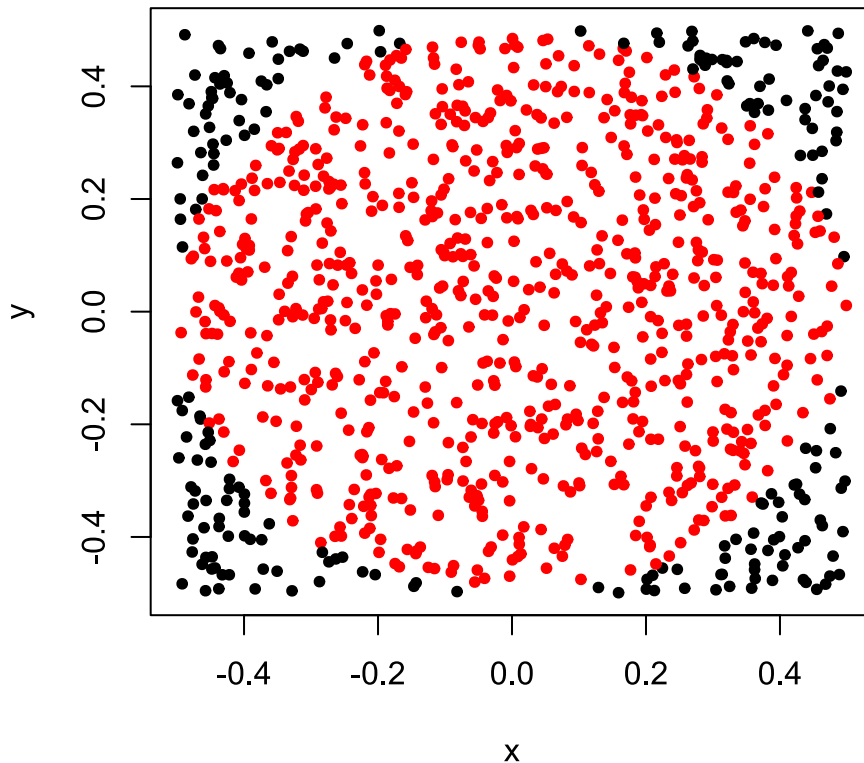


As an introduction to working with R and getting to know each other, we are going to work together to come up with a way to approximate  $\pi$ . We know that the area of a circle is:  $A_c = \pi r^2$  so we can use this to get an estimate. If we take a circle with diameter  $2r$  it will fit exactly inside a square with sides  $2r$ , so the square has an area of:  $A_s = (2r)^2 = 4r^2$ . If we take the ratio of the areas of the circle and square we get:  $\frac{A_c}{A_s} = \frac{\pi r^2}{4r^2} = \frac{\pi}{4}$  which makes:  $\pi = 4 \frac{A_c}{A_s}$ .

Now we just need to figure out how to estimate the ratio of the area of a circle to the area of a square. If we sample from a uniform distribution from  $-0.5$  to  $0.5$  we get a sample of points uniformly distributed over a square with area of 1, the area of the circle will be proportional to the number of samples that are less than or equal to a distance of  $0.5$  (this is  $r$ ) from the center of the square.

```
set.seed(120414)
x <- runif(1000, -.5, .5)
y <- runif(1000, -.5, .5)
in_circ <- ifelse(sqrt(x^2 + y^2) <= .5, 1, 0)
plot(y ~ x, pch = 20, col = c("black", "red")[in_circ + 1])
```



Let's use a larger number of samples to get a better estimate in the end!

```
x <- runif(1e4, -.5, .5)
y <- runif(1e4, -.5, .5)
in_circ <- ifelse(sqrt(x^2 + y^2) <= .5, 1, 0)
plot(y ~ x, pch = 20, col = c("black", "red")[in_circ + 1])
```

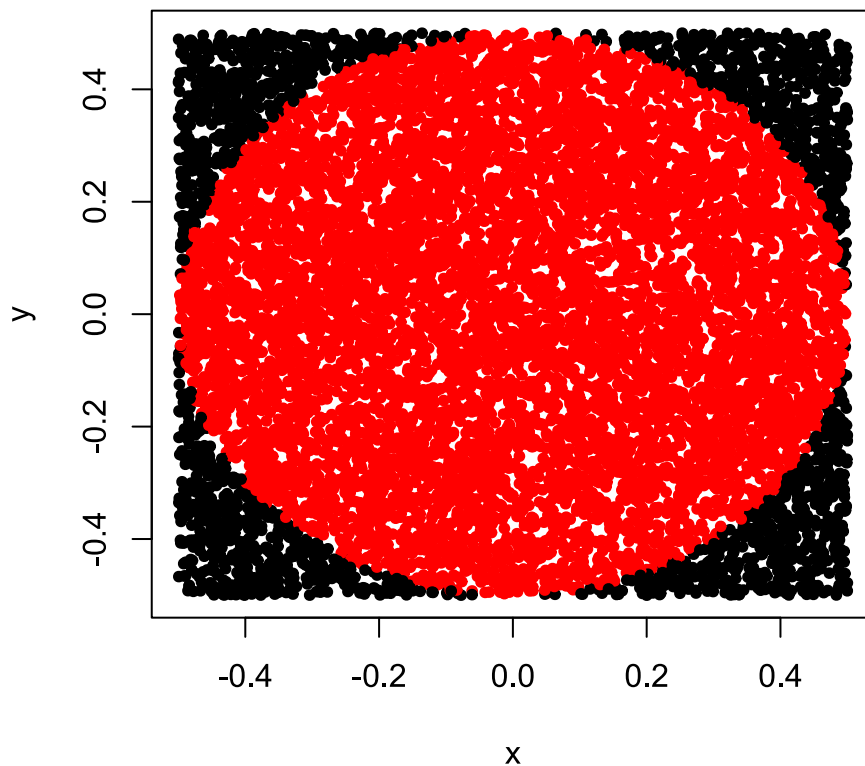


Figure 1: Looks like a circle in a square!

Now we just divide the number of points that were inside the circle by the total number of points to get an estimate of the ratio of the circle to square areas. Then we multiply by 4 and that gives us our approximation of  $\pi$ !

```
4*sum(in_circ)/1e4
```

```
[1] 3.1564
```

Pretty close! Let's replicate this 1000 times to get a distribution of approximations and see how it compares to the true value. First we need to make a function that does the simulation above.

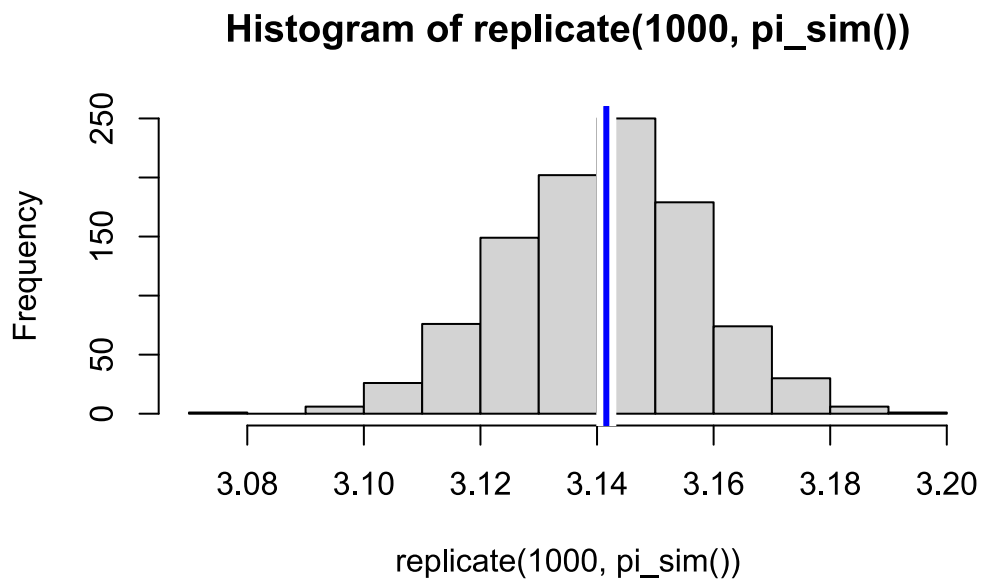
```
pi_sim <- function(){
  x <- runif(1e4, -.5, .5)
  y <- runif(1e4, -.5, .5)

  in_circ <- ifelse(sqrt(x^2 + y^2) <= .5, 1, 0)

  4 * sum(in_circ)/1e4
}
```

Now we can use `replicate` to run this function for 1000 iterations. And plot the histogram with a vertical line at the real value of  $\pi$ .

```
hist(replicate(1000, pi_sim()))
abline(v = pi, col = "white", lwd = 10)
abline(v = pi, col = "blue", lwd = 3)
```



Looks like we are pretty much centered on the true value of  $\pi$ ! Pretty cool!