Trabajo práctico: Procesos estocásticos

Licenciatura en ciencias de la computación, Universidad nacional de Rosario

Rossi, Martín

1.

Se define un suceso A tal que $P(A) = p$

y el proceso $\{N_k$: cantidad de ensayos independientes hasta obtener k éxitos consecutivos de A$\}$

$\Omega = \{\omega : \omega = (\omega_1, \omega_2, ...), \omega_i \in \{E, F\}, i \in \mathbb{N}\}$, o sea, todas las combinaciones posibles de éxitos y fallos.

$T = \mathbb{N}$ (valores que puede tomar k en N_k)

$E = \mathbb{N}$ (valores que puede tomar $N_k(\omega)$)

a.

Se simularán 10 trayectorias del proceso, para ello se fijan ω_i para $i = 0..9$ y se calculan los $N_2(\omega_i), N_3(\omega_i)$ y $N_4(\omega_i)$, $i = 0..9$.

$p = 0.25$

```
##    N2  N3   N4
## w0  5   6  624
## w1  8   9   10
## w2 17 204  205
## w3 17  43  140
## w4  3  36  214
## w5 41  61 1091
## w6  7 143  264
## w7  7   8    9
## w8 23  73   74
## w9 31  35   36
```

b.

Se aproxima $E(N_k)$ como el promedio de las 10 trayectorias:

```
##    N2    N3    N4
##  15.9  61.8 266.7
```

2.

a.

Simulación de 50 pasos del proceso $\{D_n = 2I_n - 1\}$ donde I_n es un proceso de Bernoulli con $p = 0.85$. Se empieza a observar la partícula en un momento dado que corresponderá con D_0 del proceso.

```
##    0 1  2  3  4 5  6  7  8  9
## 0  1 1 -1  1  1 1  1  1  1  1
## 10 1 1 -1  1  1 1 -1 -1  1  1
## 20 1 1  1 -1  1 1  1  1  1  1
## 30 1 1  1  1  1 1  1 -1 -1  1
## 40 1 1  1  1 -1 1 -1  1  1 -1
```

b.

Se define el proceso $\{S_n = D_0 + D_1 + ... + D_n\}$ que corresponde a la posición de la partícula al momento n.

Una realización de un proceso estocástico es la sucesión de infinitos valores que toman las variables. Se simularán los primeros 100 pasos de la misma con $p = 0.85$.

```
##     0  1  2  3  4  5  6  7  8  9
## 0   1  2  1  2  3  4  5  4  5  6
## 10  5  6  7  8  9 10 11 12 13 14
## 20 15 16 17 18 19 18 19 20 19 20
## 30 21 22 23 22 23 22 21 22 23 24
## 40 23 24 25 24 25 26 27 26 27 28
## 50 29 30 31 32 33 34 35 36 37 38
## 60 39 40 41 42 43 44 45 44 45 46
## 70 47 48 49 50 51 52 51 52 51 52
## 80 53 54 53 54 55 56 57 58 59 60
## 90 61 62 63 64 65 64 65 64 65 64
```

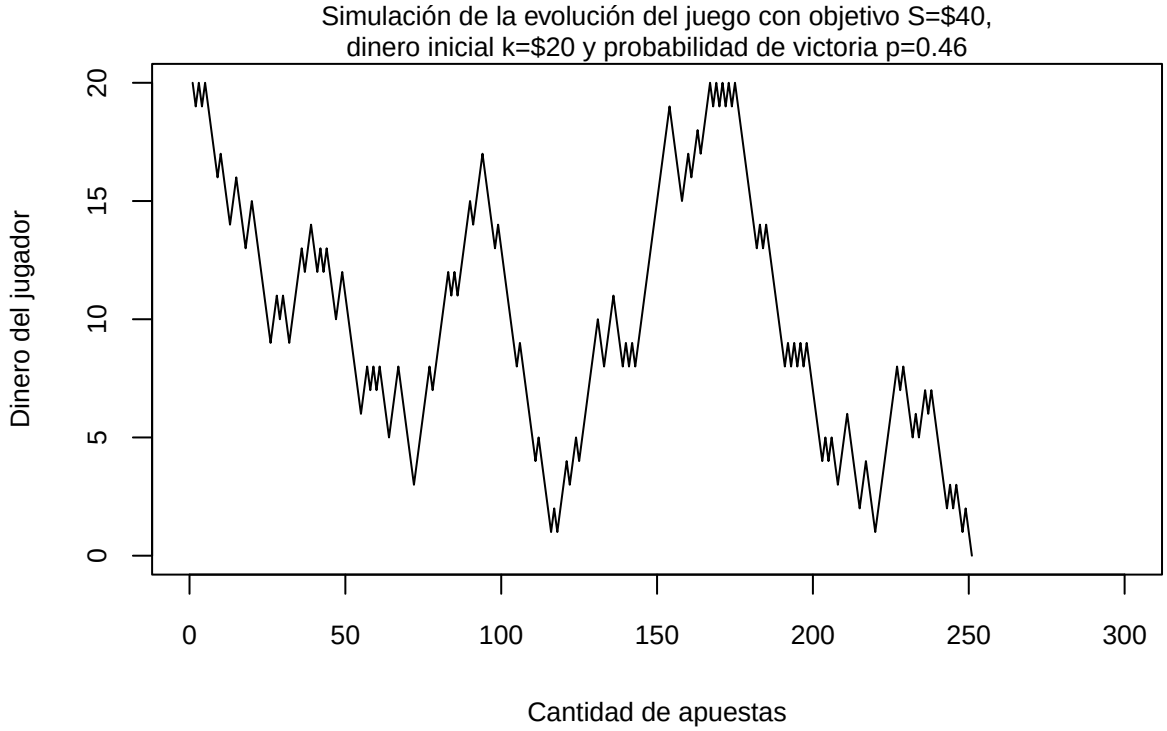
Se puede deducir que $\lim_{n\to\infty} S_n = \infty$ porque $p > 0.5$.

3.

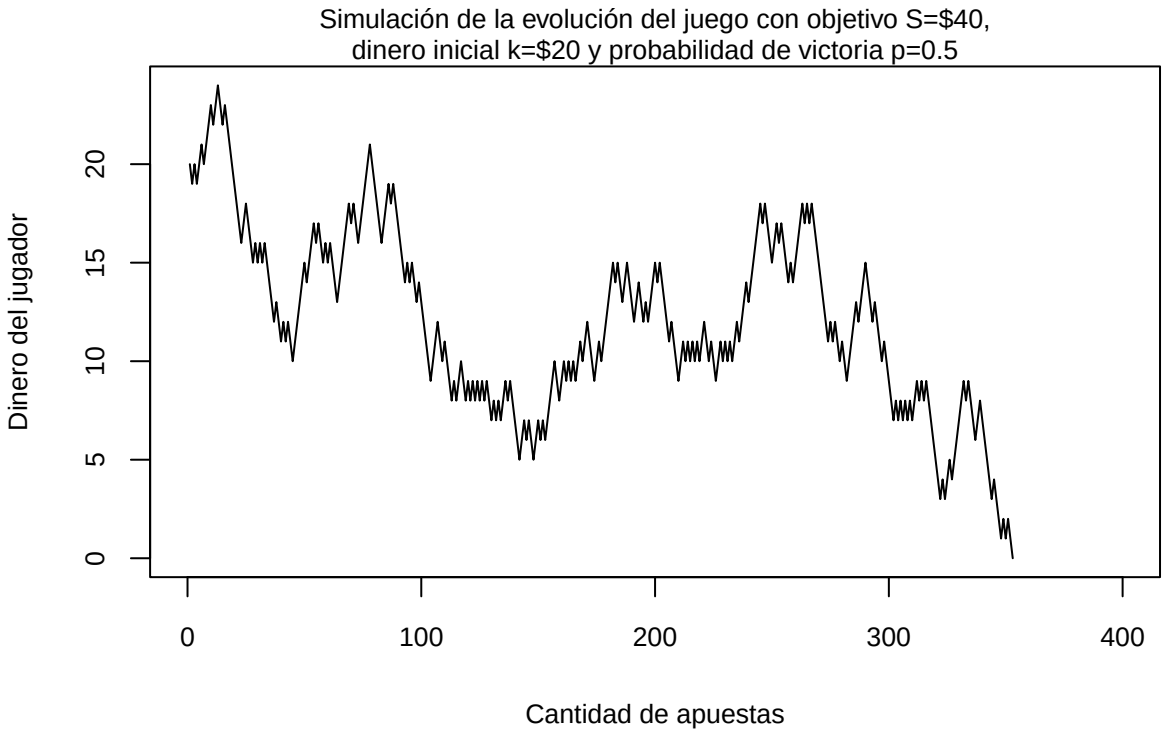
Se define el proceso de Bernoulli $\{I_n\}$ con p como la probabilidad de ganar el juego. Como en el ejercicio anterior $2I_i - 1$ es 1 o -1, lo que puede representar la ganancia del jugador en la jugada i. Entonces se puede definir el proceso estocástico $\{X_n = k + \sum_{i=0}^{n} 2I_i - 1\}$ que representa el dinero que tiene el jugador al momento n, con estado inicial $X_0 = k$.

a.

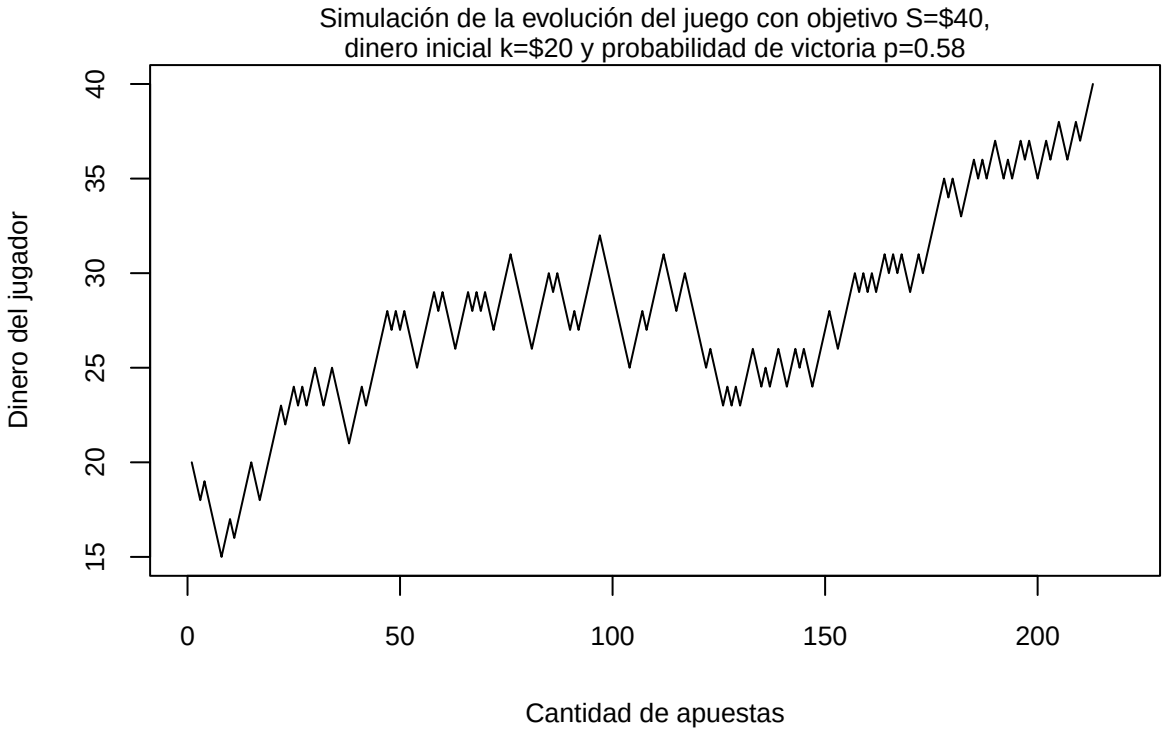
Se realizan las simulaciones para $p < 0.5 (p = 0.46)$, $p = 0.5$ y $p > 0.5 (p = 0.58)$, todas con dinero inicial $k = 20$ y objetivo $S = 40$.



El jugador finaliza con $0 en 251 apuestas. Habiendo tenido $20 como máximo y $0 como mínimo.



El jugador finaliza con $0 en 353 apuestas. Habiendo tenido $24 como máximo y $0 como mínimo.



El jugador finaliza con $40 en 213 apuestas. Habiendo tenido $40 como máximo y $15 como mínimo.

b.

Para aproximar la probabilidad de ruina del jugador se simularán 1000 trayectorias para cada p y se contarán las veces que perdió todo el dinero sobre el total. Los otros parámetros seguirán siendo los mismos, $k = 20$ y $S = 40$.

$p = 0.46$:

```
## [1] 0.96
```

$p = 0.5$:

```
## [1] 0.495
```

$p = 0.58$:

```
## [1] 0.002
```

Lo que significa que cuando $p = 0.46$, el jugador llegará a la ruina con probabilidad 0.96, si $p = 0.5$ llegará con probabilidad 0.495, y con $p = 0.58$ con probabilidad 0.002.

Calculando las probabilidades analiticamente:

El proceso $\{X_n\}$ se puede ver como una cadena de Markov, ya que tiene el conjunto de instantes de observación infinito numerable, espacio de estados discreto y cumple con la propiedad markoviana, porque el dinero que tenga el apostador en un momento n sólo dependerá del $n-1$, y no de como llegó a ese $n-1$.

Para $S = 40$

$E = \{0..40\}$

$$P = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & ... & 0 & 0 & 0 \\ q & 0 & p & 0 & 0 & ... & 0 & 0 & 0 \\ 0 & q & 0 & p & 0 & ... & 0 & 0 & 0 \\ 0 & 0 & q & 0 & p & ... & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & ... & q & 0 & p \\ 0 & 0 & 0 & 0 & 0 & ... & 0 & 0 & 1 \end{pmatrix}$$

La cadena tiene dos estados absorventes, 0 y 40, por lo que los dos forman por separado conjuntos cerrados. Los demás estados son todos transitorios. Por lo tanto no va a existir una única distribución estacionaria, pero se sabe que la matriz P^∞ tendrá la siguiente forma:

$$P = \begin{pmatrix} a_0 & 0 & ... & 0 & b_0 \\ a_1 & 0 & ... & 0 & b_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{40} & 0 & ... & 0 & b_{40} \end{pmatrix}$$

Con cada a_i como la probabilidad de ruina comenzando con $\$i$, y $b_i = 1 - a_i$ la probabilidad de llegar al objetivo.

Se aproximará con R, elevando la matriz de transición a una potencia alta, y mirando la primera y última columna de la fila que corresponde a dinero inicial \$20.

```
##          a20  b20
## p=0.46 0.96 0.04
## p=0.5  0.50 0.50
## p=0.58 0.00 1.00
```

4.

a.

Simulando el algoritmo 3000 veces en una lista de 7 elementos y sacando el promedio de comparaciones (se muestra la función porque servirá para la demostración posterior):

```
f4a<-function(A){
  if(length(A)>1){
    x<-A[runif(1,1,length(A))]
    length(A)-1+f4a(A[A<x])+f4a(A[A>x])
  }else{
    0
  }
}
sum(replicate(3000,f4a(sample(seq(1,7)))))/3000
```

```
## [1] 13.48333
```

b.

Defino las variables aleatorias C_n: cantidad de comparaciones realizadas por el algoritmo Quick-Sort en una lista de tamaño n, con $n \geq 0$.

Las esperanzas de C_0 y C_1 son triviales porque pueden tomar un solo valor.

$E(C_0) = 0$ y $E(C_1) = 0$

Si se define la nueva variable aleatoria D: posición del elemento que será usado como pivote, se puede condicionar la esperanza $E(C_n)$ por D y usar el teorema $E(X) = E(E(X|Y))$.

$E(C_n) = \sum_{d=0}^{n} E(C_n|D = d)P(D = d)$

Como cada elemento tiene la misma probabilidad de ser elegido como pivote, y hay n elementos posibles:

$E(C_n|D = d)P(D = d) = E(C_n|D = d)\frac{1}{n}$

Pero si se sabe que el d-ésimo elemento es pivote, por la definición de la función C_n es:

$C_n = n - 1 + C_d + C_{n-1-d}$

Por lo que la esperanza será:

$E(C_n|D = d) = E(n - 1 + C_d + C_{n-1-d})$

porque el elemento d se compara con todos los otros, es decir, $n - 1$ comparaciones, y después se suman las comparaciones de las dos sublistas. Por lo tanto la esperanza quedaría:

$E(C_n) = \sum_{d=1}^{n} E(n-1+C_d+C_{n-1-d})\frac{1}{n} = \frac{1}{n}\sum_{k+l=n-1} E(n-1+C_k+C_l) = n-1+\frac{1}{n}\sum_{k+l=n-1} E(C_k)+E(C_l)$

Calculando con R las esperanzas:

```
##       C0 C1 C2     C3     C4  C5   C6      C7
## E(Cn)  0  0  1 2.6667 4.8333 7.4 10.3 13.4857
```

5.

a.

Para modelar el comportamiento de visitas, se considera que el usuario tiene una probabilidad de saltar a una página cualquiera de manera aleatoria por más que no exista un enlace que lleve a ella. Se toma la probabilidad de salto como 0.15 y la de que elija un enlace 0.85, excepto en el estado g que no tiene enlaces. Y para cada acción que tome el usuario tendrá la misma probabilidad de llegar a la otra página entre las posibilidades.

Entonces para calcular la probabilidad en un paso $P(x, y)$ sería:

$$\text{Si } x \neq g\text{: } P(x,y) = \begin{cases} \frac{1}{G(x)} * 0.85 + \frac{1}{7} * 0.15 & \text{si existe el arco } x \to y \\ \frac{1}{7} * 0.15 & \text{sino} \end{cases}$$

$P(g, y) = \frac{1}{7}$, para todo y

$E = \{a, b, c, d, e, f, g\}$

La matriz de probabilidad en un paso queda:

$$P = \begin{pmatrix} \frac{3}{140} & \frac{3}{140} & \frac{3}{140} & \frac{3}{140} & \frac{25}{56} & \frac{25}{56} & \frac{3}{140} \\ \frac{32}{105} & \frac{3}{140} & \frac{32}{105} & \frac{3}{140} & \frac{3}{140} & \frac{32}{105} & \frac{3}{140} \\ \frac{3}{140} & \frac{3}{140} & \frac{3}{140} & \frac{25}{56} & \frac{3}{140} & \frac{25}{56} & \frac{3}{140} \\ \frac{3}{140} & \frac{3}{140} & \frac{3}{140} & \frac{3}{140} & \frac{3}{140} & \frac{61}{70} & \frac{3}{140} \\ \frac{131}{560} & \frac{3}{140} & \frac{3}{140} & \frac{131}{560} & \frac{3}{140} & \frac{131}{560} & \frac{131}{560} \\ \frac{25}{56} & \frac{25}{56} & \frac{3}{140} & \frac{3}{140} & \frac{3}{140} & \frac{3}{140} & \frac{3}{140} \\ \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} \end{pmatrix} = \frac{1}{1680} \begin{pmatrix} 36 & 36 & 36 & 36 & 750 & 750 & 36 \\ 512 & 36 & 512 & 36 & 36 & 512 & 36 \\ 36 & 36 & 36 & 750 & 36 & 750 & 36 \\ 36 & 36 & 36 & 36 & 36 & 1464 & 36 \\ 393 & 36 & 36 & 393 & 36 & 393 & 393 \\ 750 & 750 & 36 & 36 & 36 & 36 & 36 \\ 240 & 240 & 240 & 240 & 240 & 240 & 240 \end{pmatrix}$$

Al considerar la posibilidad de salto a cualquier nodo, todo el conjunto es cerrado irreducible y aperiódico.

b.

Se usa la función *rmarkovchain* del paquete *markovchain* para simular 100 pasos de la cadena partiendo del nodo a:

```
##   [1] "a" "e" "g" "a" "f" "a" "f" "a" "f" "a" "e" "f" "a" "e" "a" "f" "b" "a"
##  [19] "e" "f" "b" "f" "a" "e" "g" "e" "d" "f" "a" "e" "f" "b" "a" "f" "b" "a"
##  [37] "f" "b" "c" "g" "e" "a" "f" "f" "a" "e" "a" "e" "e" "f" "a" "e" "f" "b"
##  [55] "g" "f" "a" "e" "a" "f" "a" "e" "f" "b" "c" "f" "a" "f" "a" "f" "c" "d"
##  [73] "f" "a" "e" "f" "b" "a" "e" "g" "b" "c" "f" "b" "c" "a" "f" "a" "e" "g"
##  [91] "g" "e" "d" "f" "a" "e" "f" "a" "f" "b" "a"
```

c.

El rango de cada página es la probabilidad de llegar a ella a largo plazo, o sea, después de n pasos cuando $n \to \infty$. Se calcula la distribución límite π_∞ con la función *steadyStates*.

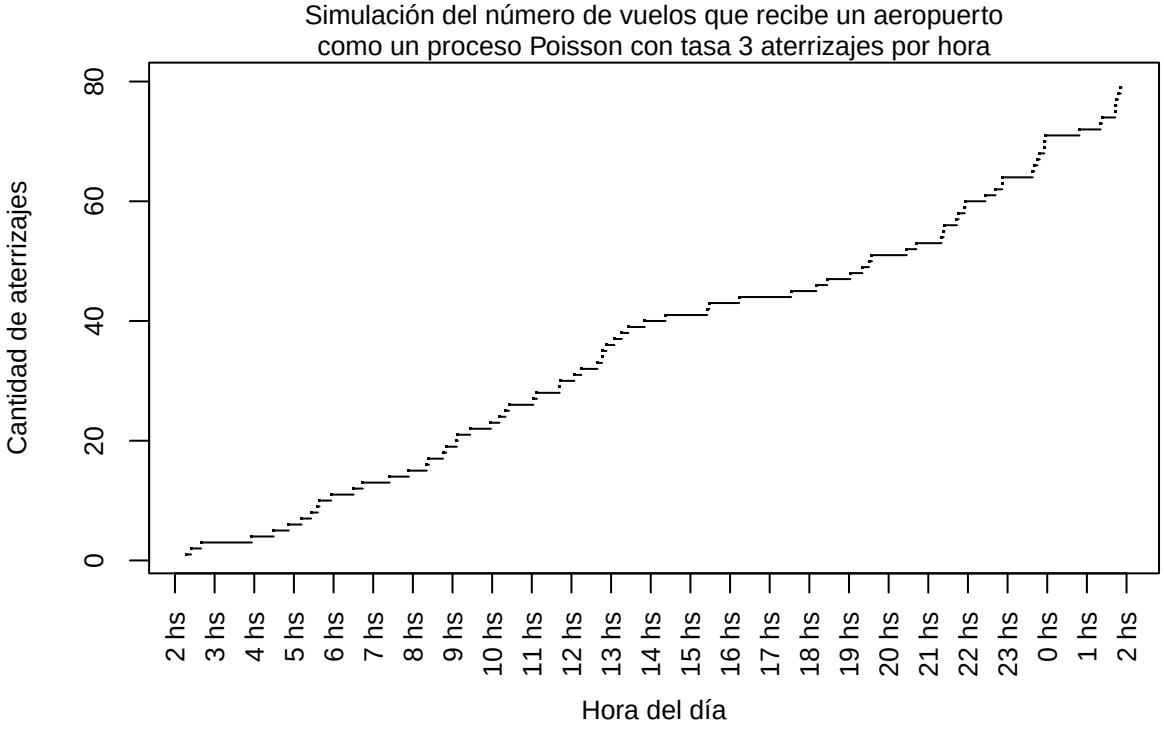
```
##            a         b          c          d         e         f          g
## [1,] 0.22198 0.1527164 0.07125255 0.08425917 0.1223244 0.2934906 0.05397684
```

Se ve que la página que tiene el mayor rango es la f, que es aproximadamente 0.29, y que también es la que más arcos de entrada tiene.

6.

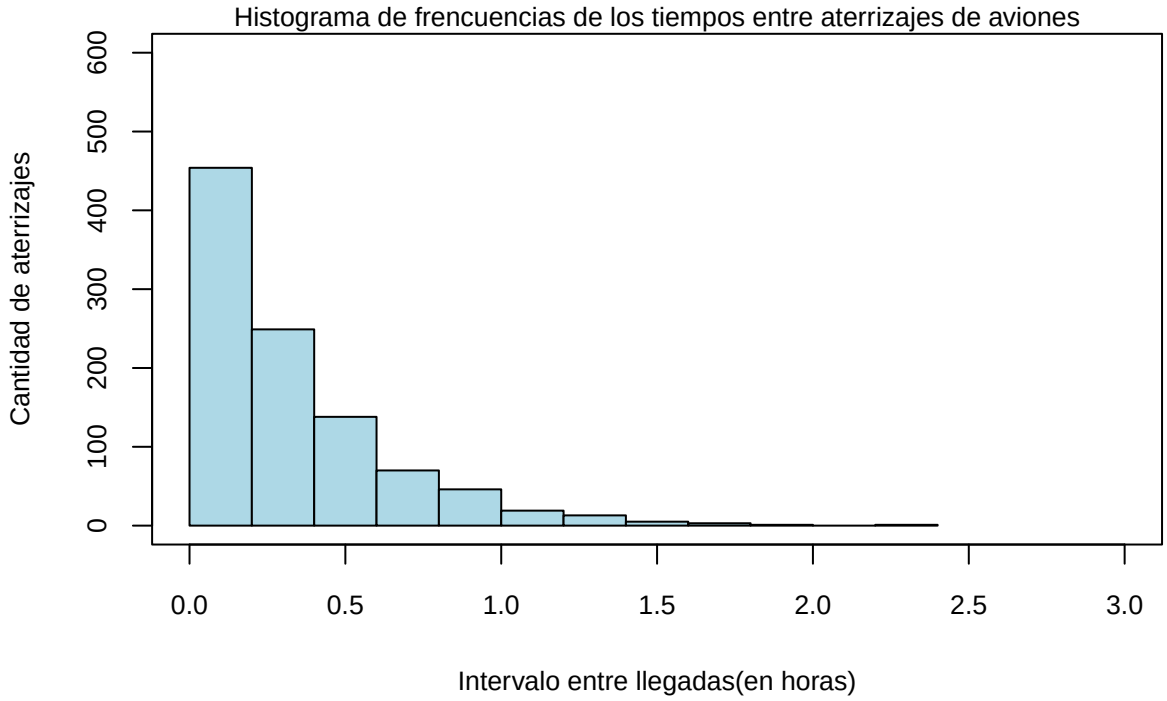
a.

Se utiliza la función *hpp.event.times* del paquete *poisson* para simular el comportamiento del proceso.



b.

Se simulan 1000 arribos del proceso de Poisson, y se calcula el tiempo entre observaciones.



El histograma corresponde a una distribución exponencial, que es el modelo de la variable del tiempo entre arribos en un proceso de Poisson.

7.

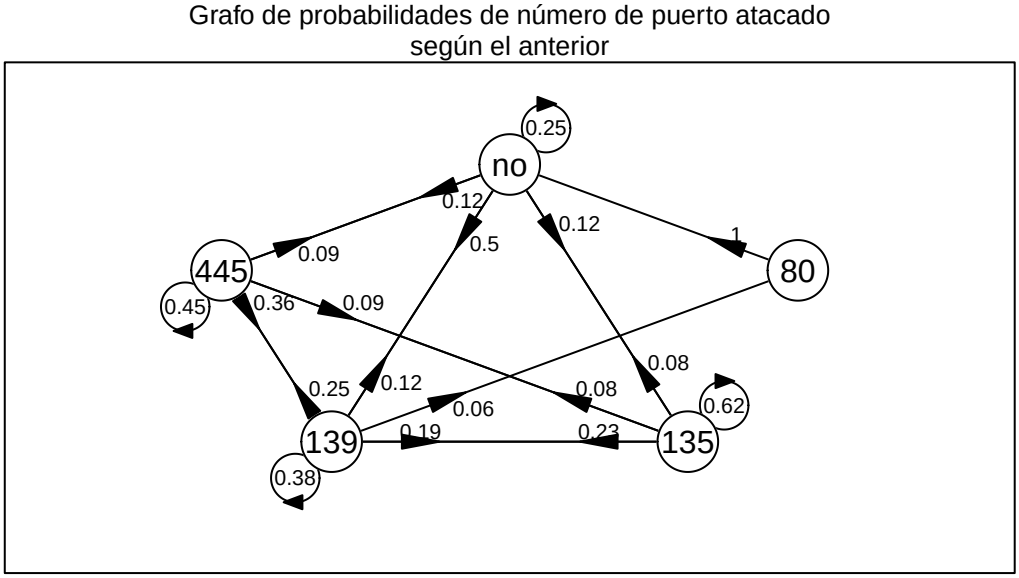
a.

Para conocer con qué probabilidad serán atacados los puertos al cabo de tres semanas se calcula la matriz de transición en tres pasos y se la multiplica por la distribución inicial, es decir, $\pi_0 P^3$.

```
##          80    135    139    445 no attack
## [1,] 0.0242 0.2423 0.3482 0.2326    0.1527
```

El puerto con más probabilidad de ser atacado (0.3482) es el 139, y el que tiene menor probabilidad (0.0241) es el 80.

b.



Para encontrar la distribución límite se busca la solución al sistema de ecuaciones:

$$\begin{cases} \pi P = \pi \\ \sum_{i \in E} \pi(i) = 1 \end{cases}$$

Como la matriz es irreducible, hay un único conjunto cerrado. Por lo tanto si existe la distribución será única para toda la cadena.

$$\pi P = \pi \iff \pi(P - I) = 0 \iff (P - I)^T \pi^T = 0^T$$

De este sistema se quita la última ecuación y se agrega la correspondiente a la condición $\sum_{i \in E} \pi(i) = 1$. El sistema a resolver queda:

$$\begin{pmatrix} -1 & 0 & \frac{1}{16} & 0 & 0 \\ 0 & -\frac{5}{13} & \frac{3}{16} & \frac{1}{11} & \frac{1}{8} \\ 0 & \frac{3}{13} & -\frac{5}{8} & \frac{4}{11} & \frac{1}{2} \\ 0 & \frac{1}{13} & \frac{1}{4} & -\frac{6}{11} & \frac{1}{8} \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} \pi(0) \\ \pi(1) \\ \pi(2) \\ \pi(3) \\ \pi(4) \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Se usa la función *solve* para resolverlo:

```
round(solve(rbind((t(M7a)-diag(5))[1:4,],c(1,1,1,1,1)),c(0,0,0,0,1)),4)
```

```
## [1] 0.0215 0.2669 0.3435 0.2273 0.1408
```

Verificando con la función *steadyStates* del paquete *markovchain*:

```
round(steadyStates(MC7a),4)
```

```
##          80    135    139    445 no attack
## [1,] 0.0215 0.2669 0.3435 0.2273    0.1408
```