Question 5

Python Code

```
import random
import matplotlib.pyplot as plt

def sim(n):
    res = []

    for i in n:
        res.append(4 * (sum(1 for _ in range(i) if random.uniform(-1, 1)**2 + random.uniform(-1,
1)**2 < 1)) / i)
        print(i, 4 * (sum(1 for _ in range(i) if random.uniform(-1, 1)**2 + random.uniform(-1, 1)**2
< 1)) / i)

    plt.plot(n, res, marker='.')
    plt.xlabel('n')
    plt.ylabel('4 * count / n')
    plt.title('Simulation')
    plt.show()

sim([50, 100, 500, 1000, 2000, 5000, 10000, 20000])
```

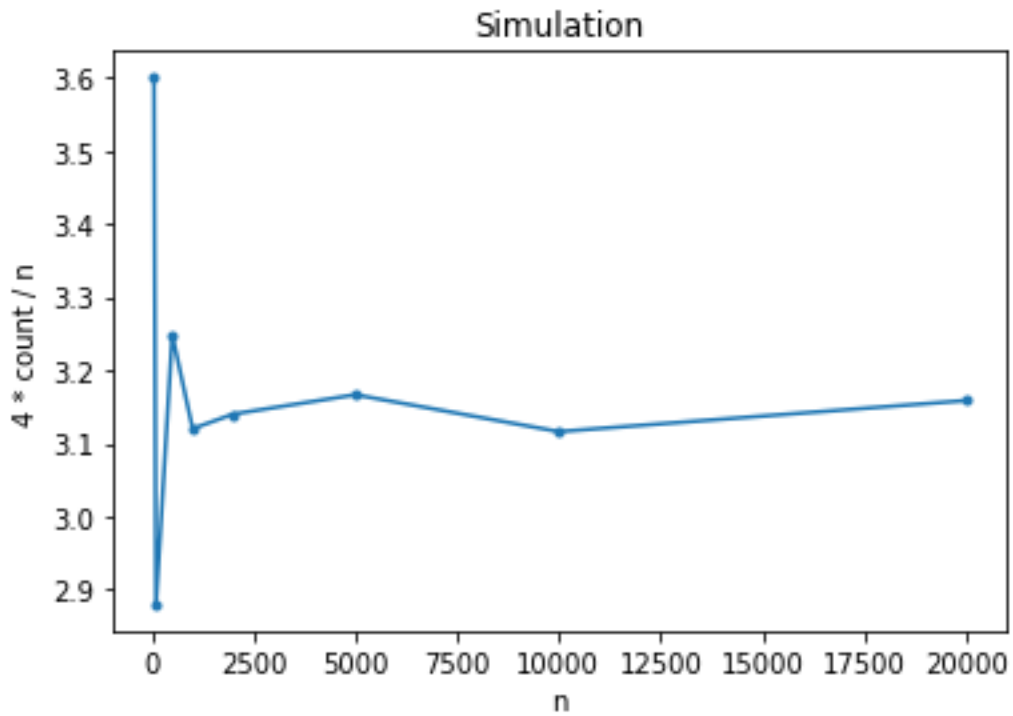
Output:

```
50 3.2
100 3.16
500 3.208
1000 3.212
2000 3.16
5000 3.1312
10000 3.1412
20000 3.1212
```

Table:

N	**4count/n**
50	3.2
100	3.16
500	3.208
1000	3.212
2000	3.16
5000	3.1312
10000	3.1412

Plot:



As n increase, the plot converges, and the value approaches the value of π, and estimations become more accurate for the ratio $4 * count/n$. The rate of increase of the ratio is not constant as evident from the plot and is slowing down as the n increases. The more simulations we run; the accurate estimations are produced. Due to randomization, we observe variability at each step of the simulation.