MONTE CARLO SIMULATION TO FIND VOLUME OF A SPHERE

Name: Ritwij Kumar

Roll no.: 220121054

Objective: To obtain the volume of a sphere from Monte Carlo simulation and check the accuracy of the Monte Carlo simulation

Theory:

One method of calculating the volume of a sphere involves examining each point within a region and determining whether it falls inside the sphere.

The Monte Carlo method employs statistical sampling, utilizing random sampling and statistical analysis to derive numerical outcomes.

For determining the volume of a sphere, consider a cube with a side length of *2a*, where a sphere S is entirely contained within the cube and touches opposite faces of the cube.

So, the radius can be given by

$$r = \frac{length\ of\ side}{2}$$

$$r = \frac{2a}{2}$$

$$r = a$$

And we know that

$$V_{cube} = (length\ of\ side)^3$$

$$V_{cube} = (2a)^3$$

And for sphere

$$V_{sphere} = \frac{4}{3}\pi r^3$$

$$V_{sphere} = \frac{4}{3}\pi a^3$$

Obtaining ratio of V_{sphere} and V_{cube}

$$x = \frac{\frac{4}{3}\pi a^3}{8a^3}$$

$$x = \frac{1}{6}\pi = 0.523598776$$

So, if we randomly project N number of points in the cubical region and measure the ratio of number of points inside the sphere to the total number of points i.e. N it should be approximately equal to the x

We'll rely on the python which uses the Mersenne Twister as the core generator for generating random values between $-a$ and $+a$.

(read more about it here)

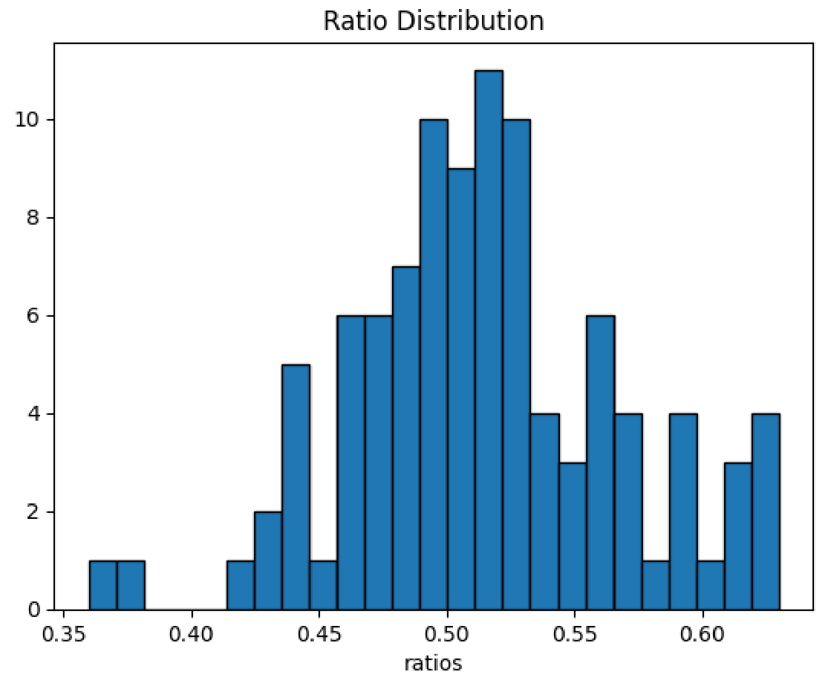
Code:

```
%matplotlib inline
from numpy import random
import numpy as np
import math
import matplotlib.pyplot as plt

#we'll be generating triplets let's say x,y,z and measuring distance from origin to check if it lies in the sphere
def func(N,a):
  #limits for the numbers
  b=-1*a

  x=random.uniform(a,b,N)
  y=random.uniform(a,b,N)
  z=random.uniform(a,b,N)

  inside=0

  for i in range(N):
    dist=(x[i]**2+y[i]**2+z[i]**2)**0.5
    if(dist<=abs(a)):
      inside+=1

  return(inside/N)
def plot(N,a):
  ratios=[ ]
  for i in range(N):
    ratios.append(func(N,a))

  plt.title("Ratio Distribution")
  plt.hist(ratios,bins=25,ec='black')
  plt.xlabel("ratios")

plot(1000,10)
```

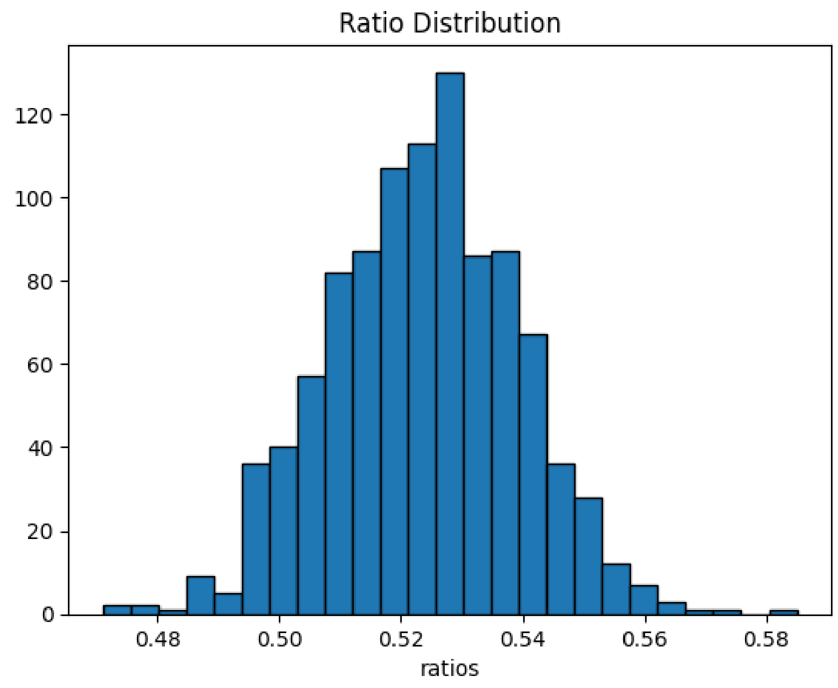
Find the link to the code here

Observations:

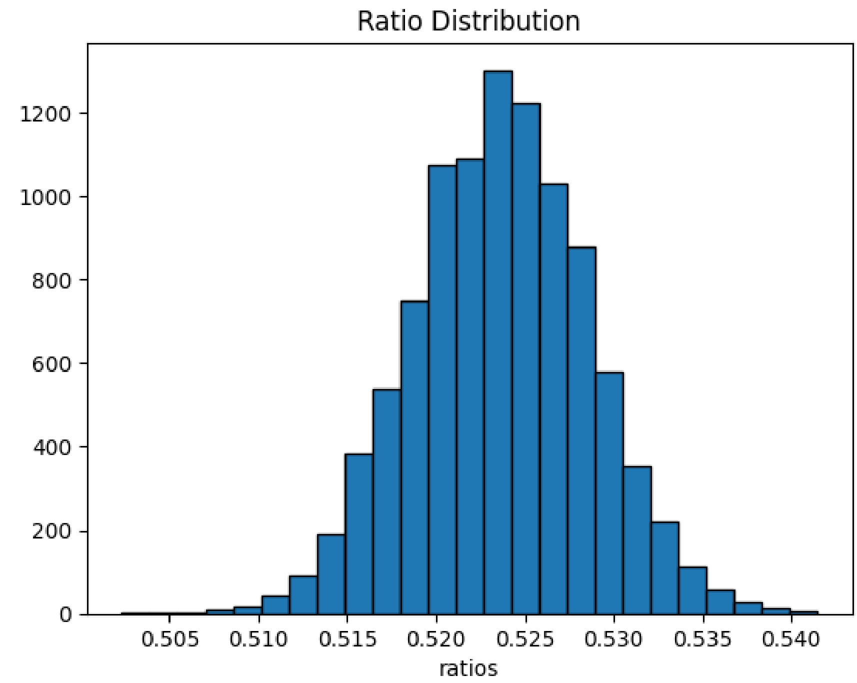
We tried to find the ratio 100 times for the r=10 and we obtained the following distribution



For N=1000 and r=10



For N=10000 and r=10



Result:

On increasing the value of iterations, we observe the peak gets closer to *x*

Therefore, we can say the volume of sphere is $\frac{4}{3}\pi r^3$ and it is verified with the monte Carlo method