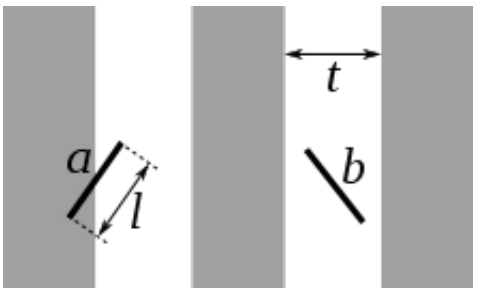
Simulation: Drop needles

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 [113…

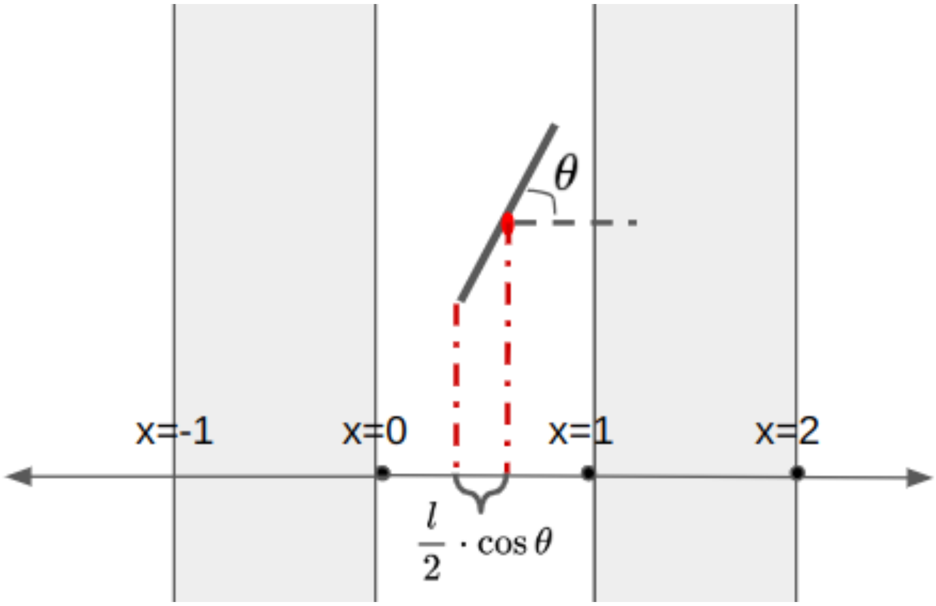
```
import numpy as np
import math
import matplotlib.pyplot as plt
import random
```

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 $-\inf$ to $\inf$. Instead, we can uniformly sample it from 0 to $2t$. Why is this the case?

We can do it from a uniform sample from 0 to $2t$ instead of -inf to inf since the pattern follows a uniform distribution from 0 to $2t$ from -inf to inf. Since the probability that the needle falls from -inf to inf is same, we can reduce the range from 0 to $2t$. All it matters is checking whether the needle crosses the strip from -t to 0, 0 to t, or t to $2t$.



[10pts]

In [114...

```
def drop_needle(strip_length, needle_length):
    """
    Simulate dropping a needle on to the floor made of parallel strips of wo
    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, an
    """
    drop_xaxis = random.uniform(0,strip_length * 2)
    drop_yaxis = random.uniform(0,strip_length * 2)
    angle = random.uniform(0,90)
    cosine_value = math.cos(math.radians(angle))

    needle_position_1 = drop_xaxis + 1/2 * cosine_value * needle_length
    needle_position_2 = drop_xaxis - 1/2 * cosine_value * needle_length
    one = math.floor(needle_position_1)
    two = math.floor(needle_position_2)

    if (two - one == 0):
        return 0
    else:
        return 1
```

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 [115...
```
std_dev = 0
def run_simulation(n, strip_length, needle_length):
    """
    Repeat drop_needle experiment for n times. Return the probability that t

    :return: float, the probability that the needle will lie across a line a
    """
    count = 0
    for x in range(n):
        count += drop_needle(strip_length, needle_length)
    prob = count / n
    return prob
```

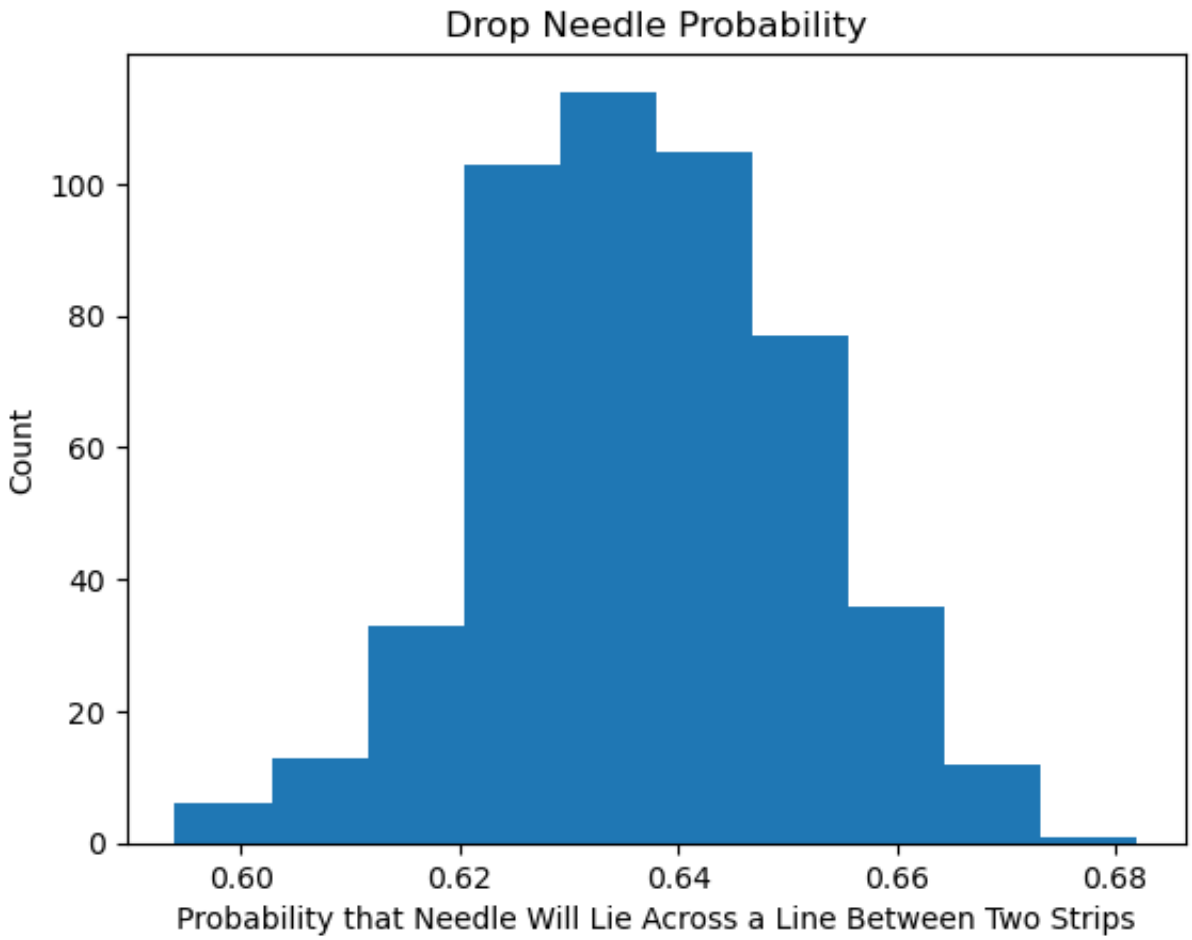
Run the simulation

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 [116...
```
# Write your code here

prob_list = []
for x in range(500):
    prob_list.append(run_simulation(1000,1,1))
```

In [117...
```
fig, ax = plt.subplots(1, 1)
ax.set_xlabel('Probability that Needle Will Lie Across a Line Between Two St
ax.set_ylabel('Count')
plt.hist(prob_list)
plt.title('Drop Needle Probability')
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:

In our case, the width t of the strips is equal to the needle length l. In other words, $l = t$. We denote the angle at the picture above as θ. In addition, the distance between the needle and the closest strip is denoted as d.

The distance between the needle and the closest strip, variable d, is minimum when the needle is dropped exactly at the position of the start of the strip (0, l, 2l, ...), which is 0. It is maximum when the needle is dropped exactly between two strips ($\frac{1}{2}l$, $\frac{3}{2}l$, ...), which is $\frac{l}{2}$.

Therefore, we can declare the bounds of the variable d as 0 $\leq d \leq \frac{l}{2}$.

Then, we need to declare the bounds of the variable θ. According to the diagram above, we need to calculate cos(θ) to calculate where the endpoints of the needle are located. Since the length of the needle cannot be negative, we have to take the absolute value of cos θ). In such case, the maximum value of |cos(θ)| occurs when θ equals 0 since |cos(0)| = 1. The minimum value of |cos(θ)| occurs when θ equals $\frac{\pi}{2}$ since |cos($\frac{\pi}{2}$)| = 0.

Therefore, we can declare the bounds of the variable θ as 0 $\leq \theta \leq \frac{\pi}{2}$.

According to the definition of probability, the probability of an event occurring is denoted by $\frac{Event}{TotalSample}$.

The value of total sample equals when the value of variables d and θ is maximum, which is $\frac{\pi}{2}$ multiplied by $\frac{l}{2}$, which is $\frac{l\pi}{4}$.

The event where the needle crosses the strip line occurs only when $d \leq \frac{l}{2}$cos(θ) within 0 $\leq \theta \leq \frac{\pi}{2}$.

Therefore, we need to take the integral of $\frac{l}{2}$cos(θ) within the bound 0 and $\frac{\pi}{2}$

$\int_0^{\frac{\pi}{2}} \frac{l}{2}cos(\theta)$ dθ

= $\frac{l}{2}$ * $\int_0^{\frac{\pi}{2}} cos(\theta)$ dθ

= $\frac{l}{2}$ * (sin($\frac{\pi}{2}$) - sin(0))

= $\frac{l}{2}$ * (1)

= $\frac{l}{2}$

Going back to the definition of probability, the probability that the needle will lie across a line between two strips is $\frac{\frac{l}{2}}{\frac{l\pi}{4}}$, which is $\frac{2}{\pi}$.

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:

Pr(|$\frac{S_n}{n}$ - μ| > ϵ) $\leq \frac{Var[\frac{S_n}{n}]}{\epsilon^2} \leq \delta$

Pr(|S_n - μn| > ϵn) $\leq \frac{Var[S_n]}{n^2 * \epsilon^2} \leq \delta$

The variance, Var[S_n], equals n *p* (1-p) = n $\frac{2}{\pi}$ (1 - $\frac{2}{\pi}$)

When we plug the value of variance, we get

$\frac{n*\frac{2}{\pi}*(1-\frac{2}{\pi})}{\epsilon^2 n^2} \leq \delta$

If we cancel out n, we get

$\frac{\frac{2}{\pi}*(1-\frac{2}{\pi})}{\epsilon^2 n} \leq \delta$

Moving the equation, we get $\frac{\frac{2}{\pi}*(1-\frac{2}{\pi})}{\epsilon^2 \delta} \leq$ n

When we plug in ϵ = 0.1 and δ = 1 - 0.95 = 0.05, we get

$\frac{\frac{2}{\pi}*(1-\frac{2}{\pi})}{0.1^2 * 0.05} \leq$ n

After calculation, we get

462.67 $\leq$ n

Since n has to be an integer that is greater than or equal to 462.67, n has to equal 463.

In summary, the value of n to get the 95% confidence interval of S_n with ϵ = 0.1 should be at least 463.

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]

The CLT states that we need at least 463 trials of dropping needle to get 95% confidence interval with ϵ value of 0.1

In [201…

```
n = 5000

mean_value_from_sample = run_simulation(n,1,1)
sample_std = math.sqrt((mean_value_from_sample) * (1-mean_value_from_sample)


actual_mean_value = 2/math.pi
actual_std = math.sqrt((2/math.pi) * (1-2/math.pi))

z_score = 1.96


clt_lower_bound = mean_value_from_sample - z_score * sample_std/math.sqrt(n)
clt_upper_bound = mean_value_from_sample + z_score * sample_std/math.sqrt(n)

print("From sample:")

print()

print("Mean: " + str(mean_value_from_sample))
print("Confidence interval: (" + (str(round(clt_lower_bound,6))) + ", " + st

print()

print("We are 95% confident that the mean number of times the needle will be
      + "two strips according to the sample data we \ncollected is between "
      + str(round(clt_lower_bound,6)) + " and "
      + str(round(clt_upper_bound,6)) + " when we run the simulation for \nn

print()


clt_lower_bound_two = actual_mean_value - z_score * actual_std/math.sqrt(n)
clt_upper_bound_two = actual_mean_value + z_score * actual_std/math.sqrt(n)

print("From given information (expected values):")

print()

print("Mean: " + str(round(actual_mean_value,6)))
print("Confidence interval: (" + (str(round(clt_lower_bound_two,6))) + ", "

print()

print("We are 95% confident that the mean number of times the needle will be
      + "two strips from the expected mean and expected standard deviation i
      + str(round(clt_lower_bound_two,6)) + " and "
      + str(round(clt_upper_bound_two,6)) + " when we run the \nsimulation f
```

```
From sample:

Mean: 0.6438
Confidence interval: (0.630526, 0.657074)

We are 95% confident that the mean number of times the needle will be
dropped across a line between two strips according to the sample data we
collected is between 0.630526 and 0.657074 when we run the simulation for
n=5000 times.

From given information (expected values):

Mean: 0.63662
Confidence interval: (0.623288, 0.649952)

We are 95% confident that the mean number of times the needle will be
dropped across a line between two strips from the expected mean and expected
standard deviation is between 0.623288 and 0.649952 when we run the
simulation for n=5000 times.
```

In [ ]: