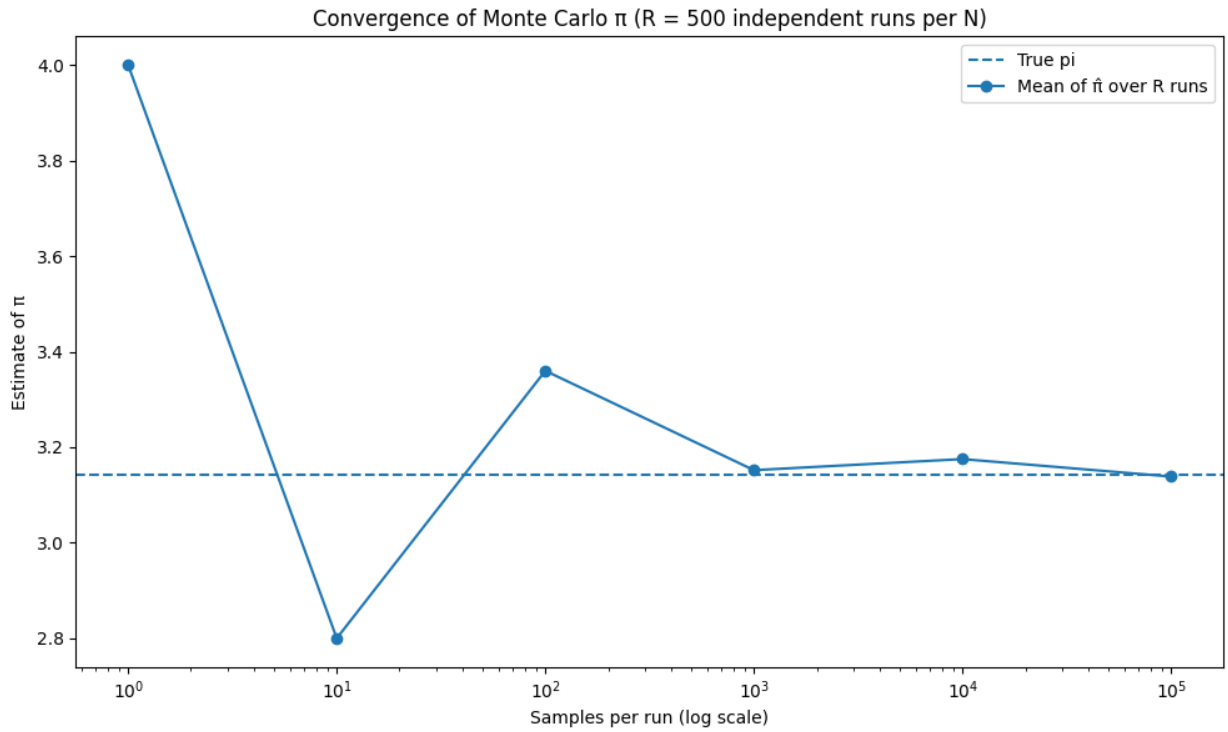
# Monte Carlo – Estimating π

I created a data type called `Coordinate` that takes inputs `x` and `y`. In the `MonteCarlos` class, I used NumPy to generate random numbers. I then implemented a method to store all the coordinates and another method to check whether each coordinate lies within the quarter circle.

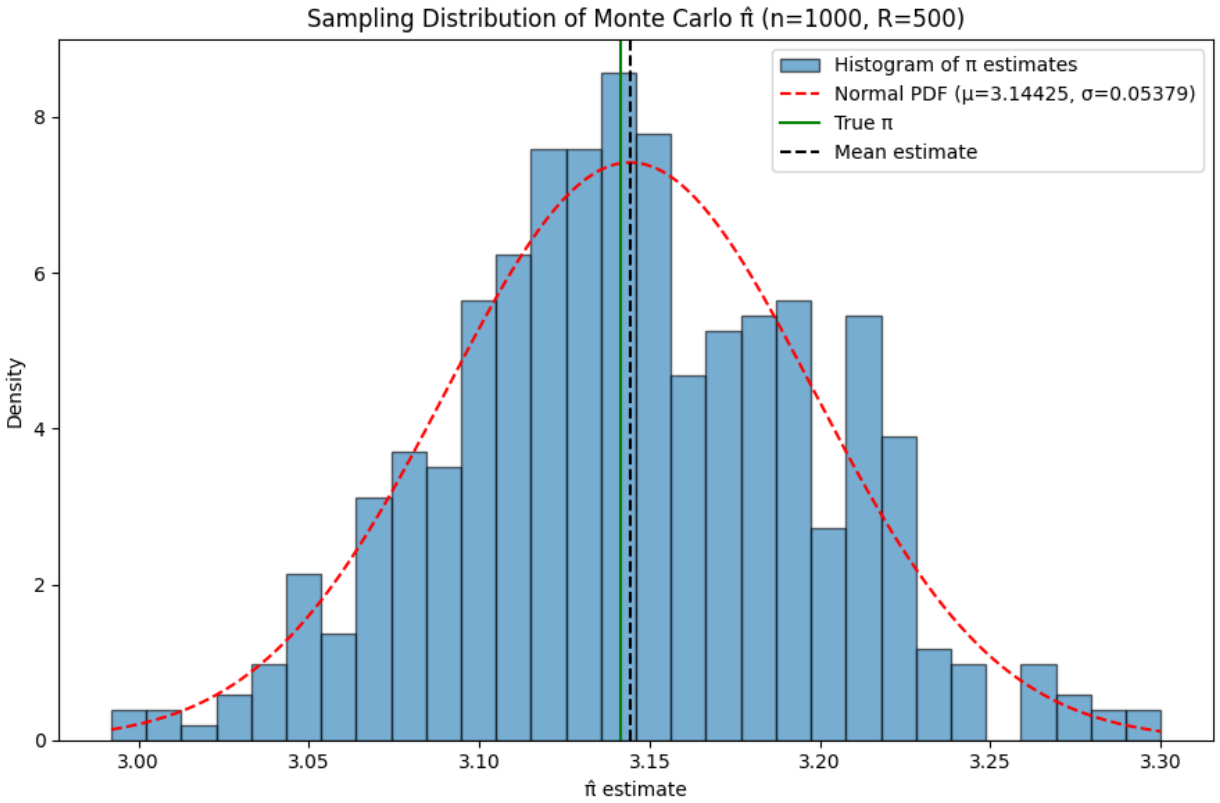
# Convergence

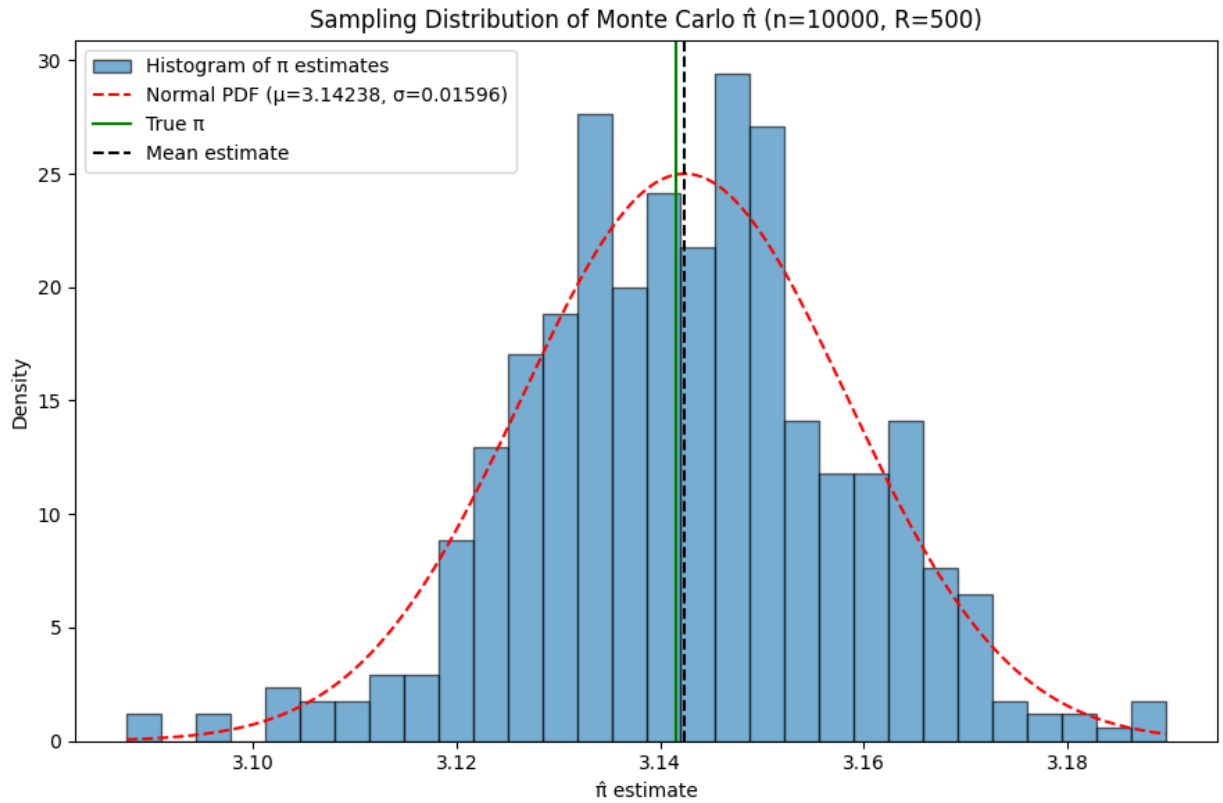
As the sample size increased, I observed that the estimated value of π converged toward the actual value of π. Initially, the estimate started near 4, and as the number of samples grew, it gradually approached the true value of π.

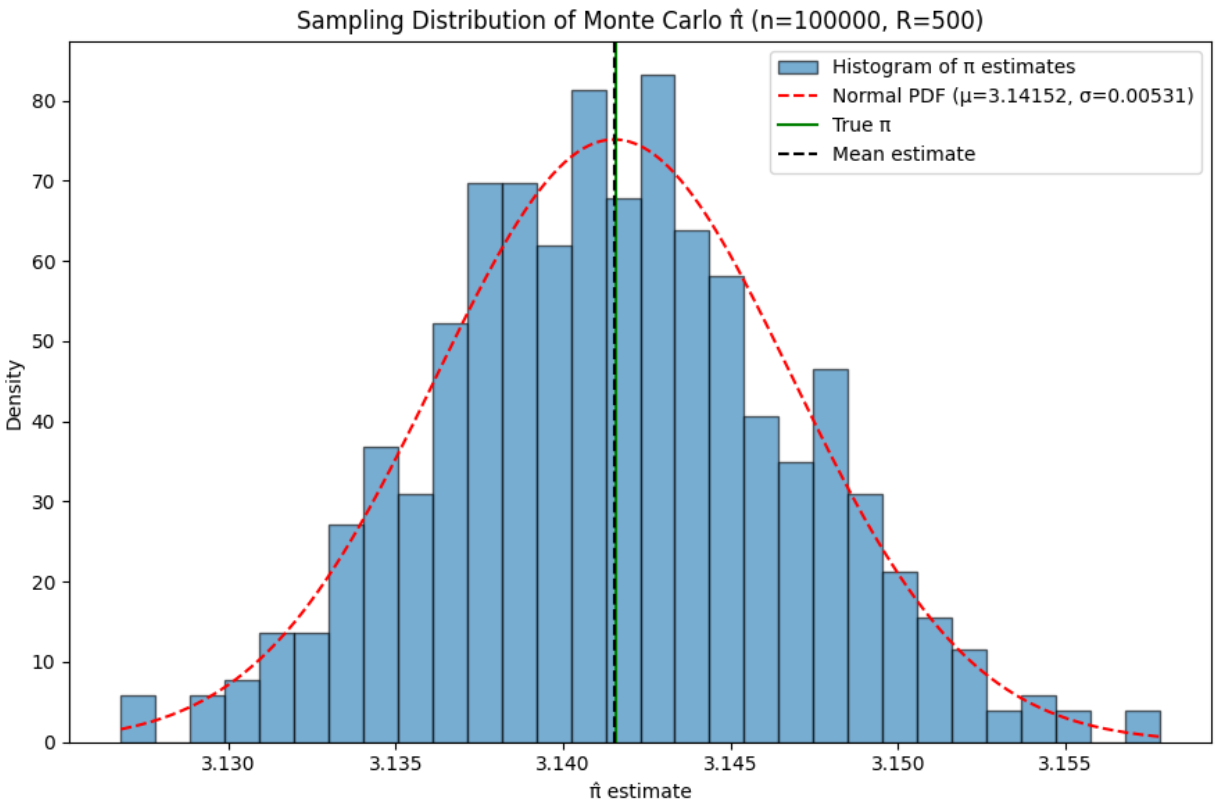



# Histogram-

As the higher the sample size goes the experiment fits better under the Normal Distribution


Sampling Distribution of Monte Carlo π̂ (n=1000, R=500)
Histogram of π estimates
Normal PDF (μ=3.14425, σ=0.05379)
True π
Mean estimate
Density
π̂ estimate
0
2
4
6
8
3.00
3.05
3.10
3.15
3.20
3.25
3.30


Sampling Distribution of Monte Carlo π̂ (n=10000, R=500)
Histogram of π estimates
Normal PDF (μ=3.14238, σ=0.01596)
True π
Mean estimate
Density
π̂ estimate
0
5
10
15
20
25
30
3.10
3.12
3.14
3.16
3.18


Sampling Distribution of Monte Carlo π̂ (n=100000, R=500)
Histogram of π estimates
Normal PDF (μ=3.14152, σ=0.00531)
True π
Mean estimate
Density
π̂ estimate
0
10
20
30
40
50
60
70
80
3.130
3.135
3.140
3.145
3.150
3.155