Año 2019

UNIVERSIDAD NACIONAL DE ROSARIO

FACULTAD DE CIENCIAS EXACTAS INGENIERÍA Y AGRIMENSURA

Cátedra:Probabilidad y Estadística

TRABAJO PRÁCTICO FINAL: Cadenas de Markov

Genga, Cristian

Ejercicio Número 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 $X_0 = k$.

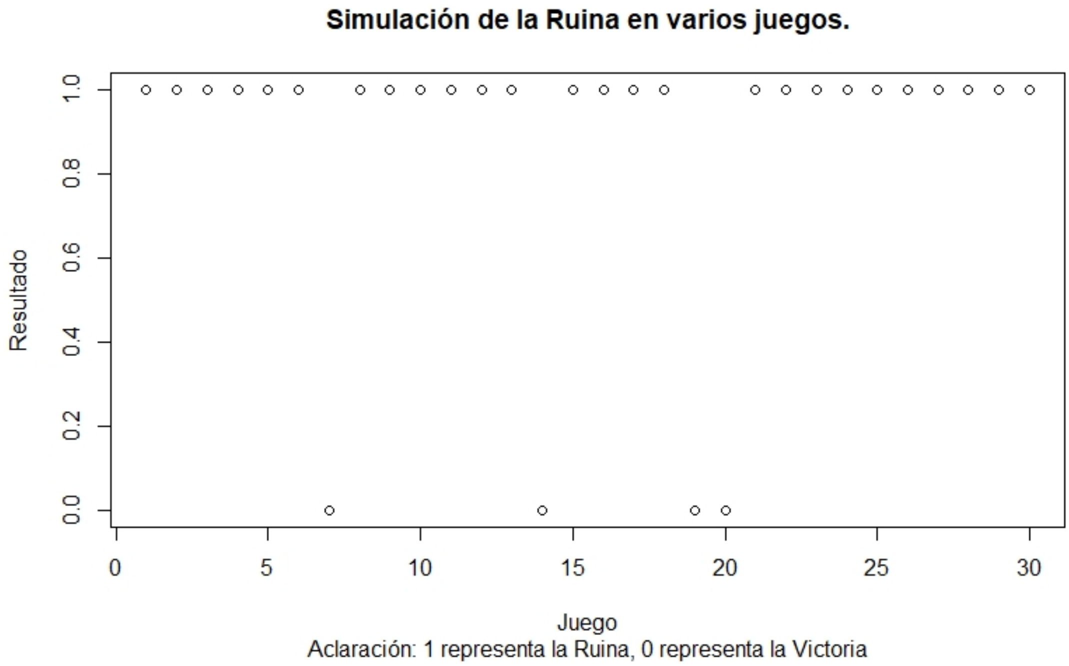
```
library(markovchain)

Juego<-function(MontoInit,MontoMax,Prob){ #Función que simula un juego.
  MontoAct<-MontoInit
  while(MontoAct>0 & MontoAct<MontoMax){ #Comprobamos si llego a la ruina o ganó la partida.
    Resultado<-sample(c(1,-1),1,c(Prob,1-Prob),replace = TRUE) #Ejecutamos una ronda.
    MontoAct<-MontoAct+Resultado
  }
  if(MontoAct==0) return(1) #LLego a la ruina.
  else return(0)
}
```

(a) Simular la ruina del jugador para una inversión inicial de \$2, jugando un juego justo.

A continuación, se simulará una única partida en la cual el jugador inicia con un monto de \$2, gana con un monto de \$15 y la probabilidad de ganar como de perder coincide en 0,5.

```
MontoInit = 2
MontoMax = 15
Prob = 0.5
Juego(MontoInit,MontoMax,Prob)
```

En el siguiente gráfico se puede observar una simulación de 30 juegos. Los cuales muestran que es más prominente llegar a la ruina que a ganar el juego.




```r
NJuegos=30 #Cantidad de juegos a simular.

Simulacion=replicate(n=NJuegos,{ #LLevamos a cabo las partidas.
  Juego(MontoInit,MontoMax,Prob)
})
```

Como se puede observar de 30 juegos simulados, solo se obtuvo la victoria en 4 de ellos.

(b) Estimar la probabilidad de que el jugador llegue a la ruina antes de ganar \$5.
A continuación se realizarán diferentes estimaciones respecto a la probabilidad de que el jugador llegue a la ruina antes de ganar \$5, partiendo de un monto inicial de \$2. Se estimara en todos los casos con 5000 juegos y luego se calculará su media para obtenerla.
Probabilidad de ruina:

- Jugando un juego justo: 0.6046
- Jugando con probabilidad de ganar igual a 0.2: 0.984
- Jugando con probabilidad de ganar igual a 0.8: 0.0646
- Jugando con probabilidad de ganar igual a 0.95: 0.022

```r
Simulaciones = replicate(5000,Juego(MontoInit,MontoMax,Prob))
mean(Simulaciones)
```

(c) Construir la matriz de transición para la cadena de markov asociada. Estimar la probabilidad en (a) calculando potencias altas de la matriz.

	$0	$1	$2	$3	$4	$5
$0	1	p	0	0	0	0
$1	1-p	0	p	0	0	0
$2	0	1-p	0	p	0	0
$3	0	0	1-p	0	p	0
$4	0	0	0	1-p	0	p
$5	0	0	0	0	0	1

Al estimar la probabilidad de caer en la ruina iniciando el juego con $2 con varias potencias altas se obtiene:

- Con potencia n=10: 0.5371094
- Con potencia n=50: 0.5999869
- Con potencia n=100: 0.6
- Con potencia n=500: 0.6

```
Estados = c("0","1","2","3","4","5")
p=0.5
MatrizT<-matrix(c(1,0,0,0,0,0,
            1-p,0,p,0,0,0,
            0,1-p,0,p,0,0,
            0,0,1-p,0,p,0,
            0,0,0,1-p,0,p,
            0,0,0,0,0,1),byrow = TRUE,nrow = 6)
MatrizT
CM<-new("markovchain",states = Estados,transitionMatrix = MatrizT) #Creacion de la cadena de markov
PI<-c(0,0,1,0,0,0)
PI*(P^n)
```

(d) Comparar los resultados con la probabilidad exacta.
En un juego justo, la probabilidad de que el jugador llegue a la ruina es (n-k)/n. La probabilidad exacta de llegar a la ruina comenzando con $2, antes de ganar $5 es (5-2)/5 = 0.6

Ejercicio Número 2:

En las 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 145- durante un año. Los estados de la cadena de markov son los cuatros puertos y se incluye un nodo indicando que ningún puerto está siendo atacado (NA). Los datos se monitorean semanalmente y el puerto más atacado durante la semana es guardado. La matriz de transición para la cadena estimada para los ataques semanales es la siguiente:

	80	135	139	145	NA
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	½	1/8	1/4

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?

Luego de dos semanas se obtuvo que el puerto con más probabilidad de ser atacado es el 139, donde $P(X_2 = 139) = 0.3868007$, a la vez que el puerto con menos probabilidad de serlo es el 80, con $P(X_2 = 80) = 0.03125$.

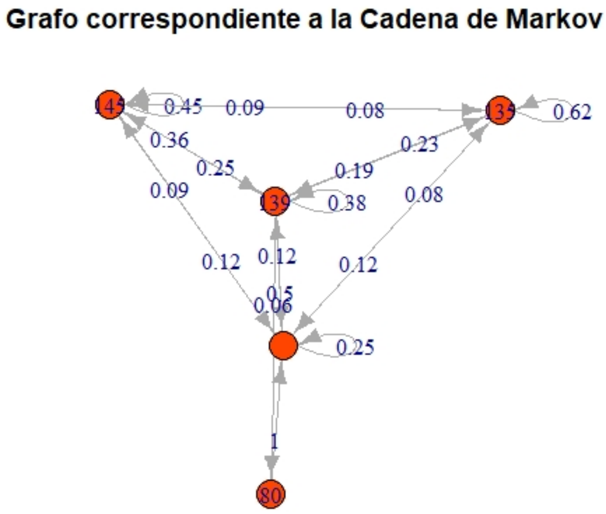
```r
library(markovchain)
states<-c("80","135","139","145",NA)
T = matrix(c(0,0,0,0,1,
             0,8/13,3/13,1/13,1/13,
             1/16,3/16,3/8,1/4,1/8,
             0,1/11,4/11,5/11,1/11,
             0,1/8,1/2,1/8,1/4),nrow = 5,byrow = TRUE)
T
pi<- c(0,0,0,0,1) #Distribución inicial
mc <- new("markovchain",states = states,transitionMatrix = T,name="Ejercicio2")#Cadena de markov
transation2 = pi*(mc*mc) # =(mc^2)
transation2
```

(b) Encontrar la distribución límite (si es que existe) de los puertos atacados. Justificar.
Al analizar la cadena correspondiente se obtuvo que es irreducible y que además su período es 1, lo cual se puede afirmar que es aperíodica. En base a estas dos condiciones se puede hallar la distribución límite de los puertos atacados, la cual corresponde a:

- P(80) = 0.02146667
- P(135) = 0.2669333
- P(139) = 0.3434667
- P(145) = 0.2273333

- P(NA) = 0.1408

```
summary(mc) #confirma que es irreducible
period(mc) #Nos dice el periodo de la cadena.
steadyStates(mc)
```



Ejercicio Número 3:

Dans et al.(2012) estudian el comportamiento de delfines en presencia de embarcaciones turísticas 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:

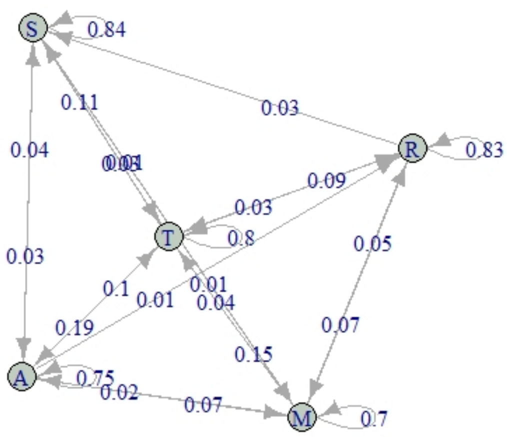
	s	t	m	f	R
s	0.84	0.11	0.01	0.04	0.00
t	0.03	0.80	0.04	0.10	0.03
m	0.01	0.15	0.70	0.07	0.07
f	0.03	0.19	0.02	0.75	0.01
r	0.03	0.09	0.05	0.00	0.83

(a) Clasificar los estados.

Todos los estados de la correspondiente cadena de markov se clasifican como recurrentes y cerrados. La periodicidad de todos los estados es igual a 1, lo cual clasifica a la cadena como aperiódica, y se obtuvo que es irreducible.

```r
library(markovchain)
Estados<-c("S","T","M","A","R")
Matriz<-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)
Matriz
mc<-new("markovchain",states = Estados,transitionMatrix = Matriz,name="Delfines")
summary(mc)
period(mc)
```

Grafo asociado a la cadena de markov



(b) Estimar la distribución a largo plazo de la actividad de los delfines.

Como se obtuvo que la cadena es irreducible y aperíodica, podemos estimar la distribución a largo plazo, la cual corresponde a:

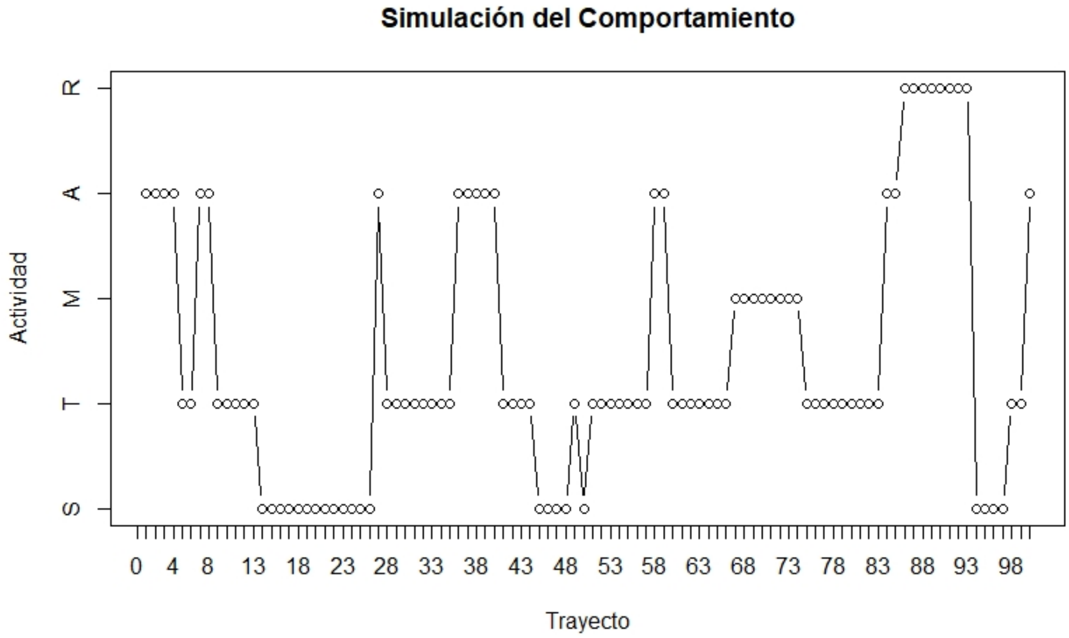
- P(S) = 0.1478358
- P(T) = 0.4149254
- P(M) = 0.0955597
- P(A) = 0.2163806
- P(R) = 0.1252985

```r
steadyStates(mc)
```

(c) Simular y graficar una trayectoria de dicho proceso.

A continuación se realizará una simulación respecto al comportamiento de los delfines de longitud n=100, y probabilidad inicial equivalente para todos los estados.

```r
Simulacion<-rmarkovchain(mc,n=100,t0=sample(Estados,"1",prob=c(1/5,1/5,1/5,1/5,1/5)))
```



Ejercicio Número 4:

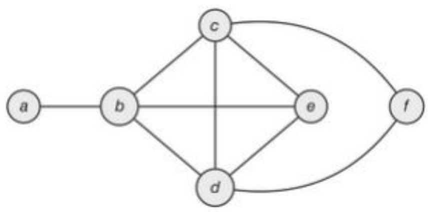


Figura 1

Simular 50 pasos de caminata aleatoria en el grafo correspondiente a la Figura 1.

```r
library(markovchain)
Estados<-c("A","B","C","D","E","F")
Matriz1<-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)
Matriz1
rw<-new("markovchain",states = Estados,transitionMatrix = Matriz1,name="Caminata")
Simulacion50<-rmarkovchain(rw,n=50,sample(Estados,"1",prob = c(1/6,1/6,1/6,1/6,1/6,1/6)))
Simulacion50
```

(a) Repetir la simulación 10 veces.¿Cuantos terminaron en el vértice c?

Se pudo observar que de las 10 simulaciones, solo 3 terminaron en el vértice C.

```r
Rep10<-replicate(10,rmarkovchain(rw,n=50,sample(Estados,1,prob = c(1/6,1/6,1/6,1/6,1/6,1/6))))
Rep10
```

(b) Comparar con el resultado exacto de la probabilidad a largo plazo de visitar a C.

La probabilidad de visitar a C en 10 simulaciones es $P(X_{10} = C) = 0.2224783$, a comparación de la probabilidad a largo plazo de visitar C, la cual corresponde a $P(X_n = C) = 0.2222222$ con n un valor muy grande.

```r
steadyStates(rw) #Probabilidad a largo plazo.
pi<-c(1/6,1/6,1/6,1/6,1/6,1/6) #Distribuciones iniciales.
pi*(rw^10)  #Probabilidad en 10 pasos.
```

Ejercicio Número 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 X_n el número de canciones no repetidas que han sido escuchadas después de la n-ésima reproducción.

(a) Mostrar que $\{X_n : n \in N\}$ es una cadena de markov y determinar la matriz de transición.

Para probar que $\{X_n : n \in N\}$ es una cadena de markov se debe corroborar lo siguiente:

1. Los instantes de observación deben ser infinitos numerables.

2. Espacio de estados discretos.

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

La condición 1) se cumple, pues los instantes de observación se basan en los números naturales, los cuales son infinitos numerables. El item 2) también es válido porque hablamos de una cantidad finita de canciones. Por último falta corroborar el item 3), a continuación se verá:

Antes hay que tener en cuenta dos resultados que ayudarán a la demostración:

- Resultado (1): $N_{n+1} = X_1 + X_2 + ... + X_n + X_{n+1}$
- Resultado (2): $P(N_{n+1} = j/N_n = i_n) = \frac{P(N_{n+1}=j, N_n=i_n)}{P(N_n=i_n} = \frac{P(X_{n+1}=j-i_n, N_n=i_n)}{P(N_n=i_n)} = \frac{P(X_{n+1}=j-i_n)P(N_n=i_n)}{P(N_n=i_n)} = P(X_{n+1} = j - i_n)$

$$P(N_{n+1} = j/N_0 = i_0, N_1 = i_1, ..., N_n = i_n) \overset{(1)}{=}$$
$$P(N_n + X_{n+1} = j/N_0 = i_0, ..., N_n = i_n) =$$
$$P(X_{n+1} + i_n = j/N_0 = i_0, ..., N_n = i_n) =$$
$$\mathrm{P}(\mathrm{X}_{n+1} = j - i_n/N_0 = i_0, ..., N_n = i_n) = P(X_n + 1 = j/N_n = i_n) \overset{(2)}{=}$$
$$P(N_{n+1} = j/N_n = i_n)$$

Por lo tanto queda demostrado el item 3). Lo cual termina mostrando que es una cadena de markov.
A continuación, la matriz de transición correspondiente:

	1	2	3	4
1	0	1	0	0
2	0	1/(k-1)	(k-2)/(k-1)	0
3	0	0	2/(k-1)	(k-3)/(k-1)
4	0	0	0	1

```
library(markovchain)
Estados<-c("1","2","3","4")
k=4
Rep=matrix(c(0,1,0,0,
             0,1/(k-1),(k-2)/(k-1),0,
             0,0,2/(k-1),(k-3)/(k-1),
             0,0,0,1),byrow=TRUE,nrow = k)
Rep
SH<-new("markovchain",states=Estados,transitionMatrix=Rep,name="Shuffle")
```

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

La probabilidad de que halla oído las cuatro canciones después de 6 reproducciones es $P(X_6 = 4) = 0.7407407$.

```
PI<-c(1,0,0,0) #Inicia con la primer canción
PI*(SH^6) #Comprobamos la probabilidad de que haya llegado a k.
```

Ejercicio Número 6:

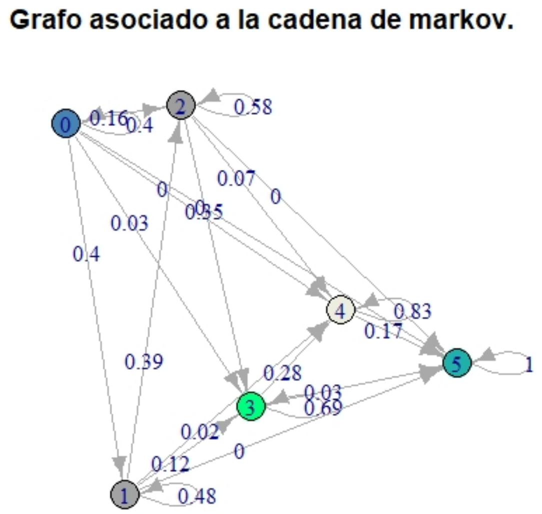
Se tiran 5 dados y se ponen a un lado aquellos que mostraron un 6. Los restantes se lanzan nuevamente y se repite el procedimiento, poniendo a un lado aquellos dados que muestran un 6, y así sucesivamente.
Sea X_n el número de dados en los que salió 6 después de n tiradas.

(a) Describir la matriz de transición P para la cadena de markov.

	0	1	2	3	4	5
0	0.4018776	0.4018776	0.1607510	0.03215021	0.003215021	0.0001286008
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

Donde Y_n se define como la cantidad de dados que mostraron 6 de un total de n dados en una tirada. La cual posee distribución binomial con parametros n y p = 1/6(Probabilidad de sacar 6 en un dado normal). $Y_n = Bi(n, 1/6)$

```
library(markovchain)
Estados<-c("0","1","2","3","4","5")
MatrizT<-matrix(c(dbinom(0,5,1/6),dbinom(1,5,1/6),dbinom(2,5,1/6),dbinom(3,5,1/6),dbinom(4,5,1/6),dbinom(5,5,1/
                  0,dbinom(0,4,1/6),dbinom(1,4,1/6),dbinom(2,4,1/6),dbinom(3,4,1/6),dbinom(4,4,1/6),
                  0,0,dbinom(0,3,1/6),dbinom(1,3,1/6),dbinom(2,3,1/6),dbinom(3,3,1/6),
                  0,0,0,dbinom(0,2,1/6),dbinom(1,2,1/6),dbinom(2,2,1/6),
                  0,0,0,0,dbinom(0,1,1/6),dbinom(1,1,1/6),
                  0,0,0,0,0,dbinom(0,0,1/6)),byrow=TRUE,nrow = 6)
MatrizT
D2<-new("markovchain",states=Estados,transitionMatrix=MatrizT,name="DadosLocos")
```



(b) Encontrar la probabilidad de obtener todos 6 en tres jugadas.

La probabilidad de obtener todos 6 en tres tiradas es $P(X_3 = 5) = 0.01327206$

```
PI<-c(1,0,0,0,0,0)
IB<- PI*(D2^3)
IB
IB[6]
```

(c) ¿Cómo se espera que sea P^{100}? Confirmar la respuesta utilizando R.

Al estimar con 100 tiradas, lo que debería ocurrir es que la probabilidad de obtener todos los 6 sea muy alta, además de que las demás probabilidades muy bajas o casi nulas, puesto que después de 100 tiradas es muy probable que hallan salido los 5 dados con 6.

```
D2100<-D2^100
D2100
```

La respuesta de R es la siguiente:

	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

La cual coincide plenamente con lo anterior dicho.

Ejercicio Número 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=1/4, p=3/4 y la distribución inicial es uniforme.

(a) Calcular la matriz de transición.

La matriz de transición respectiva es la siguiente:

	0	1	2	3
0	0	1	0	0
1	1/4	0	3/4	0
2	0	1/4	0	3/4
3	0	0	1	0

```
library(markovchain)
k<-3
p<-3/4
q<-1/4

Estados<-c("0","1","2","3")
pi<-c(1/4,1/4,1/4,1/4)
CA<-matrix(c(0,1,0,0,
             q,0,p,0,
             0,q,0,p,
             0,0,1,0),byrow=TRUE,nrow=4)

CA<-new("markovchain",states=Estados,transitionMatrix=CA,name="CaminataAleatoriaEJ7")
```

Y en la cual se puede observar un grafo del siguiente estilo:

(b) Encontrar $P(X_7 = 1|X_0 = 3, X_2 = 2, X_4 = 2)$.

$$P(X_7 = 1|X_0 = 3, X_2 = 2, X_4 = 2) = \frac{P(X_7=1) \cap P(X_0=3, X_2=2, X_4=2)}{P(X_0=3, X_2=2, X_4=2)} =$$

$$= \frac{P(X_7=1, X_0=3, X_2=2, X_4=2)}{P(X_0=3, X_2=2, X_4=2)} = \frac{P(X_0=3)*P^2(3,2)*P^2(2,2)*P^3(2,1)}{P(X_0=3)*P^2(3,2)*P^2(2,2)} =$$

$= \frac{0.25*0*0.9375*0.25}{0.25*0*0.9375} = \frac{0}{0}$ Indeterminación

<u>Cálculos Auxiliares:</u>
$P^2(3, 2) = P(3, 2) * P(2, 2) = 1 * 0 = 0$

$P^2(2, 2) = P(2, 1)*P(1, 2)+P(2, 3)*P(3, 2) = q*p+p*1 = 1/4*3/4+1/4 =$
$= 0.9375$

$P^3(2, 1) = P(2, 3) * P(3, 2) * P(2, 1) + P(2, 1) * P(1, 0) * P(0, 1) =$
$= p * 1 * q + q * q * 1 = 3/4 * 1/4 + 1/4 * 1/4 = 0.25$

El resultado dió indeterminación por lo que se puede entender que no es posible obtener tal probabilidad ya que no existe una caminata con esas condiciones.

(c) Encontrar $P(X_3 = 1, X_5 = 3)$.

$P(X_3 = 1, X_5 = 3) = P(X_3 = 1) * P(1,2) * P(2,3) = (\pi_0 * (CA^3))(1) * p * p =$
$= 0.1032715$

<u>Cálculos Auxiliares:</u>
$P(X_3 = 1) = (\pi_0 * (CA^3))(1)$

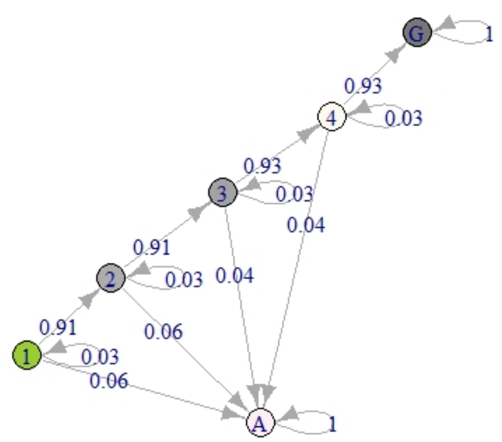
```
((CA)["1","2"]*(CA)["2","3"])*((pi*(CA^3))[1,"1"])
```

Ejercicio Número 8:

Los administradores de datos de una universidad desarrollaron un modelo markoviano para simular los índices de graduación en la institución. Los estudiantes pueden abandonar, repetir un año o pasar al año siguiente. Todos tienen un 3% de chance de repetir el año. Aquellos que se encuentran en primer o segundo año, tienen una chance del 6% de abandonar. Para estudiantes de tercer y cuarto año, el índice de abandono es de 4%.

(a) Clasificar los estados.

Grafo asociado a la cadena de markov



Se puede observar en el grafo que los estados A y G son cerrados, recurrentes y absorbentes. Mientras que los estados 1, 2, 3 y 4 son transitorios.

(b) Establecer la matriz de transición de un paso.

La matriz correspondiente al grafo anterior es:

	1	2	3	4	A	G
1	0.03	0.91	0	0	0.06	0
2	0	0.03	0.91	0	0.06	0
3	0	0	0.03	0.93	0.04	0
4	0	0	0	0.03	0.04	0.93
A	0	0	0	0	1	0
G	0	0	0	0	0	1

```r
library(markovchain)
Estados=c("1","2","3","4","A","G")
UN<-matrix(c(0.03,0.91,0,0,0.06,0,
             0,0.03,0.91,0,0.06,0,
             0,0,0.03,0.93,0.04,0,
             0,0,0,0.03,0.04,0.93,
             0,0,0,0,1,0,
             0,0,0,0,0,1),byrow = TRUE,nrow=6)
UN
UN<-new("markovchain",states=Estados,transitionMatrix=UN,name="Universidad")
```

(c) Determinar el número promedio de años que un estudiante que ingresa en primer año permanece en la institución antes de abandonar o recibirse.

Para determinar el promedio de años que un estudiante permanece en la institución, se deberán sumar los números promedios de los años que se tardan en cursar cada año universitario. En otras palabras es la suma de los promedios de visitas al estado j que la cadena realiza cuando parte de i, donde en este ejercicio $j = 1, ..., 4$ y $i = 1$.
Promedio $= R(1,1) + R(1,2) + R(1,3) + R(1,4)$, donde $R = (I - UN)^{-1}$

Donde I-UN =

0.97	-0.91	0	0	-0.06	0
0	0.97	-0.91	0	-0.06	0
0	0	0.97	-0.93	-0.04	0
0	0	0	0.97	-0.04	-0.93
0	0	0	0	0	0
0	0	0	0	0	0

Es evidente ver que la matriz I-UN es singular, por lo cual no podremos calcular R. Por lo que se calculará la matriz $R1 = (I - T)^{-1}$, donde T es la matriz de estados transitivos.

Donde (I-T) =

0.97	-0.91	0	0
0	0.97	-0.91	0
0	0	0.97	-0.93
0	0	0	0.97

La cual es no singular, por lo que si se podrá obtener R1.

R1 =

1.030928	0.967159	0.907334	0.869919
0	1.030928	0.967159	0.927276
0	0	1.030928	0.988415
0	0	0	1.030928

De aquí se podrá calcular el promedio de años de cursado de un estudiante que ingresa en primer año, cual valor corresponde a,

Promedio $= R1(1,1) + R1(1,2) + R1(1,3) + R1(1,4) =$
$= 1.030928 + 0.967159 + 0.907334 + 0.869919 = 3.77534$

Se puede observar que el tiempo promedio que permanece un alumno en la institución desde que arranca en primer año es 3.77534 años, es decir casi un total 4 años.

```
T=UN[1:4,1:4]
T
Id<-matrix(c(1,0,0,0,
             0,1,0,0,
             0,0,1,0,
             0,0,0,1),byrow=TRUE,nrow=4)
Id
R1<-solve(Id-T)
R1
PromCurso=R1[1,1]+R1[1,2]+R1[1,3]+R1[1,4]
PromCurso
```

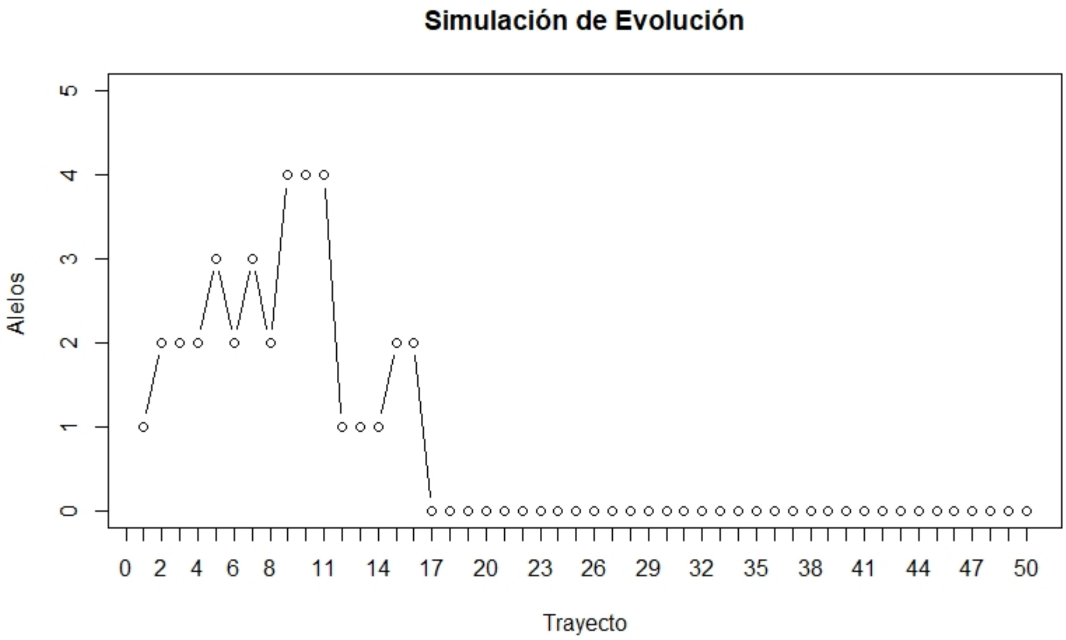
Ejercicio Número 9:

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 X_n 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 población de genes en el momento n. Por lo tanto, habiendo i alelos de tipo A en el momento n, el número de alelos en el momento $n+1$ tiene una distribución binomial con parámetros k y $p = i/k$. Esto resulta en una cadena de markov con matriz de transición definida por:

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

(a) Simular este proceso para algún valor de k.

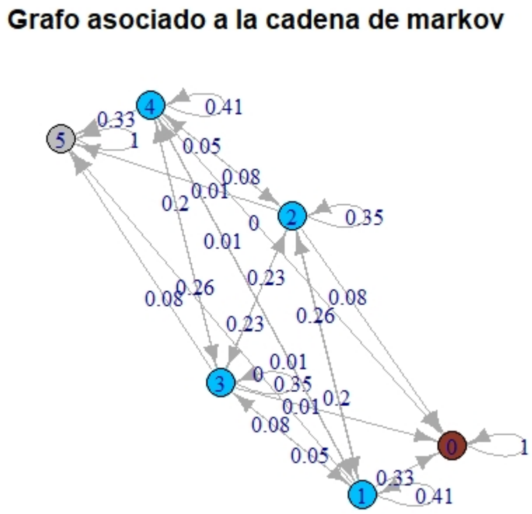
Considerar k=5 y n=50 para realizar la simulación :



Se observa que para esta simulación la población termina con todos alelos de tipo a.

```
library(markovchain)
k<-5
Estados<-c("0","1","2","3","4","5")
FuncProb<-function(i,j,k){
  dbinom(j,k,i/k)
}

WF<-matrix(c(FuncProb(0,0,5),FuncProb(0,1,5),FuncProb(0,2,5),FuncProb(0,3,5),FuncProb(0,4,5),FuncProb(0,5,5),
             FuncProb(1,0,5),FuncProb(1,1,5),FuncProb(1,2,5),FuncProb(1,3,5),FuncProb(1,4,5),FuncProb(1,5,5),
             FuncProb(2,0,5),FuncProb(2,1,5),FuncProb(2,2,5),FuncProb(2,3,5),FuncProb(2,4,5),FuncProb(2,5,5),
             FuncProb(3,0,5),FuncProb(3,1,5),FuncProb(3,2,5),FuncProb(3,3,5),FuncProb(3,4,5),FuncProb(3,5,5),
             FuncProb(4,0,5),FuncProb(4,1,5),FuncProb(4,2,5),FuncProb(4,3,5),FuncProb(4,4,5),FuncProb(4,5,5),
             FuncProb(5,0,5),FuncProb(5,1,5),FuncProb(5,2,5),FuncProb(5,3,5),FuncProb(5,4,5),FuncProb(5,5,5)),
           byrow=TRUE,nrow=6)
WF
WF<-new("markovchain",states=Estados,transitionMatrix=WF,name="Wright-Fisher")
plot(WF,main="Grafo asociado a la cadena de markov")
Simulacion<-rmarkovchain(WF,n=50)
```

Cuyo grafo se representa:



El anterior grafo clasifica los estados de esta cadena, siendo 0(Alelo a) y 5(Alelo A) estados recurrentes, cerrados y absorbentes. Mientras que los estados 1, 2, 3 y 4 son transitorios.

(b) Observar qué valor toma P_{00} y P_{kk}.

La matriz de transición correspondiente a la cadena es:

	0	1	2	3	4	5
0	1	0	0	0	0	0
1	0.32768	0.4096	0.2048	0.0512	0.0064	0.00032
2	0.07776	0.2592	0.3456	0.2304	0.0768	0.01024
3	0.01024	0.0768	0.2304	0.3456	0.2592	0.07776
4	0.00032	0.0064	0.0512	0.2048	0.4096	0.32768
5	0	0	0	0	0	1

Donde se observa el valor $P_{00} = 1$ y $P_{55} = 1$.

(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.

Como la cadena no es irreducible, no se podrá obtener la distribución limite, por lo que se estimará con un valor muy alto para obtener una aproximación lo bastante exacta. Tomando a k=5, y n=500, se tiene:

- $WF^{500}(0,5) = 0$
- $WF^{500}(1,5) = 0.2$
- $WF^{500}(2,5) = 0.4$
- $WF^{500}(3,5) = 0.6$
- $WF^{500}(4,5) = 0.8$
- $WF^{500}(5,5) = 1$

```
(WF^500)[,"5"]
```

Ejercicio Número 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?

Consideremos la variable X_n: "Número de votantes que arriban en n horas".

$P(X_3 \geq 350/X_1 = 150) = P(X_3 - X_1 \geq 350 - 150) =$

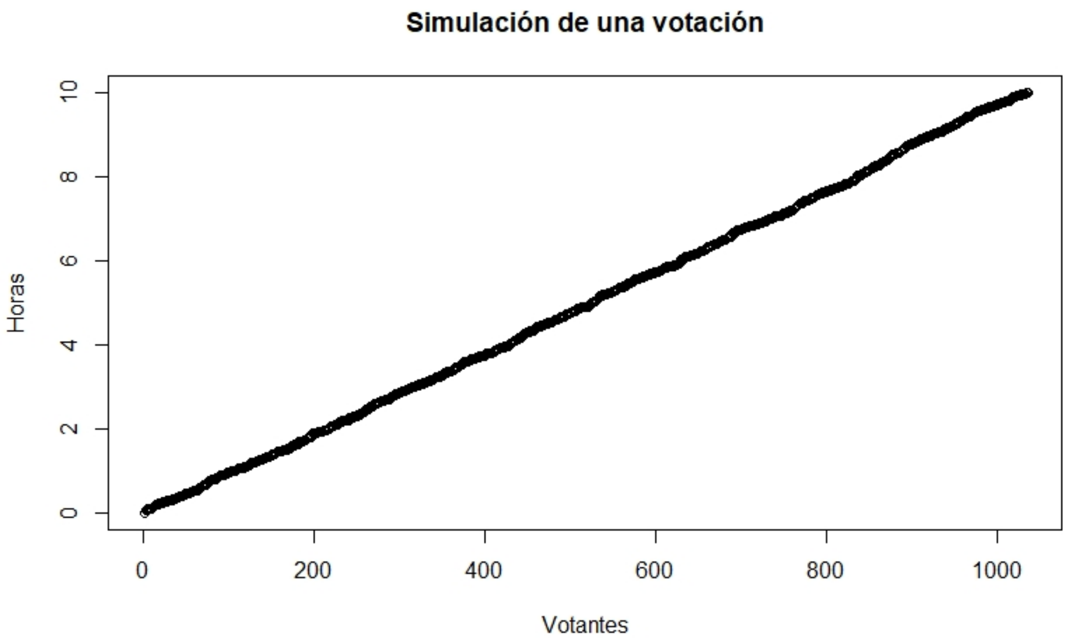
$= P(X_2 \geq 200) = 1 - P(X_2 < 200) = 1 - \sum_{j=0}^{(200-1)} \frac{e^{-\lambda s}(\lambda * s)^j}{j!} =$

$= 1 - \sum_{j=0}^{199} \frac{e^{-100*2}(100*2)^j}{j!} = 1 - \sum_{j=0}^{199} \frac{e^{-200}(200)^j}{j!} = 1 - 0.4905966 = 0.5094034$

```
library(markovchain)

ppoison<-function(suma=0,s,lambda){
  for(i in 0:199){
    suma = suma + dpois(i,s*lambda)
  }
  return(suma)
}
1 - ppoison(0,2,100)
```

(b) Simular dicho proceso y graficar una trayectoria.

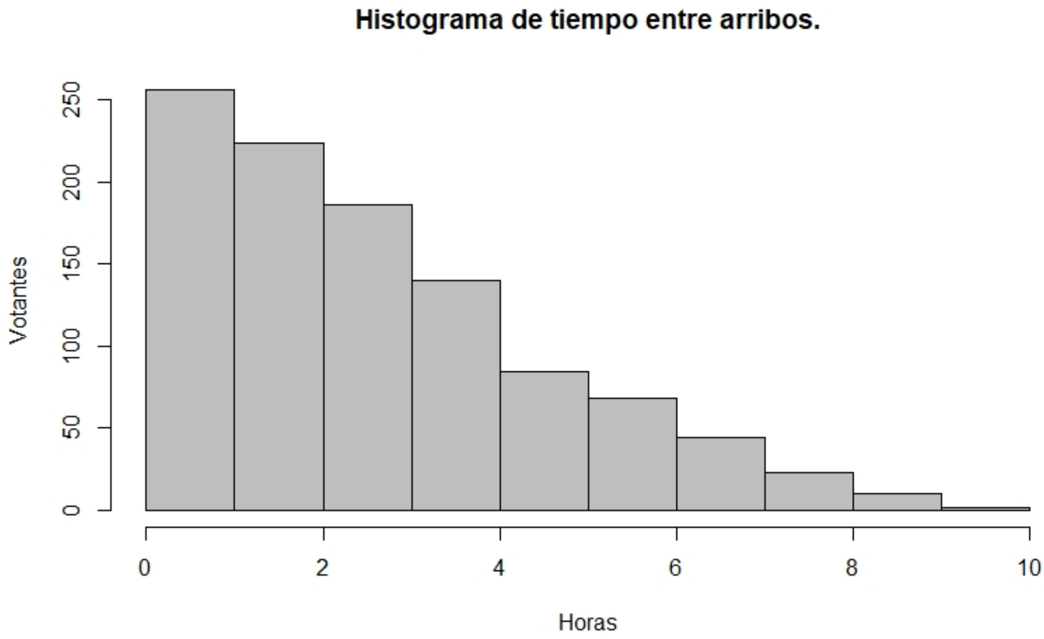
A continuación se hará una simulación de una elección, la cual será de s = 10 horas y $\lambda = 100$ votantes por hora.

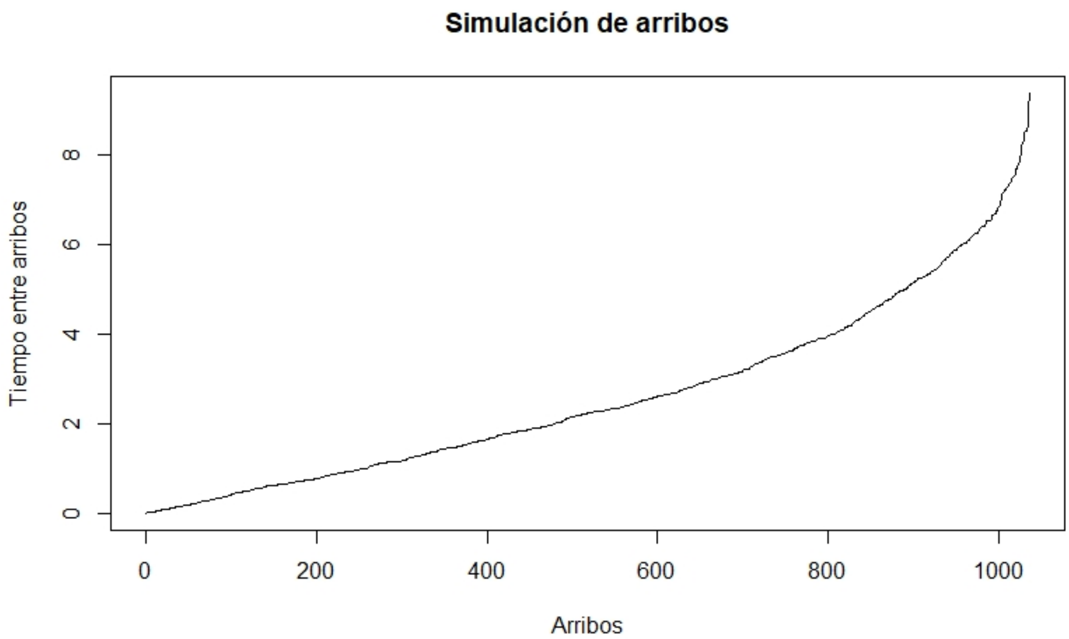



```r
votantes = rpois(1,1000)#1000 = (horas=10)*(lambda = 100)
votantes
Horas = c(0,runif(votantes,0,10))
```

(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.

```r
votantes
for(i in 1:votantes){
  Horas[i]=abs(Horas[i+1]-Horas[i])
}
```





Con el histograma anteriormente mostrado se puede observar una distribución exponencial en lo que respecta a la variable tiempo entre arribos.