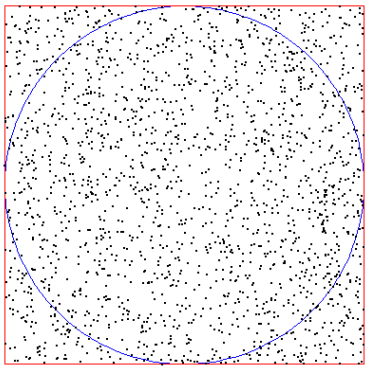
El método Montecarlo

Existe un modo curioso de calcular el valor aproximado de π. Para ello, debemos dibujar un cuadrilátero, y dentro de él un círculo.

Una vez dibujado, lo colocamos bajo la lluvia de modo que le caiga una buena cantidad de gotas. Como hoy es un día soleado, simularemos las gotas con ayuda de la computadora.



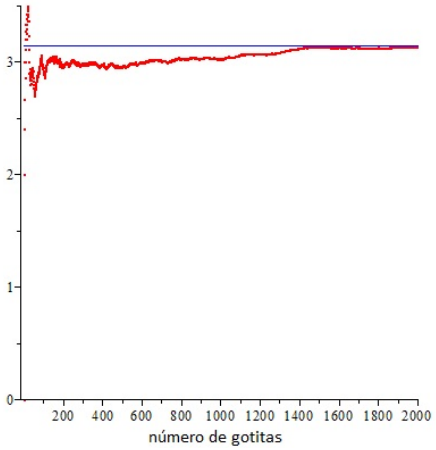
Como las gotas de lluvia se reparten al azar sobre la superficie del cuadrado, es de esperar que la probabilidad de que una gota caiga dentro del círculo sea proporcional al área del mismo, y que la probabilidad de que caiga en el cuadrado sea, también, proporcional al área del a cuadrado.

$$\frac{Gotas_{crculo}}{Gotas_{cuadrado}} \approx \frac{rea_{crculo}}{rea_{cuadrado}} \Rightarrow \frac{Gotas_{crculo}}{Gotascuadrado} \approx \frac{\pi r^2}{4r^2} = \frac{\pi}{4}$$

Por lo que podemos despejar a π como:

$$\pi \approx 4\frac{Gotas_{crculo}}{Gotas_{cuadrado}}$$

En la siguiente gráfica podemos ver cómo el valor aproximado de Pi, calculado de éste modo, se aproxima al valor real cuando el número de gotas se hace grande.



Éste tipo de métodos se utilizan muy a menudo en cálculo numérico, pero en lugar de incómodas gotas de lluvia se usan puntos al azar generados por una computadora. Se conocen como métodos de Montecarlo, en honor a sus famosos casinos (por aquello del azar).

Como a la computadora no le da pereza ponerse a contar puntitos, voy a pedirle que simule 100000 gotitas. El resultado obtenido en un caso como ese es:

$\pi \approx 4\frac{78539}{100000} = 3.14156$, que es evidentemente una buena aproximación.

Ejercicios

1. El objetivo es reproducir la gráfica de estimaciones (titulada número de gotitas).
2. El equipo decidirá cuál será el método y número de gotitas utilizadas.
3. En la sesión de clase discutiremos cómo es que la semilla aleatoria es importante en esta gráfica.
4. El equipo entregará un reporte en pdf de dos páginas con la siguiente información:
   - Explicando en qué consiste el método Montecarlo para estimación de áreas.
   - Describiendo detalladamente el método de estimación que utilizaron.
   - Presentando la gráfica donde se tiene la relación del número de gotitas contra la estimación del área deseada.
   - Presentando el código en R que se utilizó para obtener las estimaciones (incluida la gráfica presentada).

Explicación del método Montecarlo

El método montecarlo consiste en la realización de pruebas aleatorias para un experimento dado con el objetivo de poder acercarnos a la pregunta de interés, este procedimiento se basa en la probabilidad y estadística. En el ejemplo que vamos realizar consistirá en la obtención del número π, para ello utilizaremos el método montecarlo que consistirá en simular gotas de lluvias en una cartulina con un circulo dibujado en ella, y se harán varias simulaciones. Utilizaremos una semilla para generar números aleatorios que representaran las gotas de lluvia que caen dentro del circulo, por eso mismo este experimento se considera como un método de montecarlo ya que utilizaremos números aleatorios para simular las gotas de lluvia y con esto poder aproximar π. Aproximaremos π pensado en la probabilidad de que una gota caiga dentro del círculo que es proporcional al área del mismo, y que la probabilidad de que caiga en el cuadrado sea también, proporcional al área del a cuadrado.

Para este experimento, primeros necesitaremos conocer las funciones que nos ayudaran a generar numeros aleatorios, en R tenemos:

```r
# Obtencin de nmeros aleatorios con sample, devuelve entero.
sample(1:8, 2, replace = TRUE)

# Con runif() devuelve un racional.
runif(1, -1, 1)
```

Dado que nuestro experimento usaremos un circulo unitario que vaya del -1 a 1 lo mejor opción es utilizar la funcion ‘runif()‘ que devuelve un numero racional.

Para proseguir con nuestro experimento, crearemos una función que detecte las gotas que estén adentro del circulo:

```
#' Cuenta las gotas que estn dentro del circulo.
#'
#' @param n numero de gotas del experimento.
#'
#' @return Devuelve el numero de gotas que estn dentro del circulo.
gotas_dentro <- function(n){
  x <- runif(n, -1, 1)
  y <- runif(n, -1, 1)
  gotas <- 0
  for(i in 1:n){
    if(x[i]^2+y[i]^2 <= 1){
      gotas = gotas + 1
    }
  }
  return(gotas)
}
```

Para hacer nuestra simulación, utilizaremos nuestra función de números aleatorios donde cada numero simula la posición del eje x y y de la cartulina rectangular. Usando la ecuación del circulo determinaremos si el par ordenado generado por la función ‘runif()‘ esta dentro del circulo o no. Haremos esto por cada gota del experimento y devolviendo al final el numero de gotas dentro del circulo.

Para seguir con nuestra simulación, crearemos nuestra función principal la cual llamara a la función ‘gotas_dentro()‘ y calculara el número aproximado de π.

```
#' Simula el experimento de gotas de lluvia para aproximar \pi.
#'
#' @param num_gotas es el numero total de gotas del experimento.
#' @param graficar un boolean que indica si desea graficar el experimento.
#'
#' @return un vector que contiene todas la aproximaciones de \pi.
simulacion_lluvia <- function(num_gotas, graficar = TRUE) {
  x <- 1:num_gotas
  aprox_pi <- c()
  for(i in 1:num_gotas){
    aprox_pi[i]= (4*gotas_dentro(i))/i
  }
  if (graficar) graficar_simulacion(x, aprox_pi)
  return(aprox_pi)
}
```

En esta función encontraremos los valores de pi por cada experimento a realizar, creamos un vector ‘aprox_pi‘ el cual almacenará los valores de pi, creamos un ‘for‘ que es el encargado de hacer la simulación de cada experimento, cada valor de i representa el

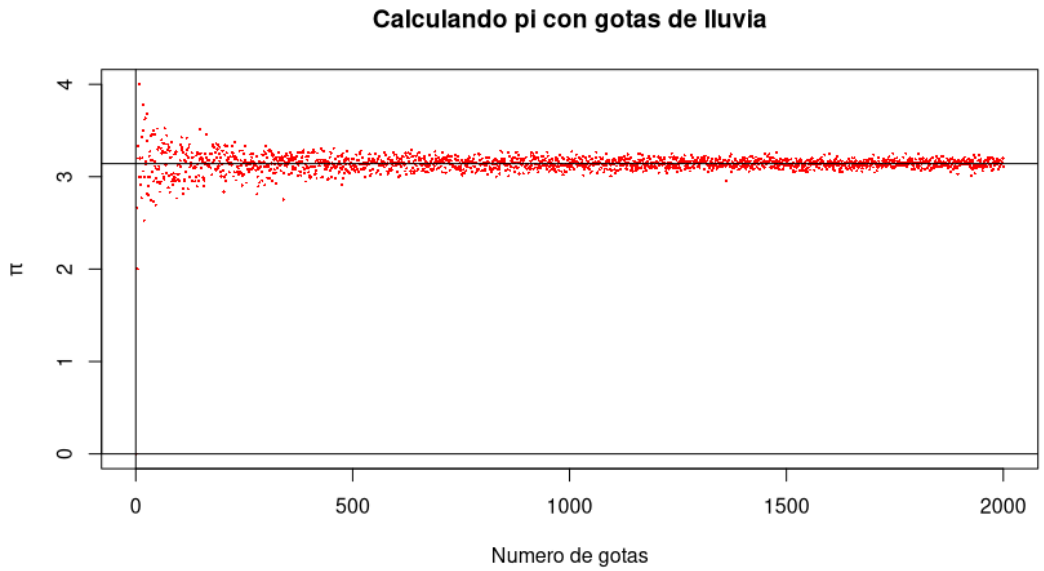
numero de gotas de un experimento. Por ultimo, verificamos si el usuario desea graficar el experimento y lo graficamos, en caso contrario, regresemos un vector que contiene todas las aproximaciones de π por cada experimento.

La función 'graficar_simulacion' solo es la encargada de de hacer la gráfica correspondiente y darle presentación de la mejor manera.

```
#' Genera una grafica que representa la simulacin.
#'
#' @param x es un vector con un numero de gotas.
#' @param y es un vector con todas la aproximaciones de pi.
graficar_simulacion <- function (x, y) {
  plot(x , y,
       main = "Calculando pi con gotas de lluvia",
       ylab = expression(pi),
       xlab = "Numero de gotas",
       pch=16, cex=.3, col="red")
  abline(0,0,pi,0)
}
```

Por último, llamaremos nuestra función y veremos los resultados:

```
aproximaciones_pi <- simulacion_lluvia(2000)
```



Podemos observar en la gráfica como se va acercando al valor de π conforme se va aumentando el número de gotas en cada experimento. Vemos que al principio al ser muy pocas gotas la probabilidad de que caiga adentro del circulo es menor, pero con forma más gotas existan en el experimento la probabilidad de que caigan dentro del circulo es mucho mayor, por lo tanto al tener más gotas dentro del circulo podemos acernos aún más al valor de π.

Conclusión

Podemos ver que conforme hacemos más experimentos se puede acercar más al valor de π debido a las probabilidades que existen como consecuencia de la relación del área del circulo con respecto al cuadrado, también un factor a considerar es la semilla que toma la función 'runif()' para la generación de números aleatorios, ya que si no se tiene una buena semilla podría verse un patrón con los números, lo cual implicaría que nuestro experimento estaría sesgado y no obtendríamos valores confiables. Por último, el método de montecarlo es muy útil cuando un problema se puede resolver haciendo un conjunto de experimentos para acercarnos más a un resultado, aunque en la vida real hacer experimentos es demasiado costoso por lo que la mejor forma de hacerlo es mediante simulaciones con un ordenador.

Referencias

1. https://naukas.com/2012/02/29/calculando-pi-con-gotas-de-lluvia/
2. https://es.wikipedia.org/wiki/Aguja_de_Buffon
3. https://fuga.naukas.com/