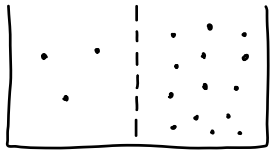
MODELO DE EHRENFEST: Modelo de difusion de particulas

Considera una urna dividida en dos por una membrana que deja pasar las particulas



X_t = nº de particulas en la parte izquierda

$E = \{0, \dots N\}$ donde N es el numero total de particulas

Transicion: se elige aleatoriamente y de manera uniforme una particula y se la cambia de lado

$$\mathbb{P}\{X_{t+1} = j \mid X_t = i\} = \begin{cases} i/N & \text{si } j = i-1 \\ N-i/N & \text{si } j = i+1 \\ 0 & \text{EN OTRO CASO} \end{cases}$$

$$P = \begin{array}{c|c} & \begin{matrix} 0 & 1 & \dots & & N \end{matrix} \\ \begin{matrix} 0 \\ 1 \\ \vdots \\ \\ N \end{matrix} & \begin{pmatrix} 0 & 1 & & & \\ 1/N & \ddots & \frac{N-1}{N} & \ddots & \\ & \ddots & \ddots & & \\ & & & & 1/N \\ & & & 1 & 0 \end{pmatrix} \end{array}$$

W8 – 11/11

Modelo de Ehrenfest

Un modelo clásico de difusión de partículas a través de una membrana es el famoso *modelo de Ehrenfest*. El modelo considera, de manera abstracta, que existen dos urnas conteniendo entre ambas un cierto número N de bolas y que, en cada paso, se cambia de urna una de las bolas, elegida al azar. De esta forma, las partículas (representadas por las bolas) fluctúan entre los dos espacios separados por la membrana (representados por las urnas) con un sentido desde el espacio con mayor concentración hacia el espacio con menor concentración de partículas.

Si nos fijamos en una de las urnas y consideramos que la variable X_t cuenta el número de bolas que contiene en el instante t, se tiene una cadena de Markov cuyos estados son los números de 0 a N.

```
N <- 4
estados_Ehrenfest <- as.character(0:N)
```

→ LOS ESTADOS DE LA CATENA NO PUEDEN SER NUMÉRICOS (PRUEBA ELIMINANDO AS. CHARACTER → ERROR)

Las probabilidades de transición se pueden calcular fácilmente

$$P(X_{t+1} = j \mid X_t = i) = \begin{cases} \frac{N-i}{N} & \text{si } j = i + 1, \\ \frac{i}{N} & \text{si } j = i - 1, \\ 0 & \text{en otro caso.} \end{cases}$$

```
probabilidad_transicion <- function(i, j) {
  i <- as.numeric(i)
  j <- as.numeric(j)
  if (j == i + 1) {
    (N - i) / N
  } else if (j == i - 1) {
    i / N
  } else {
    0
  }
}
```

→ i,j SON CARACTERES

```
transicion_Ehrenfest <-
  outer(estados_Ehrenfest, estados_Ehrenfest,
        Vectorize(probabilidad_transicion))
dimnames(transicion_Ehrenfest) <- list(estados_Ehrenfest, estados_Ehrenfest)
```

OUTER PRODUCT

TENEMOS QUE APLICAR LA FUNCIÓN PROBABILIDAD_TRANSICION (VECTORIZADA PORQUE ESTADOS_EHRENFEST ES UN VECTOR!) PARA CADA PAR DE ESTADOS!

La cadena de Markov se construye entonces como sigue:

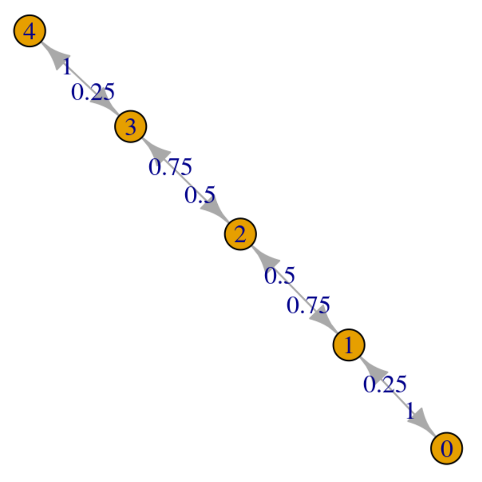
```
library(markovchain)

modelo_Ehrenfest <- new("markovchain",
                        states = estados_Ehrenfest,
                        transitionMatrix = transicion_Ehrenfest,
                        name = "Modelo de Ehrenfest")
modelo_Ehrenfest
```

```
## Modelo de Ehrenfest
##  A  5 - dimensional discrete Markov Chain defined by the following states:
##  0, 1, 2, 3, 4
##  The transition matrix  (by rows)  is defined as follows:
##      0    1    2   3    4
## 0 0.00 1.0 0.00 0.0 0.00
## 1 0.25 0.0 0.75 0.0 0.00
## 2 0.00 0.5 0.00 0.5 0.00
## 3 0.00 0.0 0.75 0.0 0.25
## 4 0.00 0.0 0.00 1.0 0.00
```

y tiene la siguiente representación gráfica:

```
plot(modelo_Ehrenfest, layout = igraph::layout.graphopt)
```



Podemos estudiar el modelo analizando realizaciones aleatorias a partir de distintos estados iniciales.

```
rmarkovchain(10, modelo_Ehrenfest, t0 = "0", include.t0 = TRUE)
```

→ SIMULACION DE LA CADENA DE MARKOV

```
##  [1] "0" "1" "2" "3" "2" "1" "2" "3" "2" "1" "2"
```

```
rmarkovchain(10, modelo_Ehrenfest, t0 = as.character(N), include.t0 = TRUE)
```

```
##  [1] "4" "3" "4" "3" "2" "3" "2" "3" "2" "3" "2"
```

En el modelo de Ehrenfest todos los estados son accesibles entre sí, aunque algunos de ellos no lo son en un único paso.

```
for (i in estados_Ehrenfest) {
  for (j in estados_Ehrenfest) {
    print(sprintf("%s es accesible desde %s: %s", j, i,
                  is.accessible(modelo_Ehrenfest, from = i, to = j)))
  }
}
```

```
## [1] "0 es accesible desde 0: TRUE"
## [1] "1 es accesible desde 0: TRUE"
## [1] "2 es accesible desde 0: TRUE"
## [1] "3 es accesible desde 0: TRUE"
## [1] "4 es accesible desde 0: TRUE"
## [1] "0 es accesible desde 1: TRUE"
## [1] "1 es accesible desde 1: TRUE"
## [1] "2 es accesible desde 1: TRUE"
## [1] "3 es accesible desde 1: TRUE"
## [1] "4 es accesible desde 1: TRUE"
## [1] "0 es accesible desde 2: TRUE"
## [1] "1 es accesible desde 2: TRUE"
## [1] "2 es accesible desde 2: TRUE"
## [1] "3 es accesible desde 2: TRUE"
## [1] "4 es accesible desde 2: TRUE"
## [1] "0 es accesible desde 3: TRUE"
## [1] "1 es accesible desde 3: TRUE"
## [1] "2 es accesible desde 3: TRUE"
## [1] "3 es accesible desde 3: TRUE"
## [1] "4 es accesible desde 3: TRUE"
## [1] "0 es accesible desde 4: TRUE"
## [1] "1 es accesible desde 4: TRUE"
## [1] "2 es accesible desde 4: TRUE"
## [1] "3 es accesible desde 4: TRUE"
## [1] "4 es accesible desde 4: TRUE"
```

Por lo tanto, hay una única clase de estados comunicantes y la cadena de Markov es irreducible.

```
communicatingClasses(modelo_Ehrenfest)
```

```
## [[1]]
## [1] "0" "1" "2" "3" "4"
```

```
is.irreducible(modelo_Ehrenfest)
```

```
## [1] TRUE
```

La cadena de Markov no es aperiódica.

```
period(modelo_Ehrenfest)
```

```
## [1] 2
```