Vamshi Jandhyala

Appeasing the Cherry Blossom Horde

4 April 2025

A nice puzzle by Xavier Durawa for the week of 3/30.

Puzzle

This weekend marks the peak bloom for D.C.'s cherry blossoms, and as always, there's a ton of foot traffic around the Tidal Basin, a circular-ish body of water just off the Potomac.

In an alternate universe, the D.C. government decided (questionably) to build a floating walking path straight across the middle of the basin. The idea was to help with pedestrian traffic flow, even if it meant ruining some of the scenic views.

While the path was being constructed, the parks department needed to fence it off to prevent premature use, so they put up a temporary barrier somewhere across the basin. The only rule: the barrier had to intersect the floating path, effectively blocking it. Other than that, the barrier's placement was completely random. (For simplicity, model the Tidal Basin as a perfect circle, the floating path as a diameter, and the barrier as a random chord that intersects the diameter.)

Now that construction is done, you, a humble parks worker, are on your way to remove the barrier and finally open the path to the public. Halfway there, you realise you forgot the map. The map would have told you which of the two ends of the floating path the barrier is closer to, so you could have picked the shorter walk.

You are not sure if it is worth turning back, so instead you ask: what is the expected ratio of the shorter segment of the path to the longer one, given that the barrier randomly intersects the path? In other words, if the barrier cuts the floating path into two segments, what is the expected value of

$$\frac{\text{length of shorter segment}}{\text{length of longer segment}}?$$

Brute force simulation

As we are dealing with ratios, assume the radius of the circle is 1 and the path is the x-axis. Given that the barrier intersects the path, we can randomly generate one point on the upper semicircle and one on the lower semicircle, calculate the point of intersection of the chord with the x-axis, and the lengths of the two segments. Repeating this process a million times yields an estimate of the expected ratio of the shorter segment to the longer.

The Python code below performs the simulation; the expected value of the ratio is 0.5619.

Python code

```
import numpy as np

def generate_points_and_calculate_ratio(num_samples=1000000):
    ratios = np.zeros(num_samples)
    for i in range(num_samples):
        theta1 = np.random.uniform(0, np.pi)
        theta2 = np.random.uniform(np.pi, 2 * np.pi)
        x1, y1 = np.cos(theta1), np.sin(theta1)
        x2, y2 = np.cos(theta2), np.sin(theta2)
        if x2 - x1 != 0:
            m = (y2 - y1) / (x2 - x1)
            b = y1 - m * x1
            x_intersect = -b / m
            y_intersect = 0
        else:
            x_intersect = x1
            y_intersect = 0
        d1 = np.sqrt((x_intersect - x1)**2 + (y_intersect - y1)**2)
        d2 = np.sqrt((x_intersect - x2)**2 + (y_intersect - y2)**2)
        shorter = min(d1, d2)
        longer = max(d1, d2)
        ratios[i] = shorter / longer
    return ratios

ratios = generate_points_and_calculate_ratio(num_samples=1_000_000)
mean_ratio = np.mean(ratios)
print(f"Expected value (mean) of the ratio: {mean_ratio:.6f}")
```

Analytical solution

Let AB be the walking path and CD be the floating barrier. For every chord CD, we have a corresponding triangle OCD with angles $\alpha = \angle OCD$ and $\beta = \angle ODC$ satisfying $0 < \alpha, \beta \leq \pi/2$.

By the sine rule, the ratio of the shorter segment to the longer segment is

$$\frac{\sin\alpha}{\sin\beta}, \qquad \alpha < \beta.$$

Let A be the random variable for α and B be the random variable for β. They are independent uniform random variables on $[0, \pi/2]$, with densities $f_A(\alpha) = 2/\pi$, $f_B(\beta) = 2/\pi$ and joint density $f_{AB}(\alpha, \beta) = (2/\pi)(2/\pi)$. The expected value of the ratio is

$$\begin{aligned}
\mathbb{E}\Big[\frac{\sin A}{\sin B}\Big| A < B\Big] &= \frac{\mathbb{E}\Big[\frac{\sin A}{\sin B}\cdot \mathbf{1}_{A<B}\Big]}{\mathbb{P}[A<B]} \\
&= \frac{\int_0^{\pi/2}\int_0^{\beta} \frac{\sin\alpha}{\sin\beta}\frac{2}{\pi}\frac{2}{\pi}\,d\alpha\,d\beta}{1/2} \\
&= \frac{8}{\pi^2}\int_0^{\pi/2}\frac{1-\cos\beta}{\sin\beta}\,d\beta \\
&= \frac{8}{\pi^2}\log 2 \;=\; 0.5618.
\end{aligned}$$

Simulation using a simplified approach

Sampling α, β uniformly on $[0, \pi/2]$ directly gives the expected ratio 0.5625, very close to the theoretical value as expected.

Python code

```
import numpy as np

def monte_carlo_conditional_expectation(num_samples=1_000_000):
    x = np.random.uniform(0, np.pi / 2, num_samples)
    y = np.random.uniform(0, np.pi / 2, num_samples)
    indicator = x < y
    ratio = np.zeros_like(x)
    ratio[indicator] = np.sin(x[indicator]) / np.sin(y[indicator])
    return np.mean(ratio[indicator])

print(monte_carlo_conditional_expectation())
```