 100 times shows us that our initial simulations were fairly typical and fall within one standard deviation of the mean, with one exception. Although our single simulations predicted a maximum of 4 nurses needed in the recovery room at any given time for both the 4 OR and the 5 OR situations, it is more likely that only 3 will be needed in the 4 OR model. Our prediction of needing 4 nurses in the recovery room for a 4 OR model is only within 2 standard deviations of the mean.

We can conclude that including an additional OR allows both the ORs and recovery rooms to close about 2 hours earlier, and nurse staffing hours are reduced by 2. The additional expense paying a surgeon in an additional OR may not be worth it just to save nursing 2 hours per day. However, assuming nurses make about $30 per hours, having 5 ORs would save about $15,600 per year in nurse salaries.

1d. Program

To create an automated version of this analysis, it is necessary to employ a series of nested for loops and if statements. The exact MATLAB code used is shown in Figure **i** of Appendix A, and the general algorithm used for these simulations is shown below:

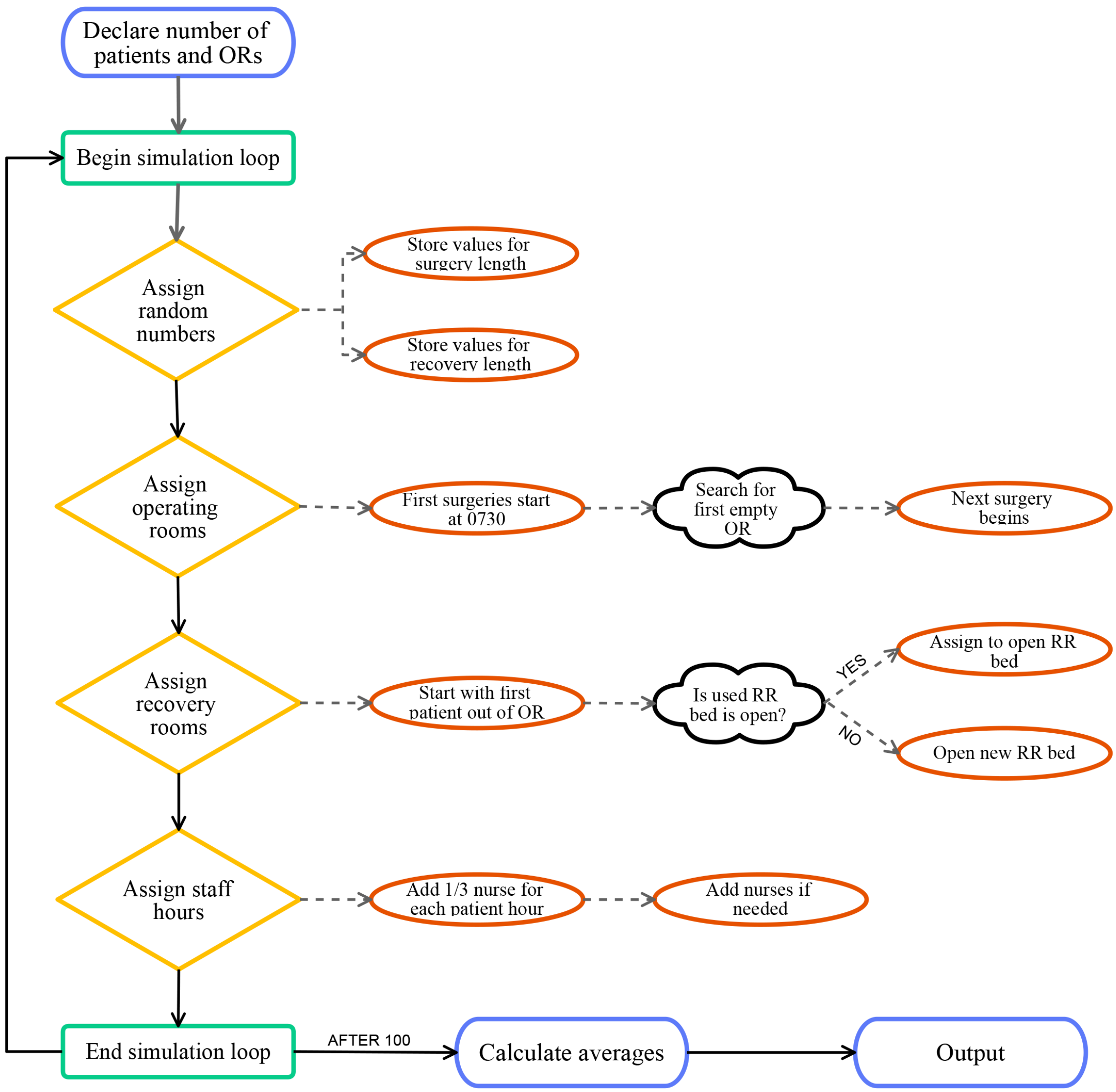

Figure 1 Flow Chart for Hospital Monte Carlo Simulation

2 Population Growth

A simple Malthusian growth model can often be sufficient for modeling populations that grow exponentially. However, a Monte Carlo simulation is discrete and accounts for variance. The following briefly examines both model types.

2a. Malthusian Growth

If we consider a Malthusian Growth model with a 4% annual growth, it will have the form

$$P_{n+1} = 1.04P_n \qquad \longrightarrow \qquad P_{n+1} = 1.04^n P_0$$

With $P_0 = 50$ and n given as years. After 10 years, the population will be approximately 74.0122. We can solve $P_{n+1} = 100$ to see that it will take 17.6730 years for the population to double. This model is easy to work with and can quickly predict unhindered growth.

2b. Birth Only Simulation

Rather than expecting the population to grow by exactly 4% each year, we can use a Monte Carlo simulation to consider each individual in the population as having a 4% chance of giving birth for a particular year. Our simulation will also include the possibility that individuals born one year can give birth the next. Table 4 below shows one simulation, which results in a population of 72 after 10 years.

Time (in years)	0	1	2	3	4	5	6	7	8	9
Population	50	52	53	58	60	63	66	67	70	72

Table 4 Population Simulation Over 10 Years

2c. Discussion

Running this simulation 1000 times, we get a mean of 74.0110 for the final population, with a standard deviation of 5.7526. Table 4 gives a final population within 1 standard deviation of the mean. We can say that the Monte Carlo simulation was likely successful (since the mean was very close to our Malthusian growth model). In this case, the Malthusian model would be better to use because it is significantly less complex yet produces good enough results.

3 A Game Of Darts

Here, we examine a standard dart board (Figure 1). On it, there are 6 areas in with to score: the inner bullseye (red), the outer bullseye (green), single rings (black and tan), double ring (outer red and green), triple ring (inner red and green) and the zero ring (black with numbers).

Figure 1: standard dart board

3a. A Single Round Of Play: Examples

In the common game of 501-darts, the goal is to reach a score of zero. Each player (or team) begins with a score of 501, and points are subtracted as they are won during rounds. The winning throw must bring the score to exactly zero and land on the double ring. The best possible score in a single round (throwing 3 darts) is 180. This corresponds to each of the three darts landing on the triple ring of the 20 section, and receiving a triple score for each hit. This score is highly unlikely, with a probability of 0.005% (if the dart board is hit randomly). For the average player with the basic skill necessary to hit the dart board for each throw, a round of play might score around 20 points. Table 5 below a simulation for 5 typical rounds of play if the board was hit randomly.

Round	Throw 1	Throw 2	Throw 3	Total Score
1	8	10	10	28
2	1	20	0	21
3	6	0	0	6
4	0	26	0	26
5	10	14	0	24

Table 5 5 Rounds of Darts

3b. A Single Round Of Play: Averages

A Monte Carlo simulation of 10,000 rounds produced an average score for three throws is 22.0484, with a standard deviation of 17.2120. For single vs. single play, it is likely that over 20 rounds must be played before a player wins.

3c. Discussion

These simulations have assumed that the dart board was hit randomly. Most players will have the intention of hitting closer to the center, and the chances of hitting the zero ring go down as the player skill goes up. A more realistic model would put less weight on the probability of hitting the zero ring. Gathering data on players of all skill levels might show that the chance of hitting the zero ring is significantly less than if the board were hit randomly. However, a double score is needed to win. Players aiming for a double score to win are much more likely to hit the zero ring. A simulation for an entire game of darts rather than single rounds could fix these issues by weighing the probability of scores differently for the winning round.

4 Stochastic Birth

A stochastic birth process is one in which the birth rate may be analyzed statistically but not calculated precisely. Here we will consider a birth-only process, where only one birth is possible in some time interval Δt. The probability that the population will be some value N after some Δt amount of time is given by:

$$P_N(t+\Delta t) = P_N(t)(1-\lambda N\Delta t) + P_{N-1}(t)\lambda(N-1)\Delta t,\ P_N(0) = \begin{cases} 0 \text{ if } N \neq N_0 \\ 1 \text{ if } N = N_0 \end{cases}$$

In the above equation, we have several values to define. First, $P_N(t)$ is the probability that the population will be N at time t. Since λ is the birth rate, the value $\lambda N\Delta t$ is the probability that the population will increase to $N+1$ after a time interval of Δt, given that the population began at N at time t. N_0 is our initial population. If we manipulate the equation above, we get:

$$\frac{P_N(t+\Delta t) - P_N(t)}{\Delta t} = -\lambda N P_N(t) + \lambda(N-1)P_{N-1}(t) = \frac{dP_N(t)}{dt}$$

By replacing N with N_0 (and noting that $P_{N_0-1} = 0$ for a birth-only process), the solution to this differential equation is $P_{N_0}(t) = e^{-\lambda N_0 t}$. As we increase N, our calculations become more complex. For $N = N_0+1$, we have:

$$P_{N_0+1}(t+\Delta t) = P_{N_0+1}(t)(1-\lambda(N_0+1)\Delta t) + P_{N_0}(t)\lambda(N_0)\Delta t$$

$$\longrightarrow \qquad \frac{dP_{N_0+1}(t)}{dt} = -\lambda(N_0+1)P_{N_0+1}(t) + \lambda(N_0)P_{N_0}(t)$$

Here, we can use the standard solution:

$$\frac{dX(t)}{dt} = a(t)X(t) + b(t) \quad \longrightarrow \quad X(t) = e^{\int_0^t a(s)ds} \int_0^t b(s) \cdot e^{\int_0^s -a(u)du} ds + X_0 e^{\int_0^t a(s)ds}$$

Where $X_0 = P_{N_0+1}(0) = 0$. Applying this standard solution to $P_{N_0+1}(t)$ and substituting $P_{N_0}(t)$ with our previous solution $e^{-\lambda N_0 t}$ gives us:

$$P_{N_0+1}(t) = e^{-\lambda(N_0+1)t} \int_0^t \lambda(N_0) e^{-\lambda N_0 s} \cdot e^{\lambda(N_0+1)s} ds + 0 \quad \longrightarrow \quad = N_0 e^{-\lambda N_0 t}(1 - e^{-\lambda t})$$

If we repeat this process for $N = N_0 + 2$ and $N = N_0 + 3$, we will see:

$$P_{N_0+2} = \frac{(N_0)(N_0+1)}{2} e^{-\lambda N_0 t}(1 - e^{-\lambda t})^2 \text{ and}$$

$$P_{N_0+3} = \frac{(N_0)(N_0+1)(N_0+2)}{2 \cdot 3} e^{-\lambda N_0 t}(1 - e^{-\lambda t})^3, \text{ and ultimately}$$

$$P_{N_0+j} = \frac{(N_0)(N_0+1)...(N_0+j-1)}{j!} e^{-\lambda N_0 t}(1 - e^{-\lambda t})^j$$

We can prove that this form hold true for all $j + 1$ with $j \in \mathbf{N}$. Our goal is to show the following:

$$\frac{dP_{N_0+j+1}(t)}{dt} = -\lambda(N_0 + j + 1)P_{N_0+j+1}(t) + \lambda(N_0 + j)P_{N_0+j}(t)$$

$$\longrightarrow \quad P_{N_0+j+1}(t) = \frac{(N_0)(N_0+1)...(N_0+j)}{(j+1)!} e^{-\lambda N_0 t}(1 - e^{-\lambda t})^{j+1}$$

4a. Proof by Induction

Let $a(t) = -\lambda(N_0 + j + 1)$ and $b(t) = \lambda(N_0 + j)P_{N_0+j}(t)$.

If $\dfrac{dP_{N_0+j+1}(t)}{dt} = a(t)P_{N_0+j+1}(t) + b(t)$ then

$$P_{N_0+j+1}(t) = e^{-\lambda(N_0+j+1)t} \int_0^t \lambda(N_0+j)P_{N_0+j}(s)e^{\lambda(N_0+j+1)s} ds + P_{N_0+j+1}(0)e^{-\lambda(N_0+j+1)t}$$

Substituting $P_{N_0+j}(s) = \dfrac{(N_0)(N_0+1)...(N_0+j-1)}{j!} e^{-\lambda N_0 s}(1 - e^{-\lambda s})^j$ we get:

$$e^{-\lambda(N_0+j+1)t}\int_0^t \lambda(N_0+j)\frac{(N_0)(N_0+1)...(N_0+j-1)}{j!}e^{-\lambda N_0 s}(1-e^{-\lambda s})^j e^{\lambda(N_0+j+1)s}ds$$

For $P_{N_0+j+1}(t)$. Using the binomial expansion theorem, we can substitute:

$$(1-e^{-\lambda s})^j = \sum_{k=0}^{j}\frac{j!}{k!(j-k)!}(-e^{-\lambda s})^k$$

Simplifying, we now have:

$$e^{-\lambda(N_0+j+1)t}\lambda\frac{(N_0)(N_0+1)...(N_0+j)}{j!}\int_0^t\sum_{k=0}^{j}\frac{j!}{k!(j-k)!}(-1)^k(e^{\lambda s})^{j+1-k}ds$$

For $P_{N_0+j+1}(t)$. Evaluating the integral, we get:

$$e^{-\lambda(N_0+j+1)t}\lambda\frac{(N_0)(N_0+1)...(N_0+j)}{j!}\sum_{k=0}^{j}\frac{j!}{k!(j-k)!}(-1)^k\tfrac{1}{\lambda(j+1-k)}[(e^{\lambda t})^{j+1-k}-1]$$

For $P_{N_0+j+1}(t)$, which we can write as:

$$e^{-\lambda N_0 t}\frac{(N_0)(N_0+1)...(N_0+j)}{(j+1)!}\sum_{k=0}^{j}\frac{(j+1)!}{k!(j+1-k)!}(-e^{-\lambda t})^k[1-e^{-\lambda(j+1-k)t}]$$

Now, our summation is almost complete for $k=0\to j+1$, however it is missing the last term $k=j+1$. This is resolved when we distribute the summation through the $[1-e^{-\lambda(j+1-k)t}]$ term. We can now state:

$$P_{N_0+j+1}(t) = e^{-\lambda N_0 t}\frac{(N_0)(N_0+1)...(N_0+j)}{(j+1)!}\left\{\left[\sum_{k=0}^{j}\frac{(j+1)!}{k!(j+1-k)!}(-e^{-\lambda t})^k\right]+(-e^{-\lambda t})^{j+1}\right\}$$

$$\longrightarrow \quad P_{N_0+j+1}(t) = e^{-\lambda N_0 t}\frac{(N_0)(N_0+1)...(N_0+j)}{(j+1)!}\sum_{k=0}^{j+1}\frac{(j+1)!}{k!(j+1-k)!}(-e^{-\lambda t})^k$$

$$\longrightarrow \quad P_{N_0+j+1}(t) = \frac{(N_0)(N_0+1)...(N_0+j)}{(j+1)!}e^{-\lambda N_0 t}(1-e^{-\lambda t})^{j+1}$$

Which is what we wanted to show. QED.

5a. Model Validity

Continuing with the same stochastic birth model, we can say that the expected population $E(t)$ is equal to:

$$E(t) = \sum_{j=0}^{\infty} (N_0 + j) P_{N_0+j}(t)$$

Each term in this summation represents the population increase by j within the time interval Δt. For $j = 0$, our first summation term is $N_0 P_{N_0}(t)$, which is simply the initial population (N_0) multiplied by the probability ($P_{N_0}(t)$) that the population will stay at N_0. The next term gives the population increased by 1 multiplied by the probability that the population will increase by 1, and so on. This fits with the general formula for expected value, which can be described as:

$$\sum_{i=0}^{n} (\text{an event } x_i \text{ occurring}) \cdot (\text{probability that } x_i \text{ occurs}).$$

For n events. Our summation has an upper limit of ∞ to account for every possible population increase.

5b. Summation Solution

We can show that the above equation reduces to $E(t) = N_0 e^{\lambda} t$. First, we take the derivative and substitute in our first definition of $dP_N(t)/dt$:

$$\frac{dE(t)}{dt} = \sum_{j=0}^{\infty} (N_0 + j)[-\lambda(N_0 + j) P_{N_0+j-1}(t) + \lambda(N_0 + j - 1) P_{N_0+j-1}(t)]$$

Examining the first few terms in this summation, we get:

$$\frac{dE(t)}{dt} = \lambda(N_0 - 1)(N_0) P_{N_0-1}(t) - \lambda(N_0)^2 P_{N_0}(t) + \lambda(N_0)(N_0 + 1) P_{N_0}(t) -$$

$$\lambda(N_0 + 1)^2 P_{N_0+1}(t) + \lambda(N_0 + 1)(N_0 + 2) P_{N_0+1}(t) - \lambda(N_0 + 2)^2 P_{N_0+2}(t) + ...$$

Since this is a birth-only model, $P_{N_0-1}(t) = 0$. We can combine terms in this summation, which reduces to:

$$\frac{dE(t)}{dt} = \lambda(N_0) P_{N_0}(t) + \lambda(N_0 + 1) P_{N_0+1}(t) + \lambda(N_0 + 2) P_{N_0+2}(t) + ...$$

Which is simply $\lambda E(t)$. To solve this differential equation, we must first find an initial condition:

$$E(0) = \sum_{j=0}^{\infty}(N_0 + j)P_{N_0+j}(0) = N_0 P_{N_0}(0) + (N_0 + 1)P_{N_0+1}(0) + (N_0 + 2)P_{N_0+2}(0) + ...$$

Since we defined $P_N(0) = 0 \; \forall \; N \neq N_0$ and $P_N(0) = 1 \; \forall \; N = N_0$, all but the first term goes to zero, leaving us with $E(0) = N_0$. Using the method described in 4a. with $a(t) = \lambda$ and $b(t) = 0$, We see that $E(t) = N_0 e^{\lambda t}$.

5c. Discussion

This result is clearly a much simpler way to compute the expected population. It is identical to the model for exponential growth and decay, implying that the expected value is not a stochastic model. It does not rely on $P_N(t)$ to predict the population.

6 Calculations

A simple example of this process can be illustrated and used to find the value of λ for some population. Let's start with an initial population of $N_0 = 10,000$ and a data point of $E(20) = 14500$. We can then solve for λ:

$$E(20) = 10000e^{\lambda \cdot 20} = 14500 \quad \longrightarrow \quad \lambda = \frac{ln(4.5)}{20} \approx 0.01682$$

This corresponds to a birth rate just under 2%. We can also use our exact value of λ to re-write our equation as $E(t) = N_0(4.5)^{t/20}$.

Appendix A: Hospital Room Assignments

Patient	Ran. No.	Sur. Len.	Sur. Begin	Sur. Ends	OR	Rec. Len.	Rec. Begin	Rec. Ends	RR
1	389	0.75	7.5	8.25	1	3	8.33	11.33	2
2	997	4	7.5	11.5	2	3	11.58	14.58	2
3	198	0.5	7.5	8	3	1.5	8.08	9.58	1
4	560	0.75	7.5	8.25	4	3	8.33	11.33	3
5	330	0.5	8.25	8.75	3	–	–	–	–
6	852	1.75	8.5	10.25	1	3	10.33	13.33	1
7	901	2.25	8.5	10.75	4	3	10.83	13.83	5
8	757	1.25	9	10.25	3	3	10.33	13.33	4
9	769	1.75	10.5	12.25	1	3	12.33	15.33	6
10	707	1.25	10.5	11.75	3	3	11.83	14.83	3
11	745	1.25	11	12.25	4	3	12.33	15.33	7
12	784	1.75	11.75	13.5	2	3	13.58	16.58	1
13	608	0.75	12	12.75	3	3	12.83	15.83	8
14	796	1.75	12.5	14.25	1	3	14.33	17.33	4
15	794	1.75	12.5	14.25	4	3	14.33	17.33	5
16	260	0.5	13	13.5	3	–	–	–	–
17	529	0.75	13.75	14.5	2	3	14.58	17.58	9
18	879	2.25	13.75	16	3	3	16.08	19.08	6
19	389	0.75	14.5	15.25	1	3	15.33	18.33	3
20	052	0.5	14.5	15	4	1.5	15.08	16.58	2
21	629	1.25	14.75	16	2	3	16.08	19.08	7
22	760	1.25	15.25	16.5	4	3	16.58	19.58	10
23	418	0.75	15.5	16.25	1	3	16.33	19.33	8
24	234	0.5	16.25	16.75	3	1.5	16.83	18.33	1
25	764	1.25	16.25	17.5	2	3	17.58	20.58	4
26	225	0.5	16.5	17	1	1.5	17.08	18.58	2
27	806	1.75	16.75	18.5	4	3	18.58	21.58	3
28	471	0.75	17	17.75	3	3	17.83	20.83	5
29	601	0.75	17.25	18	1	3	18.08	21.08	9
30	775	1.75	17.75	19.5	2	3	19.58	22.58	1
31	823	1.75	18	19.75	3	3	19.83	22.83	6
32	637	1.25	18.25	19.5	1	3	19.58	22.58	2

Table i Hospital Simulation with 4 Operating Rooms

Patient	Ran. No.	Sur. Len.	Sur. Begin	Sur. Ends	OR	Rec. Len.	Rec. Begin	Rec. Ends	RR
1	389	0.75	7.5	8.25	1	3	8.33	11.33	2
2	997	4	7.5	11.5	2	3	11.58	14.58	2
3	198	0.5	7.5	8	3	1.5	8.08	9.58	1
4	560	0.75	7.5	8.25	4	3	8.33	11.33	3
5	330	0.5	7.5	8	5	–	–	–	–
6	852	1.75	8.25	10	3	3	10.08	13.08	4
7	901	2.25	8.25	10.5	5	3	10.58	13.58	6
8	757	1.25	8.5	9.75	1	3	9.83	12.83	1
9	769	1.75	8.5	10.25	4	3	10.33	13.33	5
10	707	1.25	10	11.25	1	3	11.33	14.33	7
11	745	1.25	10.25	11.5	3	3	11.58	14.58	3
12	784	1.75	10.5	12.25	4	3	12.33	15.33	9
13	608	0.75	10.75	11.5	5	3	11.58	14.58	8
14	796	1.75	11.5	13.25	1	3	13.33	16.33	1
15	794	1.75	11.75	13.5	2	3	13.58	16.58	5
16	260	0.5	11.75	12.25	3	–	–	–	–
17	529	0.75	11.75	12.5	5	3	12.58	15.58	10
18	879	2.25	12.5	14.75	4	3	14.83	17.83	7
19	389	0.75	12.5	13.25	3	3	13.33	16.33	4
20	052	0.5	12.75	13.25	5	1.5	13.33	14.83	11
21	629	1.25	13.5	14.75	1	3	14.83	17.83	2
22	760	1.25	13.5	14.75	3	3	14.83	17.83	3
23	418	0.75	13.5	14.25	5	3	14.33	17.33	6
24	234	0.5	13.75	14.25	2	1.5	14.33	15.83	12
25	764	1.25	14.5	15.75	5	3	15.83	18.83	11
26	225	0.5	14.5	15	2	1.5	15.08	16.58	8
27	806	1.75	15	16.75	4	3	16.83	19.83	12
28	471	0.75	15	15.75	1	3	15.83	18.83	9
29	601	0.75	15	15.75	3	3	15.83	18.83	10
30	775	1.75	15.25	17	2	3	17.08	20.08	1
31	823	1.75	16	17.75	5	3	17.83	20.83	5
32	637	1.25	16	17.25	1	3	17.33	20.33	4

Table ii Hospital Simulation with 5 Operating Rooms

```
% Author: Geneva Porter
% V1 Date: 21 November 2018
%
% This script was created to solve the classic hospital expansion problem
% put forth by Schmitz ans Kwak in their 1971 paper "Monte Carlo
% Simulation of Operating-Room and Recovery-Room Usage." In this paper,
% they outline a simulation based on the number of operating rooms and
% recovery rooms planned for a new hospital wing ICU.

p = input('\n Number of Patients per Day: ');
o = input(' Number of Operating Rooms Available: ');
s = input(' Number of Simulations to Run: ');

A = zeros(s,6); % for storing values from simulations

for x = 1:s

    M = zeros(p,12); % time table for surgeries and recovery
    M(:,1) = (1:1:p)'; % patient number
    M(:,2) = round(1000*rand(p,1)); % assigned surgery

%     M(:,3) = length of operation
%     M(:,4) = operation begins
%     M(:,5) = operation ends
%     M(:,6) = operating room number
%     M(:,7) = operating room is available for next patient
%     M(:,8) = recovery time needed
%     M(:,9) = recovery Begins
%     M(:,10) = recovery Ends
%     M(:,11) = recovery room number
%     M(:,12) = recovery room is available for next patient

    for i=1:p
        if M(i,2) < 158 % probability distribution for surgery type
            M(i,3) = 0.5; M(i,8) = 1.5; % operation / recovery  time
        elseif M(i,2) < 242
            M(i,3) = 0.5; M(i,8) = 1.5;
        elseif M(i,2) < 327
            M(i,3) = 0.5; M(i,8) = 0;
        elseif M(i,2) < 385
            M(i,3) = 0.5; M(i,8) = 0;
        elseif M(i,2) < 621
            M(i,3) = 0.75; M(i,8) = 3;
        elseif M(i,2) < 767
```

```
            M(i,3) = 1.25; M(i,8) = 3;
        elseif M(i,2) < 857
            M(i,3) = 1.75; M(i,8) = 3;
        elseif M(i,2) < 912
            M(i,3) = 2.25; M(i,8) = 3;
        elseif M(i,2) < 946
            M(i,3) = 2.75; M(i,8) = 3;
        elseif M(i,2) < 967
            M(i,3) = 3.25; M(i,8) = 3;
        elseif M(i,2) < 980
            M(i,3) = 3.75; M(i,8) = 3;
        else
            M(i,3) = 4; M(i,8) = 3;
        end
    end

    for i=1:o % operations when OR first opens
        M(i,4) = 7.5; % operation begins
        M(i,5) = M(i,4)+ M(i,3); % operation ends
        M(i,6) = i; % OR number
        M(i,7) = M(i,5) + .25; % OR is available for next patient
    end

    for i = o+1:p % subsequent operations
        [time0,index0] = min(M(1:i-1,7),[],1); % next operation room ready
        M(i,4) = time0; % operation begins
        M(i,5) = time0 + M(i,3); % operation ends
        M(i,6) = M(index0,6); % OR number
        M(i,7) = M(i,5) + .25; % OR is available for next patient
        M(index0,7) = inf; % available OR time will not repeat
    end

    M(:,9) = M(:,5) + .08*ones(p,1); % Recovery Begins
    M(:,10) = M(:,9) + M(:,8); % Recovery Ends
    M(:,12) = M(:,10) + .25*ones(p,1); % RR is available for next patient

    n = inf; % used to weed out patients with no recovery

    for i = 1:p
        if M(i,8) == 0 % eliminating patients who don't use a recovery bed
            M(i,9) = n;
            M(i,10) = 0;
            M(i,11) = 0;
            M(i,12) = n;
```

```
        end
    end

    rm = 0; % number of recovery rooms needed
    for i = 1:p
        if M(i,8) > 0
            [time1,index1] = min(M(:,9),[],1); % next recovery patient
            [time2,index2] = min(M(:,12),[],1); % next recovery room
            if time1 < time2 % there is no used bed available
                rm = rm+1; % new bed is used
                M(index1,11) = rm; % patient is added to new bed
                M(index1,9) = n; % patient will not repeat
            elseif time1 < n % there is a used bed available
                M(index1,11) = M(index2,11); % patient assigned used bed
                M(index1,9) = n; % patient will not repeat
                M(index2,12) = n; % RR ready time will not repeat
            end
        end
    end

    % Subbing back values that were previously changed for convenience
    M(:,7) = M(:,5) + .25*ones(p,1);
    M(:,9) = M(:,5) + .08*ones(p,1);
    M(:,12) = M(:,10) + .25*ones(p,1);
    for i = 1:p
        if M(i,8) == 0
            M(i,9) = 0;
            M(i,12) = 0;
        end
    end

open = ceil(max(M(:,10))); % max nurse shift
T = [(8:1:open-1); (9:1:open); zeros(1,open-8)]'; %patients by hour

for i = 1:length(T) % for each hour that the recovery room is open
    for j = 1:p % for each patient
        if floor(M(j,9)) == T(i,1) % in recovery during ith hour
            T(i,3) = T(i,3)+1/3; % nurse can take 3 patients
            if M(j,8) == 3 % recovery time is 3 hours
                T(i+1:i+2,3) = T(i+1:i+2,3) +[1/3;1/3]; % add next hours
            elseif M(j,8) == 1.5 % recovery time is 1.5 hours
                T(i+1,3) = T(i+1,3) + 1/3; % add next hour
                if floor(M(j,10)) == T(i+2,1) % recovery starts after .5 hr
                    T(i+2,3) = T(i+2,3) + 1/3; % add next hour
```

```
                end
            end
        end
    end
end

T(:,3)=ceil(T(:,3)); % must have an integer number of nurses
for i = 1:length(T) % recovery room must be staffed by at least 2 nurses
    if T(i,3) < 2
        T(i,3) = 2;
    end
end

A(x,1) = max(M(:,5)); % latest time OR is open
A(x,2) = mean(M(:,3)); % average time OR is used
A(x,3) = max(M(:,10)); % latest time recovery room is open
A(x,4) = max(M(:,11)); % recovery beds used
A(x,5) = sum(T(:,3)); % staff hours needed
A(x,6) = max(T(:,3)); % nurses needed at peak times

end

r1 = mean(A(:,1)); std1 = std(A(:,1));
r2 = mean(A(:,2)); std2 = std(A(:,2));
r3 = mean(A(:,3)); std3 = std(A(:,3));
r4 = mean(A(:,4)); std4 = std(A(:,4));
r5 = mean(A(:,5)); std5 = std(A(:,5));
r6 = mean(A(:,6)); std6 = std(A(:,6));

fprintf('\n This hospital sees %d surgical patients per day\n',p);
fprintf(' and uses %d operating rooms. The following are the\n',o);
fprintf(' result of %d simulations: \n\n',s);
fprintf(' The average operating room closure is at %.4f hours, \n',r1);
fprintf(' with a standard deviation of %.4f.\n',std1)
fprintf(' Each operating room is used an average of %.4f hours, \n',r2);
fprintf(' with a standard deviation of %.4f.\n',std2);
fprintf(' The average recovery room closure is at %.4f hours, \n',r3);
fprintf(' with a standard deviation of %.4f.\n', std3);
fprintf(' An average of %.4f beds are used for recovery per day, \n',r4);
fprintf(' with a standard deviation of %.4f.\n', std4);
fprintf(' It takes an average of %.4f staf hours for this ICU, \n',r5);
fprintf(' with a standard deviation of %.4f.\n', std5);
fprintf(' An average of %.4f nurses are needed for peak hours, \n',r6);
fprintf(' with a standard deviation of %.4f.\n', std6);
```

Figure i MATLAB code for Hospital Monte Carlo Simulation