ICMA393 Discrete Simulation: HW1

Mahidol University International College
Austin Jetrin Maddison
September 25, 2024

1. Birthday Problem

The simulations ran were set to $seed = 27$ and $N = 400000$.

a)

$$p_{10} = 0.11686$$
$$CI = [0.11685, 0.11885]$$

b)

$$p_{20} = 0.41126$$
$$CI = [0.41125, 0.41430]$$

c)

$$p_{30} = 0.70595$$
$$CI = [0.70594, 0.70877]$$

d) Found n that satisfies the condition $p_n \geq 0.5$ from running the simulation on a range of n's [0, 50] and selecting the first $p_n \geq 0.5$.

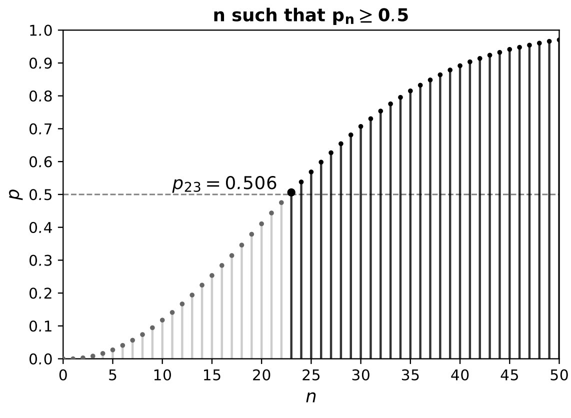

Figure 1: The first point $\geq p = 0.5$ was $p_{23} = 0.506$ which satisfies the condition.

e) We can model p_n using a product function. We do this by getting the product of probabilities where we always pick somebody with a unique birthday. Then we invert the result by minus 1 to get the probability of picking somebody with a non-unique birthday.

$$p(n) = 1 - \prod_{i=0}^{n} \frac{365 - i}{365}$$

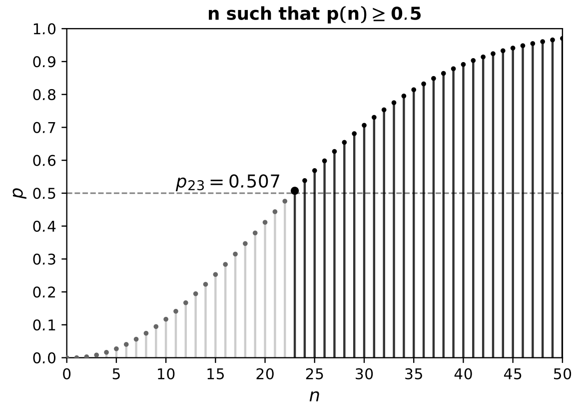

Figure 2: $p(n)$ results are similar to the simulation. The difference between simulation p_n and $p(n)$ is MSE = 2.013×10^{-7}.

For all intensive purposes $p(n)$ models the results in simulation p_n.

2. Alice and Bob Play a Game

The strategy I let Alice have is the following...

Case 1: Found Exclusive Output

She presses the button recording n output values $x_1, x_2, \ldots x_n$. As she records each output she checks whether x_i is exclusive to one of the buttons output range. If x_i is exclusive she returns the corresponding button as the answer.

Case 2: No Exclusive Output

If she presses the button n times with no exclusive output appearing. She calculates $\bar{x}$ and finds the minimum difference between the mean output range of the 2 buttons. She returns the corresponding button that gets the minimum distance.

Psuedocode

Although the source code differs because loops are removed for speed, the idea is the same.

```
xs = []
for i in range(0, n):
    x_n = button_unknown.get_next_value()
    xs.append(x_n)

    # Case 1: Found exclusive output.
    if(x_n == 1)
        return 1  # it is button 1
    if(x_n == 100)
        return 2  # it is button 2

# Case 2: No exclusive output. Evaluate the
    minimum distance of means.
x_mean = sum(xs) / n
button_1_mean = (1 + 99)/2
button_2_mean = (2 + 100)/2

return argmin([abs(x_mean - button_1_mean), abs(
    x_mean - button_2_mean)]) + 1
```

Find n such that Alice is correct ≥ 0.99

I sampled Alice's strategy varying n between $[0, 500]$ and extracted the first n that results in probability atleast 0.99. I set the $seed$ = 27 and ran 400000 trials for all n's.

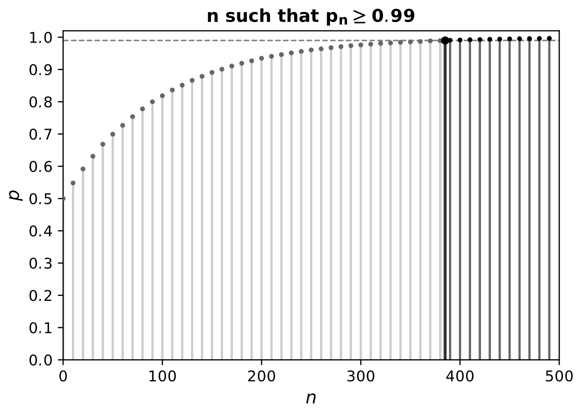

Figure 3: The first point to ≥ p = 0.99 was p_{384} = 0.990 which satisfies the condition.

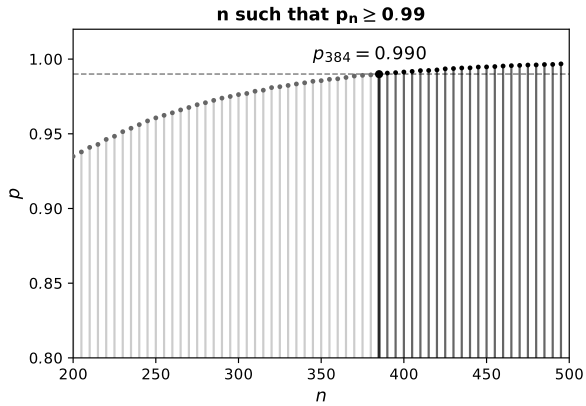

Figure 4: Crop of Figure 3.

3. Practice with Uniform and Geometric Distributions

First find E[X] and V[X] of the geometric distribution. Since X is a fair dice $p\{X = 5\} = 1/6$.

$$E[X] = \frac{1}{1/6} = 6$$
$$V[X] = \frac{1-p}{(1/6)^2} = 30$$

Then I do the same for uniform distribution of Y while equating the corresponding known values from X.

$$E[Y] = \frac{n-1}{2} = 6$$
$$V[Y] = \frac{(n^2-1)}{12} = 30$$

Since n is determined by the setting of a and b we can rewrite it like this.

$$E[Y] = \frac{(b-a+1)-1}{2} = 6$$
$$V[Y] = \frac{(b-a+1)^2-1}{12} = 30$$

Now it's easy to find a and b by solving the system of equations. However I am not going to do that because that is not how I found it at first.

What I did is rearranged $E[Y]$ such that $a+b = 12$. Then I did a search of pairs of a, b st, $a + b = 12$ and $a < b$ until I got $V[Y] = 30$.

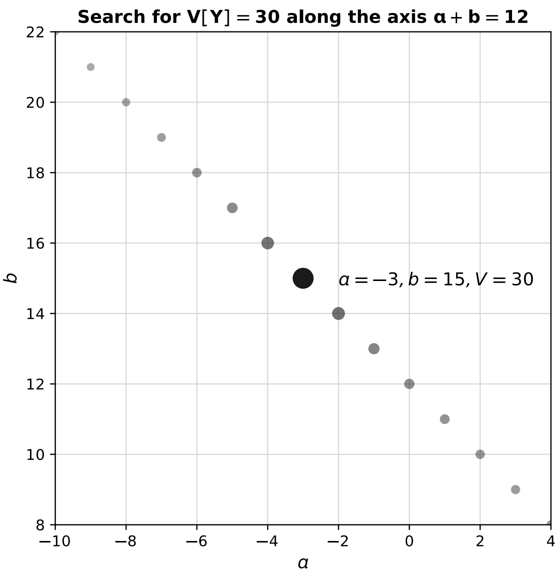

Figure 5: The size and opacity of the discs is the inverse distance away from V[Y] = 30.

For $E[X] = E[Y]$ and $V[X] = V[Y]$,

$$a = -3, b = 15$$

Source Code

https://github.com/AustinMaddison/discrete-simulation/tree/main/src/hw1