ISyE 6644 — Spring 2021 — Mini-Projects 1!

Timeline:

- **M 2/8/21**: Suggested topics and ground rules posted.
- **F 2/12/21**: Select topic and group members (handled as an assignment in Canvas)
- **M 3/8/21, 11:59 pm**: Turn in Mini-Project 1.

Some ground rules:

- Pick *one* project that you find interesting. The projects are all intended to be a little open-ended, but do not pick a project that will kill you time-wise!
- You are allowed to substitute any other interesting topic of your own choice as long as you run it by us first (scroll to the end for some details).
- You are allowed to work in groups that are reasonably sized, though you will have to form the groups on your own. (Suggested group sizes are given in the text of each problem.)
- The topics range from computer-intensive to more statistically oriented. When computer work is necessary, you are allowed to use whatever computer languages you'd like, e.g., R, Minitab, Matlab, Python, even Excel.
- Good luck, and have fun!

Some Suggested Projects for ISyE 6644:

Applications-Oriented Problems

1. (2 group members) The African Tanker Problem. If you like tricky little problems, then this is the one for you! This famous exercise concerns the simulation of a somewhat complicated port, and is described in Law (2015) (Chapter 2 exercises). You should model this and conduct a little more output analysis than is asked for in the text of the problem.

2. (2 members) Go to some intersections and model a couple of *traffic lights*. Collect some real data. Try to answer as many interesting questions as you can think of. For instance, what would the effect of synchronization be? Or, how do cell phones affect driver behavior? Justify everything you conclude by conducting careful output analysis. This could be a great project!

3. (2–4 members) Run a simulation having direct applicability for *your job*. Some people in the class have access to interesting "real-world" problems, either through their jobs, or perhaps as an assignment in another course here at Georgia Tech. If you'd like to do a full-scale simulation analysis on a particular nontrivial simulation (no M/M/1 queues, please!), then this project may be the one for you. I would expect some input analysis (i.e., choosing input distributions, planning run-length, etc.), some modeling and programming, and some output analysis (e.g., confidence intervals). This would be a good project for a group with members having diverse interests.

4. (2–3 members) *Pandemic Flu Spread.* Consider a classroom of 21 elementary school kids. 20 of the kids are healthy (and susceptible to flu) on Day 1. Tommy (the 21st kid) walks in with the flu and starts interacting with his potential victims. To keep things simple, let's suppose that Tommy comes to school every day (whether or not he's sick) and will be infectious for 3 days. Thus, there are 3 chances for Tommy to infect the other kids — Days 1, 2, and 3. Suppose that the probability that he infects any individual susceptible kid on any of the three days is $p = 0.02$; and suppose that all kids and days are independent (so that you have i.i.d. Bern(p) trials).

   If a kid gets infected by Tommy, he will then become infectious for 3 days as well, starting on the next day.

   (a) What is the distribution of the number of kids that Tommy infects on Day 1?

   (b) What is the expected number of kids that Tommy infects on Day 1?

   (c) What is the expected number of kids that are infected by Day 2 (you can count Tommy if you want)?

   (d) Simulate the number of kids that are infected on Days 1,2,.... Do this many times. What are the (estimated) expected numbers of kids that are infected by Day i, $i = 1, 2, \ldots$? Produce a histogram detailing how long the "epidemic" will last.

Language- and Modeling-Oriented Problems

5. (3 members) Learn and implement the self-contained simulation language SIMLIB from Chapter 2 of the Law (2015) text. Although SIMLIB is not the most-efficient simulation language around, it will really teach you how an actual simulation language is written. In your project write-up, give your comments and opinions on the language, provide an easy user's guide, and demonstrate how SIMLIB works on one or two example models. Note: I'll allow you to replicate the functionality of SIMLIB using any language such as C++, Python, etc.

6. (2 members) A different modeling paradigm. Read and comment on Lee Schruben's fundamental paper, "Simulation Modeling with Event Graphs," *Communications of the ACM*, Volume 26, pp. 957–963 (1983). Then do the same for the Sargent (1988) and Som and Sargent (1989) articles cited in the References section of the Law (2015) text. (The latter two papers extend and refine the event graphs technique, and they provide lots of nice examples.)

Programming-Oriented Problems

7. (1 member) Let's *gamble*! Implement a blackjack simulation; in fact, simulate a few reasonable strategies. Statistically analyze these strategies by playing lots of independent blackjack games. Then determine which choice maximizes your profit. (Assume that the House plays by the usual rules.) Include all code and commentary. Be creative!

8. (1–2 members) Let's gamble some more! Same thing as in Problem 7, except for solitaire. (This isn't easy, but it's fun.)

9. (1–2 members) Another game. There are two players, A and B. At the beginning of the game, each starts with 4 coins, and there are 2 coins in the pot. A goes first, then B, then A,. . . . During a particular player's turn, the player tosses a 6-sided die. If the player rolls a:

   - 1, then the player does nothing.
   - 2: then the player takes all coins in the pot.
   - 3: then the player takes half of the coins in the pot (rounded down).
   - 4,5,6: then the player puts a coin in the pot.

   A player loses (and the game is over) if they are unable to perform the task (i.e., if they have 0 coins and need to place one in the pot). We define a cycle as A and then B completing their turns. The exception is if a player goes out; that is the final cycle (but it still counts as the last cycle). We are trying to determine the expected number (and maybe even the distribution) of cycles the game will last for. I'm guessing that you can use "first-step" analysis to get the expected value. Simulation seems the easiest thing to do to get the entire distribution.

Theory-Oriented Problems

10. (2 members) Work on generalizations of the "Birthday Problem" that we did in class. E.g., what's the probability that at least *three* kids in the room will have the same birthday? What happens when the birth patterns change a bit over the year (e.g., more babies born in the spring)?

11. (2 members) Prepare a tutorial on Monte Carlo integration techniques and give several examples.

And...

12. **Any Other Problem That You Choose** — as long as you can convince us. Succinctly describe the problem and why you think it would be cool for this class. Needs to be nontrivial, useful, and interesting.