

UNIVERSIDAD NACIONAL DE COLOMBIA SEDE BOGOTA

PROCESOS ESTOCASTICOS

Parical 1-B Implementacion Computacional

Cesar Augusto Prieto Sarmiento - CC: 1065843742

13 de marzo de 2024

```
#install.packages("markovchain")
#install.packages("DiagrammeR")
#install.packages("diagram")
```

```
library(markovchain)
library(diagram)
library(DiagrammeR)
```

Extra: El siguiente trozo de codigo fue tomado del repositorio de GitHub donde esta subido el paquete markovchain, y modificado en algunos aspectos para que este pueda funcionar de manera correcta a la hora de graficar las cadenas descritas en los siguientes ejercicios. Para más ingresar en el link: ??

```
# plot a diagram using DiagrammeR for a markovchain object
.plotDiagrammeR <- function(object, node_fill_color = "lightblue", ...) {
  if(is(object,"markovchain")){
    mat <- object@transitionMatrix
  } else if(is(object,"ctmc")){
    mat <- object@generator
  }
  names <- rownames(mat)

  # names of nodes
  nodes <- ''
  for(i in 1:nrow(mat)) {
    nodes <- paste0(nodes, names[i], ' [fillcolor = "', node_fill_color, '"]; ')
  }

  # store edges
  edges <- ''
  for(i in 1:nrow(mat)) {
    for(j in 1:ncol(mat)) {
      if (mat[i,j] != 0) {
        edges <- paste0(edges, names[i], "->", names[j], " [label = ", mat[i,j], "] ")
      }
    }
  }

  # extract extra parameter
```

```

dots <- list(...)
args <- ""
for(name in names(dots)) {
  args <- paste0(args, name, "=\"", dots[[name]], "\" ")
}

# print(args)
if (requireNamespace("DiagrammeR", quietly = TRUE)) {
  res <- DiagrammeR::grViz(paste0("
digraph circles {
  graph [overlap = true, fontsize = 10]

  node [shape = circle,
fixedsize = true,
width = 0.9] // sets as circles
  ", nodes, "

  ", edges, args,"

  // labelfontsize = 20 labelloc='t' label ='Weather transition matrix'
}
"))

  return (res)
} else {
  print("Diagrammer unavailable")
}
}

```

Punto 1

Implementar los siguientes ejercicios en programación de R u otra programación. Enviar por correo electrónico el miércoles, 13 de marzo antes de las 07:00 a.m. Datos de transición:

$$\begin{pmatrix}
 0 & 0 & 1 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 1 \\
 0 & 0 & 0 & 0 & 1 & 0 \\
 \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 & 0 & 0 \\
 1 & 0 & 0 & 0 & 0 & 0 \\
 0 & \frac{1}{2} & 0 & 0 & 0 & \frac{1}{2}
 \end{pmatrix}$$

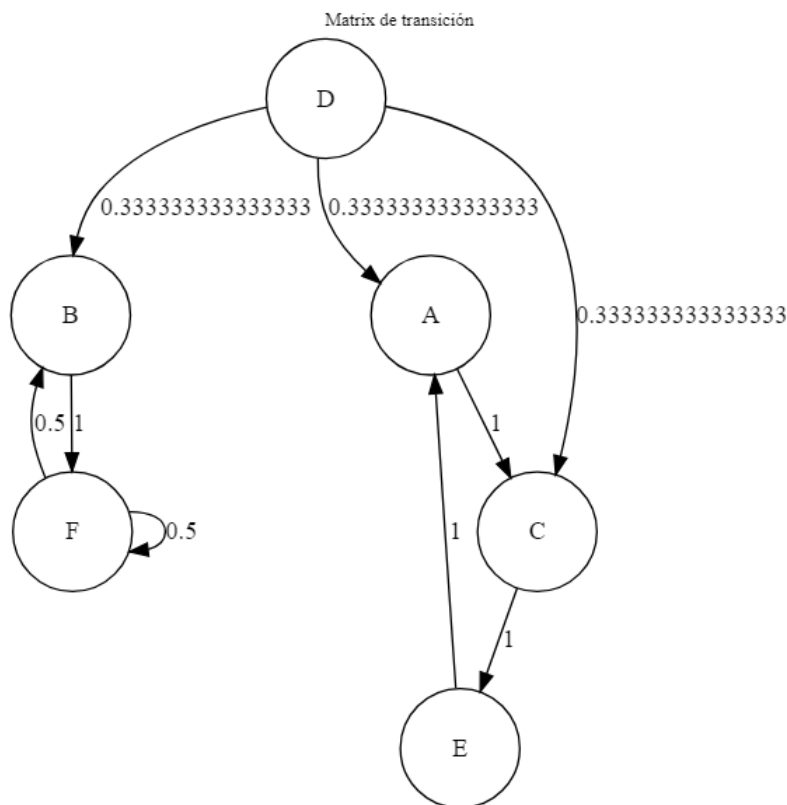
```

# Definir la matriz de transición
P <- matrix(c(0, 0, 1, 0, 0, 0,
              0, 0, 0, 0, 0, 1,
              0, 0, 0, 0, 1, 0,
              1/3, 1/3, 1/3, 0, 0, 0,
              1, 0, 0, 0, 0, 0,
              0, 1/2, 0, 0, 0, 1/2), nrow = 6, byrow = TRUE)

# Crear objeto markovchain
mc <- new("markovchain", states = c("A", "B", "C", "D", "E", "F"), transitionMatrix = P)

```

```
#Visualizacion de la cadena
set.seed(2)
.plotDiagrammeR(mc, node_fill_color = "lightblue", label = "Matrix de transición", label
```



Observemos que los datos quedan almacenados dentro del objeto mc. Utilizando la función summary, podemos explorar su contenido:

```
str(mc);show(mc) #Se explora la clase del objeto que se creo con la funcion anterior
```

Formal class 'markovchain' [package "markovchain"] with 4 slots

```
..@ states      : chr [1:6] "A" "B" "C" "D" ...
..@ byrow       : logi TRUE
..@ transitionMatrix: num [1:6, 1:6] 0 0 0 0.333 1 ...
.. ..- attr(*, "dimnames")=List of 2
.. .. ..$ : chr [1:6] "A" "B" "C" "D" ...
.. .. ..$ : chr [1:6] "A" "B" "C" "D" ...
..@ name        : chr "Unnamed Markov chain"
```

Unnamed Markov chain

A 6 - dimensional discrete Markov Chain defined by the following states:

A, B, C, D, E, F

The transition matrix (by rows) is defined as follows:

	A	B	C	D	E	F
A	0.0000000	0.0000000	1.0000000	0	0	0.0
B	0.0000000	0.0000000	0.0000000	0	0	1.0
C	0.0000000	0.0000000	0.0000000	0	1	0.0
D	0.3333333	0.3333333	0.3333333	0	0	0.0

```
E 1.0000000 0.0000000 0.0000000 0 0 0.0
F 0.0000000 0.5000000 0.0000000 0 0 0.5
```

```
summary(mc) # Se explora el contenido del objeto mc
```

```
Unnamed Markov chain Markov chain that is composed by:
```

```
Closed classes:
```

```
A C E
```

```
B F
```

```
Recurrent classes:
```

```
{A,C,E},{B,F}
```

```
Transient classes:
```

```
{D}
```

```
The Markov chain is not irreducible
```

```
The absorbing states are: NONE
```

De esta salida, podemos observar que la cadena de Markov es irreducible y no tiene estados absorbentes, lo cual será útil para responder algunas preguntas más adelante.

a Clasifica los estados en clases.

Solucion:

```
# 1. Clasificar los estados en clases comunicantes
communicating_classes <- communicatingClasses(mc)
print("Clases de estados comunicantes:")
print(communicating_classes)
```

```
[[1]]
[1] "A" "C" "E"
```

```
[[2]]
[1] "B" "F"
```

```
[[3]]
[1] "D"
```

Existen 3 estados los cuales son: $\{A, C, E\}$, $\{B, F\}$ y $\{D\}$

b Estudia la recurrencia o transitoriedad de cada uno de los estados de la cadena.

Solución:

```
# 2. Determinar la recurrencia de cada estado
recurrent_states <- recurrentStates(mc)
print("Estados recurrentes:")
print(recurrent_states)
```

```
[1] "A" "B" "C" "E" "F"
```

De la salida del código podemos decir que las clases recurrentes son $\{A, C, E\}$, $\{B, F\}$ y por otra parte la clase D es transitoria.

- c Calcular el periodo de cada una de las clases recurrentes.

Solución:

```
# 3. Calcular el período de cada clase recurrente
C1 <- new("markovchain", transitionMatrix = P[c(1,3,5), c(1,3,5)])
C2 <- new("markovchain", transitionMatrix = P[c(2,6), c(2,6)])

cbind(period(C1), period(C2))

      [,1] [,2]
[1,]    3    1
```

De la salida del código podemos concluir que para las clases 1 y 2 los periodos son 3 y 1 respectivamente.

- d Identificar los estados ergódicos.

Solución:

Como el periodo de $C2 = 1$ entonces podemos decir que es aperiódico, por lo tanto es un estado ergódico.

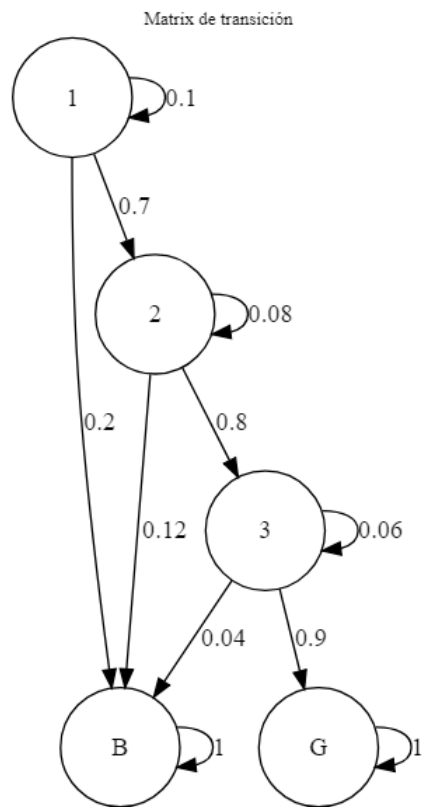
Punto 2

En la oficina de Admisiones de la Universidad Nacional se ha obtenido la información necesaria para las siguientes estadísticas sobre un programa de Maestría que dura tres niveles: el 70 % de los estudiantes que ingresan al primer nivel pasan con éxito al segundo nivel, el 10 % lo repite y el 20 % restante se retira por diferentes motivos; de todos los estudiantes que pasan al segundo nivel, el 80 % accede al tercer nivel, el 8 % repite y el 12 % restante sale del programa por bajo nivel académico u otras razones; de todos los estudiantes que ingresan al tercer nivel, el 90 % se gradúa, el 6 % lo repite y el 4 % restante no puede optar al título y son retirados por no cumplir las normas estipuladas.

```
P <- matrix(c(1,0,0,0,0,
             0.2,0.1,0.7,0,0,
             0.12,0,0.08,0.8,0,
             0.04,0,0,0.06,0.9,
             0,0,0,0,1), nrow = 5, byrow = TRUE)

mc <- new("markovchain", transitionMatrix = P,
         states = c("B","1","2","3","G"))
#Donde B y G significan Baja y Grado

set.seed(2)
plotDiagrammeR(mc, node_fill_color = "lightblue", label = "Matrix de transición", label
```



Ahora, definamos la matriz canonica y hagamos un par de operaciones extra para poder responder a las siguientes preguntas.

```

CanoniMa <- matrix(c(1.00,0.00,0.00,0.00,0.00,
                     0.00,1.00,0.00,0.00,0.00,
                     0.20,0.00,0.10,0.70,0.00,
                     0.12,0.00,0.00,0.08,0.80,
                     0.04,0.90,0.00,0.00,0.06),
                    nrow = 5, byrow = TRUE,
                    dimnames = list(c("B","G","1","2","3"),
                                    c("B","G","1","2","3")))

M <- solve(diag(1,3,3) - CanoniMa[c(3,4,5), c(3,4,5)]); M

      1      2      3
1 1.111111 0.8454106 0.7194984
2 0.000000 1.0869565 0.9250694
3 0.000000 0.0000000 1.0638298

```

- a) ¿Cuántos alumnos lograrán el título de Maestría de un grupo de 100 aspirantes que se matricularon en el primer nivel?

textbfSolucion:

```

#Probabilidades de Absorcion
Pr <- absorptionProbabilities(mc);Pr

```

	B	G
1	0.35245143	0.6475486
2	0.16743756	0.8325624
3	0.04255319	0.9574468

Dado la respuesta obtenida por la funcion implementada, podemos concluir que de un grupo de 100 aspirantes que se matricularon en primer nivel solo el 64 % de ellos lograran el titulo de la maestria en este caso 64 estudiantes.

- b) Si cada nivel dura un semestre, ¿durante cuántos años se deberá ofrecer esta Maestría si el país necesita, aproximadamente, 500 especialistas en esta área, sabiendo que esta universidad solo está en capacidad de recibir, como máximo, 50 alumnos nuevos cada semestre?

Solucion:

```
n <- ceiling(500/Pr[1,2]);n
Años <- ceiling(n/50/2);Años

[1] 8
```

Así mismo, tambien podemos concluir que para lograr la meta de 500 especialistas en el area, se necesitara un total de 8 años de oferta de la maestria.

Punto 3

Considera una caminata aleatoria con los estados $S = \{-2, -1, 0, 1, 2\}$. Si el proceso está en estado i ($i = -1, 0, 1$), entonces se mueve a cualquier $i - 1$ o $i + 1$ con igual probabilidad. Si el proceso se encuentra en estado -2 o 2 en el tiempo n , entonces se mueve al estado $-1, 0$ o 1 en el tiempo $n + 1$ con probabilidad igual.

- a) Escribe la matriz de probabilidad de transición para esta caminata aleatoria.

Solucion:

```
P <- matrix(c(0,1/3,1/3,1/3,0,
              0.5,0,0.5,0,0,
              0,0.5,0,0.5,0,
              0,0,0.5,0,0.5,
              0,1/3,1/3,1/3,0), byrow = TRUE, ncol = 5)

#Visualizacion de la cadena
set.seed(2)
.plotDiagrammeR(mc, node_fill_color = "lightblue", label = "Matrix de transición",

mc <- new("markovchain", transitionMatrix = P, states = c("-2","-1","0","1","2"));m
```

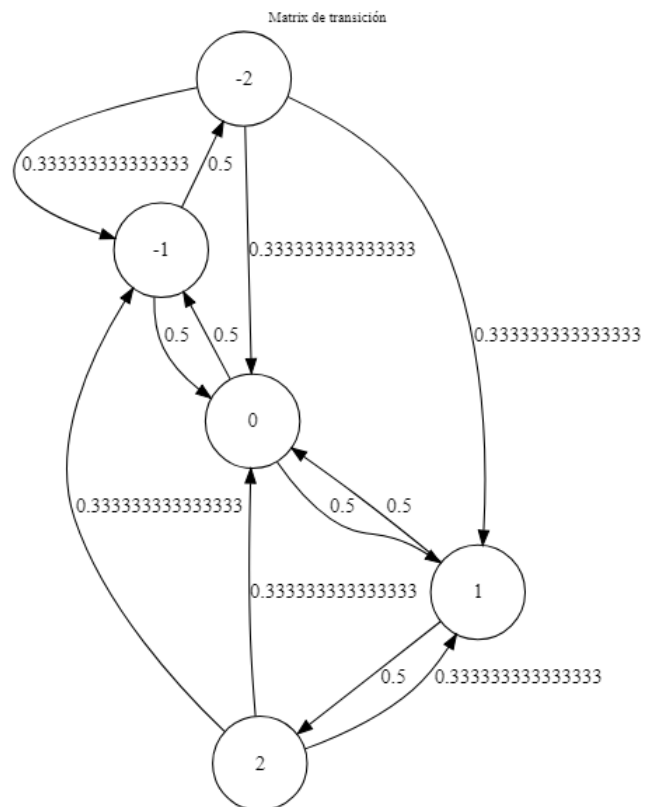
Unnamed Markov chain

A 5 - dimensional discrete Markov Chain defined by the following states:

```

-2, -1, 0, 1, 2
The transition matrix (by rows) is defined as follows:
  -2      -1      0      1      2
-2 0.0 0.3333333 0.3333333 0.3333333 0.0
-1 0.5 0.0000000 0.5000000 0.0000000 0.0
0  0.0 0.5000000 0.0000000 0.5000000 0.0
1  0.0 0.0000000 0.5000000 0.0000000 0.5
2  0.0 0.3333333 0.3333333 0.3333333 0.0

```



La matrix que se pide en el inciso a es la siguiente:

	-2	-1	0	1	2
-2	0,0	1/3	1/3	1/3	0,0
-1	0,5	0,0	0,5	0,0	0,0
0	0,0	0,5	0,0	0,5	0,0
1	0,0	0,0	0,5	0,0	0,5
2	0,0	1/3	1/3	1/3	0,0

b) Calcula la distribución estacionaria.

Solucion: Al calcular la distribucion estacionaria obtenemos lo siguiente:

```
steadyStates(mc)
```

```

-2      -1      0      1      2

```



```
[1,] 0.1153846 0.2307692 0.3076923 0.2307692 0.1153846
```

Punto 4

Un jugador llega a un casino con 200 dólares. Decide jugar hasta que su capital sea igual a 700 dólares o hasta que se arruine. En cada ronda de juego, el jugador gana o pierde 100 dólares con probabilidades 0.4 y 0.6 respectivamente. ¿Cuál es la probabilidad de que el jugador logre alcanzar su meta de los 700 dólares? ¿Cuál es la probabilidad de ruina del jugador?

Solucion:

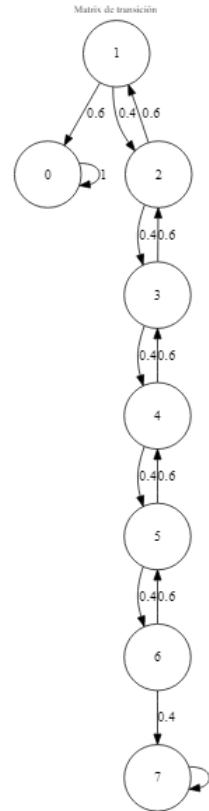
```
P <- matrix(c(1,0,0,0,0,0,0,0,
              0.6,0,0.4,0,0,0,0,0,
              0,0.6,0,0.4,0,0,0,0,
              0,0,0.6,0,0.4,0,0,0,
              0,0,0,0.6,0,0.4,0,0,
              0,0,0,0,0.6,0,0.4,0,
              0,0,0,0,0,0.6,0,0.4,
              0,0,0,0,0,0,0.6,0,0.4,
              0,0,0,0,0,0,0,1), nrow = 8, byrow = TRUE)

mc <- new("markovchain", transitionMatrix = P,
          states = c("0","1","2","3","4","5","6","7"))

#Visualizacion de la cadena
set.seed(2)
.plotDiagrammeR(mc, node_fill_color = "lightblue", label = "Matrix de transición", label)

summary(mc)

Unnamed Markov chain  Markov chain that is composed by:
Closed classes:
0
7
Recurrent classes:
{0},{7}
Transient classes:
{1,2,3,4,5,6}
The Markov chain is not irreducible
The absorbing states are: 0 7
```



```
M <- absorptionProbabilities(mc); M
```

	0	7
1	0.9689169	0.03108305
2	0.9222924	0.07770763
3	0.8523555	0.14764449
4	0.7474502	0.25254978
5	0.5900923	0.40990772
6	0.3540554	0.64594463

Entonces, la probabilidad de que el jugador alcance su meta es de 0.077, es decir el 7.7% y la probabilidad de que alcance la ruina es de 0.922, es decir, del 92.2 %.