Classic Riddler

29 May 2020

Riddle:

One Friday morning, suppose everyone in the U.S. (about 330 million people) joins a single Zoom meeting between 8 a.m. and 9 a.m.—to discuss the latest Riddler column, of course. This being a virtual meeting, many people will join late and leave early.

In fact, the attendees all follow the same steps in determining when to join and leave the meeting. Each person independently picks two random times between 8 a.m. and 9 a.m.—not rounded to the nearest minute, mind you, but *any* time within that range. They then join the meeting at the earlier time and leave the meeting at the later time.

What is the probability that at least one attendee is on the call with everyone else (i.e., the attendee's time on the call overlaps with every other person's time on the call)?

Extra credit: What is the probability that at least *two* attendees are on the call with everyone else?

Solution:

My strategy from the beginning was to turn this into a simulation, since trying to extract an analytic solution, even for 3–4 people seemed extremely difficult. My thought was to create two arrays, one for the start time and one for the end time of all participants (starting with 2 and eventually 330 milllion). Then I could run through the arrays and compare the start and end times one-by-one against the end and start times of all other participants. The code I used is found in `zoom_call.C`.

To perform the simulation, I just needed to generate lots and lots of random numbers. The numbers were all between 0 and 1, of course, since it is a straightforward linear transformation to get a time between 8:00 and 9:00. For my notation, I designated x_i as the start time and y_i as the end time of participant i. In the code, I first assign all the numbers, and always make sure that $x_i < y_i$.

At some point while playing with and trying to optimize the code, I realized that instead of checking every start time against every end time (and vice versa), I could just keep track of the latest start time (x_{max}) and earliest end time (y_{min}). That way, I only have to run through the participant list once and do the checks at the same time. The two necessary conditions for success that I needed to check for each participant are $x_i < y_{min}$ and $y_i > x_{max}$. This means that participant i logged on before the first person to log off did so, then logged off after the last person to log on did so. So I made these two checks at the same time in the code.

Another realization was that I didn't need to make this check for every single participant. Obviously, once I found one participant, I could call it a success and quit. But I also became aware of the conditions available to be successful. I decided to track the participant with the longest call, since that participant was most likely to overlap all other calls. But this in itself wasn't enough to get the most overlap. Other calls could be shorter in total length, but extend earlier or later than the longest call. This means they had the chance to see callers not seen by the longest caller. So I only bothered to check the two success conditions if a participant 1) had the longest call, 2) started before the longest caller, or 3) ended after the longest caller. Any other participants had shorter calls and didn't see as many other callers as the longest caller.

Here I was satisfied with my code logic, and the only thing to do was to try it many times. I decided that 10 million trials should work to get a statistically meaningful result, but out of caution, I also decided to limit that to 1 million for at least 2 thousand participants. It turns out I didn't try that many participants, anyway.

When I started out with the easiest test, 2 participants, the output result was clearly $^2/_3$. Good, it looked like a simple answer for a simple problem, and it was not unreasonable. (Note: I didn't actually check this result with an analytic solution.) When I moved to 10, 100, and 1,000 participants, the simulation started to take a couple of minutes, but the result stayed the same. I didn't (and still don't) see anything wrong with the logic or the calculation, so it seems that the answer for any number of participants is $\boxed{\mathbf{^2/_3}}$.