# Método de Monte Carlo

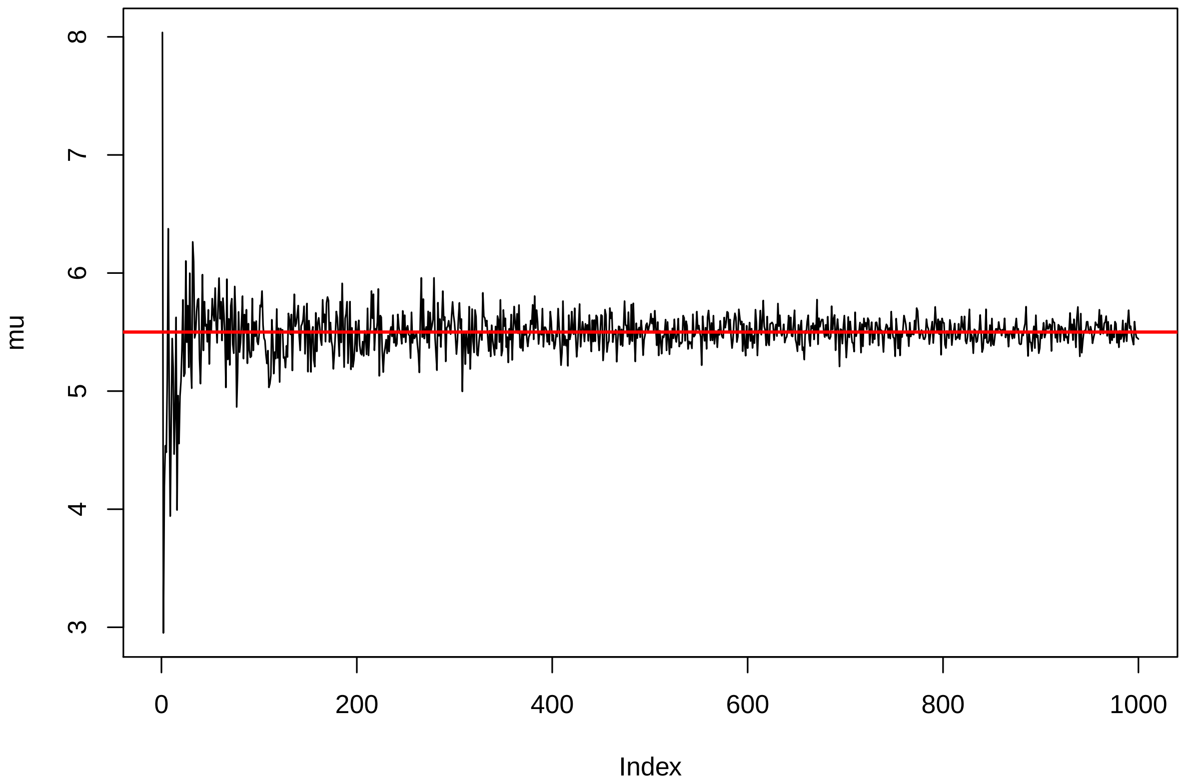
Análisis Numérico

Joaquin Cavieres G.

## Método de Monte Carlo

### Ejemplo 1

```
n  = 1000                            # N de iteraciones
mu = 0                               # Vector de medias
for(i in 1:n){
  mu[i] = mean(runif(i,1,10))        # Muestreando desde i samples con ~unif(1,10)
}
plot(mu, type ='l')                  # Graficar las muestras
abline(h = 5.5, col='red', lwd = 2) # Graficar la media de la poblacion
```




**Ejemplo 2**

Imaginemos un circulo en un cuadrado de lado 1 y el radio del circulo debería ser inicialmente $r = 0{,}5$. Sabemos que el área de un circulo es:

$$\pi * r^2$$

y el área del cuadrado en terminos del radio debería ser:

$$(2 * r)^2$$

Finalmente, el radio de las áreas:

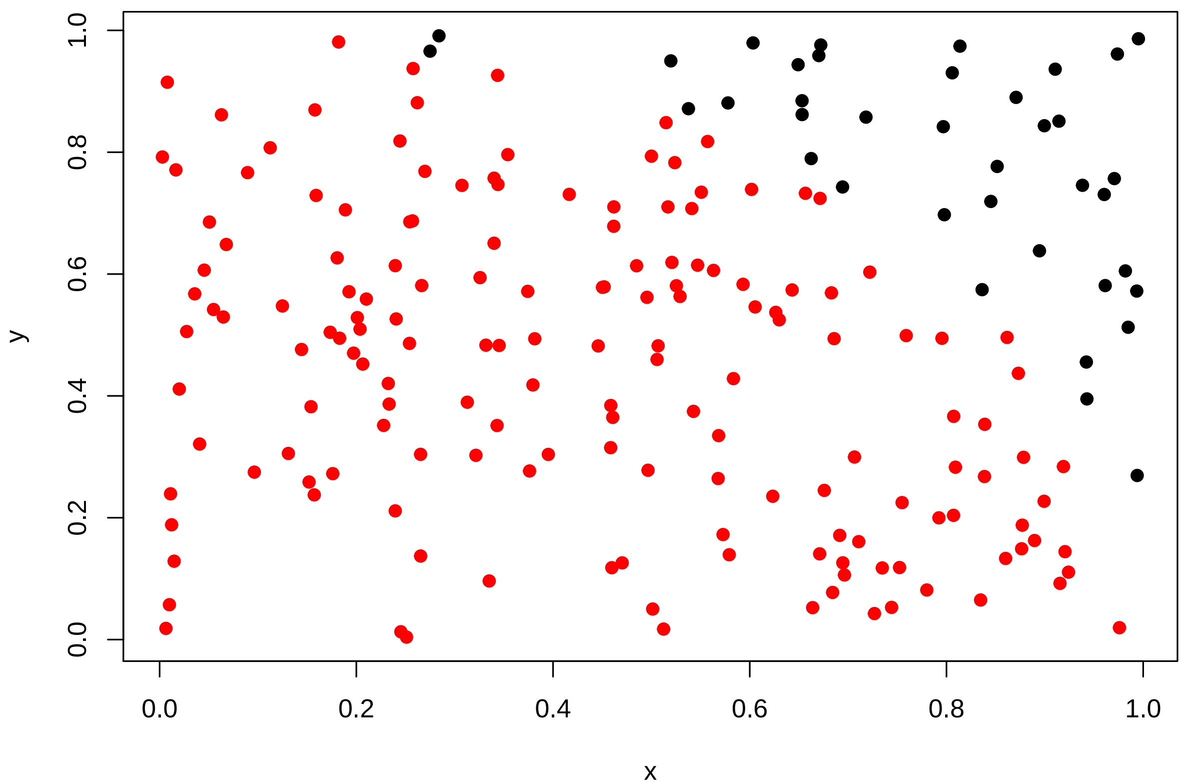
$$\rho = \frac{\text{Área del circulo}}{\text{Área del cuadrado}} = \frac{\pi * r^2}{(2 * r)^2} = \frac{\pi}{4}$$

así que:

$$\pi = 4 * \rho$$

Para aproximar $\rho$ debemos generar puntos aleatorios en $[0-1, 0-1]$ y calculamos la razón de puntos que no "caen" dentro de la circunferencia y los puntos que caen dentro de la circunferencia. Hagamos la simulación para 200 puntos:

```
n<- 200
x<- runif(n,0,1)
y<- runif(n,0,1)
d <- (x^2+y^2<1)
plot(x,y, col= d+1, pch=19)
```




Vemos los que caen en el circulo

```
adentro <-sum(d)
adentro
```

```
## [1] 162
```

y afuera del circulo:

```
afuera <-(n - adentro)
afuera
```

```
## [1] 38
```

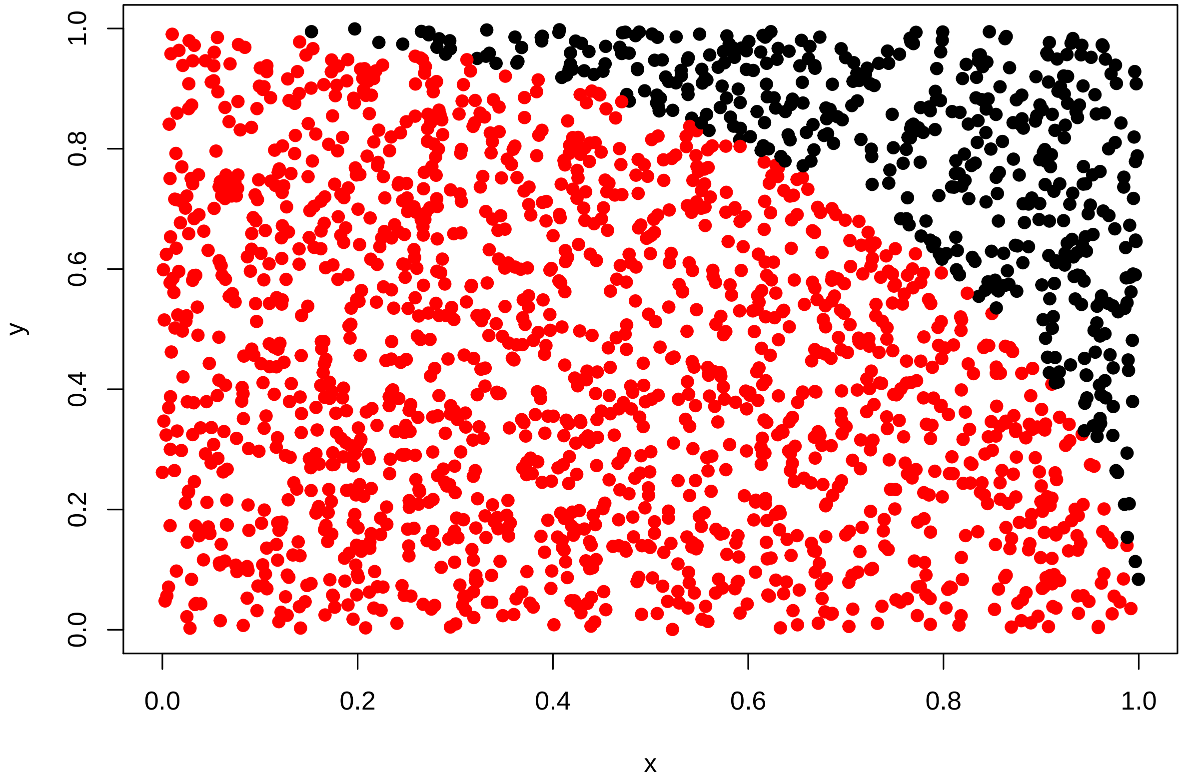
Calculamos $\pi$

```
pi <- (adentro / n)*4
pi
```

```
## [1] 3.24
```

y repetimos el ejercicio para 2000 puntos:

```
n<-2000
x<-runif(n,0,1)
y<-runif(n,0,1)
d<-(x^2+y^2<1)
plot(x,y, col= d+1, pch=19)
```




Repetimos lo hecho previamente:

```
adentro <-sum(d)
adentro
```

```
## [1] 1543
```

y afuera:

```
afuera <-(n - adentro)
afuera
```

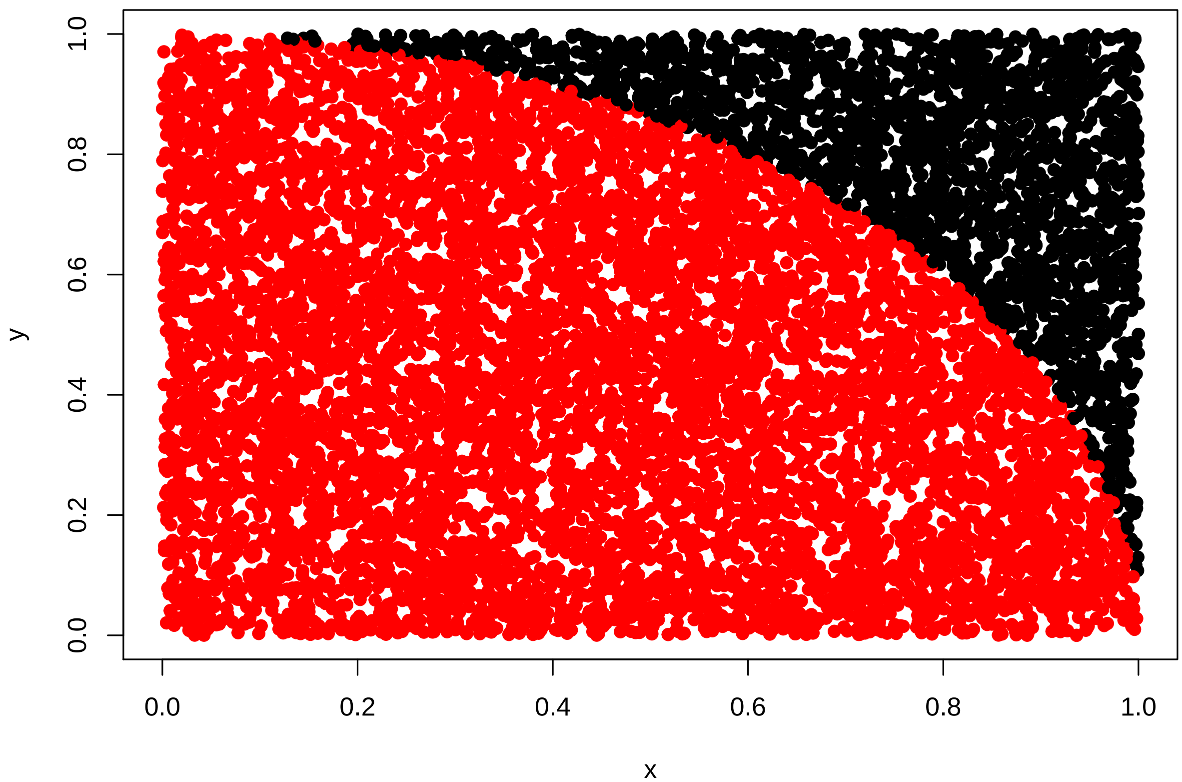
```
## [1] 457
```

Calculamos $\pi$ nuevamente

```
pi2 <- (adentro / n)*4
pi2
```

```
## [1] 3.086
```

```
n <- 10000
x <- runif(n,0,1)
y <- runif(n,0,1)
d <- (x^2+y^2<1)
plot(x, y, col = d+1, pch=19)
```




en el circulo,

```
adentro <-sum(d)
adentro
```

```
## [1] 7829
```

y fuera del circulo

```
fuera <- (n - adentro)
fuera
```

```
## [1] 2171
```

$\pi$ para los 10000 puntos

```
pi3 <- (adentro/n)*4
pi3
```

```
## [1] 3.1316
```

Hagamos un calculo para 500000 observaciones:

```
n <- 500000
x <- runif(n,0,1)
y <- runif(n,0,1)
d <-(x^2+y^2<1)

adentro <-sum(d)
adentro
```

```
## [1] 393252
```

```
afuera <-(n - adentro)
afuera
```

```
## [1] 106748
```

```
pi4 <- (adentro/n)*4
pi4
```

```
## [1] 3.146016
```

Veamos los mismos resultados pero ahora en una función:

```
montecarlo <- function (f, a, b, m = 1000) {
x <- runif (m, min = a, max = b)
y.hat <- f(x)
pi <- (b - a) * sum(y.hat) / m
return (pi)
}
```

Hacemos la evaluación:

```r
f <- function(x) { sqrt(1 - x^2) }
montecarlo(f, 0, 1, m = 1e3) * 4
```

```
## [1] 3.088719
```

```r
# Aumentemos el m (núm de iteraciones)
montecarlo(f, 0, 1, m = 1e6) * 4
```

```
## [1] 3.141154
```