LARSON—MATH 353–CLASSROOM WORKSHEET 30
Optimal Stopping & Variance

1. Log in to CoCalc.

   (a) Start the Chrome browser.

   (b) Go to `https://cocalc.com`

   (c) Login (**your VCU email address** is probably your username).

   (d) You should see an existing Project for our class. Click on that.

   (e) Click "New", then "Worksheets", then call it **c30**.

Optimal Stopping.

A coin is flipped n times. Let s_n be the number of Heads that come up. At any point you can stop and collect $\frac{s_n}{n}$ dollars (so the max would be \$1). If you get a Heads on the first flip you should stop. If you get Tails on the first flip, you should flip again.

What *strategy* should you use to decide whether it is optimal to stop (whether you would win the *expected* maximum if you stopped at that point).

Zeilberger notes that the game must stop after some finite number N of flips. Write code that simulates the average maximum value of $\frac{s_n}{n}$ given that k heads and m tails have already been flipped (so there are $N-k-m$ flips remaining and we know $\frac{s_n}{n} \geq \frac{k}{k+m}$ We also know the expected average maximum value when $N=1$ is $\frac{1}{2}$).

2. We will need to produce sequences of "coin flips". We may as well use 0s and 1s. Let's let 1 be "heads" and 0 be "tails".

   Write a function `coin_flips(n)` that produces a list of n random 0s and 1s. Test it!

3. Write a function `max_ratio(flips)` that inputs a string of heads and tails and outputs the maximum value of $\frac{s_n}{n}$ for any position n in the string.

   Test it for strings where the true value is easy to compute.

4. Now write a function `max_ratio_extended(k,m, flips)` that takes integers k, m and a string of heads and tails and outputs the maximum value of $\frac{s_n}{n}$ for any position n in the string, starting from k (previous) heads and $k+m$ (previous) flips (so the input flips are assumed to be new—past the point where the previous flips occurred).

5. Now write a function `experiment_optimal(k,m,N)` that takes integers k, m and N (with $N \geq k+m$), produces $N-k-m$ random flips and outputs the maximum value of $\frac{s_n}{n}$ for any position n in the string, starting from k (previous) heads and $k+m$ (previous) flips (so the input flips are assumed to be new—past the point where the previous flips occurred).

6. Test this function by repeating `experiment_optimal(0,0,1)`. What should you get?

7. Test this function by repeating `experiment_optimal(1,0,1)`. What should you get?

8. Test this function by repeating `experiment_optimal(0,1,1)`. What should you get?

9. Repeat `experiment_optimal(0,0,2)`. Can you explain the output?

10. How can we use information that we get from our simulations to design an optimal stopping *strategy*?

Variance & Standard Deviation

Suppose you flip a coin 100 times. You'd expect 50 heads. You'd also expect to rarely get exactly 50 heads.

11. What is the *average* difference from 50?

12. What can you say about the *distribution* of differences?

13. **Getting your classwork recorded**

    When you are done, before you leave class...

    (a) Click the "Make pdf" (Adobe symbol) icon and make a pdf of this worksheet. (If Cocalc hangs, click the printer icon, then "Open", then print or make a pdf using your browser).

    (b) Send me an email with an informative header like "Math 353—c30 worksheet attached" (so that it will be properly recorded).

    (c) Remember to attach today's classroom worksheet!