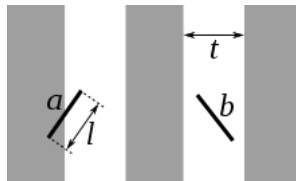


Simulation: Drop needles

Suppose we have a floor made of parallel strips of wood, each the same width t , and we drop a needle with length $l = t$ onto the floor. What is the probability that the needle will lie across a line between two strips?

Below is an example of two needles dropped. Needle a falls across a line, while needle b does not.



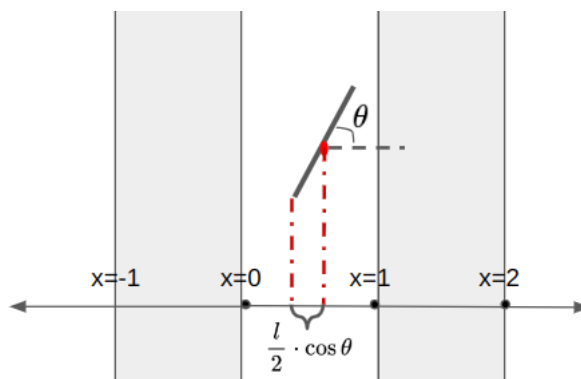
In this coding homework, we will simulate such experiments and connect them with the estimation of π .

```
In [1]: import numpy as np
import math
import matplotlib.pyplot as plt
```

The first thing to write is a function `drop_needle`. It simulates dropping a needle onto the floor we described and returns whether the needle lies across a line between two strips.

Now the question is how to describe the position of a needle using random variables. The figure below visualizes a needle sampled, with $t = l = 1$ (see figure above). Remember that the needle should have an equal probability of landing in any position. In fact, we can uniformly sample the position of the needle's mass center and then uniformly sample the angle formed by the needle and the x-axis. Specifically, we only focus on the mass center's position with respect to (w.r.t.) the x-axis since we can assume the strip is long enough.

Besides, we do not need to sample the x-value of the center from $-\infty$ to ∞ . Instead, we can uniformly sample it from 0 to $2t$. Why is this the case?



[10pts]

Answer: Since the length of the strip is the same as the length of the needle, considering the boundary of where the needle can touch the strip, the mass center of the needle should be at least $l/2$ away from the strip, on both side. Therefore, for a single view of the strip where needle is dropped, it should be $2l$ for all possibilities, which makes the summation of probabilities of all cases equal 1.

```
In [2]: def drop_needle(strip_length, needle_length):
        """
        Simulate dropping a needle on to the floor made of parallel strips of woods.
        Return whether the needle lie across a line between two strips.

        :return: An Integer that equals to 1 if the needle lie across a line, and 0 otherwise.
        """
        t = strip_length
        l = needle_length
        mc = np.random.uniform(0, 2*t)
        theta = np.random.uniform(0, np.pi)
        left = mc - abs(l/2.0 * math.cos(theta))
        right = mc + abs(l/2.0 * math.cos(theta))
        if mc < t:
            if left <= 0 or right >= t:
                return 1
            else:
                return 0
        else:
            if left <= t or right >= 2*t:
                return 1
            else:
                return 0

        # write your code here
```

Next, write a function `run_simulation` that calls `drop_needle` repetitively for `n` times. The function should return the probability that a dropped needle lies across a line based on the `n` trials. [5pts]

```
In [3]: def run_simulation(n, strip_length, needle_length):
        """
        Repeat drop_needle experiment for n times. Return the probability that the needle will lie across a line.

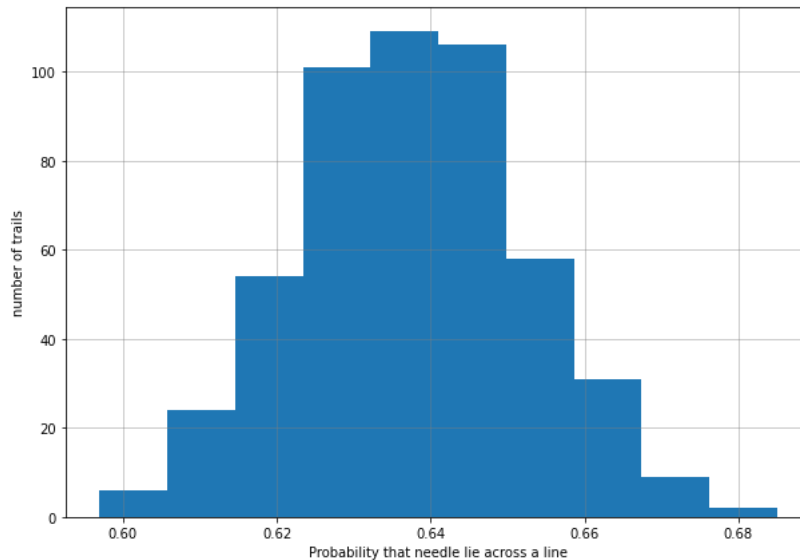
        :return: float, the probability that the needle will lie across a line according to the n experiments.
        """
        # Write your code here
        count = 0
        nsim = n
        while nsim > 0:
            count += drop_needle(strip_length, needle_length)
            nsim -= 1

        return 1.0 * count / n
```

Run the simulation

Run the `run_simulation` function 500 times with parameters `n=1000`, `strip_length=1`, and `needle_length=1`. Each time the function is going to return a probability of the needle lying across the line. Plot a histogram of those 500 probabilities. [5pts]

```
In [4]: # Write your code here
record = []
nsim = 500
while nsim > 0:
    record.append(run_simulation(1000,1,1))
    nsim -= 1
fig, ax = plt.subplots(figsize=(10,7))
ax.hist(record)
ax.grid(b = True, color = 'grey', alpha = 0.5)
plt.xlabel("Probability that needle lie across a line")
plt.ylabel("number of trails")
plt.show()
```

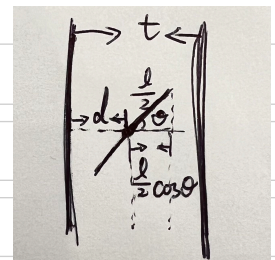


Estimate π based on the returned probability

This experiment can be used to estimate the value of π . In the case where the needle length l equals to the width t of the strips, the probability of a needle lies across a line is $p = \frac{2}{\pi}$. Try to prove why this holds. [15pts]

Your proof:

Let l denote the needle length and t be the distance between two strips. Since the strips are parallel and equally spaced, we can simplify our proof to how far away our needle lunge from the nearest vertical strip. Denote d as the distance to the nearest strip, and thus, $d \sim \text{Unif}(0, t/2)$, and since the needle can be placed randomly by angels, $\Theta \sim \text{Unif}(0, \pi)$ relative to the horizon. The diagram below visually shows all the denotations:



Therefore, a crossing will occur in case $d < l/2 * \sin(\theta)$. Also, based on the fact that d and θ are independent uniform random variables on $[0, t/2]$, and $[0, \pi]$ respectively,

$$P(\text{crossing}) = \int_0^\pi \int_0^{\frac{l}{2} \sin(\theta)} \frac{1}{\frac{t}{2}} \frac{1}{\pi} dd d\theta$$

$$P(\text{crossing}) = \frac{2}{t\pi} \int_0^\pi \frac{t}{2} \sin(\theta) d\theta$$

$$P(\text{crossing}) = \frac{2l}{t\pi}$$

Since we have $l = t$,

$$P(\text{crossing}) = \frac{2}{\pi}$$

Therefore, we can use this probability of crossing to estimate π as $\hat{p} = \frac{\text{number of crossing}}{\text{number of throws}}$. When we plug the empirical probability of crossing on the theoretical one,

$$\hat{\pi} = 2 * \frac{\text{number of crossing}}{\text{number of throws}}$$

Q.E.D.

Assume you know $p = \frac{2}{\pi}$, and you can call the function `run_simulation`. Let X_i be the indicator random variable of dropping the i -th needle, i.e., $X_i = 1$ if the i -th needle lies across a line, and $X_i = 0$ otherwise. Let $S_n = \sum_{i=1}^n X_i$. How many needles do you need to drop, i.e., what value of n should you take, to get the 95% confidence interval of S_n ? Use Chebyshev's inequality and let $\epsilon = 0.1$. [15pts]

Your answer:

By WLLN, $\lim_{n \rightarrow \infty} \Pr(|S_n - n\mu| > n\epsilon) = 0$. By Chebyshev's inequality, $P(|S_n - E(S_n)| \geq n\epsilon) \leq \frac{\text{Var}(S_n)}{(n\epsilon)^2} \leq 1 - 0.95 = 0.05$.

Since the needle can either be crossing the strip or not, it is a bernoulli trial following binomial distribution. In this way, we have $X = S_n$, $\mu = np$, and $\sigma = \sqrt{np(1-p)}$. Since $p = \frac{2}{\pi}$,

$$\begin{aligned} \frac{\text{Var}(S_n)}{(n\epsilon)^2} &\leq 0.05 \\ \frac{n * 2/\pi * (\pi - 2)/\pi}{n^2 \epsilon^2} &\leq 0.05 \\ n &\geq 462.67008 \end{aligned}$$

Therefore, at least 463 needles should be dropped to get the 95% confidence interval of S_n .

What does the CLT say about S_n ? Run the `run_simulation` function with parameters `n=5000`, `strip_length=1`, and `needle_length=1`, and report a 95% confidence interval of S_n based on CLT (see lecture slides page 4 from Feb 2). [10pts]

Answer: By CLT, as n increases, the distribution of S_n that follows a binomial distribution with n trials and probability p of success gets closer and closer to a normal distribution.

```
In [5]: # Write your code here.
m = np.mean(record)
s = np.std(record)
Cleft = m - 2 * s
Cright = m + 2 * s
print("95% confidence interval: (", Cleft, ", ", Cright, ")")
```

95% confidence interval: (0.6077474579032859 , 0.6667485420967142).

In []: