Consider picking tickets at random from a box (so each ticket in the box is equally likely each time you pick). Suppose the box has *four* tickets marked 0, 1, 2, and 3 respectively.

Let A be the event that the *first pick* yields an even number; B be the event that the *second pick* is greater than or equal to one.

1. Pick two numbers without replacement. Find $\mathrm{P}(B \mid \text{first pick is } 0)$.

2. Pick two numbers without replacement. Find $\mathrm{P}(B \mid \text{first pick is } 2)$.

3. Pick two numbers with replacement. Find $\mathrm{P}(B \mid A)$.

Consider a fair, eight-sided die.

4. I roll the die four times. What is the probability that I roll the **same** number on all four rolls?

5. I roll the die twice. What is the probability that the rolls are **different**?

6. My dog Bella has two toys that she loves: an orange ball, and a thick rope. Each time she picks out a toy, she chooses it independently of all the other times (like a coin toss). That day she picked out a toy three times. Let event be A the event that Bella picked the rope *at most* one time and let event B be the event that the toys she picked that day included *both* the rope and the ball. Are A and B independent?

A European roulette wheel has thirty-seven pockets, *numbered 0 through 36*. Eighteen of these pockets are red, eighteen are black, and one is green: pocket number 0. In each round, the wheel is spun and a white ball lands in one of these 37 pockets.

7. What is the probability of getting at the ball landing in the green pocket *at least once* in five spins of the wheel?

8. Write R code to simulate five spins of this wheel.

9. Create three vectors: one which contains 100 simulated spins of the European roulette wheel (call this `one_hundred`), one which contains 1,000 such spins (call this `one_thousand`), and another which contains 10,000 such spins (call this `ten_thousand`). Write the code to create the `one_hundred` vector below.

10. Create a new vector that returns `TRUE`/`FALSE` values for each element in `one_hundred`, where `TRUE` means that the ball landed in the green pocket, and save it. Repeat these steps with the `one_thousand` and `ten_thousand` vectors. Write the code where you worked with the `one_hundred` vector below.

11. In each of the three simulations, find the proportion of spins that landed in the green pocket (write the code and the proportion). *Hint: how can you take a proportion of a logical vector?*

12. Comment on how the proportions changed with respect to the true probability of landing in the green pocket (which you calculated earlier) across the three simulations.