

PMDS HW1

E44091144 Shen-Ching Feng

Random Walk Simulation

Simulation process (Use 2D case as an example)

Import the packages

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

Define the function to randomly generate 1 & -1

```
def generate_random_sign():
    n = random.choice([-1, 1])
    return n
```

Calculate the L1 distance (Manhattan distance) and L2 distance (Euclidean distance)

$$L1 = |x| + |y|$$

$$L2 = \sqrt{x^2 + y^2}$$

```
## Define the number of trials of simulation
## and the samples of random walks in the single trials
num_trial, num_samples = 1000, 10000

l1_dist = []
l2_dist = []

for i in range(num_trial):
    x, y = 0, 0

    for i in range(num_samples):

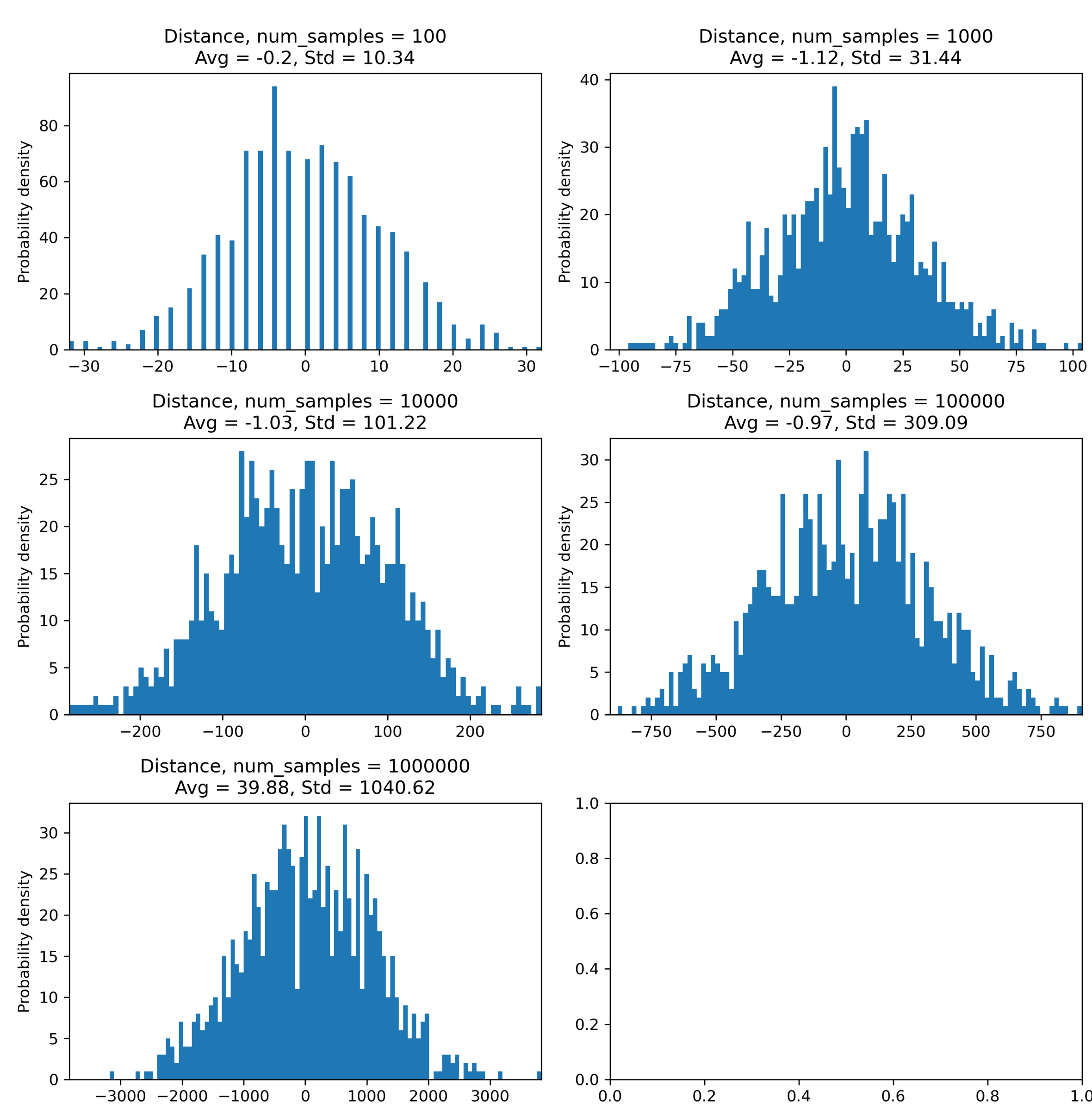
        x1 = generate_random_sign()
        y1 = generate_random_sign()

        x += x1
        y += y1

    ## Distance calculation
    ## L1 distance
    l1 = abs(x) + abs(y)
    l1_dist.append(l1)

    ## L2 distance
    l2 = np.sqrt(x**2 + y**2)
    l2_dist.append(l2)
```

1D Cases

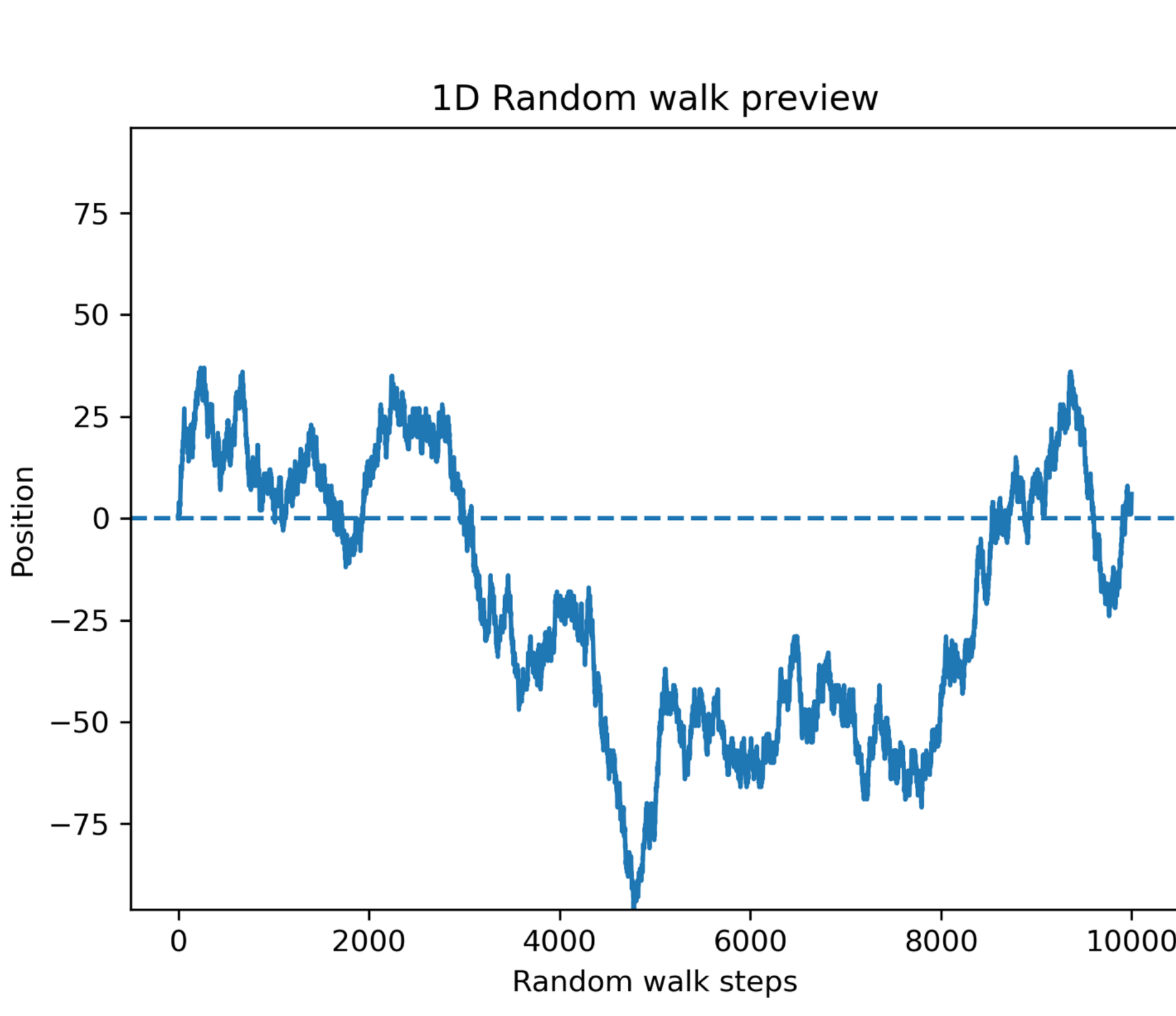


1D Preview

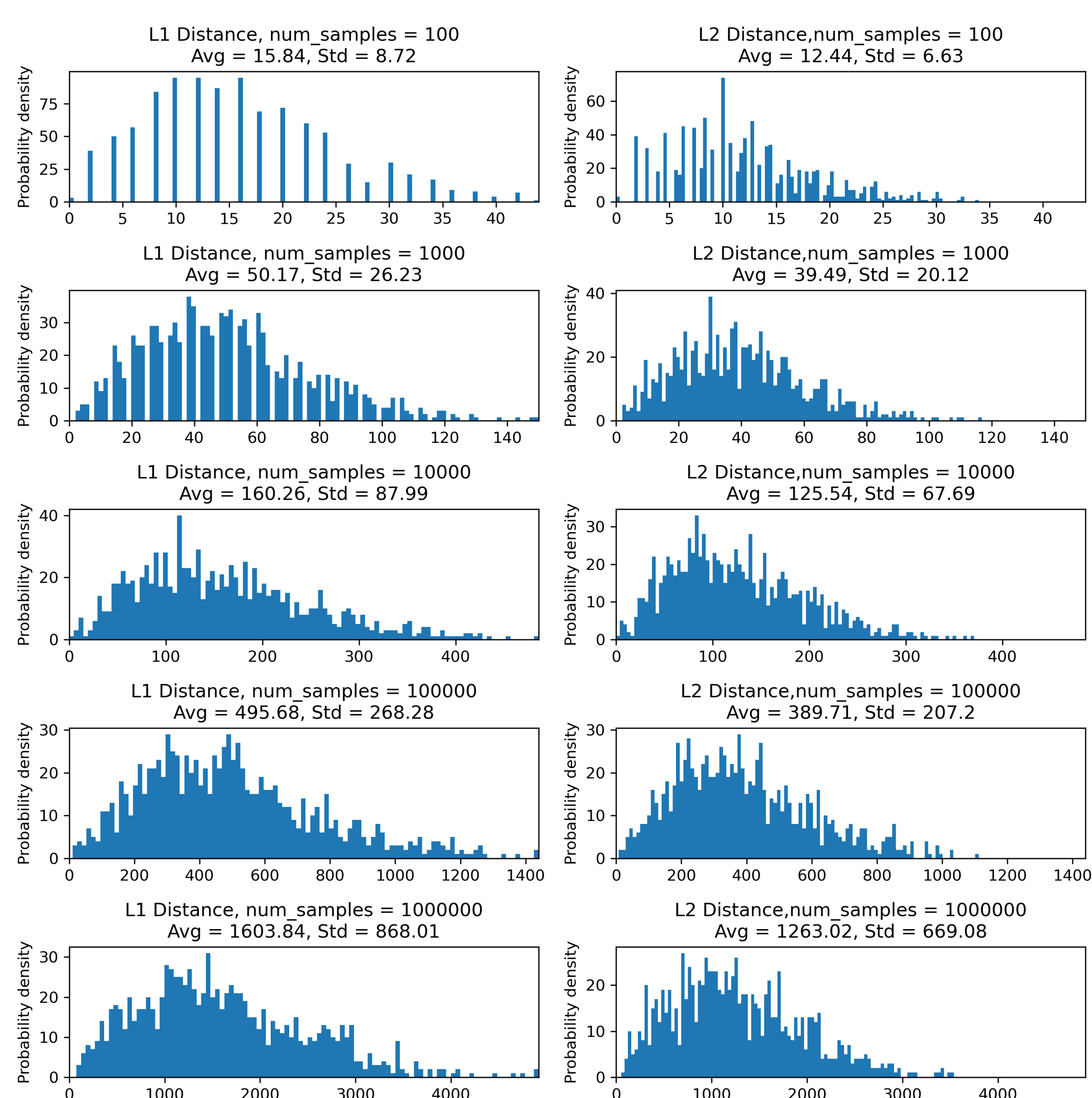
A simulation process of 1D random walk process

Final position: 6

How many times the dot comes back to the origin: 93



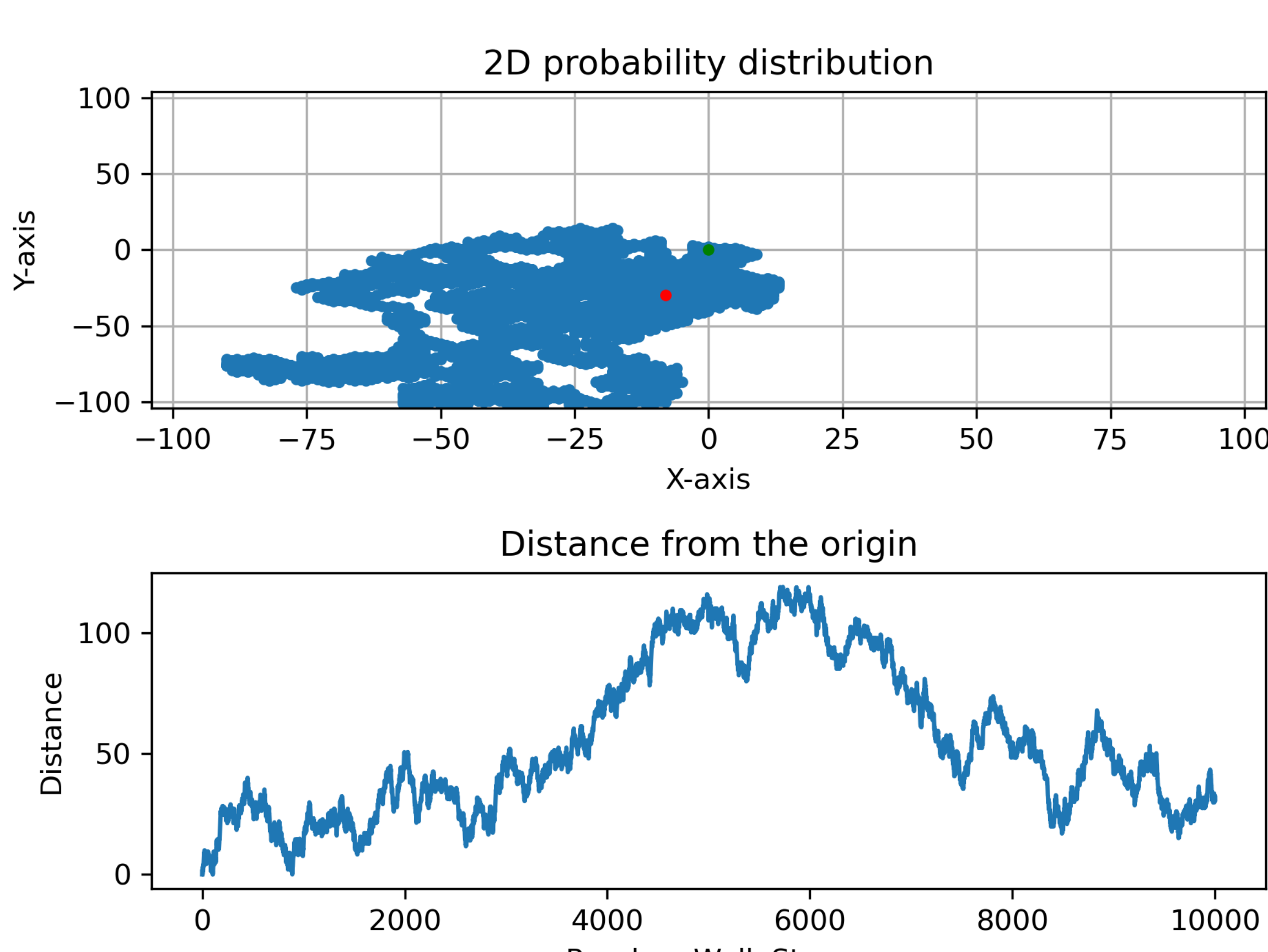
2D Cases



2D Preview

A simulation process of 2D random walk process (Green point: The origin ; Red point: Final position)

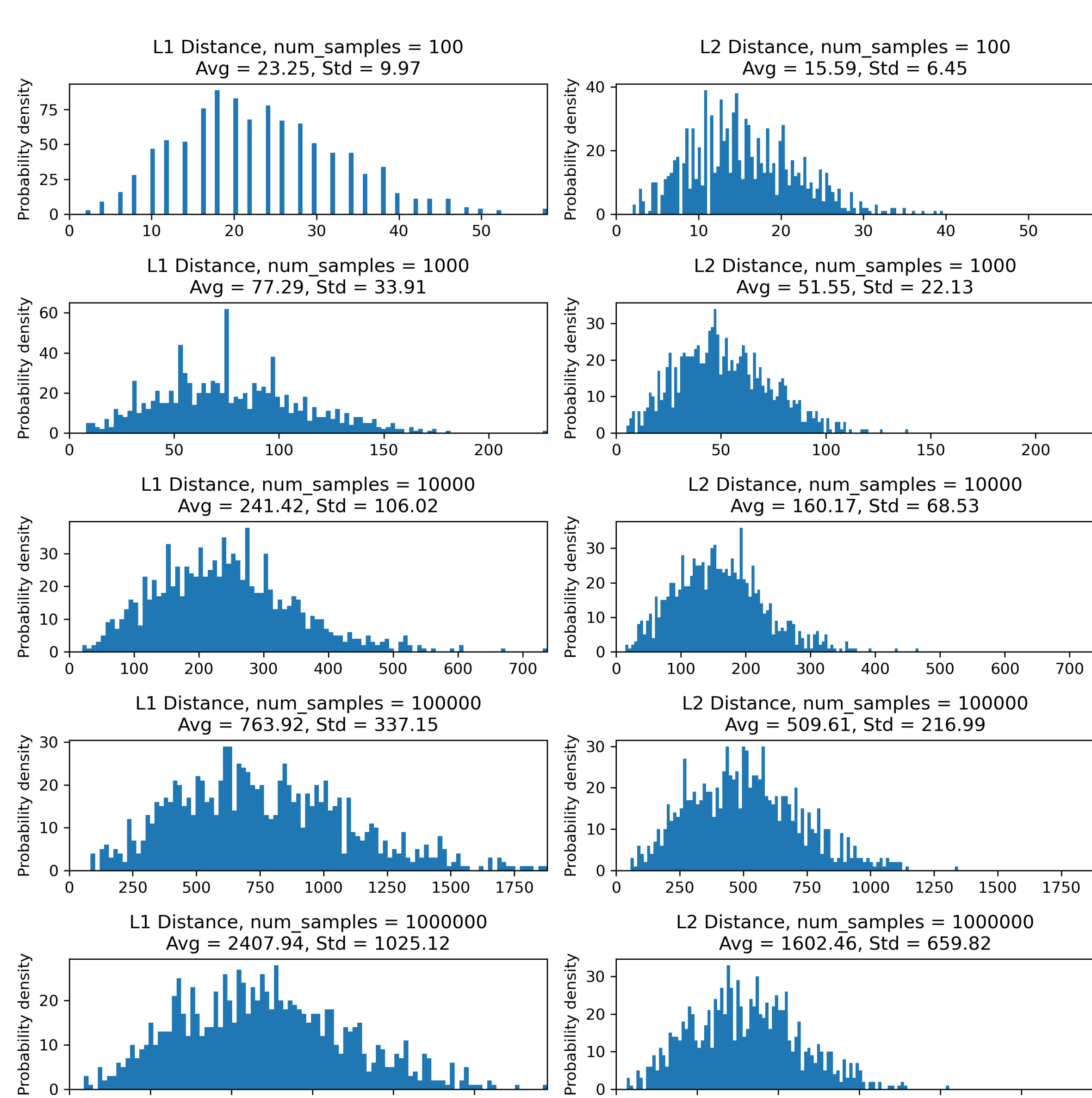
Final position: (-8, -30)



The time spent in each quadrant

Quadrant	-	+
+	3.28 %	0.05 %
-	88.69 %	6.76 %

3D Cases



4D Cases

