Scott Fullenbaum
Math 87 HW 6
Due 3/09/22

---

1. Let's look at the impact of a juror's likelihood to vote correctly and the chance of a wrongful conviction. A jury consists of 20 members, and each member must vote either guilty or not guilty. Assume that the vote of each member is independent. Suppose that the probability that a juror votes correctly (i.e. in line with the actual innocence/guilt of the person on trial) is 60%.

   (a) What is the probability that there is an inconclusive (tied) voting outcome?
   If tied, then there are 10 people that vote guilty (or vice versa). This is a binomial distribution with $p = 0.6$. Let X be the random variable representing the number of guilty votes. $P(X = 10) = \binom{20}{10}(0.6)^{10} * (0.4)^{10} = 0.117$, which is the probability of a tie.

   (b) If there is an inconclusive verdict, then a judge breaks the tie. Suppose a judge votes correctly 70% of the time. What is the probability of a tie AND a judge voting correctly?
   It is reasonable to assume that how the Judge votes is independent from the jury votes, if that point is reached. So, we can take the product of $0.7 * 0.117$ to get P(tie and judge correct) = 0.082

   (c) A verdict is reached if at least 11 jury members vote the same way, or a tie is broken in the manner given above. What is the probability that the correct verdict is reached?
   If at least 11 vote correctly, then can write that sum as $\sum_{n=11}^{20} \binom{20}{n} 0.6^n 0.4^{20-n} = 0.755$. Then, we have to account for chance of correct verdict under a tie, which is $P(X = 10)$ from part a, then multiplying by 0.7. So, add 0.82 and get the final answer of making a correct verdict as 0.837.

   (d) Of the times where the jury reaches the incorrect verdict, suppose 60% of people on trial are guilty but not convicted, and 40% are innocent but convicted. What is the probability of a wrongful conviction?
   The probability of an incorrect conviction is $(1 - p(correct)) = 1 - 0.837 = 0.163$. Multiply this by the proportion that were innocent but convicted, which is 0.4, and you get the probability of a wrongful conviction as 0.065.

   (e) What is the expected number of people wrongfully convicted in 5 million trials?
   We can treat this as a binomial distribution, where X is a random variable representing number of people wrongly convicted. So X is a binomial distribution, with n = 5000000 and p = 0.065. From the definition of a binomial distribution, the $E(x) = np = 325000$

   (f) Write a python function that takes in the likelihood of a juror to vote correctly and returns the probability of a wrongful conviction. You may use scipy.stats.binom or compute your own binomial distribution. Include your code in your report.
   So we just have to generalize our results from the previous parts, meaning we find p(correct), then use that to find p(incorrect) (Code on next page).

```
def binom(p):
  p_tot = 0
  for i in range (11, 21): #probability of correct conviction at x = i
    x = scipy.special.binom(20, i)
    p_comp = (p**i)*(1-p)**(20-i)
    combine = x*p_comp
    p_tot = p_tot + combine
  tie = scipy.special.binom(20,10)*(p)**10*(1-p)**10 #finds p(x = 10)
  p_tot = p_tot+0.7*tie
  return (1-p_tot)*(0.4) #P(wrongful, which is returned)
```

(g) Test your function on likelihoods 50%, 60%, 70%, 80%, and 90%. Report both the probability of a wrongful conviction, and the expected number of people wrongfully convicted in 2 million trials. Discuss your results.
Quick note, I rounded probabilities to 3 decimal places when those weren't all zeroes, and expected value to 2 decimal places. Expected values were obtained using the actual probability from my code.

Expected number of wrongful is just 2000000*p(wrongful):
p = 0.5: p(wrongful) = 0.186, E(wrongful) = 371808.47
p = 0.6, p(wrongful) = 0.065, E(wrongful) = 130130.96
p = 0.7, p(wrongful) = 0.011, E(wrongful) = 21111.95
p = 0.8, p(wrongful) = 0.0005, E(wrongful) = 938.27
p = 0.9, p(wrongful) = 1.06*10^{-6}, E(wrongful) = 2.11

It appears that as p(wrongful) increases E(wrongful) exponentially approaches zero. This makes sense as if p = 1, then there are no wrongful convictions as every juror makes the right call. Furthermore, as probability of correct increases, it becomes harder to make a wrongful conviction, as you need at least 10 people for a wrongful conviction to be possible, which becomes incredibly low as the p(correct) gets closer to 1.

(h) What are some flaws in the assumptions of the problem?

The main flaw in the problem's assumptions is that each juror is independent. If we're following the American judicial system, jurors have the ability to talk with each other after debate. Because of this, juror's may be influenced to change their votes, meaning their decision is not independent. Secondly, the problem assumes that they are able to accurately ascertain whether the jury made the correct decision. For example, if a person were convicted of a crime, but never admitted to it, would that be considered the right or wrong decision? You don't neccesarily have definitive proof that the jury made the right decisison.

2. Let C be the circle defined by $(x - 1.25)^2 + (y - 1.25)^2 = 0.4$. We know that its interior has area 0.4π, but suppose that we didn't. Your task is to compute this area by Monte Carlo simulation. Here is an outline: Define a region D containing C with known area. Write a program that uniformly samples from this region D. Using this, and a Monte Carlo "rejection sampling" method, estimate the area within C. Your algorithm should let you choose the number of samples you take.

(a) Run this experiment with sample count (1, 5, 10, 50, 100, 1000, 5000, 10000, 50000, 100000, 500000, 1000000). Compute the error (relative the true area) both as a value and as a percentage.

To perform Monte Carlo simulation, let's define the closed region $0 \leq x \leq 2$ and $0 \leq y \leq 2$, which is a square of area 4. We can uniformly distribute points, and any point of distance ≤ 0.4 from $(1.25, 1.25)$, is within the circle.

Here is the code I used for performing the Monte Carlo simulation

```
import random
import math

def area(n): #Returns the area via Monte Carlo method, takes in samples
  in_count = 0
  for i in range(0, n):
    rand_x = random.uniform(0, 2) #Gets a random x value
    rand_y = random.uniform(0, 2) #Gets a random y value
    distance = math.sqrt((1.25-rand_x)**2+(1.25-rand_y)**2)
    if (distance < math.sqrt(.4)): #If inside circle
      in_count = in_count + 1
  area_r = 4*in_count/n
  return(area_r)

print(area(1000000)) #Runs with 1000000 samples
```

Before I show results, due to the random nature of the sampling, the result for each number of samples is not the same fore each trial. I just used the first value I received. All values rounded to three digits (except for last one as otherwise it is zero.

Monte Carlo results

Num samples	Calculated Area	Error Value (absolute value)	Percentage Error (%
1	0	1.256	100
5	0.8	0.457	36.338
10	2.4	1.14	90.986
50	1.68	0.423	33.690
100	0.8	0.457	36.338
1000	1.308	0.051	4.087
5000	1.267	0.011	0.841
10000	1.231	0.025	2.024
50000	1.272	0.015	1.203
100000	1.265	0.008	0.634
500000	1.260	0.003	0.228
1000000	1.257	0.0006	0.046

(b) Let Cbe the circle defined by $(x-0.75)^2+(y-0.75)^2 = 0.3$. Consider the area contained inside the union of C $\cup$ $C^{'}$. Since the circles overlap, this area should be smaller than that of either individual circle. Modify your algorithm so that it computes the area of the region bounded by C $\cup$ $C^{'}$. Compute this area by running the algorithm with at least 1000000 samples, and compare this to the sum of the area of C and C$C^{'}$ separately.

We can update our algorithm by checking if the point is within distance from the center of either circle. We can use this using the or statement. As it will be true if the point is within the area of either circle or both.

```
import random
import math

def area(n):
  in_count = 0
  for i in range(0, n):
    rand_x = random.uniform(0, 2)
    rand_y = random.uniform(0, 2)
    dist1 = math.sqrt((1.25-rand_x)**2+(1.25-rand_y)**2)
    dist2 = math.sqrt((0.75-rand_x)**2+(0.75-rand_y)**2)
    if (dist1 < math.sqrt(.4) or dist2 < math.sqrt(.3)):
      in_count = in_count + 1
  area_r = 4*in_count/n
  return(area_r)

x = area(1000000)
print("Calculated Area: ", x)
```

Running with 1,000,000 samples, we get the area of the union as 1.888. This makes sense, as the area of C $\cup$ $C^{'}$ $\leq = 0.7\pi = 2.199$, as the circles overlap.