**Triage Activity Instructions**

Contained in this document are directions for conducting the triage activity as part of the Computational STEM Lab workshop.

Necessary Items:

1. The triage slides :these can be found in the “Slides” sub-directory within the “triage” directory. You may want to edit these a bit (we’ll talk more about this later).
2. The Jupyter notebook entitled “triage\_exercise.ipynb” from the “triage” directory.
3. The file entitled “triage\_funcs.py” from the “triage” directory. Note that you will never need to open this file but it must be in the same folder as “triage\_exercise.ipynb” in order for the activity to function properly.
4. A set of index cars with at least two colors, preferably blue and orange OR yellow and red.

Directions:

1. Open the triage slides. (Make sure you’ve looked through the slides several times before actually giving the workshop).
2. Slides 1-2: discuss what a mass casualty event is, and explain that we will be using computers to try and improve the efficacy with which casualties are given help.
3. Slides 3-5: explain what triage is, and then explain that START is one of the most common methods of triage. I would only briefly show slide 5. We’ve changed the Yellow/Red colors to Blue/Orange since a member of our team suffers from color-blindness. Feel free to change them back.
4. Slides 6-7: initiate the emergency. These slides are very Chicago specific so feel free to change them to something more local. If you have trouble altering the slides because you’re unfamiliar with LaTeX/Beamer, please feel free to reach out. Discuss how you will be creating a policy for the local trauma unit since there has just been a local mass-casualty event.
5. Slide 8: Discuss how an admission policy will need to be designed. Most of the relevant information is on the slide.
6. Slide 9: Before advancing past the first line of text “Only admit…” it is a great idea to allow students to “think, pair, share”. I.e. allow them to discuss possible policies amongst themselves and then contribute their own ideas. Once you’ve done this for a few minutes proceed to discuss the listed policies. I you feel confident enough to discuss it, talk about how an active area of research is coming up with optimal admission policies where your admission decision is dependent on the time a patient arrives, the number of available beds, and their triage status.
7. Slide 10: Begin discussing how we will build a model to help us decide which policy is best. Again ask the class what kind of information they would like to have in order to help them make a decision and have them think, pair, share before moving on and discussing each of the individual bullet points.
8. Slide 11: This slide is optional and I typically reserve it for the end. This is basically an example of a paper doing research into this exact problem.
9. Open the “triage\_exercise” notebook with your class.
10. You want to have the following general road map in your heard throughout this exercise. We are going to slowly build up a model for triage by:
    1. Defining how patients arrive.
    2. Defining which patients survive.
    3. Putting it all together with a walkthrough (using the cards).
    4. Running the model many times in the notebook.
11. First allow students to put their name into the notebook and discuss how the first box of code needs to be run in order to import the necessary packages.
12. Discuss how you will now be developing a model for how patients arrive. The first block of code in this section will create a black curve. This represents how the intensity of patient arrivals changes over time. Try an explain what this means to the students and have them try out many different values of *num\_pat* and *peak* to see how it changes the shape of the curve.
13. After this is complete start discussing how the IMMEDIATE and DELAYED patients may have different arrival patterns. An interesting thing to mention is that IMMEDIATE patients typically arrive a little bit later than DELAYED patients. This is due to several reasons (e.g. IMMEDIATE patients are more likely to be trapped under rubble sometimes DELAYED patients can be transported to the hospital without an ambulance and don’t need to wait for one to become available, etc…).
14. Have students play with the four parameters in this code block and see how it changes the two curves.
15. The code block after this will generate samples of arrival from these intensity curves. This may be a good time to emphasize why we may want to run simulations. Just because we know the intensities of arrivals doesn’t mean we know exactly when each patient will arrive.
16. We will now discuss how we decide whether a patient survives or not. You should emphasize why we need the computer to generate random numbers for us and that humans are really bad at being truly “random”. To demonstrate open the link to the mindreader app included in the notebook. Once open have a one or two students try and beat the mind reader. The basic idea is that the students will try and alternate pressing left and right without forming any sort of pattern which the computer will be able to guess. This is really hard for humans to do. I recommend that you play with this website a bit before you actually open it in the workshop as the interface can be a bit confusing the first time you look at it. Also I recommend selecting the hardest difficulty level (a combination of the fastest and smartest algorithms).
17. The next code block shows how we can generate a sequence of numbers representing which patients survive or not. Have the students play with the *prob* parameter and see how it changes the output. Also emphasize that even if they don’t change *prob* the output will change every time. Ask why this is and then explain the concept of random numbers.
18. We will now put this all together in a simulation. Lay out the index cards with some writing implements. Have each student (and each instructor except yourself) to run the first code block. It give them directions for which color to choose and tell them to write number on the front, and a 0 or 1 on the back. We will now do a walkthrough of a simulation. Each person represents a patient, with the color of their index card representing their injury severity, the number of the front representing the time they arrive at the hospital, and the 0 or 1 on the back representing whether they survive if they turned away.
19. The walkthrough:
    1. First set aside some number of desks or chairs representing the hospital beds. I typically try to aim for roughly 1/3 the number of beds vs the number of people with cards.
    2. Have all the people with cards line up in order. Namely have them line up so that people with lower numbers are at the front of the line and people with larger numbers are at the back of the line.
    3. Now walk through the simulation doing FCFS first. For example, for FCFS, assign people in line to beds until there are not more free beds and assign everyone else to the other side of the room. Then have the everyone with a 1 raise their hands and count how many survivors there are (including those in beds).
    4. Repeat this process for ORANGE ONLY. Walking through the line only assigning the ORANGES to beds until there are no more free beds. Repeat the process of counting survivors and compare with the FCFS policy. Sometime there will be free beds when using ORANGE ONLY. If this is the case make sure to point it out.
20. Now explain that this was just one simulation and in order to truly compare the policies we would want to walk through this simulation many, many times. This can be accomplished by running simulations on the computer. The second code black will do just that. Allow the students to play with the parameters in this code block but beware if they set a number too large of *num\_reps* then it may take a long (OR REALLY LONG) time to run.
21. Discuss which policy works better and instruct the students to see if they can find different combinations of parameters where each policy is better.
22. The final code black isn’t too important but will display images of the output of the simulations run above.