MATH 20 – LAB 1

due Wednesday, August 2

Instructions: Complete all sections. Start this lab **early**, so you have plenty of time to ask questions.

Each experiment asks you to submit code. You must name the files as instructed and submit them with your completed assignment. Please comment your code!

The experiments ask you to perform some kind of simulation and describe your results. When describing, use the data you generated to make guesses at the expected value, etc. What does the distribution look like? Do you see a pattern?

You don't have to write a lot for each part! Just a few sentences (with pictures if necessary) will suffice.

Submission: You will submit your written answers (without your code) in class as a single document on Wednesday, August 2. **This must be typed.** It is okay to leave blanks for math equations and handwrite those if you need them. All graphs involved should be generated by R. Submit your code, in three separate files as outlined below, to me via email, with the subject line `Math 20 Lab 1 - [your first and last name]`. Your code should be submitted by the time class starts on August 2. (It's fine to send it in earlier, too.)

EXPERIMENT 1 — STUBBORN CANDLES

Today is Archer's 35th birthday. His friends have baked him a cake and placed on it 35 lit candles. After the traditional ritual singing, Archer tries to blow out the candles. Each time he blows, he successfully extinguishes between one and N candles (inclusive), where N is the number of currently lit candles. Each of these N outcomes is equally probable.

For example, after his first attempt there are somewhere between 0 and 34 candles that remain lit, and each of these occurs with equal probability; he had $1/35$ chance of blowing out just one, $1/35$ of two, etc. After his second attempt (if it was necessary), there are somewhere between 0 and 33 candles that remain lit, though now these are no longer equiprobable events.

(a) Write an R program that simulates this. Run this simulation many times, recording the number of attempts needed to extinguish all the candles. Describe your findings, including some kind of plot that demonstrates your conclusion.

(b) Edit the previous R program so that you can input the age of the person as the variable N and perform the same simulation (you probably want to use functions for this). Pick a few values of N (they don't have to actually be ages of any human!) and perform the simulations. Describe your findings, including some kind of plot.

(c) Use the code from the last step to do the following: For each N value in the set $\{1, 2, \ldots, 100\}$, run the simulation many times. Plot all the results together in some way. (There are many options on how to represent information in a graph. Pick one that you think works best for this kind of problem!) What seems to be the behavior of $\mathbb{E}[\text{number of attempts}]$ as N increases?

(d) Submit your code for this problem in a file named `candles-yourlastname.R` (for example, mine would be `candles-pantone.R`). Make sure you've added comments so I understand what it does and how to run it.

Experiment 2 — Unicellular Replication

Lana is a microbiologist who studies colonies of unicellular organisms. She knows that a particular organism has the following behavior. Once per minute, the cell either divides into two identical cells (with probability p) or simply dies (with probability $1 - p$).

She starts a colony with a single organism, and observes the resulting population over time. For example, if $p = 1/2$, then after the first minute there is a 50% probability that there are two organisms and a 50% probability there are no organisms. The actions of each organism are independent of the actions of each other organism.

(a) Pick a few p values (some extreme, some moderate, etc), and simulate the population of the colony over time. (To be more specific, play around and try to determine how long you should simulate for to get a good idea what's going to happen.) Describe your results.

(b) In the last step, you should have discovered that for some value of p, the probability that the colony goes extinct is virtually 1 (more precisely, as time goes to infinity, it will always go extinct), while for other values of p the colony sometimes goes extinct and sometimes doesn't. Use your program to estimate, for each p value in the set $\{0, 0.01, 0.02, \ldots, 0.99, 1.00\}$, the probability that the colony goes extinct. Describe your results, including some kind of reasonable plot.

(c) Submit your code for this problem in a file named `colonies-yourlastname.R`.

Experiment 3 — The Real Number Casino

Cyril, a casino mogul, has invented a new casino game. The problem is that he has no idea what the expected payout is, so he doesn't know what to charge for it! The game works like this. The player starts with a score of 0. The computer spits out a random real number between 0 and 1 inclusive.[1] The number is added to the players score and the player gets \$10. For as long as the player's score is not 1 or larger, this is repeated—the computer picks a random real number between 0 and 1, adds it to the players score, and the player gets \$10.

For example, suppose the computer outputs the numbers 0.2153, 0.451148, 0.71. The player wins \$30, and then the game stops because the total score is 1 or larger.

(a) Write a simulation to estimate the expected payoff of this game.

(b) What if, instead, the real number returned by the computer is squared before it's added to the players score? Now what is the expected payoff?

(c) Invent some other variant of this game (add a rule, change the payout structure, etc) and simulate the expected payout now. While you're at it, come up with a snappy name for this game!

(d) Describe your results from above, including some kind of plot if appropriate, and submit your code in a file named `casino-yourlastname.R`.

[1] This number is chosen *uniformly at random*, which just means that each real number is equally likely to be chosen. When we cover distributions, we'll talk more about what this means. For now, checkout out the `runif` command in `R`.