

Simulación de π mediante variables aleatorias uniformes

Andrés Miniguano Trujillo

Curso de L^AT_EX, 201

Contenido

1 Proceso para estimar π

2 Aproximación de π

Proceso para estimar π

Ley de los grandes números

Mediante la ley de los grandes números se puede estimar el valor de π por medio de un algoritmo semejante a la regla de Laplace:

Proceso

- 1 Fijar un número natural n suficientemente grande e inicializar $N = 0$.
- 2 Generar $U_i \sim \mathcal{U}(0,1)$, vectores de dos componentes para cada $i \in \llbracket 1, n \rrbracket$.
- 3 Si $U_{i,1}^2 + U_{i,2}^2 \leq 1$, entonces $N \leftarrow N + 1$.
- 4 Calcular $\tilde{\pi}_1 = 4N/n$. Este valor es una aproximación de π .

- 1 Fijar un número natural n suficientemente grande e inicializar $N = 0$.
- 2 Generar $U_i \sim \mathcal{U}(0,1)$, vectores de dos componentes para cada $i \in \llbracket 1, n \rrbracket$.
- 3 Si $U_{i,1}^2 + U_{i,2}^2 \leq 1$, entonces $N \leftarrow N + 1$.
- 4 Calcular $\tilde{\pi}_1 = 4N/n$. Este valor es una aproximación de π .

- 1 Fijar un número natural n suficientemente grande e inicializar $N = 0$.
- 2 Generar $U_i \sim \mathcal{U}(0,1)$, vectores de dos componentes para cada $i \in \llbracket 1, n \rrbracket$.
- 3 Si $U_{i,1}^2 + U_{i,2}^2 \leq 1$, entonces $N \leftarrow N + 1$.
- 4 Calcular $\tilde{\pi}_1 = 4N/n$. Este valor es una aproximación de π .

- 1 Fijar un número natural n suficientemente grande e inicializar $N = 0$.
- 2 Generar $U_i \sim \mathcal{U}(0,1)$, vectores de dos componentes para cada $i \in \llbracket 1, n \rrbracket$.
- 3 Si $U_{i,1}^2 + U_{i,2}^2 \leq 1$, entonces $N \leftarrow N + 1$.
- 4 Calcular $\tilde{\pi}_1 = 4N/n$. Este valor es una aproximación de π .

La idea es bastante sencilla: En el cuadrado de lado 1 y centrado en $(1/2, 1/2)$ generamos dos variables aleatorias uniformes (coordenadas de puntos). Luego contamos los puntos que caen dentro del círculo de radio 1 y centrado en $(0, 0)$. Finalmente, una aproximación de π está dada por 4 veces la razón entre los puntos dentro del círculo y el número total de puntos simulados. Pondré a prueba esta forma de simular π :

Forma para simular π

```
> n <- 5000;  
> U <- runif(2 * n, 0, 1);  
> V <- -(U[1 : n]^2 + U[(n + 1) : (2 * n)]^2)  
> a <- -V <= 1  
> pi1 <- 4 * sum(a)/n
```

Aproximación de π

Aproximación de π

Obtengo la siguiente aproximación con su respectivo error:

π_1	error	error relativo
3.1592	0.01761	0.0056

Aproximación de π

Ahora realizaré esta prueba varias veces:

```
> m < -2000; pi1 < -rep(0, m);  
> for(kin1 : m){  
+   U < -runif(2 * n, 0, 1);  
+   V < -(U[1 : n]^2 + U[(n + 1) : (2 * n)]^2)  
+   a < -V <= 1; pi1[k] < -4 * sum(a)/n  
+ }
```

Aproximación de π

Presento a continuación el histograma de frecuencia relativa de los errores relativos:

```
> hist(abs(pi1-pi)/pi, density=100, border="beige",  
main="Histograma de error relativo", + xlab=,  
ylab="Frecuencia relativa", freq=FALSE)
```

Aproximación de π

Histograma de error relativo

