# Práctica 8: Modelo de urnas

*Alumno: José Adrian Garcia Fuentes*
*Profesor: Satu Elisa Schaeffer*

*Universidad Autónoma de Nuevo León, Facultad de Ingeniería Mecánica y Eléctrica*
*16 de junio de 2021*

## 1. Introducción

El modelo de la urna es uno de los elementos de mayor uso en la aplicación de probabilidades estadísticas ya que tiende a ser un concepto que permite la facilidad de comprender gráficamente las distintas formas en las cuales puede aplicarse dicho modelo, un modelo de urnas es aquel que trata de simular un contenedor con elementos para calcular la probabilidad de extraer un elemento de la jornada, alguna característica del elemento, por ejemplo, podemos tener una urna con bolas de un tamaño y bolas de otro donde se desea conocer cuál es la probabilidad de una bola de menor tamaño. En la práctica se simula un sistema en donde se abordan los fenómenos de coalescencia y fragmentación de partículas, donde las partículas se unen y se descomponen para formar cúmulos, estos fenómenos son de gran utilidad al realizar análisis en muchas áreas como en física y química. Esto puede servir en la práctica de laboratorio como para lograr predecir qué cantidad de partículas quedaran atrapadas en un filtro de cierta apertura de poro, por ejemplo, supongamos que tenemos alguna solución y deseamos filtrar una determinada partícula una de las características más relevantes de dicha partícula es su tamaño se cuenta con una red que sólo captura las partículas de tamaño que se especifique, mediante el procedimiento solicitado por el uso del modelo de urnas aplicando el principio para este modelo generaríamos un número de enteros distribuido de tal manera que se agrupen al tamaño de los cúmulos originalmente iniciado.

## 2. Objetivo

Graficar el porcentaje que se logra filtrar en cada iteración.

## 3. Resultados

Para la simulación se toman en cuenta dos parámetros principales, que son el número de partículas $n = 16k$, $32k$, $64k$, $128k$ y el número de cúmulos $k = 1000$ [1]. La metodología empleada se realizó a través de Rstudio [2] llevando a cabo los pasos señalados en la práctica 8: modelo de urnas [1], a partir del código en el repositorio de Schaeffer [3], se realizaron modificaciones, los resultados de la experimentación los podemos ver en la figura 1 donde el eje vertical nos indica el porcentaje de las partículas que se logran filtrar y en el eje horizontal la iteración, correspondiendo los colores a la cantidad de cúmulos.

```
vectorn <- c(16*k,32*k,64*k,128*k)

 for (replica in 1:30) {
   basefiltrados <- c()
     for (n in vectorn) {
```

Los valores se agregan al vector n solicitado por el número de cúmulos y se realizan un total de 30 réplicas, se crea la variable filtrado para guardar los valores.

```
filtrado <- c()
```

```
 h <- cumulos[cumulos > c]
     filtrado[paso] <- sum(h) / n
   }
   basefiltrados <- cbind(basefiltrados,
   filtrado)
 }
 colnames(basefiltrados) <- vectorn

```

Para mejorar la resolución de la imagen se agregó formato `.tiff` sustituyendo el formato `.png`, se añadió una leyenda en la parte inferior derecha para una mayor comprensión visual del gráfico.

```
 tiff(paste("p8_", replica, ".tiff", sep=""),
     width=12, height=12, units="cm", res=600,
     pointsize = 10)
```

```
plot(basefiltrados[,1], type = "l", col= "
  red", ylim=c(min(basefiltrados), max(
  basefiltrados)),
     main = paste("Replica", replica), xlab
  = "Iteraciones", ylab = "Porcentaje de
  filtraciones")
lines(basefiltrados[,2], type = "l", col= "
  blue")
lines(basefiltrados[,3], type = "l", col= "
  black")
lines(basefiltrados[,4], type = "l", col= "
  green")
legend("bottomright",legend = c("16k","32k",
   "64k", "128k"),fill=c("red","blue", "
  black", "green"))
dev.off()
print(replica)
}
```

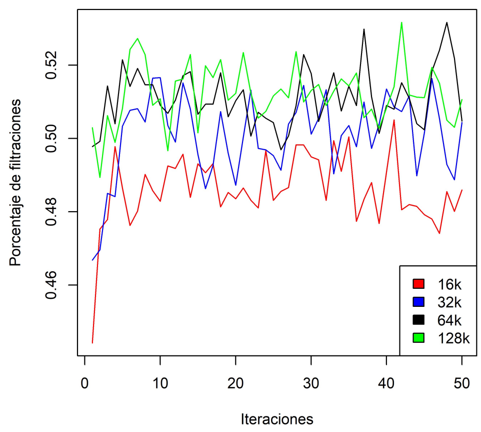


Figura 1: Porcentaje de filtración de los cúmulos en la réplica 1.

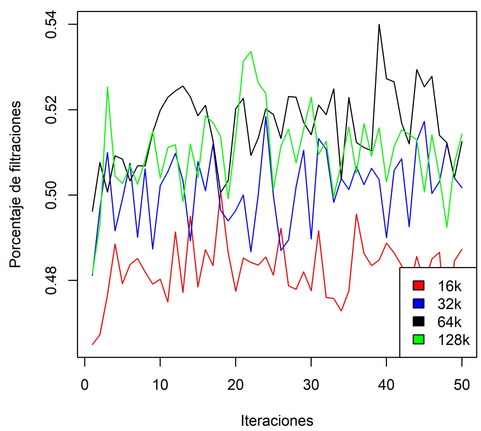


Figura 2: Porcentaje de filtración de los cúmulos en la réplica 30.

# 4. Conclusión

En la figura 1 se muestra que los cúmulos de mayor tamaño ($64k$, $128k$) tienen un mayor porcentaje de filtración que los de menor tamaño ($16k$, $32k$). A diferencia de lo que se puede observar en la figura 2 el valor más alto de la curva se encuentra en los cúmulos principales en este caso en la iteración 40 de 64 cúmulos (color negro) y a medida que avanza la iteración se observa que el filtrado avanza, por tanto los cúmulos más grandes se filtran con mayor facilidad pero tardan más en formarse que los cúmulos pequeños.

# 5. Reto 1

Determina si algún intervalo de iteraciones en el que el filtrado alcance un óptimo. Realiza réplicas para determinar si el momento en el cual se alcanza el máximo tiene un comportamiento sistemático. Incluye visualizaciones para justificar las conclusiones.

```
x1 <- c()
  for (m in 1:ncol(basefiltrados)) {
    x<-which.max(basefiltrados[,m])
    x1<-c(x1, x)
  }
  x2 <- rbind(x2,x1)
}
colnames(x2) <- vectorn
rownames(x2) <- seq(1:replica)
```

Mediante which.max se obtienen los valores más altos de las 50 iteraciones agrupadas en filas para los 4 diferentes tamaños de $n$ (número de partículas) los resultados obtenidos se muestran en las figuras 3, 4, 5, 6 obteniendo resultados variables.

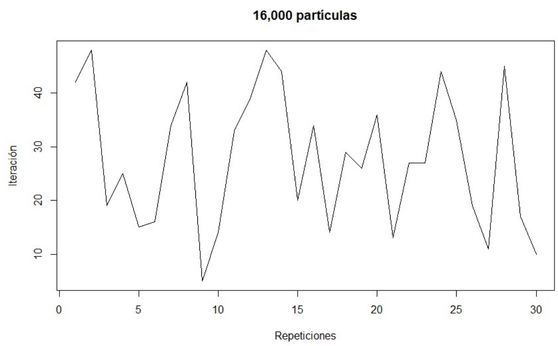


Figura 3: Gráfico de réplicas filtrando los mejores valores de $16k$.

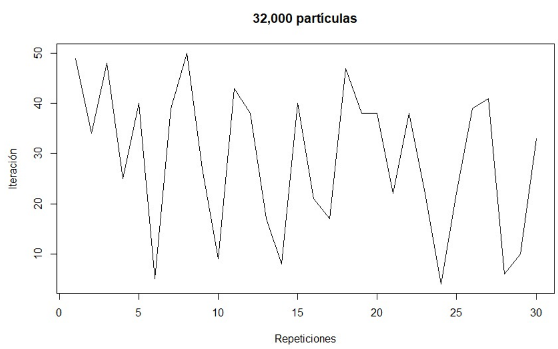


Figura 4: Gráfico de réplicas filtrando los mejores valores de $32k$.

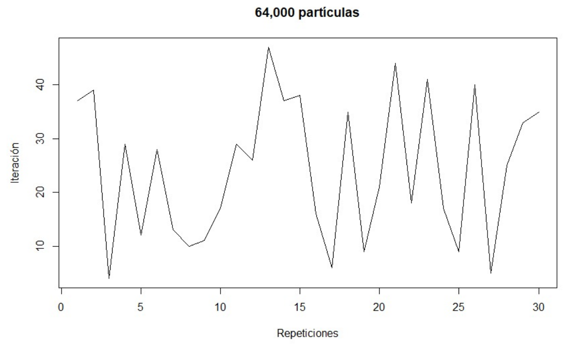


Figura 5: Gráfico de réplicas filtrando los mejores valores de $64k$.

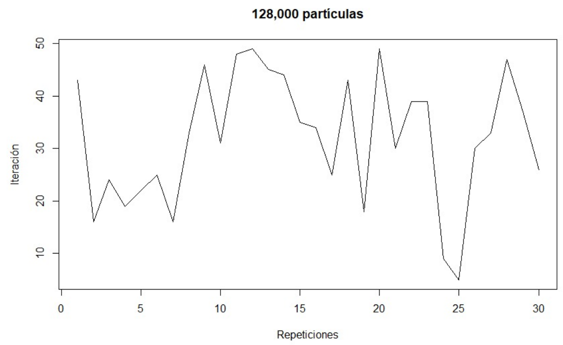


Figura 6: Gráfico de réplicas filtrando los mejores valores de $128k$.

Para el reto uno se puede concluir que el intervalo de iteraciones realmente es muy variable sin embargo para el caso de $32k$ y $128k$ puede llegar a alcanzar picos más altos que los cúmulos de $16k$ y $64k$, en la figura 4 se muestra un comportamiento uniforme en el largo y ancho de los picos comparado con el resto.

## 6. Reto 2

Determina cómo los resultados de la tarea y del primer reto dependen del valor de $c$. ¿Qué cambia y cómo si $c$ ya no se asigna como la mediana inicial sino a un valor menor o mayor?

```
for (q in 1:2) {
```

Primero se genera un ciclo for para la variable $q$ cuando $q$ sea igual a 1 la variable $c$ tomará el valor y se le restará la desviación estándar en otro caso se sumará, provocando la modificación de $c$.

```
if (q == 1) {
      c <- median(cumulos) - sd(cumulos) #
    tamanio critico de cumulos
    } else {c <- median(cumulos) + sd(cumulos)
    }
```

En las figuras 7 y 8 se muestran los gráficos con valores mayores y menores que $c$ respectivamente, comparando con la tarea base y el primer reto los valores de filtración son más uniformes salvo a excepción del menor número de cúmulos $16k$ para el caso de $c$ menor.

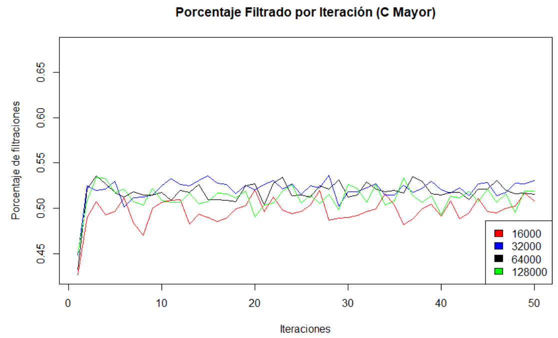


Figura 7: Gráfico del porcentaje filtrado por iteracción $c$ mayor.

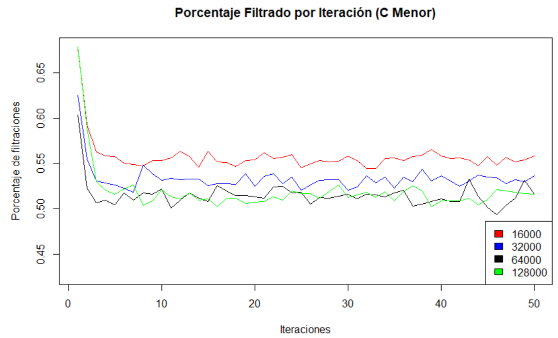


Figura 8: Gráfico del porcentaje filtrado por iteracción $c$ menor.

Para el caso de los mejores filtrados y llevando a cabo los pasos del reto 1 se observan en las figuras 9 y 10 que si $c$ es menor el porcentaje de filtración será muy bajo sin embargo si $c$ es mayor el porcentaje de filtración será variable.

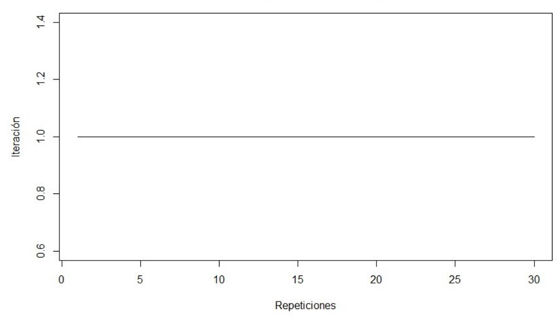


Figura 9: Gráfico de $16k$ con un valor de $c$ menor.

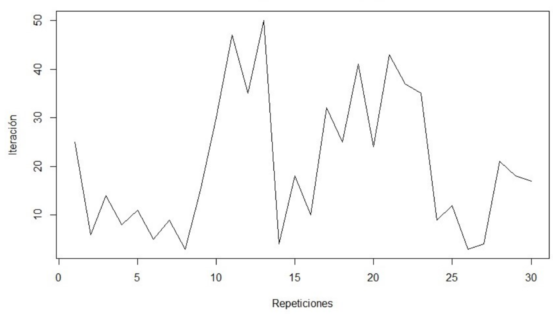


Figura 10: Gráfico de $16k$ con un valor de $c$ mayor.

Para el reto dos se puede concluir que al modificar el valor de $c$ ya sea mayor o menor se obtiene un porcentaje de filtrado más uniforme para las 4 cantidades diferentes de $n$ salvo alguna excepción para el caso de $c$ menor, pero esto puede deberse a la cantidad de $n$ en $16k$.

# Referencias

[1] E. Schaeffer, "Práctica 8: Modelo de urnas," abril 2021. https://elisa.dyndns-web.com/teaching/comp/par/p8.html.

[2] J. J. Allaire, "Rstudio," mayo 2021. https://rstudio.com.

[3] E. Schaeffer, "Práctica 7: Búsqueda local," ABRIL 2021. https://github.com/fuentesadrian/Simulation/tree/master/LocalSearch.