Simulación: Tarea 4

Luis Gerardo Martínez Valdés Emiliano Pizaña Vega Fausto Membrillo Fuentes

Otoño 2021

.

Pregunta 1

(a)

Sean X_{1n}, X_{2n} = # componentes en el armario 1 y 2 al tiempo n, respectivamente (ó después de n horas)

$n \in \{0, 1, 2, ...\}$

$X_{in} \in \{0, 1, 2\} \quad \forall n \; \forall i$

Defino $X_n = (X_{1n}, X_{2n})$

Veamos que $\{X_n | n \in \{0, 1, ...\}\}$ es cadena de Markov y sus posibles estados

S = espacio de estados = $\{(0,0), (0,1), (0,2), (1,0), (2,0)\}$ ó solo $\{1, 2, 3, 4, 5\}$

Matriz de prob. de transición:

$$P = \begin{pmatrix} 0.5 & 0.35 & 0.15 & 0 & 0 \\ 0.15 & 0.5 & 0 & 0 & 0.35 \\ 0.35 & 0 & 0.5 & 0.15 & 0 \\ 0 & 0 & 0.5 & 0.5 & 0 \\ 0 & 0.3 & 0 & 0 & 7 \end{pmatrix}$$

tiene entradas P_{ij} donde

$P_{1,1} = a_1 a_2 + (1-a_1)(1-a_2)$
$= (0.3)(0.5) + (0.7)(0.5) = .5$

$p_{12} = (1-a_1)a_2 = .35$

$p_{1,3} = a_1(1-a_2) = .15$
$p_{2,1} = a_1(1-a_2) = .15$
$p_{2,2} = p_{1,1} = .5$
$p_{2,5} = (1-a_1)a_2 = .35$
$p_{3,3} = a_2$
$p_{3,4} = 1-a_2$
$p_{4,2} = a_1$
$p_{4,5} = 1-a_1$

II.
Con $a_1 = .3$ $a_2 = .5$

En R generé 500 pasos de la cadena, con cond iniciales $X_0 = (0,0)$ y semilla 1234

Las proporciones de tiempo apagadas son la proporción que pasa en $(0,2)$ ó $(0,2)$.

(1.94% y 41.46%)

```
#Ejercicio simular cadena de markov donde Xn=(X1n,X2n).
#Xin = num componentes en el armario i al tiempo n (o despues de n horas)

#Espacio de estados S = {(0,0), (0,1), (0,2), (1,0), (2,0) }.
#La matriz de probabilidades de transicion entre esos 5 estados es
#P = [.5 .35 .15 0 0; .15 .5 0 0 .35; .35 0 .5 .15 0; 0 0 .5 .5 0; 0 .3 0 0 .7]

set.seed = 1234  #para poder replicar los resultados, ya que dependen de
#numeros aleatorios
probs = c(.5,.35,.15,0,0,.15,.5,0,0,.35,.35,0,.5,.15,0,0,0,.5,.5,0,0,.3,0,0,.7)
P <- matrix(data=probs,nrow=5,ncol=5,byrow=TRUE)

#------------------------------------
#funcion auxiliar para ver el estado estacionario de la cadena de markov
matrizPotencia <-function(P,pot){
  resultado=P
  for (j in 2:pot){
    resultado=resultado%*%P
  }
  matrizPotencia = resultado
}
###################################
```

```r
numPasos = 500
pasoActual = 0
estadoActual=c(0,0)
numEdoAct = 1
print(numEdoAct)
```

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

```r
listaEstadosVisito = rep(0,numPasos+1)
listaEstadosVisito[1] = 1
for (pasoActual in 1:numPasos){
  #numEdoAct = numEstadoActual(estadoActual[1],estadoActual[2])
  #ver si estoy en el estado 1,2,3,4 o 5
  #print(numEdoAct)
  numEdoAct = sample(1:5, 1, prob = P[numEdoAct,1:5])
  #print(numEdoAct)
  listaEstadosVisito[pasoActual] = numEdoAct
}
print(listaEstadosVisito)
```

```
##   [1] 2 2 5 5 2 5 5 5 5 2 5 2 2 1 1 1 3 3 4 3 3 1 1 2 1 1 2 2 5 5 5 5 5 5 5 5 5
##  [38] 2 2 5 2 2 2 2 2 1 2 2 5 5 2 2 1 1 2 5 5 5 5 5 5 5 5 5 5 5 5 2 2 5 5 2 5 5
##  [75] 5 5 2 2 2 2 1 3 3 3 1 1 1 1 2 2 5 5 2 5 5 2 5 5 5 5 2 2 2 5 5 5 2 1 1 1 1
## [112] 2 1 2 2 1 2 5 5 2 5 2 5 5 5 2 5 2 2 5 5 5 5 5 5 5 2 2 5 5 5 2 2 2 5 5 5 5
## [149] 2 5 5 5 5 5 5 5 2 5 5 2 5 5 5 5 5 2 5 5 2 5 5 5 2 5 5 5 2 2 5 2 2 5 5 2 2
## [186] 2 5 5 5 5 2 2 5 5 5 5 5 5 2 5 5 2 2 5 2 2 2 5 5 2 1 1 2 5 5 5 2 2 1 2 2 2
## [223] 2 1 2 5 5 5 5 5 2 5 2 2 1 1 2 5 5 5 5 5 2 2 5 5 2 2 5 2 2 1 1 3 4 4 3 1 2
## [260] 1 1 1 2 1 1 1 1 2 1 1 1 1 1 1 1 2 2 2 2 5 5 5 2 5 5 2 2 2 2 2 2 2 5 5 2 2
## [297] 5 5 5 5 2 5 5 5 5 5 5 5 5 5 5 2 5 5 5 5 5 2 5 5 5 5 5 2 1 1 1 1 2 2 2 5 5
## [334] 5 5 5 5 5 2 1 2 5 2 2 5 5 5 5 5 5 2 1 1 3 1 2 1 1 1 1 2 2 2 5 2 5 2 1 1 2
## [371] 5 2 2 5 5 2 2 2 2 2 2 2 2 5 5 5 2 5 5 5 5 5 2 2 5 2 5 2 2 5 2 5 2 1 2 2 2
## [408] 2 5 5 2 2 2 5 5 5 2 1 2 2 1 2 1 2 2 2 1 1 2 5 5 5 5 5 2 5 5 5 5 5 5 5 5 5
## [445] 5 5 5 5 2 1 1 3 1 1 1 2 5 5 5 5 5 2 2 5 5 5 5 2 5 5 5 2 5 5 2 5 5 2 1 1 1
## [482] 3 3 1 2 2 2 5 2 5 5 5 5 5 5 5 5 2 5 5 0
```

```r
proporcionApagadaMaq1 = sum(listaEstadosVisito==4)/numPasos
#proporcion de tiempo que esta en estado 4 ( X=(2,0) )

proporcionApagadaMaq2 = sum(listaEstadosVisito==5)/numPasos
#proporcion de tiempo que esta en estado 5 ( X=(0,2) )

print(proporcionApagadaMaq1)
```

```
## [1] 0.006
```

```r
print(proporcionApagadaMaq2)
```

```
## [1] 0.476
```

```r
#salieron 0.002 y 0.484
#ESAS SON PARA ESTA MUESTRA, LAS PROPORCIONES EXACTAS SE OBTIENEN CON EL VECTOR
#DE ESTADO ESTACIONARIO (sale de resolver (pi1,pi2,...,pi5)=(pi1,pi2,...,pi5)*P junto con pi1+...+pi5=1
#IGUAL SE PUEDE OBTENER ELEVANDO LA MATRIZ P A UNA POTENCIA GRANDE Y VIENDO SUS COLUMNAS:
PotenciadeP = matrizPotencia(P,100)   #estoy calculando P^100
print(PotenciadeP)
```

```
##            [,1]      [,2]       [,3]       [,4]      [,5]
```

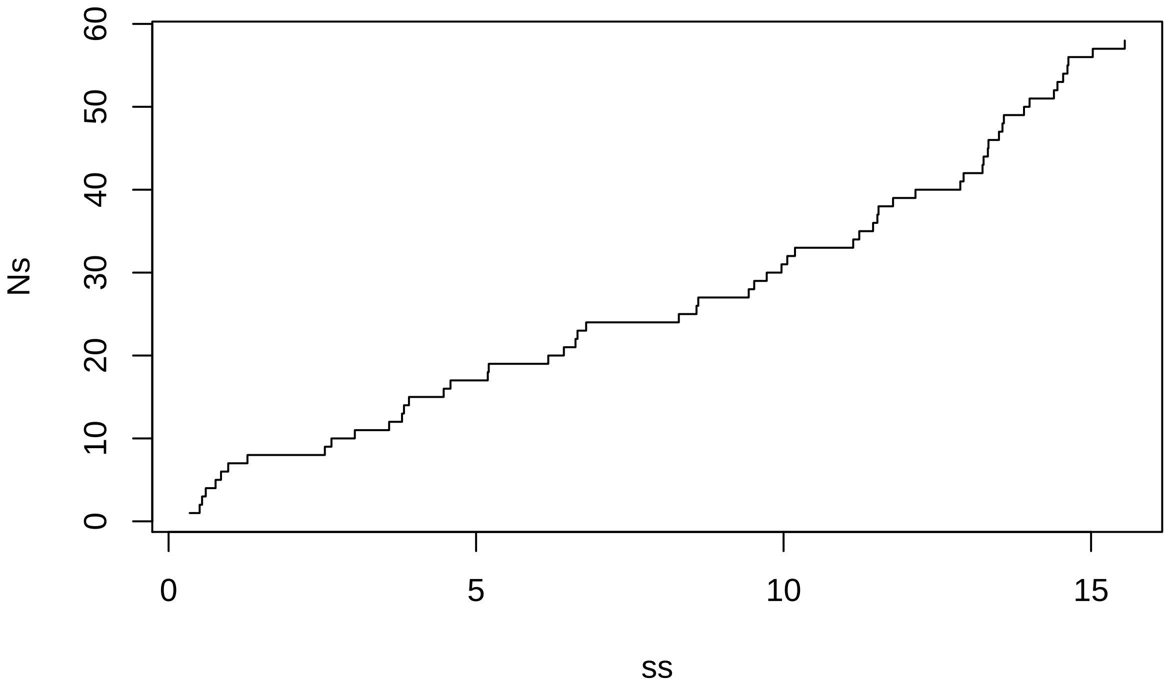
```
## [1,] 0.1512242 0.3528565 0.06481037 0.01944311 0.4116659
## [2,] 0.1512242 0.3528565 0.06481037 0.01944311 0.4116659
## [3,] 0.1512242 0.3528565 0.06481038 0.01944311 0.4116659
## [4,] 0.1512242 0.3528564 0.06481038 0.01944312 0.4116658
## [5,] 0.1512242 0.3528565 0.06481037 0.01944311 0.4116659
```

```
#ahi se ve (columnas 4 y 5) que las probabilidades (proporciones)
#son cercanas a 0.0194 y 0.4116  (1.94% y 41.16%)
```

Pregunta 2

##(a)

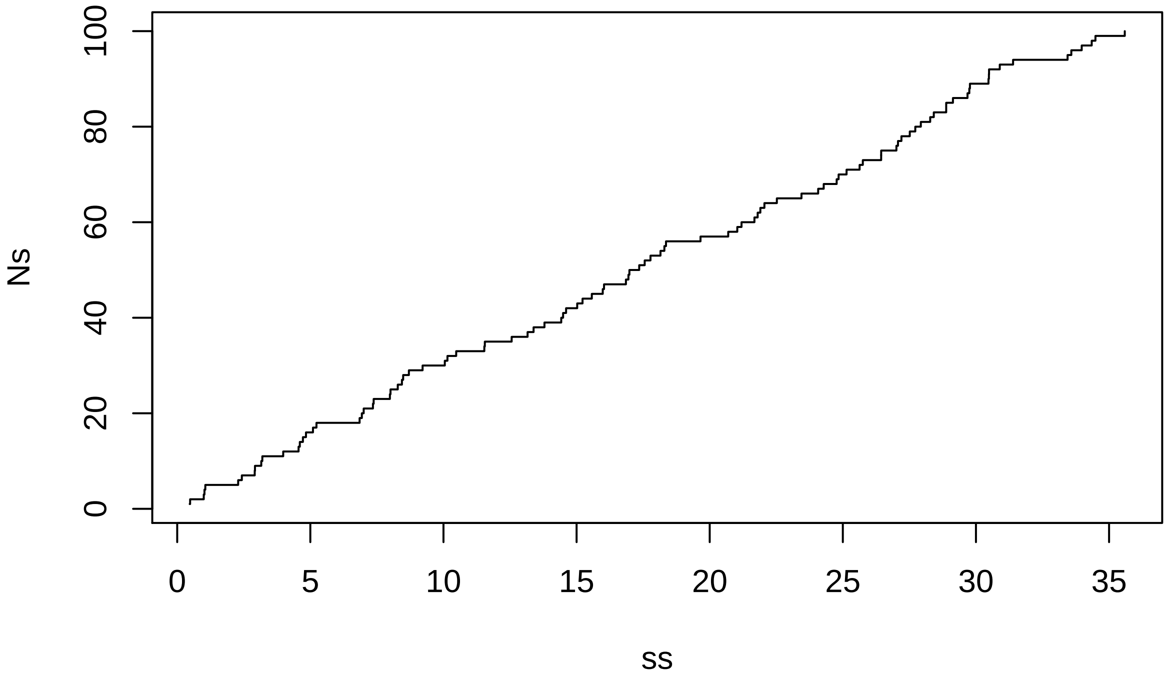
```
lambdat <- function(t, T0){
  for(i in 0:T0){
    if(t <= ((2*i) + 1) & t > 2*i) {
      z <- 3
      return(z)
    }
    else
      z <- 5
  }
  return(z)
}

p.nohomogeneo <- function(lambdat,n){

  lambda <- 5 #constante que mayoriza
  TT <- rexp(n,lambda)
  s <- cumsum(TT)
  u <- runif(n)

  ss <- s[u <= lambdat(s,n)/lambda]
  Ns <- 1:length(ss)
  plot(ss, Ns, type = "s")
  return(list(epocas = ss, cuenta= Ns))
}
x <- p.nohomogeneo(lambdat,100)
```



(b)

```
lambdat <- function(t, T0){
  for(i in 0:T0){
    if(t <= ((2*i) + 1) & t > 2*i) {
      z <- 3
      return(z)
    }
    else
      z <- 5
  }
  return(z)
}

p.nohomogeneo <- function(lambdat){

  lambda <- 5 #constante que mayoriza
  i <- 1
  ss <- NULL
  while(length(ss) != 100)
  {
    TT <- rexp(i,lambda)
    s <- cumsum(TT)
    u <- runif(i)

    ss <- s[u <= lambdat(s,i)/lambda]
    i <- i + 1
  }
  Ns <- 1:length(ss)
  plot(ss, Ns, type = "s")
  return(list(epocas = ss, cuenta= Ns))
```

```r
}

x <- p.nohomogeneo(lambdat)
```



(c)

```r
ppois(2,6.75,lower.tail = FALSE) #lambda nos queda como 6.75
```

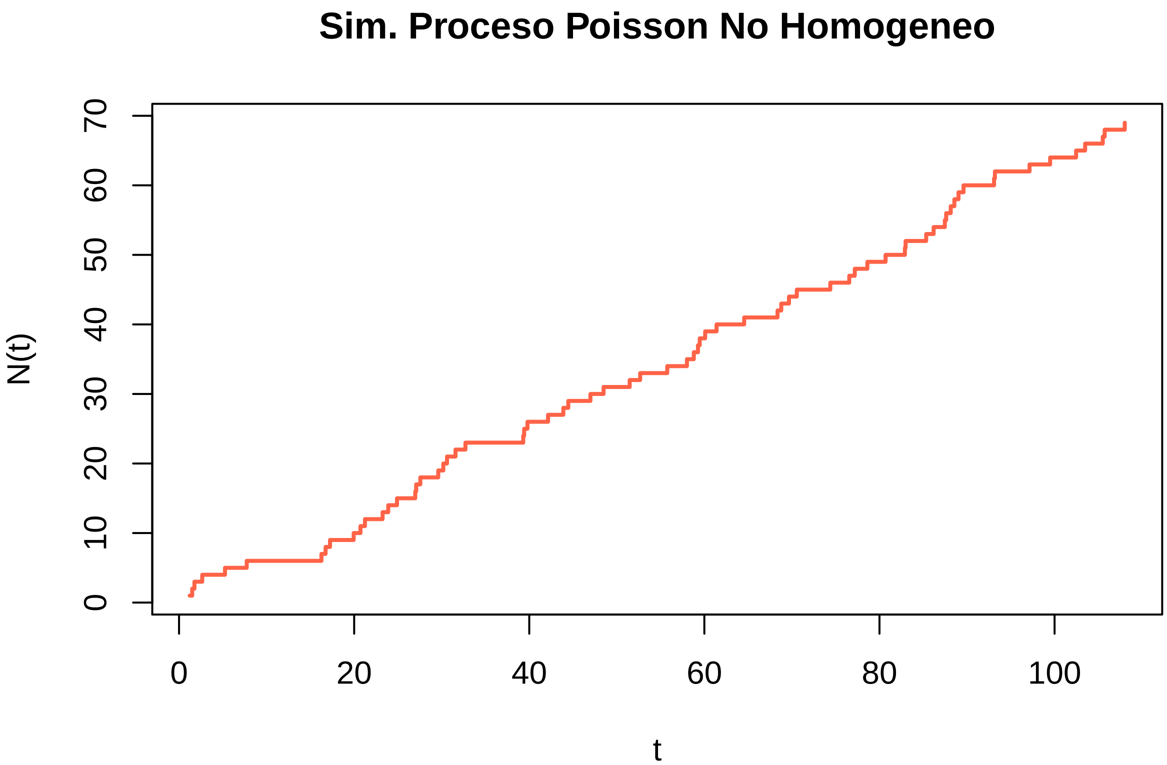
```
## [1] 0.9642516
```

Pregunta 3

Queremos simular un Proceso Poisson no-homogéneo con función de intensidad $\lambda = |sen(t)|$

```r
#Creamos funcion intencisidad
lambda.t<- function(t){abs(sin(t))}


# Creamos el proceso Poisson N-Hom
ppoisson.nh<- function(lambda.t,n){
  lambda<- 1 #Aqui mayoritamos la funcion lambda.t
  T.exp<- rexp(n, lambda) #Generamos va's exponenciales
  tiempos<- cumsum(T.exp) #Suma Acumulativa de tiempos
  u<- runif(n)  #Generamos va's uniformes

  #Verificamos los tiempos que complen la condicion de aceptacion
  tiempos.a<- tiempos[u <= lambda.t(tiempos)/lambda]
  N.t<- 1:length(tiempos.a) #Contador

  #Graficamos
  plot(tiempos.a, N.t, type = 's',col='tomato',lwd=2, xlab = 't', ylab = 'N(t)',
      main= "Sim. Proceso Poisson No Homogeneo")
```

```r
  return(list(t=tiempos.a, contador= N.t))

}

Nt<- ppoisson.nh(lambda.t,100)
```



Pregunta 4

Datos:

- $N_1(t) \sim \ Poiss(3/día)$
- $N_2(t) \sim \ Poiss(4/día)$
- Máquina shock 1 falle: $p_1 = 0.011$
- Máquina shock 2 falle: $p_2 = 0.005$
- $N(t) \equiv \#$ de reemplazos de la máquina sobre el intervalo $(0, t]$

(a)

Sabemos que la distribución Poisson tiene la propiedad aditiva sobre su tasa. Entonces:

$N_1(t) + N_2(t) \sim \ Poiss(3 + 4 = 7/día) = Poiss\left(\frac{7}{3} \ por \ turno \ de \ 8 \ hrs.\right)$

Por lo tanto,

$\mathbb{E}[N_1(t) + N_2(t)] = \frac{7}{3}$; $Var[N_1(t) + N_2(t)] = \frac{7}{3} \ hrs.$

(b)

Sabemos que el promedio de shocks en un día es $3+4=7$, por lo que el promedio de shocks en una hora es $7/24$. Definimos $X \sim \ Poiss(\frac{7}{24})$. Así,

$$\mathbb{P}(X=2) = \frac{e^{-.292} * .292^2}{2!} \approx 0.03183624$$

(c)

(c)

$N(t) \xrightarrow[\text{shock 1}]{.011} N_1(t) \sim Poiss(.011(3)t) = Poiss(.033t)$

$N(t) \xrightarrow[.005]{\text{shock 2}} N_2(t) \sim Poiss(.005(4)t) = Poiss(.02t)$

$$\begin{aligned} E[N(t)] = E[N_1(t) + N_2(t)] &= E[N_1(t)] + E[N_2(t)] \\ &= .033t + .02t \\ &= .053t \end{aligned}$$

$\therefore N(t) \sim Poiss(.053t)$ con parámetro $.053t$

Pregunta 5

Sea $T \equiv$número de eventos en un periodo de tiempo.

$$\mathbb{P}(T \geq 2) = 0.28 \iff \mathbb{P}(T < 2) = \mathbb{P}(T=0) + \mathbb{P}(T=1) = 0.72$$

Así,

$$\frac{e^{-\lambda}\lambda^0}{0!} + \frac{e^{-\lambda}\lambda^1}{1!} = e^{-\lambda} + \lambda e^{-\lambda} = (1+\lambda)e^{-\lambda} = 0.72$$

Resolviendo la eqcuación se tiene que $\lambda = 1.04285$ ya que por definición $\lambda > 0$.

Pregunta 6

Primero para cada instante t tenemos que $X(t)$ es el número de clientes que ya fueron atendidos y $Y(t)$ el número de clientes que se estan atendiendo en el momento t. Queremos calcular la esperanza y varizna de $X(t)$ para poder determinar su distrubución. Por lo que tenemos que:

$$E(x(t)) = 5\int_0^t e^{(-x/40)}\,dx \ = 5 * 40(1-e^{(-t/40)})$$

Al tomar $t = 120$ tenemos:

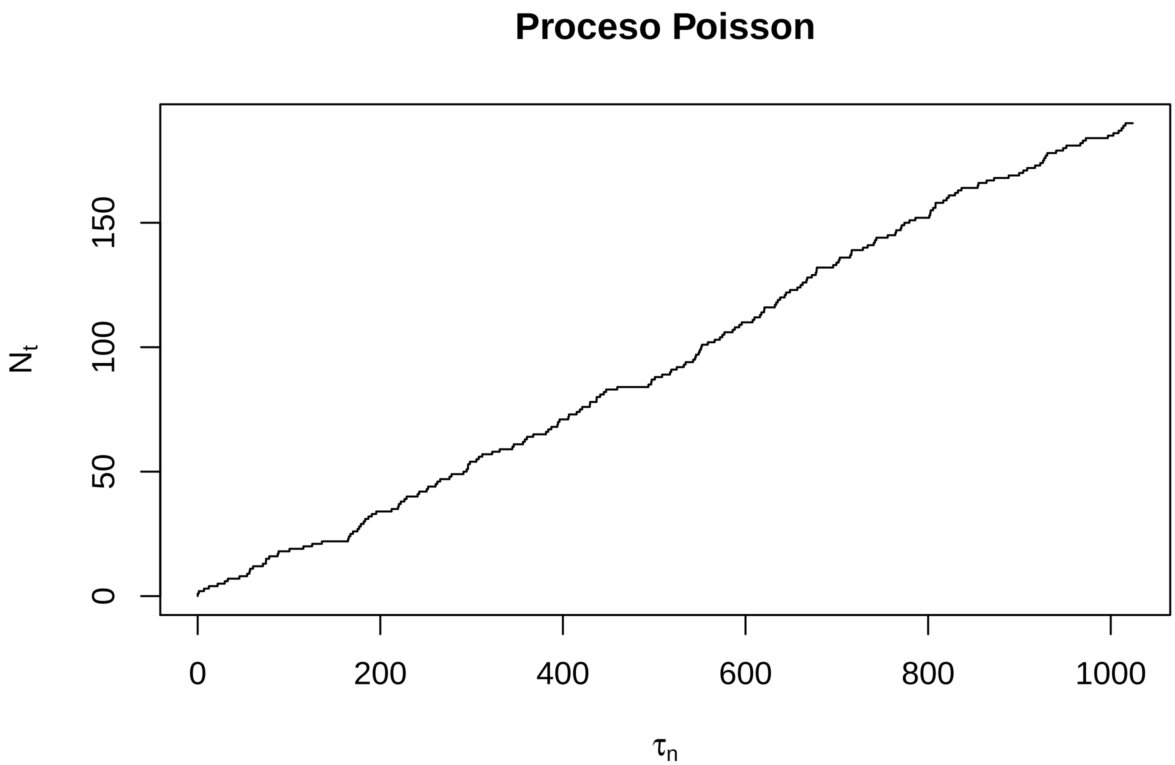
$$40(1-e^{(-t/40)}) = 5 * 38 = 190$$

Y por otro lado tenemos

$$V(x(t)) = 190$$

Por lo tanto tenemos que $X \sim Poiss(190)$

```
n <- 190
lambda = 5
TA <- rexp(n,rate=1/lambda)
tau <- cumsum(TA) #tiempos en donde ocurren los eventos
Nt <- 1:length(tau)
plot(c(0,tau), c(0,Nt), type="S",main="Proceso Poisson",
     xlab = expression(tau[n]), ylab=expression(N[t]))
```



Pregunta 7

(a)

1 $dS = a(S,t)\,dt + b(S,t)\,dt$

$$dG = \left(\frac{\partial G}{\partial S} \cdot a + \frac{\partial G}{\partial t} + \frac{1}{2} \frac{\partial^2 G}{\partial S^2} b^2 \right) dt + \frac{\partial G}{\partial S}\, dz$$

$$dX_t = -\frac{1}{3}\, dt + \frac{1}{2}\, dz$$

$S_t = e^{X_t}$ $\qquad a = 1/3 \quad b = 1/2$

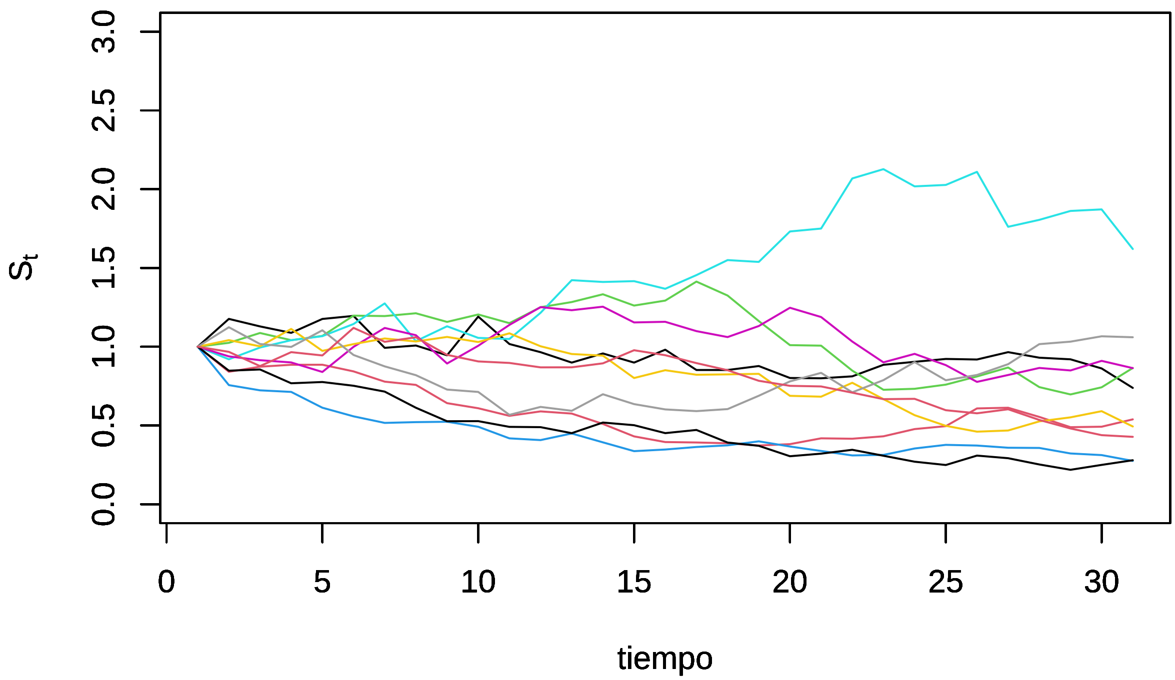
$$dS_t = \left[e^x \left(-\frac{1}{3}\right) + \frac{1}{2} e^x \left(\frac{1}{4}\right) \right] dt + e^x \left(\frac{1}{2}\right) dz$$

$$dS_t = \left[-\frac{1}{3} S_t + \frac{1}{8} S_t \right] dt + \frac{1}{2} S_t\, dz$$

$$dS_t = \frac{-5}{24} S_t\, dt + \frac{1}{2} S_t\, dz$$

(b) y (c)

```r
options(width = 150, digits = 3)
BGeo <- function(n, TT, a, b, S0 = 100){
  #Función para generar un proceso Browniano Geométrico
  #n es el número de puntos de partición del intervalo [0,TT]
  #a es el drift y b la volatilidad
  dt <- TT/n #incremento de los intervalos para cubrir [0,TT]
  S <- S0 #valor inicial
  for(i in 2:(n+1)){
    S <- append(S, S[i-1]*exp((a-b^2/2)*dt + b*sqrt(dt)*rnorm(1)))
  }
  return(S)
}
suma=0
for (i in 1:10){
par(new=TRUE)
bg<-BGeo(30, 1, -5/24, 0.5, 1)
plot(bg, type = "l",col= i, ylim = c(0,3),
     xlab = "tiempo", ylab = expression(S[t]))
suma=suma+bg[30]
}
```




```
promedio<-suma/i
# Se puede concluir que St tiene una distribución lognormal para t grande y fija
```

(d) y (e)

```
options(width = 150, digits = 3)
BGeo <- function(n, TT, a, b, S0 = 100){
  #Función para generar un proceso Browniano Geométrico
  #n es el número de puntos de partición del intervalo [0,TT]
  #a es el drift y b la volatilidad
  dt <- TT/n #incremento de los intervalos para cubrir [0,TT]
  S <- S0 #valor inicial
  for(i in 2:(n+1)){
    S <- append(S, S[i-1]*exp((a-b^2/2)*dt + b*sqrt(dt)*rnorm(1)))
  }
  return(S)
}
suma=0
for (i in 1:100){
  par(new=TRUE)
  bg<-BGeo(30, 1, -5/24, 0.5, 1)
  #plot(bg, type = "l",col= i, ylim = c(0,3),
      # xlab = "tiempo", ylab = expression(S[t]))
  suma=suma+bg[30]
}
promedio100<-suma/i
suma1=0
for (i in 1:1000){
  par(new=TRUE)
  bg1<-BGeo(30, 1, -5/24, 0.5, 1)
  #plot(bg, type = "l",col= i, ylim = c(0,3),
  # xlab = "tiempo", ylab = expression(S[t]))
```

```
  suma1=suma1+bg1[30]
}
promedio1000<-suma1/i
#Podemos decir con claridad que cuando n tiende a infinito el promedio tiende
#a la media de una distribución lognormal
```

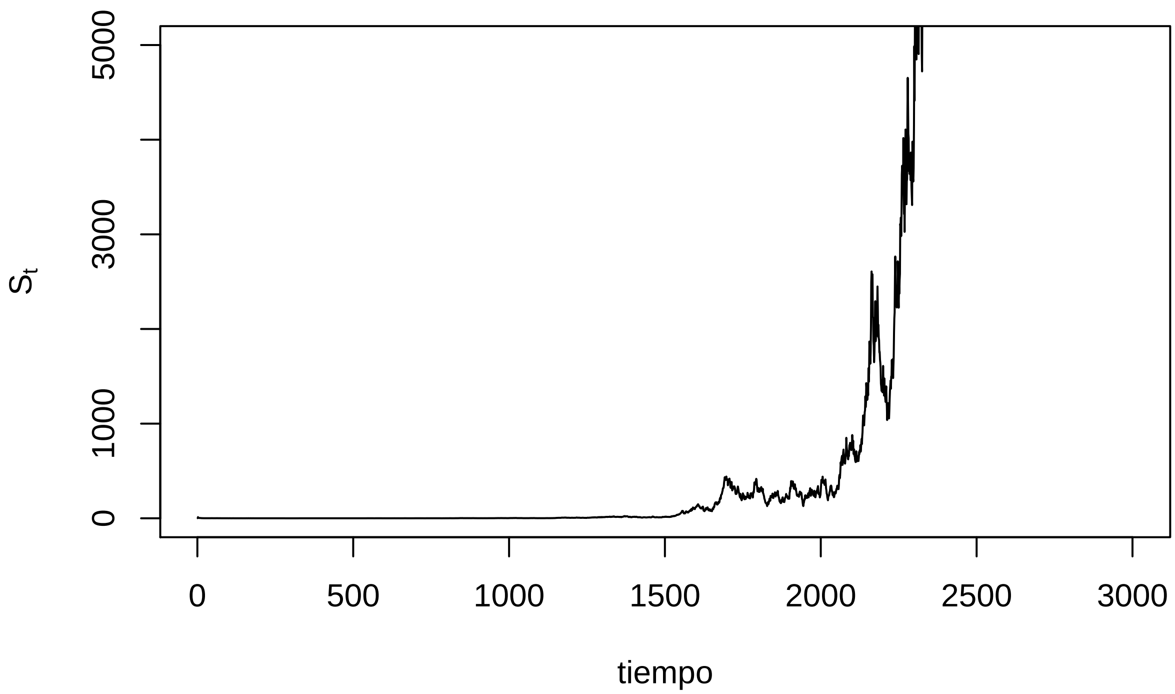
Pregunta 8

(a)

```
WienerGeo <- function(n,TT,a,b,S0){
  dt <- TT / n
  S <- S0
  for(i in 2:(n+1)){
    S <- append(S, S[i-1]*exp((a-b^2/2)*dt + b*sqrt(dt)*rnorm(1))) }
  return(S)
}
WienerGeo(5000,5000/12, .10, .30 ,1) #Muestra
```

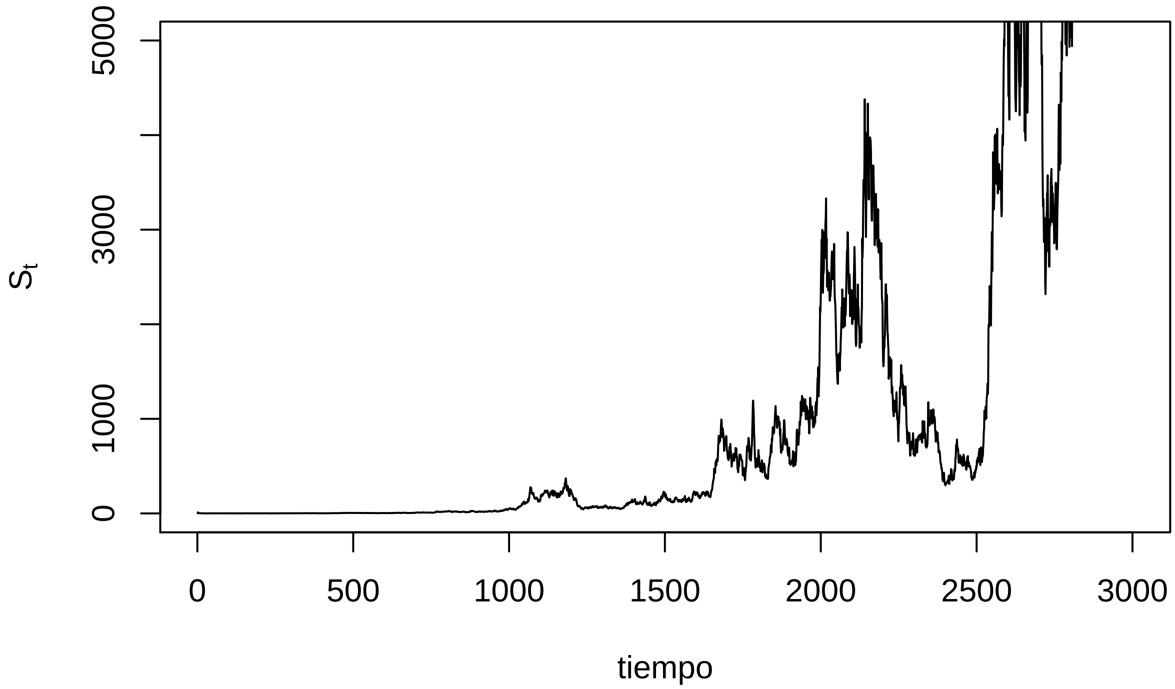
(b)

```
WienerGeo_mod <- function(n,TT,a,b,S0){
  dt <- TT / n
  S <- S0
  for(i in 2:(n+1)){
    k <- (S[i-1]*exp((a-b^2/2)*dt + b*sqrt(dt)*rnorm(1)))
    S <- append(S, k)
  }
  for(i in 2:(n+1)){
    t <- (i-1)*dt
    S[i] <- S[i]/t
  }

  return(S)
}

X <- WienerGeo_mod(3000,3000/12, .10, .30 ,1)

plot(X, type = "l", ylim = c(0,5000), xlab = "tiempo", ylab = expression(S[t]))
```



(c)

```
WienerGeo_mod1 <- function(n,TT,a,b,S0){
  dt <- TT / n
  S <- S0
  for(i in 2:(n+1)){
    k <- (S[i-1]*exp((a-b^2/2)*dt + b*sqrt(dt)*rnorm(1)))
    S <- append(S, k)
  }
  q <- max(S)
  for(i in 2:(n+1)){
    t <- (i-1)*dt
    S[i] <- ((S[i] - q*t)^2)/t
  }

  return(S)
}

X <- WienerGeo_mod(3000,3000/12, .10, .30 ,1)

plot(X, type = "l", ylim = c(0,5000), xlab = "tiempo", ylab = expression(S[t]))
```



S_t
0
1000
3000
5000
0
500
1000
1500
2000
2500
3000
tiempo