Coding :

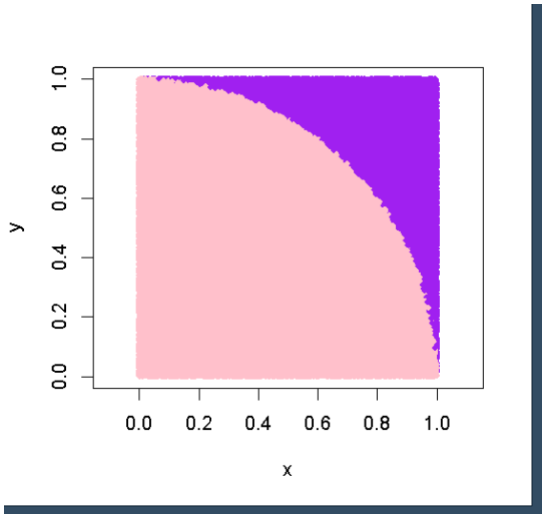
```r
options(scipen = 20)                          #Forcing the program to not use scientific notations

set.seed(277)                                 #Set seed for the randomization as the last 3 digits of my ID to make it uni

my.pi <- function(n){                         #Build "my.Pi" : a function to estimate Pi value

  x = runif(n, min = 0, max = 1)              #Use runif(n) to generate n random coordinate (x,y) points

  y = runif(n, min = 0, max = 1)              #Use runif(n) to generate n random coordinate (x,y) points

  r = sqrt(x^2 + y^2)                         #Calculate radius using the distance between the generated coordinate (x,y)

  num.circle.dots = sum(r <= 1)               #count the dots inside the circle

  num.square.dots = n                         #amount of dots

  ratio = num.circle.dots / num.square.dots   #Find the ratio of a quarter of unit circle and unit square

  my.pi = ratio * 4                           #Multiply by 4 quadrant

  plot(x, y, col = ifelse(r <= 1, "pink", "purple"), asp = 1, pch = 20)  #Plot graph to display the ratio in the first quadrant

  return(my.pi)
}

cat("My pi =", my.pi(2000000), "\n")              #print out the value of estimated pi
```

Table :

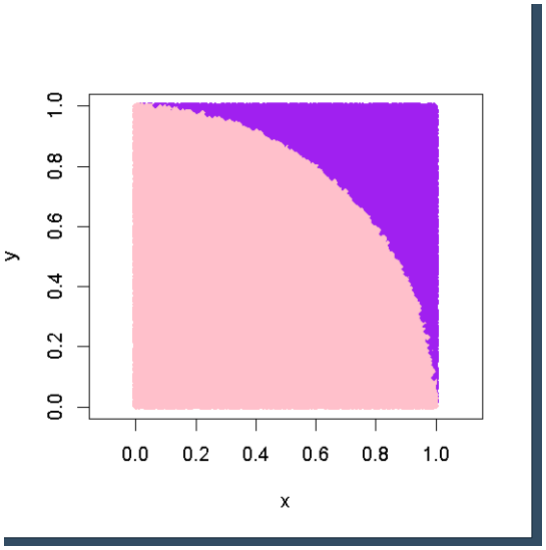
Number of dots	Pi value	Difference from actual Pi
500,000	3.141736	0.000143346411
1,000,000	3.141856	0.000263346411
2,000,000	3.142364	0.000771346411

Graphs :

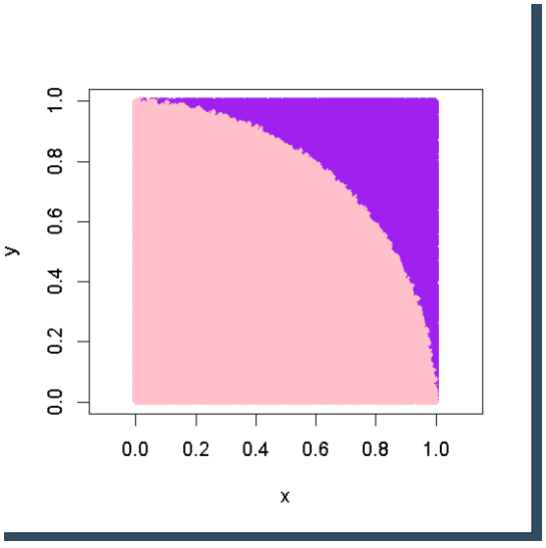
500,000



1,000,000



2,000,000



Conclusion :

By generating random dots in range (0,1), we can estimate pi value with the area of the quadrant of the circle. The range 1 circle will have the area of πr^2 = π(1)^2 = π. Which means our ratio of ¼ circle inside the (1,1) square will have the area of around ¼ π.