Trabajo Práctico Final 2019
Probabilidad y Estadística

Alumnos: - Blanco, Fernando
- Navall, Nicolás Uriel

Ejercicio 1

En cada ronda de un juego, un jugador gana $1, con probabilidad p, o pierde $1, con probabilidad 1-p. El jugador comienza con $k. El juego se detiene cuando el jugador pierde todo su dinero o gana un total de $ n, donde n >k. Las fortunas sucesivas del jugador forman una cadena de Markov en {0,1,. . . , n} con X0 = k.
(a) Simular la ruina del jugador para una inversión inicial de $2, jugando un juego justo.
(b) Estimar la probabilidad de que el jugador llegue a la ruina antes de ganar $5.
(c) Construir la matriz de transición para la cadena de markov asociada. Estimar la probabilidad en a) calculando potencias altas de la matriz.
(d) Comparar los resultados con la probabilidad exacta.

Ayuda: ver rutina gamblesruin.R.

```
# gamblersruin.R
library(markovchain)
# gamble(k, n, p)
#   k: Estado inicial del apostador
#   n: El apostador juega hasta $n o quedar en la ruina
#   p: Probabilidad de ganar $1 en cada jugada
#   La función devuelve 1 si el jugador es arruinado eventualmente
#                     retorna 0 si el jugador eventualmente gana $n

gamble <- function(k,n,p) {
  stake <- k
  while (stake > 0 & stake <k+n) { #gana $n cuando tiene $k+n
    bet <- sample(c(-1,1),1,prob=c(1-p,p))
    stake <- stake + bet
  }
  if (stake == 0) return(1) else return(0)
}
```

a) Simulamos dándole al jugador un estado inicial de $2 y un ‘n’=5 y una probabilidad de 0,5 (juego justo).

```
k <- 2
n <- 5
```

```r
p <- 0.5
gamble(k,n,p)
```

b) Estimamos la ruina con realizando varias simulaciones y luego calculando la media, dado que las simulaciones resultaran en 1 si el jugador termina en la quiebra y 0 en caso contrario. $P(Xn = 0) = 0.717$.

```r
trials <- 1000
simlist <- replicate(trials, gamble(k, n, p))
mean(simlist) # Probabilidad estimada de que el jugador llegue a la ruina
```

c) Matriz de transicion

	0$	*1$*	*2$*	*3$*	*4$*	*5$*	*6$*	*7$*
0$	1	0	0	0	0	0	0	0
1$	q	0	p	0	0	0	0	0
2$	0	q	0	p	0	0	0	0
3$	0	0	q	0	p	0	0	0
4$	0	0	0	q	0	p	0	0
5$	0	0	0	0	q	0	p	0
6$	0	0	0	0	0	q	0	p
7$	0	0	0	0	0	0	0	1

La primera columna y la primera fila indican el estado correspondiente a las filas y a las columnas respectivamente.

$P(Xn = 0) = x0 * P^n(0)$

Consideramos 50 un número suficientemente grande, luego

$P(X_{50} = 0) = x0 * P^{50}(0) = 0.711763$

```r
#Estados
states <- c("0",1:7)
```

```r
#Matriz
q <- 1-p
P <- matrix(c(1,0,0,0,0,0,0,0,
            q,0,p,0,0,0,0,0,
            0,q,0,p,0,0,0,0,
            0,0,q,0,p,0,0,0,
            0,0,0,q,0,p,0,0,
            0,0,0,0,q,0,p,0,
            0,0,0,0,0,q,0,p,
            0,0,0,0,0,0,0,1),byrow = TRUE, nrow = 8)

#Cadena de Markov
P <- new("markovchain",states = states, transitionMatrix = P)

#Distribución inicial
init <- c(0,0,1,0,0,0,0,0)

init*P^50
```

d) La probabilidad de que el jugador llegue a la ruina, en un juego justo (p=0.5), sería de $\frac{n}{k+n}$ exactamente. Luego, la probabilidad exacta de llegar a la ruina con un monto inicial igual a \$2 y haber ganado menos de \$5 es $\frac{5}{2+5} = \frac{5}{7} = 0.7142857$.

```r
#Probabilidad exacta
n/(n+k)
```

Ejercicio 2

En aplicaciones de seguridad informática, un honeypot (o sistema trampa) es una herramienta dispuesta en una red o sistema informático para ser el objetivo de un posible ataque informático, y así poder detectarlo y obtener información del mismo y del atacante. Los datos del honeypot son estudiados utilizando cadenas de markov. Se obtienen datos desde una base de datos central y se observan ataques contra cuatro puertos de computadoras - 80, 135, 139 y 445- durante un a˜no. Los estados de la cadena de markov son los cuatro puertos y se incluye un nodo indicando que ning´un puerto est´a siendo atacado (NA). Los datos de monitorean semanalmente y el puerto más atacado durante la semana es guardado. La matriz de transición para la 1 cadena estimada para los ataques semanales es la siguiente:

$$
\begin{array}{c|ccccc}
 & 80 & 135 & 139 & 145 & NA \\
\hline
80 & 0 & 0 & 0 & 0 & 1 \\
135 & 0 & 8/13 & 3/13 & 1/13 & 1/13 \\
139 & 1/16 & 3/16 & 3/8 & 1/4 & 1/8 \\
145 & 0 & 1/11 & 4/11 & 5/11 & 1/11 \\
NA & 0 & 1/8 & 1/2 & 1/8 & 1/4
\end{array}
$$

La distribución inicial es a = (0,0,0,0,1).

(a) Después de dos semanas, ¿cuáles son los puertos con más y menos probabilidad de ser atacados?
(b) Encontrar la distribución límite (si es que existe) de los puertos atacados. Justificar.

```
#Ejercicio 2
library(markovchain)

#Estados
states <- c("80","135","139","145","NA")
#Matriz
P <- matrix(c(0,0,0,0,1,
              0,8/13,3/13,1/13,1/13,
              1/16,3/16,3/8,1/4,1/8,
              0,1/11,4/11,5/11,1/11,
              0,1/8,1/2,1/8,1/4),byrow = TRUE,nrow = 5)

#Distribución inicial
x0 <- c(0,0,0,0,1)

#Cadena de Markov
P <- new("markovchain",states = states, transitionMatrix = P,name =
"Honeypot")
```

a) El puerto con más probabilidad de ser atacado es el 139 con $P(X_2 = 139) = 0.3868007$ y el puerto con menos probabilidad es el 80 con $P(X_2 = 80) = 0.03125$.

```
P2 <- x0*P*P
```

b) Como la cadena es irreducible y aperiódica entonces tiene distribución al límite la cual es:
$P(80) = 0.02146667$ $P(135) = 0.2669333$ $P(139) = 0.3434667$ $P(145) = 0.2273333$ $P(NA) = 0.140$

```
#Distribución al límite
if (is.irreducible(P) && (period(P) == 1)) steadyStates(P)
```

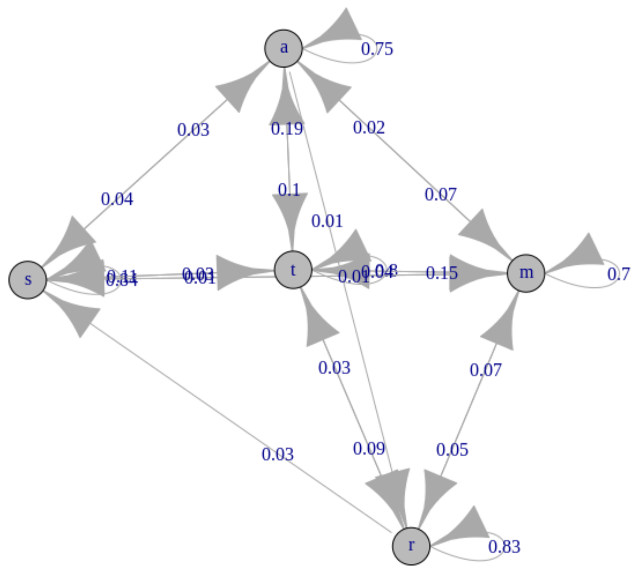
Ejercicio 3

Dans et al. (2012) estudian el comportamiento de delfines en presencia de embarcaciones tur´ısticas en la Patagonia Argentina. Para ello postulan un modelo de cadena de Markov, con espacio de estados conformado por las 5 actividades primarias de los delfines : socializar (s), viajar (t), merodear (m), alimentarse (f), descansar (r), obteniendo la siguiente matriz de transición:

$$
\begin{array}{c c}
 & \begin{array}{ccccc} s & t & m & a & r \end{array} \\
\begin{array}{c} s \\ t \\ m \\ a \\ r \end{array} &
\begin{pmatrix}
0{,}84 & 0{,}11 & 0{,}01 & 0{,}04 & 0{,}00 \\
0{,}03 & 0{,}80 & 0{,}04 & 0{,}10 & 0{,}03 \\
0{,}01 & 0{,}15 & 0{,}70 & 0{,}07 & 0{,}07 \\
0{,}03 & 0{,}19 & 0{,}02 & 0{,}75 & 0{,}01 \\
0{,}03 & 0{,}09 & 0{,}05 & 0{,}00 & 0{,}83
\end{pmatrix}
\end{array}
$$

(a) Clasificar los estados.
(b) Estimar la distribución a largo plazo de la actividad de los delfines.
(c) Simular y graficar una trayectoria de dicho proceso.

```
library(markovchain)

#Comportamiento de delfines en la Patagonia Argentina

#Estados
states <- c("s","t","m","a","r")

#Matriz
P <- matrix(c(0.84, 0.11, 0.01, 0.04, 0,
              0.03, 0.8, 0.04, 0.1, 0.03,
              0.01, 0.15, 0.7, 0.07, 0.07,
              0.03, 0.19, 0.02, 0.75, 0.01,
              0.03, 0.09, 0.05, 0, 0.83),byrow=TRUE,nrow = 5)

#Cadena de Markov
P <- new("markovchain",states = states,transitionMatrix = P,name =
"Comportamiento de delfines")
```



a) Todos los estados son recurrentes aperiódicos.

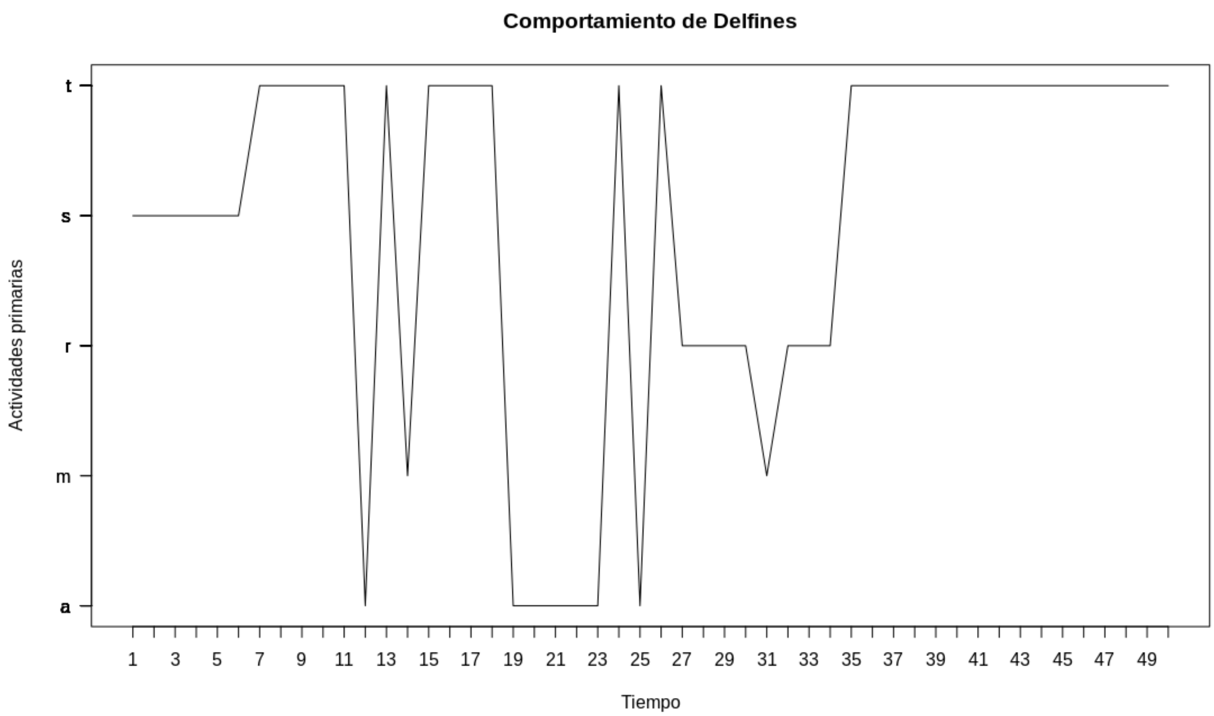
```
summary(P)
period(P)
```

b) Por el punto a) sabemos que la cadena es irreducible, ya que el conjunto de todos los estados es un conjunto cerrado, y aperiódico, por lo que tiene distribución a largo plazo, la cual es:
$P(s) = 0.1478358$ $P(t) = 0.4149254$ $P(m) = 0.2163806$ $P(a) = 0.2163806$ $P(r) = 0.1252985$

```
# Distribución al límite
steadyStates(P)
```

c)

```
sim <- rmarkovchain(P,n = 50)
sim <- factor(sim)
plot(1:length(sim),sim,type = "l",xlab = "Tiempo",ylab = "Actividades
primarias",main = "Comportamiento de Delfines",axes = FALSE)

# Eje Y
axis(2, las =1,labels=sim, at=sim)

# Eje X
axis(1,at=1:length(sim))
```



Ejercicio 4

Simular 50 pasos de caminata aleatoria en el grafo correspondiente a la Figura 1.
(a) Repetir la simulación 10 veces. ¿Cuántas terminaron en el vértice c?
(b) Comparar con el resultado exacto de la probabilidad a largo plazo de visitar a c.

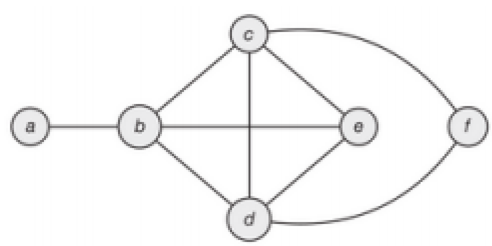


Figura 1

```r
#Ejercicio 4
library(markovchain)

#Estados
states <- c("a","b","c","d","e","f")

#Matriz
P <- matrix(c(0,1,0,0,0,0,
              0.25,0,0.25,0.25,0.25,0,
              0,0.25,0,0.25,0.25,0.25,
              0,0.25,0.25,0,0.25,0.25,
              0,1/3,1/3,1/3,0,0,
              0,0,0.5,0.5,0,0),byrow=TRUE,nrow=6)

#CM
P <- new("markovchain",states = states,transitionMatrix = P)

#Simular 50 pasos de caminata
sim <- rmarkovchain(P,n = 50,t0 = sample(states,1,prob =
c(1/6,1/6,1/6,1/6,1/6,1/6)))
```

a) 5 de las simulaciones terminaron en el estado c.

```r
#a)
a <- replicate(10,rmarkovchain(P,n = 50,t0 = sample(states,1,prob =
c(1/6,1/6,1/6,1/6,1/6,1/6))))
```

b) $P(Xn = c) = 0.2222222$
Los resultados no son similares ya que 10 es un número pequeño de simulaciones para sacar conclusiones respecto a la probabilidad a largo plazo.

```r
if (is.irreducible(P) && (period(P) == 1)) steadyStates(P)
```

Ejercicio 5

Hay k canciones en el reproductor de música de María. El reproductor está seteado en un modo shuffle, en el cual las canciones se eligen aleatoriamente, de forma uniforme, con reemplazo. Por lo tanto, las repeticiones de canciones es posible. Sea Xn el número de canciones no repetidas que han sido escuchadas después de la n-ésima reproducción.

(a) Mostrar que {Xn : n ∈ N}es una cadena de markov y determinar la matriz de transición.

(b) Si María tiene cuatro canciones en su reproductor de música, encontrar la probabilidad de que todas las canciones sean escuchadas despu´es de 6 reproducciones.

a) Veamos que el proceso $\{Xn:\ n \in N\}$ es efectivamente una cadena de markov ya que cumple con los siguientes puntos:
   - Los instantes de observación son infinitos numerables
   - Espacio de estados es discreto
   - $P(N_{n+1} = j \,/\, N_0 = i_0,\ N_1 = i_1,\ \ldots,\ N_n = i_n) \ = P(N_{n+1} = j \,/\, N_n = i_n)$

Los primeros dos puntos se cumplen, ahora pasaremos a demostrar el tercer y último punto:

$P(N_{n+1} = j \,/\, N_0 = i_0,\ N_1 = i_1,\ \ldots,\ N_n = i_n) =^{(1)} P(N_n + \ X_{n+1} = j \,/\, N_0 = i_0,\ N_1 = i_1,\ \ldots,\ N_n = i_n) \ =$
$P(i_n + \ X_{n+1} = j \,/\, N_0 = i_0,\ N_1 = i_1,\ \ldots,\ N_n = i_n) \ = P(X_{n+1} = j - i_n) =^{(2)} P(N_{n+1} = j \,/\, N_n = i_n)$

$\therefore P(N_{n+1} = j \,/\, N_0 = i_0,\ N_1 = i_1,\ \ldots,\ N_n = i_n) \ = P(N_{n+1} = j \,/\, N_n = i_n)$

(1) $N_{n+1} = X_1 + X_0 + \ \ldots \ + \ X_n \ + X_{n+1}$

(2) $P(N_{n+1} = j \,/\, N_n = i_n) = \frac{P(N_{n+1}=j,\, N_n=i_n)}{P(N_n=i_n)} =$

$\frac{P(X_{n+1}=j-i_n,\, N_n=i_n)}{P(N_n=i_n)} = \frac{P(X_{n+1}=j-i_n)P(N_n=i_n)}{P(N_n=i_n)} = P(X_{n+1} = j - i_n)$

La matriz de transición es la siguiente:

	k	k-1	k-2	k-3	...	1	0
k	0	1	0	0	...	0	0
k-1	0	1/k	k-1/k	0	...	0	0
k-2	0	0	2/k	k-2/k	...	0	0
...	...	...	...	...	...	...	...
1	0	0	0	0	...	k-1/k	1/k
0	0	0	0	0	...	0	1

La primera columna y la primera fila corresponden a los estados del proceso.

b) Si el reproductor tiene 4 canciones, la probabilidad de haber escuchado todas luego de 6 reproducciones es

$P(X_6 = 0) + P(X_5 = 0) + P(X_4 = 0) = X_0 * P(X_6 = 0) + X_0 * P(X_5 = 0) + X_0 * P(X_4 = 0) = 0.7089844.$

```
#Estados
states <- c("4","3","2","1","0")

#Matriz
P <- matrix(c(0,1,0,0,0,
            0,1/4,3/4,0,0,
            0,0,2/4,2/4,0,
            0,0,0,3/4,1/4,
            0,0,0,0,1),byrow=TRUE,nrow=5)
```

Ejercicio 6

Se tiran 5 dados y se ponen a un lado aquellos que mostraron un 6. Los restantes se lanzan nuevamente y se repite el procedimiento, poniendo a un lado aquellos dados que muestran un 6, y así sucesivamente. Sea Xn el número de dados en los que salió 6 despu´es de n tiradas.

(a) Describir la matriz de transición P para la cadena de markov.
(b) Encontrar la probabilidad de obtener todos 6 en tres jugadas.
(c) ¿Cómo se espera que sea P^{100}? Confirmar la respuesta utilizando R.

a) La probabilidad de sacar un 6 en un dado es de $\frac{1}{6}$,

Sea Y_n:"Cantidad de dados que salieron 6 en una tirada de n dados" es una variable aleatoria con distribución binomial con parámetros (n,$\frac{1}{6}$).

Luego en el proceso estocástico la probabilidad en un paso $P(i,j)$ es la probabilidad de teniendo i dados con 6 en la próxima tirada tenga j dados con 6 es $P(Y_{5-i} = j - i)\ si\ j - i \geq 0\ y\ 0\ en\ otro\ caso.$

La matriz de transición es la siguiente:

	0	1	2	3	4	5
0	0.4018776	0.4018776	0.1607510	0.0321502 1	0.0032150 21	0.0001286 008

1	0	0.4822531	0.3858025	0.11574074	0.015432099	0.0007716049
2	0	0	0.5787037	0.34722222	0.069444444	0.0046296296
3	0	0	0	0.69444444	0.277777778	0.0277777778
4	0	0	0	0	0.833333333	0.1666666667
5	0	0	0	0	0	1

```
#Probabilidad de sacar 6 en un dado
p <- 1/6

#Matriz
P<-matrix(c(dbinom(0,5,p),dbinom(1,5,p),dbinom(2,5,p),dbinom(3,5,p),dbinom(
4,5,p),dbinom(5,5,p),

0,dbinom(0,4,p),dbinom(1,4,p),dbinom(2,4,p),dbinom(3,4,p),dbinom(4,4,p),
            0,0,dbinom(0,3,p),dbinom(1,3,p),dbinom(2,3,p),dbinom(3,3,p),
            0,0,0,dbinom(0,2,p),dbinom(1,2,p),dbinom(2,2,p),
            0,0,0,0,5/6,1/6,
            0,0,0,0,0,1),byrow = TRUE,nrow=6)
```

b) La probabilidad de obtener todos 6 en 3 jugadas es de $P(X_3 = 5) = 0.01327206$.

```
#Cadena de Markov
P <- new("markovchain", states =states, transitionMatrix = P,name =
"Dados")
#Distribución inicial
x0 <- c(1,0,0,0,0,0)

#b)
x3 <- x0*P^3
x3[6]
```

c) La matriz P^100 debería tener altas probabilidades de caer en el estado 5 en cada una de las filas, indicando de esta manera que sin importar el estado inicial, eventualmente se llegará al estado 5.

```
P^100
#Salida de R
             0            1            2            3            4         5
0 2.566711e-40 1.062849e-31 1.76046e-23 1.457977e-15 6.037336e-08 0.9999999
1 0.000000e+00 2.125698e-32 7.04184e-24 8.747864e-16 4.829869e-08 1.0000000
2 0.000000e+00 0.000000e+00 1.76046e-24 4.373932e-16 3.622402e-08 1.0000000
3 0.000000e+00 0.000000e+00 0.00000e+00 1.457977e-16 2.414935e-08 1.0000000
4 0.000000e+00 0.000000e+00 0.00000e+00 0.000000e+00 1.207467e-08 1.0000000
5 0.000000e+00 0.000000e+00 0.00000e+00 0.000000e+00 0.000000e+00 1.0000000
```

Efectivamente la matriz P^100 es como se esperaba, con todos elementos en 0 exceptuando la última columna la cual tiene elementos de 1.

Ejercicio 7

Considerar la caminata aleatoria en {0,...,k}, la cual se mueve a la izquierda o a la derecha con probabilidades p y q, respectivamente. Si el proceso está en 0, transiciona a 1 en el siguiente paso. Si el proceso está en k, transiciona a k-1 en el siguiente paso. Esto se llama caminata aleatoria con límites reflectantes.
Asumir que k = 3, q = ¼, p = ¾ y la distribución inicial es uniforme.
(a) Calcular la matriz de transición.
(b) Encontrar $P(X_7 = 1|X_0 = 3, X_2 = 2, X_4 = 2)$
(c) Encontrar $P(X_3 = 1, X_5 = 3)$

a) La matriz de transición es la siguiente:

	"0"	"1"	"2"	"3"
"0"	0	1	0	0
"1"	q	0	p	0
"2"	0	q	0	p
"3"	0	0	1	0

```
#Probabilidades
```

```r
p <- 3/4
q <- 1/4

#Matriz
P <- matrix(c(0,1,0,0,
            q,0,p,0,
            0,q,0,p,
            0,0,1,0),byrow = TRUE,nrow = 4)
```

b)

$$P(X_7 = 1/X_0 = 3, X_2 = 2, X_4 = 2) = \frac{P(X_7=1,X_4=2,X_2=2,X_0=3)}{P(X0=3,X2=2,X4=2)} =$$

$$\frac{P^3(2,1)P^2(2,2)P^2(3,2)P(X0=3)}{P^2(2,2)P^2(3,2)P(X0=3)} = \frac{0.593*0.375*0*0,25}{0,375*0*0,25} = \frac{0}{0} \text{Abs!}$$

Concluimos entonces que $P(X_7 = 1/X_0 = 3, X_2 = 2, X_4 = 2) = 0$, ya que no es posible que exista dicha traza.

c) $P(X_3 = 1, X_5 = 3) = P^2(1,3) * P(X_3 = 1)$ $(propiedad\ de\ CM)$

$P^2(1,3) * ((X_0 * P^3)(1))$ $(propiedad\ de\ MC)$

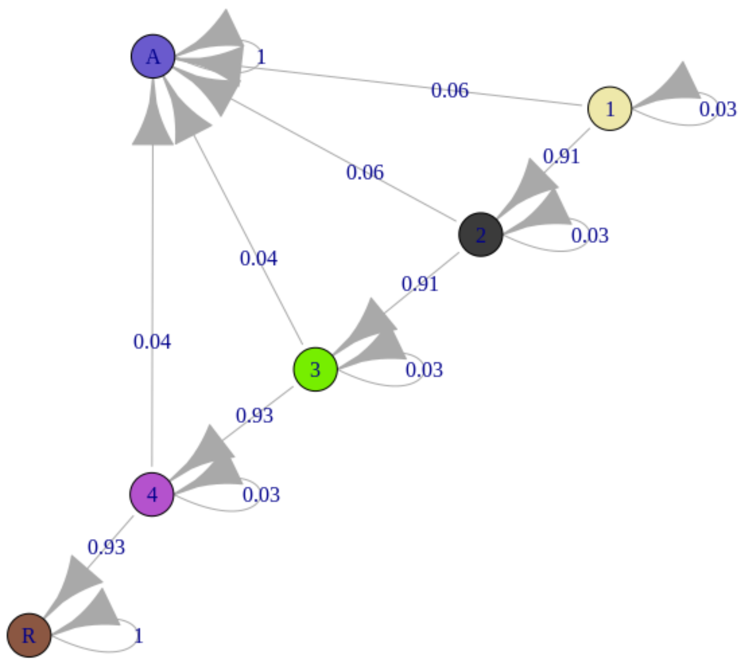
Utilizamos R para realizar el cálculo anterior y concluimos que
$P(X_3 = 1, X_5 = 3) = 0.1032715.$

```r
#Estados
states <- c("0","1","2","3")

#CM
P <- new("markovchain",states = states, transitionMatrix = P, name =
"Caminata Aleatoria")

#Distrinución inicial
x0 <- c(1/4,1/4,1/4,1/4)
#c)
((P^2)["1","3"])*((x0*(P^3))[1,"1"])
```

Ejercicio 8

Los administradores de datos de una universidad desarrollaron un modelo markoviano para simular los ´ındices de graduación en la institución. Los estudiantes pueden abandonar, repetir un año o pasar al a˜no siguiente. Todos tienen un 3 % de chance de repetir el a˜no. Aquellos que se encuentran en primer o segundo a˜no, tienen una chance del 6 % de abandonar. Para estudiantes de tercer y cuarto a˜no, el ´ındice de abandono es de 4 %.

(a) Clasificar los estados.
(b) Establecer la matriz de transición de un paso.
(c) Determinar el número promedio de años que un estudiante que ingresa en primer año permanece en la institución antes de abandonar o recibirse.



a) Los estados R y A (recibido y abandona) son absorbentes y el resto son transitorios.

b) La matriz de transición en un paso es la siguiente:

	1º año	*2º año*	*3º año*	*4º año*	*Recibido*	*Abandona*
1º año	0.03	0.91	0	0	0	0.06
2º año	0	0.03	0.91	0	0	0.06
3º año	0	0	0.03	0.93	0	0.04
4º año	0	0	0	0.03	0.93	0.04
Recibido	0	0	0	0	1	0
Abandon	0	0	0	0	0	1

a						

La primera columna y la primera fila indican el estado correspondiente a las filas y a las columnas respectivamente.

```r
#Matriz
P <- matrix(c(0.03,0.91,0,0,0,0.06,
            0,0.03,0.91,0,0,0.06,
            0,0,0.03,0.93,0,0.04,
            0,0,0,0.03,0.93,0.04,
            0,0,0,0,1,0,
            0,0,0,0,0,1),byrow = TRUE,nrow = 6)
```

c)

El número promedio de años es la suma del número promedio de años que cursa cada grado lo cual es equivalente al número de visitas a cada estado partiendo desde el primer año.

$R(1, 1) + R(1, 2) + R(1, 3) + R(1, 4)$

$R = (I - P)^{-1}$

Como no existe la inversa de esa matriz no podemos calcular R de esa forma, por lo cual para calcular dicha suma pasaremos a calcular la matriz N. Para ello realizaremos los siguientes cálculos:

```r
Q <- P[1:4,1:4]

det(diag(x=1,nrow=4,ncol=4)-Q)
#0.8852928

N  <- inv(diag(x=1,nrow=4,ncol=4)-Q)
N%*%matrix(c(1,1,1,1),byrow=TRUE,nrow=4)
#        [,1]
#[1,] 3.775341
#[2,] 2.925363
#[3,] 2.019343
```

```
#[4,] 1.030928
```

Q es la matriz de transición de los estados transitivos. Verificamos que existe la inversa de la matriz (I-Q) calculando su determinante, el cual es diferente de 0.

$N = (I - Q)^{-1}$ es la parte de los estados transitivos de la matriz R, es decir que $N(i, j)$ es el número promedio que un alumno cursa el año j, habiendo partido desde el año i.

Con esta información a disposición para saber el año promedio que permanece en la institución un alumno que empieza en primero sólo tenemos que sumar los $N(1, j),\ j \in [1, 4]$ que serían el promedio de año que cursa cada grado.

El número promedio de años que un estudiante que empieza primer año permanece en la institución es 3.775341

Ejercicio 9

El modelo Wright-Fisher describe la evolución de una población fija de k genes. Los genes pueden ser de uno de dos tipos, llamados alelos: A o a. Sea Xn el número de alelos A en la población en el momento n, donde el tiempo se mide por generaciones. Bajo este modelo, el número de alelos A en el momento n+1 se obtiene muestreando con reemplazo desde la poblaci´on de genes en el momento n. Por lo tanto, habiendo i alelos de tipo A en el momento n, el número de alelos A en el momento n+1 tiene una distribución 4 binomial con par´ametros k y p = i/k. Esto resulta en una cadena de markov con matriz de transición definida por:

$$P_{ij} = \binom{k}{j} (\frac{i}{k})^j (1 - \frac{i}{k})^{k-j},\ para\ 0 \leq i, j \leq k. \qquad (1)$$

(a) Simular este proceso para algún valor de k.
(b) Observar qué valor toma P00 y Pkk.
(c) Cuando la cadena progresa, la población, en algún momento, termina con todos alelos a (estado 0) o todos alelos A (estado k). Determinar cuál es la probabilidad de que la población evolucione al estado k.

a) Simulamos el proceso para un k=3.

```
library(markovchain)

#Estados
states = c("0",1:3)
```

```r
#Matriz
p <- function(k,i,j){
  dbinom(j,k,i/k)
}

P <- matrix(c(p(3,0,0),p(3,0,1),p(3,0,2),p(3,0,3),
              p(3,1,0),p(3,1,1),p(3,1,2),p(3,1,3),
              p(3,2,0),p(3,2,1),p(3,2,2),p(3,2,3),
              p(3,3,0),p(3,3,1),p(3,3,2),p(3,3,3)),byrow = TRUE,nrow = 4)
#Matriz resultante
#            [,1]      [,2]      [,3]       [,4]
#[1,] 1.00000000 0.0000000 0.0000000 0.00000000
#[2,] 0.29629630 0.4444444 0.2222222 0.03703704
#[3,] 0.03703704 0.2222222 0.4444444 0.29629630
#[4,] 0.00000000 0.0000000 0.0000000 1.00000000

#MC
P <- new("markovchain",states = states, transitionMatrix = P, name =
"Wright-Fisher")

rmarkovchain(P,n = 42,t0 = "1") #k = 3
```

a) Al correr la simulación obtuvimos la siguiente traza de estados:

```
 [1] "2" "2" "1" "1" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0"
"0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0"
[30] "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0"
```

Lo cual significa que no existen más alelos de tipo A.

b) Observamos que P_{00} y P_{kk}son igual a 1,
En el caso 00 como no se tiene ningún gen tipo A no se podría transmitir ese gen a la próxima generación.
En el caso kk se tiene todos genes tipo A y ningún de otro tipo, por lo que en la próxima generación sólamente habrá de ese tipo.

c) La probabilidad de que la población evolucione al estado k sería la distribución al límite. Como la cadena no es irreducible, pues tiene 2 estados absorbentes y 2 transitivo por lo cual tiene subconjuntos cerrados, no tiene distribución invariante, por lo tanto para aproximar la probabilidad realizamos potencias altas de la matriz de transición. Al realizar $P' = P^n$, siendo n un número muy grande, la probabilidad de $P(k) \approx P'(,k)$

Siguiendo con k=3 que tomamos en el punto a) y considerando que 1000 es un número suficientemente grande la distribución de probabilidad $P(3)$ no queda aproximadamente igual a:

Estado inicial	P(i,3)
0	0
1	0.3333333
2	0.6666667
3	1

```
(P^1000)[,"3"]
```

Ejercicio 10

El día de la elección, las personas participaron en un centro de votaciones de acuerdo con un proceso de Poisson. El promedio, 100 votantes llegan cada hora.
(a) Si 150 personas arribaron durante la primer hora, ¿qué probabilidad hay de que al menos 350 votantes arriban antes de las 3 horas?
(b) Simular dicho proceso y graficar una trayectoria.
(c) Considerar la variable aleatoria: tiempo entre arribos. Obtener los valores a través de simulación y graficar. Determinar qué distribución tiene el tiempo entre arribos de votantes.

a) Sea Ns: "Cantidad de votantes que arriban en s horas"

$$P(N_3 \geq 350 / N_1 = 150) = P(N_3 - N_1 \geq 350 - 150) = P(N_2 \geq 200) =$$

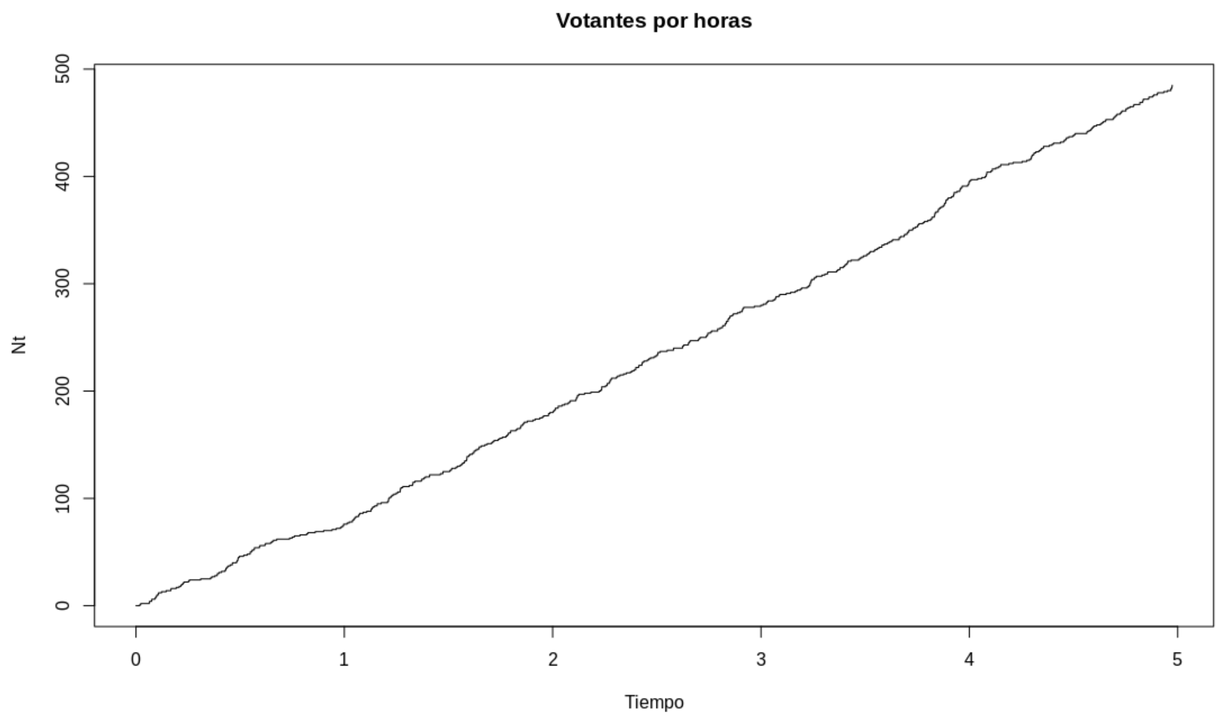
$$1 - P(N_2 \leq 200) = 1 - \sum_{i=0}^{200} \frac{e^{-2*100}*(2*100)^i}{i!} = 1 - \sum_{i=0}^{200} \frac{e^{-200}*200^i}{i!} = 0.4812057$$

La probabilidad de que al menos 350 votantes arriben antes de las 3 horas, sabiendo que en la primer hora arribaron 150, es de 0.4812057.

```r
#P(Ns <= k)
pmenor <- function(s,k,lambda){
  r <- 0
  for (i in 0:k){
      r <- r + dpois(i,s*lambda)
  }
  return(r)
}

1 - pmenor(2,200,100)
```

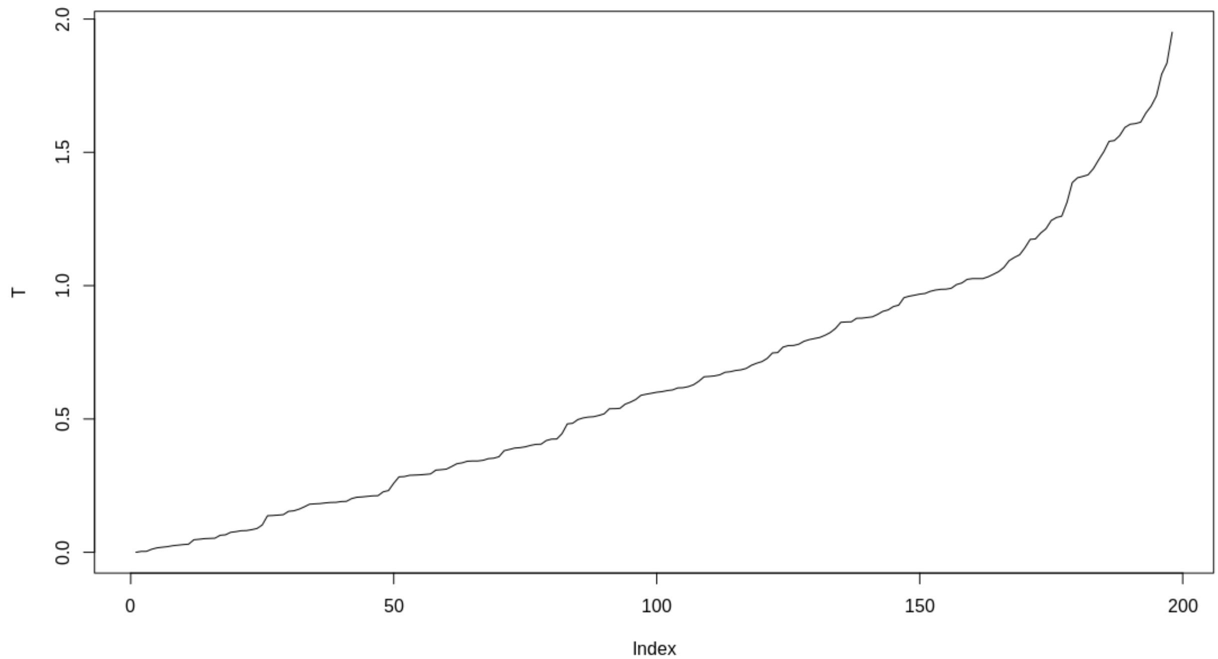
b)

```r
 ProcesoPois<- function(t,lambda){
  N<- rpois(1,t*lambda) #(Nt)
  C<- sort(runif(N,0,t)) #(Tk , 0<=k<=Nt)
  data.frame(x=c(0,C),y=c(0:N))
}
N <- c(0)

P <- ProcesoPois(5,100)
plot(P,type = "s",xlab="Tiempo",ylab="Nt",main = "Votantes por horas")
```



c)

```
#T:"Tiempo entre arribos"
#t = tiempo hasta el cual voy a simular
processtime <- function(t,lambda){
  N <- rpois(1,lambda = t*lambda) #Número de arribos total
  T <- c(0)
  for (i in 1:N) { #Simulo todos los Tk (tiempo hasta el k arribo)
      T <- c(T,runif(1,0,t-T[i]))
  }
  T <- sort(T)
}

T <- processtime(2,100)
plot(T,type = "l")
```



A partir del ploteo resultante de la simulación podemos concluir que la distribución es exponencial.

***

Ejercicio 9 c

La probabilidad de que la cadena evolucione al estado k nos la brinda la matriz F que tiene la probabilidades de ir de i a j.
Siguiendo con nuestro ejemplo con k = 3 tenemos:
$F(0,3) = 0 \; y \; F(3,3) = 1$ por ser estados absorbentes
Luego para las $F(1,3) \; y \; F(2,3)$ calculamos la matriz G la cual es:

F(1,0)	F(1,3)
F(2,0)	F(2,3)

$G = (I - Q)^{-1} * B$

B Es la matriz con las probabilidades de paso de los estados transitorios a los absorbentes y $(I - Q)^{-1}$ es la matriz R (promedio de número de visitas) de los estados transitorios

Tenemos entonces:

$(I - Q)^{-1} =$

2.1428571	0.8571429
0.8571429	2.1428571

$B =$

0.29629630	0.03703704
0.03703704	0.29629630

Luego:

$G =$

0.6666667	0.3333333
0.3333333	0.6666667

Y la probabilidad de que la población evolucione al estado k es

Estado inicial	F(i,3)
0	0
1	0.3333333
2	0.6666667
3	1