PRÁCTICA 1: ANÁLISIS Y SIMULACIÓN DE CADENAS DE MARKOV

PROCESOS ESTOCÁSTICOS Y SERIES TEMPORALES

GRADO EN CIENCIA E INGENIERÍA DE DATOS

Sumario: En esta práctica, aprenderemos a analizar y manipular cadenas de Markov utilizando *R*. Emplearemos métodos para identificar clases irreducibles y clasificar los estados. Además, veremos cómo calcular probabilidades de transición de n pasos, probabilidades y tiempos medios de absorción, y distribuciones estacionarias. También aprenderemos a simular trayectorias de la cadena de Markov. Finalmente, exploraremos ejemplos de aplicación de las cadenas de Markov, aplicando los métodos discutidos.

1. Creación y representación de objetos tipo cadena de Markov

Usaremos el paquete *markovchain* de *R*, el cual es usado para el análisis y simulación de cadenas de Markov. Procederemos a instalar y cargar dicho paquete:

```
library("markovchain")
```

```
## Package:  markovchain
## Version:  0.9.5
## Date:     2023-09-24 09:20:02 UTC
## BugReport: https://github.com/spedygiorgio/markovchain/issues
```

Una vez cargado el paquete, podemos crear objetos de tipo *markovchain*. Estos objetos, como su nombre indica, son cadenas de Markov. Para definir una cadena de Markov deberemos especificar el espacio de estados y la matriz de transición entre los estados.

Por ejemplo, consideremos el problema de la ruina del jugador, donde apuesto un euro a cara en sucesivos lanzamientos de moneda. Paro de jugar si me arruino o si llego a la riqueza objetivo de 4 euros. Dicha cadena tiene espacio de estados {0, 1, 2, 3, 4} correspondientes a las diferentes cantidades de dinero que puedo obtener, y matriz de transición

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 \\ 0 & 1/2 & 0 & 1/2 & 0 \\ 0 & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Creamos el objeto de tipo cadena de Markov mediante el código:

```
estados <- c("0", "1", "2", "3", "4")
P <- rbind(c(1, 0, 0, 0, 0),
           c(1/2, 0, 1/2, 0, 0),
           c(0, 1/2, 0, 1/2, 0),
           c(0, 0, 1/2, 0, 1/2),
           c(0, 0, 0, 0, 1))
mc <- new("markovchain", states = estados, transitionMatrix = P, name = "Ruina del jugador")
```

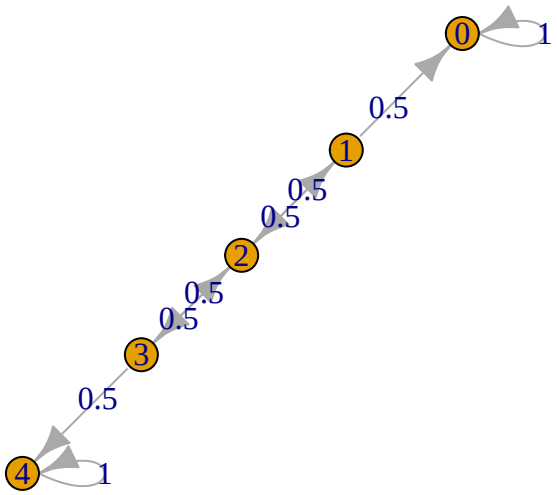
Para visualizar la cadena de Markov podemos hacer:

```
print(mc)
```

```
##     0   1   2   3   4
## 0 1.0 0.0 0.0 0.0 0.0
## 1 0.5 0.0 0.5 0.0 0.0
## 2 0.0 0.5 0.0 0.5 0.0
## 3 0.0 0.0 0.5 0.0 0.5
## 4 0.0 0.0 0.0 0.0 1.0
```

Podemos representar gráficamente la cadena mediante un grafo usando:

```
set.seed(1234)
plot(mc)
```



La disposición de los estados en el grafo es establecida al azar. Para fijar dicha disposición, hemos fijados la semilla de los números aleatorios.

Mediante la función *summary()* podemos hacer un análisis general de la cadena de Markov, identificando las clases irreducibles y clasificando los estados:

```
summary(mc)
```

```
## Ruina del jugador  Markov chain that is composed by:
## Closed classes:
## 0
## 4
## Recurrent classes:
## {0},{4}
## Transient classes:
```

```
## {1,2,3}
## The Markov chain is not irreducible
## The absorbing states are: 0 4
```

Observamos que se obtienen tres clases irreducibles $A := \{0\}$, $B := \{4\}$, y $C := \{1, 2, 3\}$. Las clases A y B tienen un único elemento absorbente correspondientes a la ruina "0" y la riqueza objetivo "4", respectivamente. La clase C tiene elementos transitorios. También nos informa de que las clases A y B son cerradas. Una clase es cerrada cuando, una vez que la cadena alcanza algún estado de dicha clase ya no puede salir de ella.

2. Probabilidades de transición

Dado un estado, podemos seleccionar las probabilidades de transición desde dicho estado a los demás. Por ejemplo, si poseemos una riqueza de 3, podemos calcular las probabilidades para diferentes riquezas en la siguiente jugada:

```
conditionalDistribution(mc, "3")
```

```
##   0   1   2   3   4
## 0.0 0.0 0.5 0.0 0.5
```

Nos dice que en la siguiente jugada tendremos una riqueza de 2 con probabilidad 0.5, o una riqueza de 4 con probabilidad de 0.5.

Imaginemos que queremos conocer las probabilidades de las diferentes riquezas en dos jugadas. Para ello podemos usar la operación producto de cadenas de Markov:

```
conditionalDistribution(mc*mc, "3")
```

```
##    0    1    2    3    4
## 0.00 0.25 0.00 0.25 0.50
```

En este caso, vemos que, en dos jugadas, tendremos una riqueza de 1 con probabilidad 0.25, una riqueza de 3 con probabilidad 0.25, o una riqueza de 4 con probabilidad 0.5.

Recordemos que el producto matricial en R se obtiene usando el operardor %*%. Por ejemplo, para obtener la matriz de transición en dos pasos, que sabemos que viene dada por P^2, haremos:

```
P %*% P
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,] 1.00 0.00  0.0 0.00 0.00
## [2,] 0.50 0.25  0.0 0.25 0.00
## [3,] 0.25 0.00  0.5 0.00 0.25
## [4,] 0.00 0.25  0.0 0.25 0.50
## [5,] 0.00 0.00  0.0 0.00 1.00
```

Y mirando la cuarta fila de la matriz anterior (correspondiente al estado 3), obtenemos la probabilidad de transición en dos pasos (dos jugadas) cuando partimos de una riqueda de 3.

Si queremos conocer las probabilidades para las diferentes riquezas dentro de cinco jugadas, podemos usar la operación potencia:

```
conditionalDistribution(mc^5, "3")
```

```
##      0      1      2      3      4
## 0.1875 0.0000 0.1250 0.0000 0.6875
```

En este caso, vemos que, en cinco jugadas, tendremos una riqueza de 0 con probabilidad 0.1875, una riqueza de 2 con probabilidad 0.1250, o una riqueza de 4 con probabilidad 0.6875.

Y la matriz de transición en 5 pasos es P^5:

```
library(expm)
```

```
## Cargando paquete requerido: Matrix
```

```
##
## Adjuntando el paquete: 'expm'
```

```
## The following object is masked from 'package:Matrix':
##
##     expm
```

```
P %^% 5
```

```
##        [,1]  [,2]  [,3]  [,4]   [,5]
## [1,] 1.0000 0.000 0.000 0.000 0.0000
## [2,] 0.6875 0.000 0.125 0.000 0.1875
## [3,] 0.3750 0.125 0.000 0.125 0.3750
## [4,] 0.1875 0.000 0.125 0.000 0.6875
## [5,] 0.0000 0.000 0.000 0.000 1.0000
```

Para obtener dichas probabilidades dentro de 20 jugadas hacemos:

```
conditionalDistribution(mc^20, "3")
```

```
##            0            1            2            3            4
## 0.2495117188 0.0004882812 0.0000000000 0.0004882812 0.7495117188
```

3. Probabilidad de absorción y tiempo medio de absorción

También podemos calcular la probabilidad de absorción por estados absorbentes. Esto lo haremos con la función *absorptionProbabilities*, que en realidad sirve para obtener la probabilidad de absorción por estados recurrentes:

```
absorptionProbabilities(mc)
```

```
##      0    4
## 1 0.75 0.25
## 2 0.50 0.50
## 3 0.25 0.75
```

Lo cual nos da las distintas probabilidades de absorción por los estados absorbentes 0 y 4, empezando en cada uno de los estados transitorios 1, 2 y 3. Por ejemplo, si partimos de una riqueza de 3 euros tendremos una probabilidad de 1/4 de arruinarnos y una probabilidad de 3/4 de alcanzar la riqueza objetivo.

También podemos calcular el tiempo medio de absorción. En clase vimos cómo calcular el tiempo medio de absorción para cadenas con un solo punto absorbente. En nuestro caso tenemos dos estados absorbentes. En realidad, podemos aplicar el mismo método para obtener la probabilidad de ser absorbido por cualquiera de los dos estados sin distinción entre ellos, considerando el conjunto de los estados absorbentes como un único estado absorbente. En nuestro caso podemos hacerlo mediante el siguiente código:

```
meanAbsorptionTime(mc)
```

```
## 1 2 3
## 3 4 3
```

Para los estados transitorios 1, 2, y 3 nos da los tiempos medios de absorción 3, 4, y 3, respectivamente. Esto quiere decir que si, por ejemplo, tenemos una riqueza de 2 euros de media tardaremos 4 lanzamientos de moneda hasta que acabe el juego, bien por arruinarnos o bien por haber alcanzado la riqueza objetivo.

4. Simulación

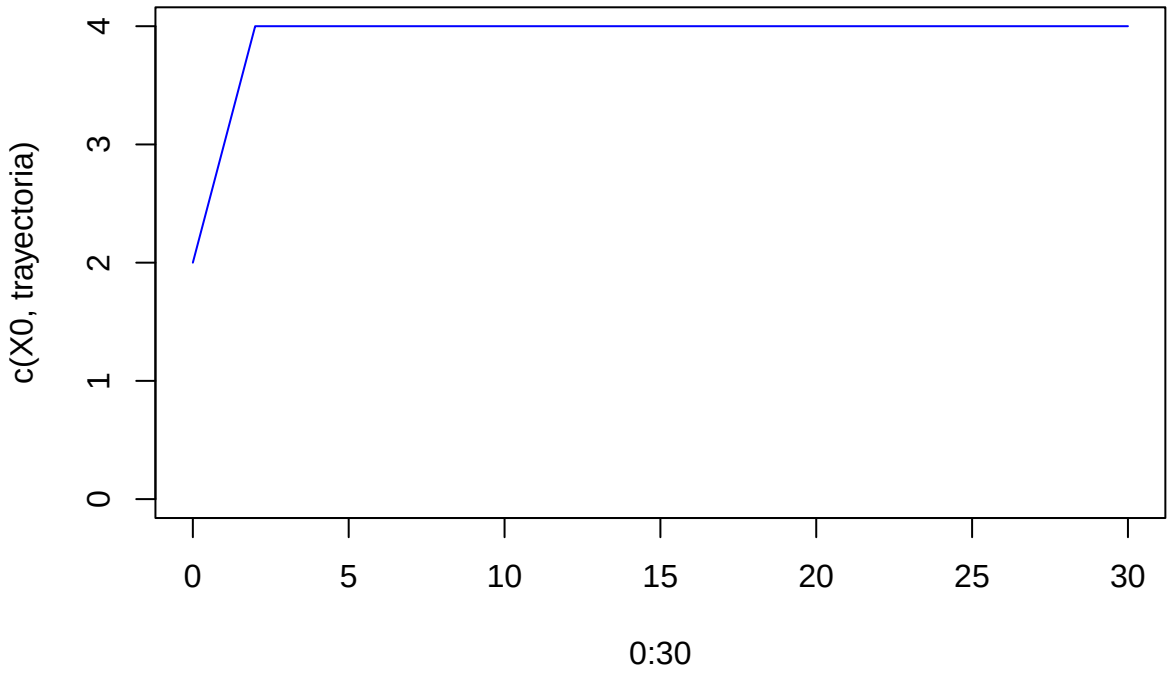
También podemos simular trayectorias de cadenas de Markov fácilmente. Esto lo haremos mediante la función *rmarkovchain*. El siguiente código simula 30 jugadas consecutivas suponiendo que partimos de una riqueza inicial de 2 euros:

```
set.seed(111)
X0 <- "2"
trayectoria <- rmarkovchain(n = 30, object = mc, t0 = X0)
```

Hemos especificado el estado del que partimos. Si no le diésemos ningún valor al parámetro t0, la cadena empezaría en cualquier estado al azar.

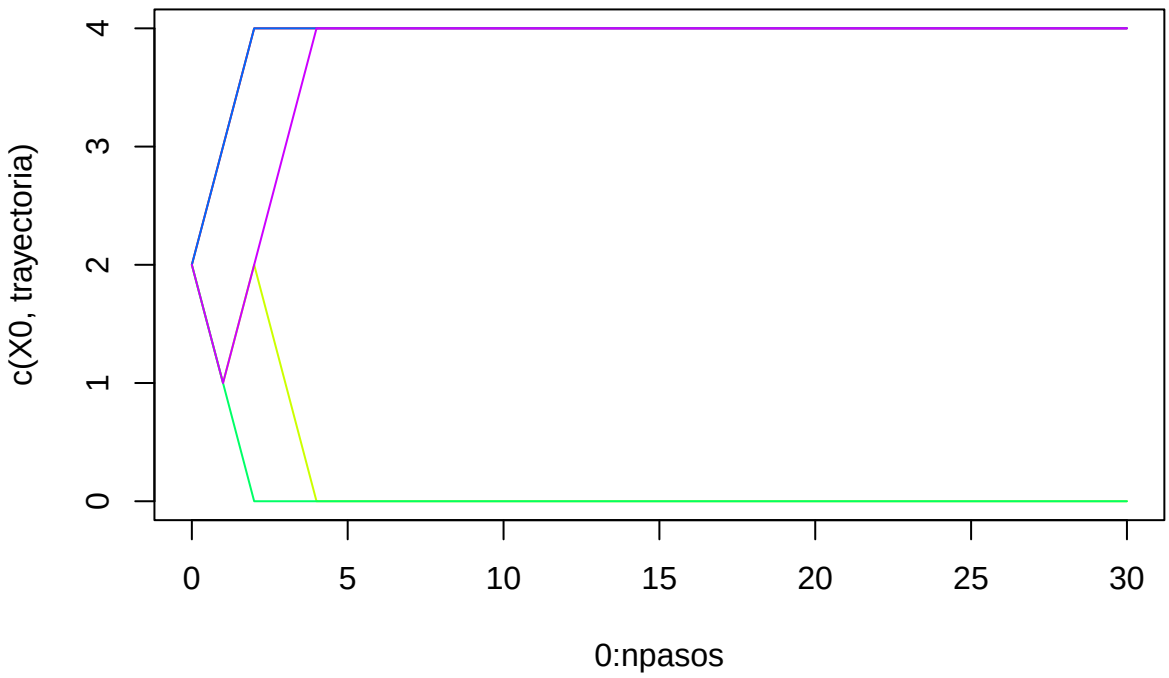
Podemos dibujar la trayectoria obtenida:

```
plot(0:30, c(X0, trayectoria), type = "l", lty = 1, col="blue", ylim=c(0,4))
```



Podemos simular varias trayectorias:

```
set.seed(111)
npasos <- 30
trayectoria <- rmarkovchain(n = npasos, object = mc, t0=X0)
colores <- rainbow(5)
plot(0:npasos, c(X0, trayectoria), type = "l", lty = 1, col=colores[1], ylim=c(0, 4))
for(i in 2:5){
  trayectoria <- rmarkovchain(n = npasos, object = mc, t0=X0)
  lines(0:npasos, c(X0, trayectoria), col=colores[i])
}
```



Vemos que unas trayectorias conducen a la riqueza objetivo mientras que otras, acaban en la ruina.

5. Distribuciones estacionarias y límite

En este apartado vamos a considerar un ejemplo distinto. Consideremos la cadena de Markov describiendo los cambios en el tiempo atmosférico de un día para otro en cierta región.

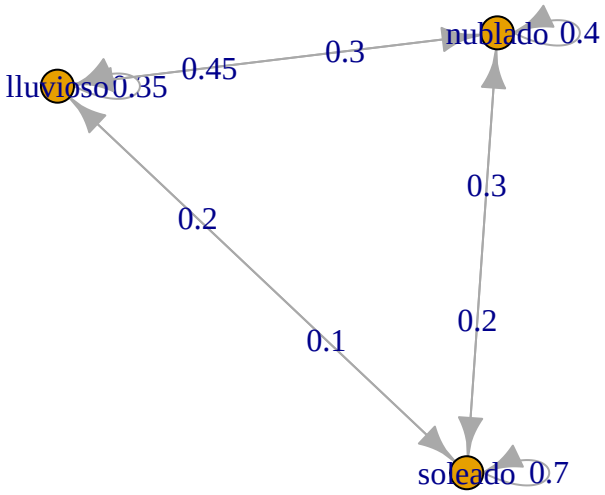
	Soleado	**Nublado**	**Lluvioso**
Soleado	0.7	0.2	0.1
Nublado	0.3	0.4	0.3
Lluvioso	0.2	0.45	0.35

Definimos en *R* la cadena de Markov correspondiente:

```
estados <- c("soleado", "nublado", "lluvioso")
P <- rbind(c(0.7, 0.2, 0.1),
           c(0.3, 0.4, 0.3),
           c(0.2, 0.45, 0.35))
mc.tiempo <- new("markovchain", states = estados, transitionMatrix = P, name = "Tiempo atmosférico")
```

Podemos dibujar el grafo de la cadena:

```
set.seed(123)
plot(mc.tiempo)
```



Queremos saber con qué probabilidad (o con qué frecuencia) se da cada uno de los tiempos atmosféricos a largo plazo. Para ello vamos a calcular la distribución límite.

Vimos en clase que la distribución límite existe si la cadena es irreducible y aperiódica. Podemos comprobar estas condiciones fácilmente en *R*. Comprobamos que es irreducible mediante:

```
is.irreducible(mc.tiempo)
```

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

Vemos que, en efecto, es irreducible.

Para ver si es aperiódica, calculamos el periodo mediante:

```
period(mc.tiempo)
```

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

Al ser el periodo 1, tenemos que es aperiódica, por lo que la distribución límite $\pi_j = \lim_{n\to\infty} \mathbb{P}(X_n = j)$ existe.

Vimos que la distribución límite es una disttribución estacionaria. Para calcular dicha distribución estacionaria hacemos:

```
steadyStates(mc.tiempo)
```

```
##        soleado   nublado  lluvioso
## [1,] 0.4636364 0.3181818 0.2181818
```

Concluimos que, en el largo plazo, el 46.36% de los días serán soleados, el 31.81% de los días estará nublado, y el 21,82% de los días será lluviosos.

Podemos comprobar que la distribución límite coincide con las filas de la matriz de transición en n pasos, cuando n tiende a infinito. Calculemos algunas potencias de la matriz de transición y veamos cómo las filas se estabilizan en la distribución estacionaria (y distribución límite):

```
P %^% 3
```

```
##          [,1]     [,2]     [,3]
## [1,] 0.51150 0.294250 0.194250
## [2,] 0.43050 0.334750 0.234750
## [3,] 0.41025 0.344875 0.244875
```

```
P %^% 5
```

```
##           [,1]      [,2]      [,3]
## [1,] 0.4733287 0.3133356 0.2133356
## [2,] 0.4569262 0.3215369 0.2215369
## [3,] 0.4528256 0.3235872 0.2235872
```

```
P %^% 10
```

```
##           [,1]      [,2]      [,3]
## [1,] 0.4638152 0.3180924 0.2180924
## [2,] 0.4635125 0.3182437 0.2182437
## [3,] 0.4634369 0.3182816 0.2182816
```

```
P %^% 30
```

```
##           [,1]      [,2]      [,3]
## [1,] 0.4636364 0.3181818 0.2181818
## [2,] 0.4636364 0.3181818 0.2181818
## [3,] 0.4636364 0.3181818 0.2181818
```

```
P %^% 50
```

```
##           [,1]      [,2]      [,3]
## [1,] 0.4636364 0.3181818 0.2181818
## [2,] 0.4636364 0.3181818 0.2181818
## [3,] 0.4636364 0.3181818 0.2181818
```

Simulemos el tiempo atmosférico durante 1000 días seguidos:

```
set.seed(1111)
trayectoria <- rmarkovchain(n = 1000, object = mc.tiempo)
```

Veamos los 30 primeros pasos:

```
trayectoria[1:30]
```

```
##  [1] "lluvioso" "nublado"  "lluvioso" "soleado"  "nublado"  "nublado"
##  [7] "soleado"  "soleado"  "soleado"  "nublado"  "lluvioso" "lluvioso"
## [13] "soleado"  "soleado"  "lluvioso" "lluvioso" "soleado"  "nublado"
## [19] "nublado"  "soleado"  "nublado"  "lluvioso" "nublado"  "soleado"
## [25] "soleado"  "soleado"  "soleado"  "soleado"  "nublado"  "lluvioso"
```

Para ver con qué número de días se da cada tipo de tiempo atmosférico, podemos hacer una tabla de frecuencias absolutas:

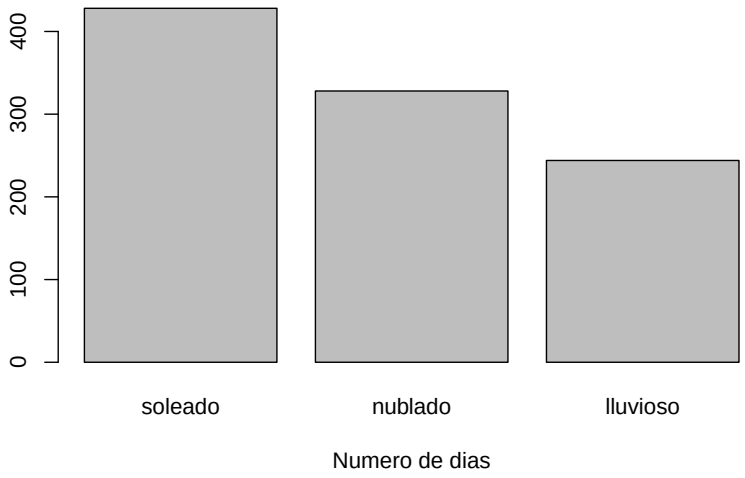
```
tabla <- table(trayectoria)
tabla <- tabla[estados]
tabla
```

```
## trayectoria
```

```
##  soleado  nublado lluvioso
##      428      328      244
```

Podemos hacer un gráfico de barras, para visualizar las frecuencias anteriores:

```
barplot(tabla,  xlab="Numero de dias")
```



Calculemos también las frecuencias relativas:

```
tabla / 1000
```

```
## trayectoria
##  soleado  nublado lluvioso
##    0.428    0.328    0.244
```

Vemos que las frecuencias relativas son parecidas a los valores de la distribución estacionaria:

```
steadyStates(mc.tiempo)
```

```
##        soleado   nublado  lluvioso
## [1,] 0.4636364 0.3181818 0.2181818
```

Si aumentamos el número de pasos a 10.000, la aproximación será mejor.

```
set.seed(123)
trayectoria <- rmarkovchain(n = 10000, object = mc.tiempo)
table(trayectoria)[estados] / 10000
```

```
## trayectoria
##  soleado  nublado lluvioso
##   0.4599   0.3244   0.2157
```

6. Ejemplos de aplicación

Veamos dos ejemplos de aplicación de las cadenas de Markov.

6.1. Cálculo de PageRank de páginas de internet

Recordemos que en la cadena de Markov del algoritmo pageRank de Google, un "surfeador de internet" cliqueaba al azar los enlaces en un conjunto de páginas. Vamos a considerar por ejemplo que tenemos el conjunto de páginas A, B, C, D, E donde:

- A tiene enlaces a C, D, y E.
- B tiene enlace a E.
- C tiene enlace a B y D.
- D itene enalce a B.
- E tiene B y C.

De esta manera, tenemos la matriz de transición:

$$P = \begin{bmatrix} 0 & 0 & 1/3 & 1/3 & 1/3 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1/2 & 0 & 1/2 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1/2 & 1/2 & 0 & 0 \end{bmatrix}$$

Creamos la correspondiente cadena de Markov en R

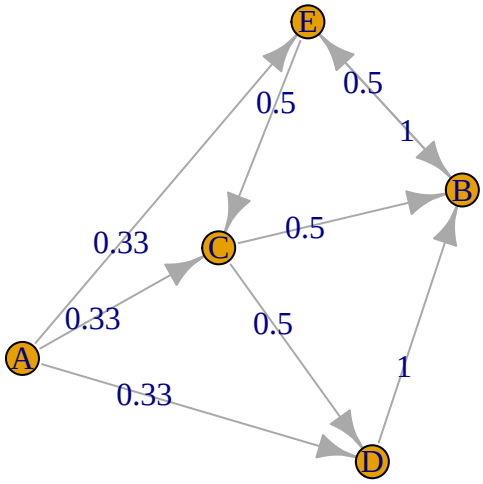
```
# Definimos los estadosP
estados <- c("A","B", "C", "D", "E")

# Definimos la matriz de transición
P <- rbind(c(0, 0, 1/3, 1/3, 1/3),
           c(0, 0, 0, 0, 1),
           c(0, 1/2, 0, 1/2, 0),
           c(0, 1, 0, 0, 0),
           c(0, 1/2, 1/2, 0, 0))

# Definimos el objeto de tipo cadena de Markov
mc.pageRank <- new("markovchain", states = estados, transitionMatrix = P, name = "PageRank")
```

Para ver qué aspecto tiene la cadena, dibujaremos el grafo. Esto se puede hacer simplemente aplicando *plot*. La disposición del gráfico es aleatoria, por lo que fijamos la semilla para quedarnos con una disposición particular.

```
set.seed(122)
plot(mc.pageRank)
```



Para conocer las clases irreducibles y clasificar los estados en recurrentes, transitorios y absorbentes usaremos la función *summary*:

```
summary(mc.pageRank)
```

```
## PageRank  Markov chain that is composed by:
## Closed classes:
## B C D E
## Recurrent classes:
## {B,C,D,E}
## Transient classes:
## {A}
## The Markov chain is not irreducible
## The absorbing states are: NONE
```

Esta cadena contiene dos clases irreducibles: $C_1 = \{B, C, D, E\}$ formada por estados recurrentes y $C_2 = \{A\}$ formada por el estado transitorio A. En este contexto, podemos calcular la probabilidad de que el estado transitorio alcance cada estado recurrente (y en cierto modo quede absorbido por éste) y el tiempo medio (número medio de pasos) para que el estado transitorio sea absorbido:

```
absorptionProbabilities(mc.pageRank)
```

```
##   B         C         D         E
## A 0 0.3333333 0.3333333 0.3333333
```

```
meanAbsorptionTime(mc.pageRank)
```

```
## A
## 1
```

Vemos que la cadena no es irreducible, por lo que la distribución límite podría no existir. Para conseguir una cadena irreducible vamos a considerar un factor de amortiguación d=0.85. Por lo que modificamos la cadena de Markov con el siguiente código:

```
d <- 0.85
n <- 5
M <- d*P + (1-d)*(1/n)*matrix(1, n, n)
# Redefinimos la cadena teniendo en cuenta la nueva matriz de transición
mc.pageRank <- new("markovchain", states = estados, transitionMatrix = M, name = "Pagerank")
```

Sabemos que dicha cadena es irredicible y aperiódica. No obstante, podemos comprobarlo:

```
is.irreducible(mc.pageRank)
```

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

```
period(mc.pageRank)
```

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

Al ser irreducible y aperiódica, la distribución límite existe.

Finalmente, calculamos dicha distribución límite la cual nos da el pageRank:

```
steadyStates(mc.pageRank)
```

```
##          A        B         C         D         E
## [1,] 0.03 0.344565 0.1793366 0.1147181 0.3313803
```

De esta manera, obtenemos los pageRanks:

$$\pi_A = 0.03, \quad \pi_B = 0.3446, \quad \pi_C = 0.1793, \quad \pi_D = 0.1147, \quad \pi_E = 0.3313.$$

Ordenado de mayor a menor, tenemos que el buscador de Google mostrará las páginas en el orden B, E, C, D, A.

6.2. Finanzas: riesgo de crédito

Los **ratings de crédito** son evaluaciones que indican la solvencia o capacidad de una entidad para cumplir con sus obligaciones financieras. Estas evaluaciones son emitidas por agencias de calificación crediticia y son cruciales tanto para inversores como para prestatarios. Una de las escalas de calificación empleadas es:

Rating	**Descripción**
AAA	Riesgo mínimo
AA	Riesgo muy bajo
A	Riesgo bajo
BBB	Riesgo moderado-bajo
BB	Riesgo moderado
B	Riesgo moderado-alto
CCC	Riesgo alto
D	Quiebra

Para modelar cómo evoluciona cada año la solvencia de las entidades financieras se usan cadenas de Markov.

```
rc <- c("AAA", "AA", "A", "BBB", "BB", "B", "CCC", "D")
P <- rbind(c(90.81, 8.33, 0.68, 0.06, 0.08, 0.02, 0.01, 0.01),
           c(0.70, 90.65, 7.79, 0.64, 0.06, 0.13, 0.02, 0.01),
           c(0.09, 2.27, 91.05, 5.52, 0.74, 0.26, 0.01, 0.06),
           c(0.02, 0.33, 5.95, 85.93, 5.30, 1.17, 1.12, 0.18),
```

```
           c(0.03, 0.14, 0.67, 7.73, 80.53, 8.84, 1.00, 1.06),
           c(0.01, 0.11, 0.24, 0.43, 6.48, 83.46, 4.07, 5.20),
           c(0.21, 0, 0.22, 1.30, 2.38, 11.24, 64.86, 19.79),
           c(0, 0, 0, 0, 0, 0, 0, 100))/100

ratings <- new("markovchain", states = rc, transitionMatrix = P, name = "Rating")
```

Hagamos un análisis previo mediante *summary*:

```
summary(ratings)
```

```
## Rating  Markov chain that is composed by:
## Closed classes:
## D
## Recurrent classes:
## {D}
## Transient classes:
## {AAA,AA,A,BBB,BB,B,CCC}
## The Markov chain is not irreducible
## The absorbing states are: D
```

Vemos que se trata de una cadena con dos clases irreducibles: una de estados transitorios y otra con un estado absorbente. Dicho estado absorbente es la quiebra de la entidad.

Supongamos que una compañia tiene rating BBB. Entonces, a partir de dicha cadena podemos hacer diversos análisis.

Por ejemplo, las diferentes probabilidades para los diferentes ratings dentro de un año son:

```
conditionalDistribution(ratings, "BBB")
```

```
##    AAA     AA      A    BBB     BB      B    CCC      D
## 0.0002 0.0033 0.0595 0.8593 0.0530 0.0117 0.0112 0.0018
```

En particular, hay una probabilidad de quiebra de 0.0018.

Dentro de 20 años las probabilidades son:

```
conditionalDistribution(ratings^20, "BBB")
```

```
##         AAA          AA           A         BBB          BB           B
## 0.006441889 0.063066668 0.236399355 0.212230902 0.121199065 0.110119238
##         CCC           D
## 0.024364998 0.226177886
```

Vemos que la probabilidad de quiebra es mucho más alta siendo 0.2262.

También podemos calcular el tiempo medio en el que se producirá la quiebra. Para ello calculamos el tiempo medio de absorción por el estado absorbente "quiebra":

```
meanAbsorptionTime(ratings)
```

```
##       AAA        AA         A       BBB        BB         B       CCC
## 106.52488  97.22364  87.23272  72.32740  55.79107  37.27407  22.40553
```

Obtenemos el tiempo medio hasta la quiebra para cada uno de los ratings. En particular, nuestra entidad tardará de media 72.33 años en quebrar.

Sabemos que esta cadena no es irreducible, lo podemos comprobar con R. También sabemos que, al tener un estado absorbente y el resto transitorios, a largo plazo la cadena permanecerá en el estado absorbente. Es

decir, en este contexto la distribución límite es inmediata, teniendo todos los estados transitorios probabilidad nula y el estado absorbente probabilidad 1.

```
is.irreducible(ratings)
```

```
## [1] FALSE
```

```
steadyStates(ratings)
```

```
##      AAA AA A BBB BB B CCC D
## [1,]   0  0 0   0  0 0   0 1
```

Referencias

1. Spedicato, Giorgio Alfredo, Tae Seung Kang, Sai Bhargav Yalamanchi, Deepak Yadav, and Ignacio Cordón. The markovchain Package: A Package for Easily Handling Discrete Markov Chains in R. 2021. https://github.com/spedygiorgio/markovchain/.