EE 598
Probabilistic Simulation Models
Project #2a
Pooja Voladoddi- 6571916065
02/21/2014

The problem asks to generate 100 samples of IID 2-D uniform random variables in the unit square, and count how many of these samples fall within the quarter unit-circle centered at the origin inscribing the unit square.

This is one of the applications of Brute force Monte Carlo Simulation or Monte Carlo methods of estimation.

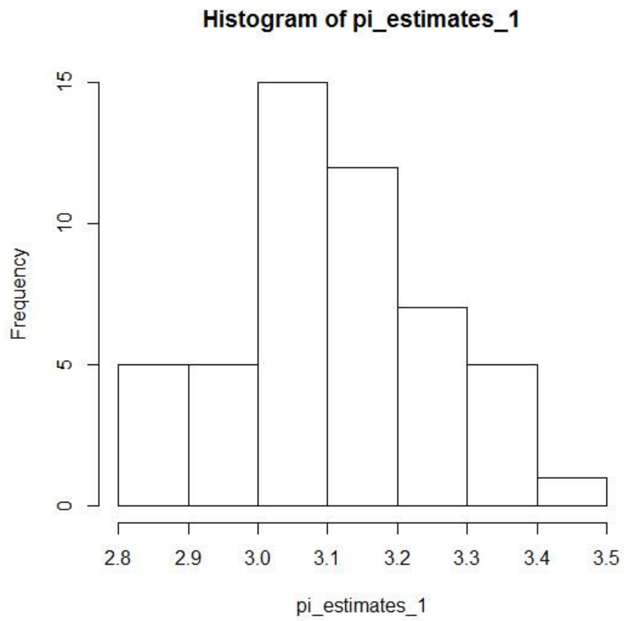
The goal here is to estimate the value of "PI" using MC method. This approach requires us to map the area covered by the uniformly generated samples (essential to maintain the condition of IID) in the inscribed quarter circle to that of the unit square in total. This is done so that this ratio can be used to estimate the value of "PI" using :

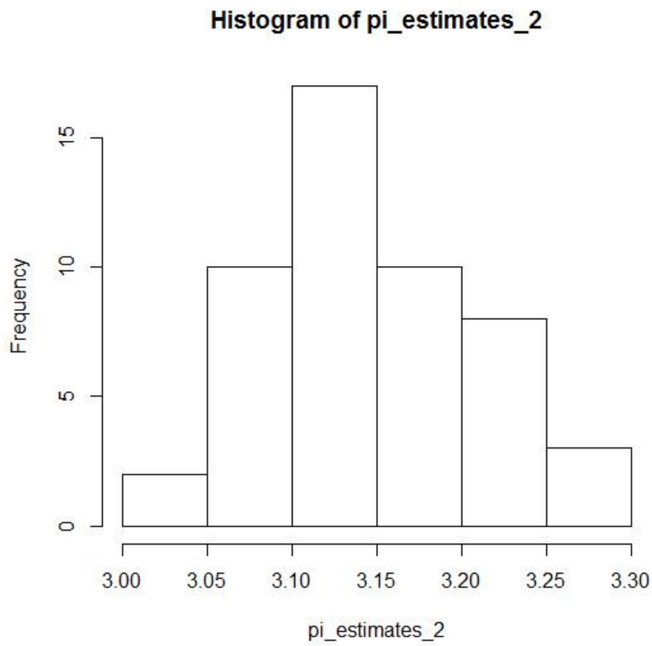
(#Samples inside the quarter circle) / (#Total Samples inside the unit square) = pi / 4

The LHS estimated ratio can be multiplied by "4" of RHS to estimate the value of pi.

Setting the seed = 5 and running the program [appendix A] gives the count (#samples inside the quarter circle out of the total 100 samples) is equal to "76".

i.)	Area estimate for n = 100, k = 50 runs	:	**0.8 sq.units**
	Pi estimate	:	**3.2**
ii.)	Area estimate for n =1000, k = 50 runs	:	**0.779 sq.units**
	Pi estimate	:	**3.116**
iii.)	Area estimate for n = 5000, k = 50 runs	:	**0.7708 sq.units**
	Pi estimate	:	**3.0832**
iv.)	Area estimate for n =10000, k = 50 runs	:	**0.7882 sq.units**
	Pi estimate	:	**3.1528**
v.)	Area estimate for n =50000, k = 50 runs	:	**0.78454 sq.units**
	Pi estimate	:	**3.13816**

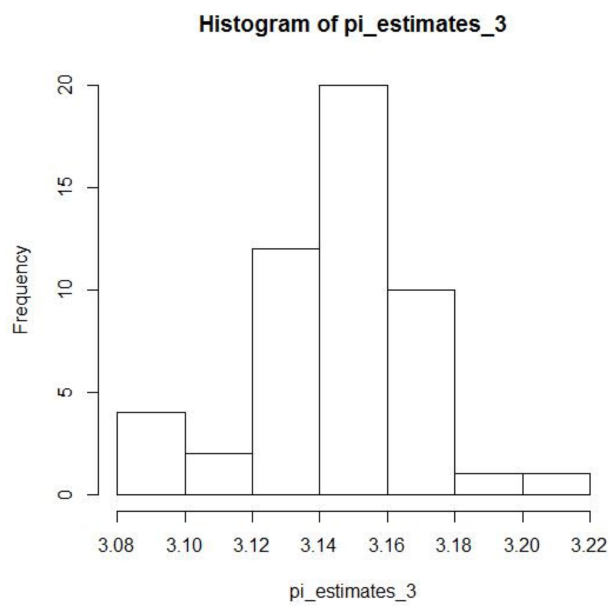


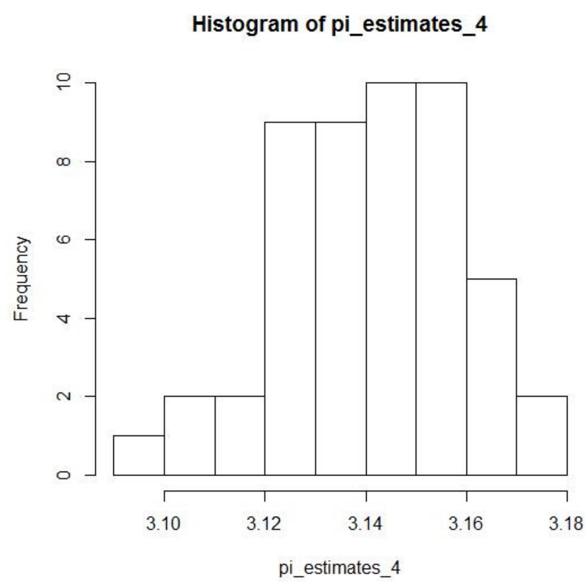


Histogram of (i.) – n= 100

Histogram of (ii) – n = 1000

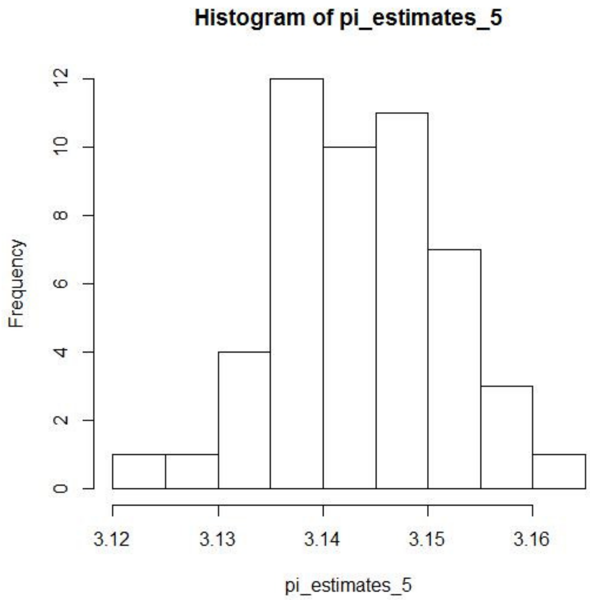
On carefully observing the two histograms, it is easy to decipher that the when n = 100 samples, the value of pi estimate is farther than its actual value when compared to that of the pi estimate in the case of n=1000 samples. The histogram on the right is slightly more towards the 3.15 whereas the histogram on the left shows much variance (from 3.0 to 3.2).





Histogram of (iii) – n = 5000

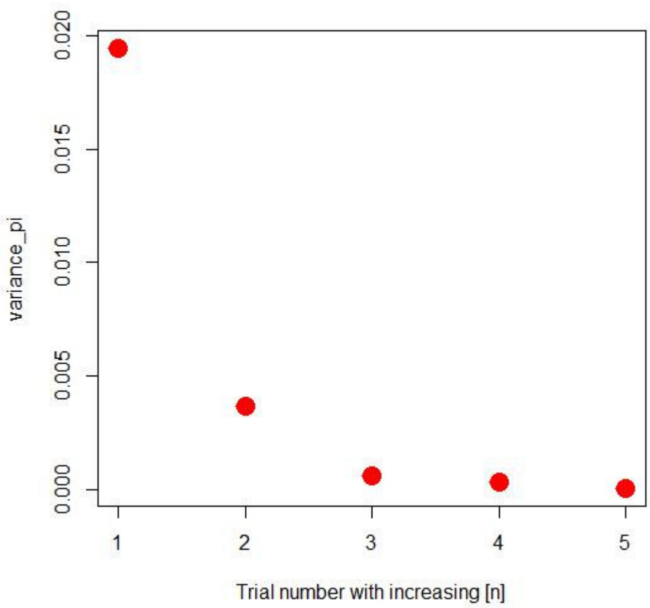
Histogram of (iv) – n = 10000



Histogram of (v) – n = 50000

It is noticed that when the number of trials increase, the number of pi estimates are closer to the actual value.

This is a little clearer on observing the variance plot for the five trials:



Notice how the variance reduces with increasing value of ‘n’ (100, 1000, 5000, 10000, 50000). It indicates that the **estimated** value of pi is getting closer to the real value.

The disadvantage of the MC method is that requires recursive computation and for a large number of trials in order to estimate accurately, and hence this requires a lengthy computation time.

[APPENDIX A] – [programming language : R]

```
#********Monte Carlo Simulation for estimation of PI********

pi_estimates_1 <- NULL

for (trial in 1:5){
for (k in 1:50){
if (trial == 1){
n <- 100
}
else if (trial == 2){
n <- 1000
}
else if (trial == 3){
n <- 5000
}
else if (trial == 4){
n <- 10000
}
else{
n <- 50000
}

count <- 0
samples_x <- runif(n)
samples_y <- runif(n)

for (i in 1:n){
if ((samples_x[i]^2 + samples_y[i]^2)<= (1)){
count = count + 1
}
}
count


##########################
area_estimate <- count/n                    #area of inscribed circle estimate
pi_estimate_1 <- area_estimate * 4          #pi estimate
pi_estimates_1[k] <- pi_estimate_1
#########################
}
x11()
hist(pi_estimates_1)                        #histogram of pi estimates
}

######################################################################
###
#########************variance of pi_estimates*************############
variance_pi <-
c(var(pi_estimates_1),var(pi_estimates_2),var(pi_estimates_3),var(pi_estimates_4),va
r(pi_estimates_5))
plot(variance_pi,xlab='Trial number with increasing [n]',col='red',lwd='10')
```