Procesos Estocásticos II. Tarea de Programación A

Andrés Limón Cruz

2024-04-24

El objetivo es implementar el proceso de ruina, basado en el modelo de Cramer Lunderberg, pero donde el donde el proceso que cuenta las llegadas de las reclamaciones es un proceso de renovación general.
Para lograrlo se debe implementar lo siguiente:

1

Codigo que simule trayectorias de un proceso de renovación
Inputs: Tiempo final t y parámetros de la distribución de los tiempos de interarribo. (Por ejemplo α y β si $T_n = \Gamma(\alpha, \beta)$).
Outputs: Tiempos de arribo (las W_n)

```r
#Simulación de un proceso de renovación
simulacion_renovacion <- function(t, alpha,beta){
  #Generamos los tiempos de arribo
  W <- c(0,rgamma(t, alpha,beta))
  return(cumsum(W))
}

#Ejemplo
simulacion_renovacion(10, 1,1)
```

```
##  [1]  0.0000000  0.1969958  0.4402853  1.4599108  2.8717127  3.6931832
##  [7]  5.4840104  6.5956245  8.7567049  8.9417811 10.7956967
```

2

Código para simular la trayectoria de un proceso de poisson compuesto $X_t = \sum_1^{N_t} Y_n$.
Inputs: Tiempo final t, intensidad del Poisson λ y parámetros de la distribución de los saltos (por ejemplo α y β si $Y_n = \Gamma(\alpha, \beta)$).
Outputs: Tiempos de salto y tamaño de los saltos

```r
#Simulación de un proceso de Poisson compuesto
simulacion_poisson_compuesto <- function(t, lambda, alpha, beta){
  #Generamos los tiempos de salto
  N <- simulacion_renovacion(t,alpha = 1,beta = lambda)
  #Generamos los saltos
  Y <- simulacion_renovacion(t,alpha,beta)
  return(list(N_t = N, Y = Y))
}

#Ejemplo
simulacion_poisson_compuesto(10, 10, 1, 1)
```

```
## $N_t
##  [1] 0.00000000 0.07100506 0.17036321 0.21505265 0.22711689 0.29464006
##  [7] 0.30325423 0.38791505 0.42412739 0.44195565 0.52863890
##
## $Y
##  [1]  0.000000  1.481692  2.981225  5.018382  5.789832  7.581775  8.429474
##  [8]  9.870387 10.882411 12.206503 14.800887
```

3

Codigo para simular la trayctoria del proceso de Cramer-Lundberg

$$R_t = x + ct - \sum_{n=1}^{N_t} Y_n$$

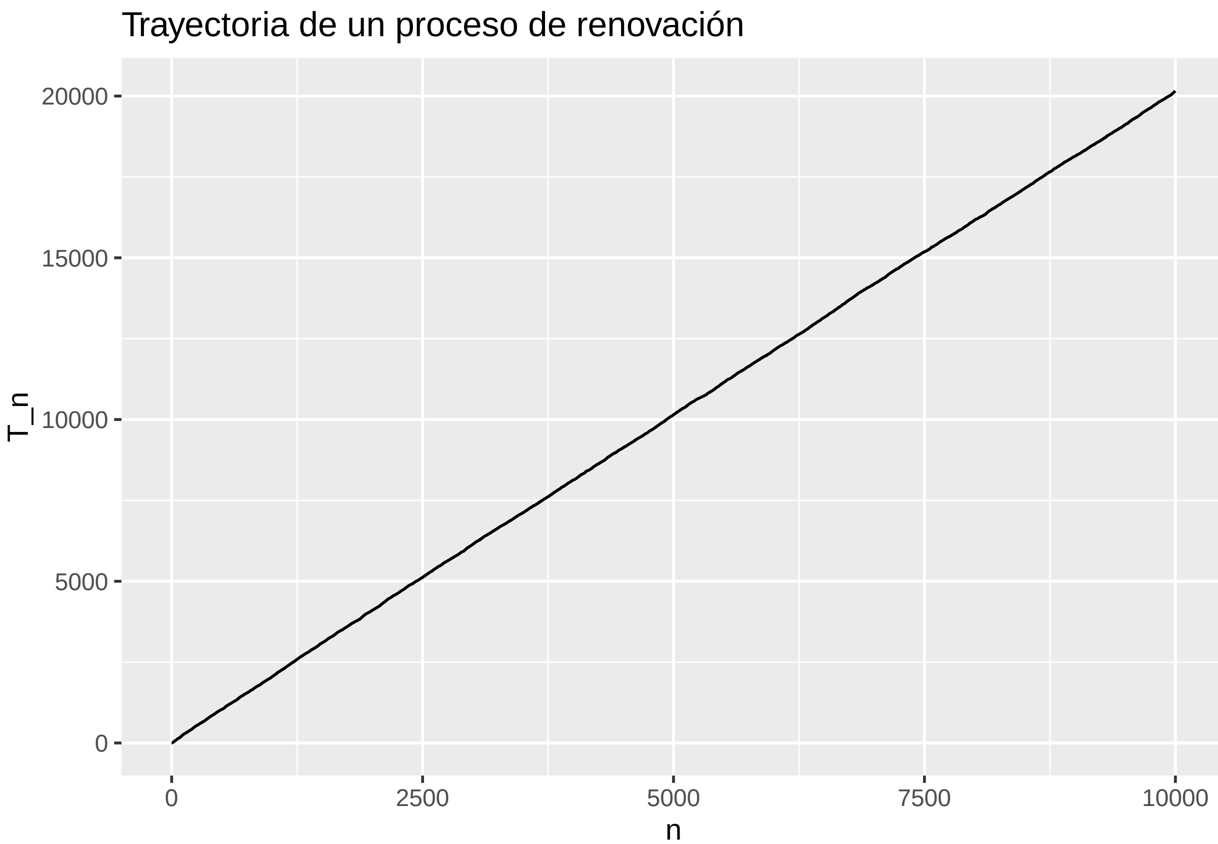
Inputs: Tiempo final t, capital inicial u, prima c, intensidad del poissson λ y parámetros de la distribución de los saltos (por ejemplo α y β si $Y_n = \Gamma(\alpha, \beta)$).
Outputs: Tiempos de salto y tamaño de los saltos

```r
#Simulación de un proceso de Cramer-Lundberg
simulacion_cramer_lundberg <- function(t, u, c, lambda, alpha){
  #Generamos los tiempos de salto
  N <- simulacion_renovacion(t,alpha = 1, beta = lambda)
  #Generamos las perdidas
  Y <- simulacion_renovacion(t,alpha = 1, beta = alpha)
  #Calculamos el proceso

  R <- u+c*N - Y

  return(list(N_t = N,R_t= R))
}

#Ejemplo
simulacion_cramer_lundberg(10, 0, 1, 1, 1)
```

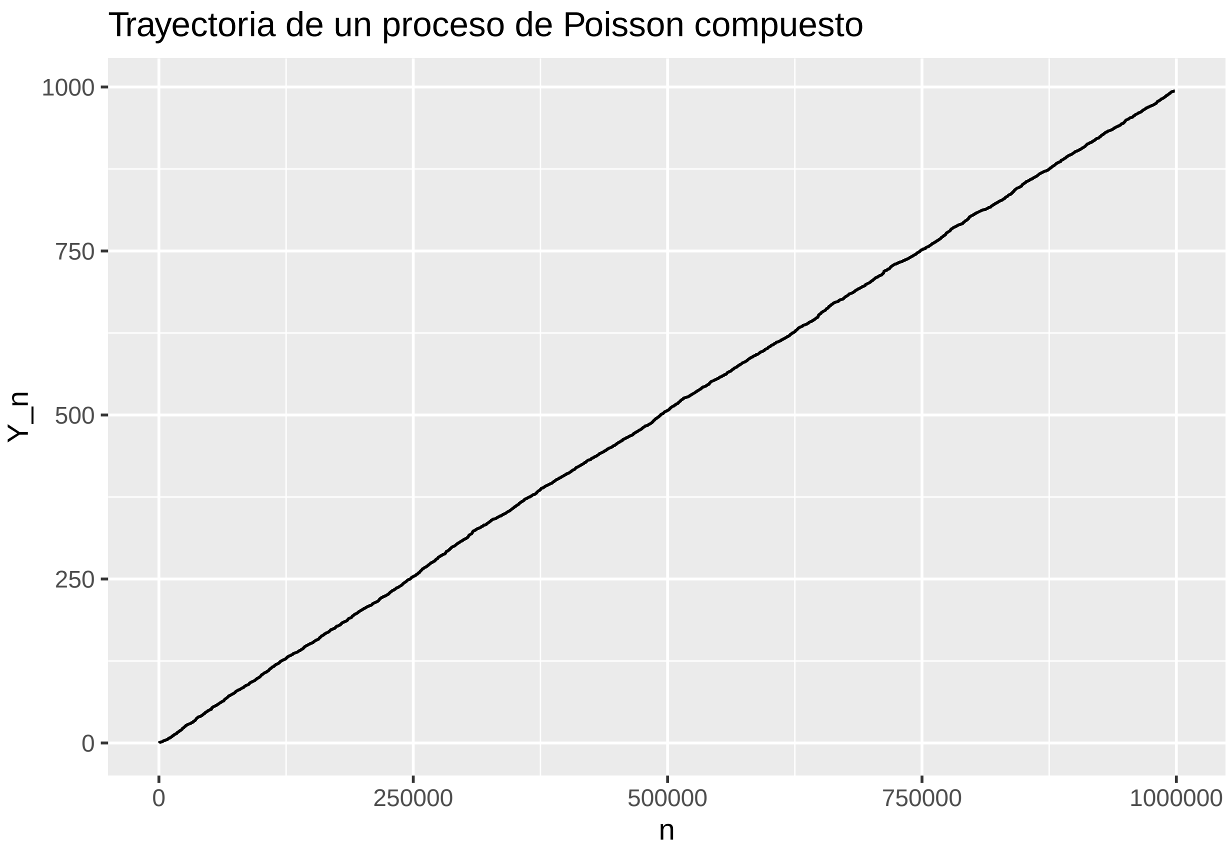
```
## $N_t
##  [1]  0.000000  1.505633  1.907598  3.698265  5.302875  5.474292  6.794559
##  [8]  7.308125  9.222925 11.304614 11.518083
##
## $R_t
##  [1] 0.000000 1.374613 1.335459 2.046708 3.313531 3.483384 4.770098 4.542001
##  [9] 5.568960 7.386112 7.388428
```

Para verificar si los codigos son correctos:
Para 1 y 2 verifiquen que se cumple el teorema elemental de renovación y grafique una trayectoria para cada 1

```r
Simulacion_1 = data.frame(T_llegada = simulacion_renovacion(10000, 2, 1))
ggplot(data = Simulacion_1) +
  geom_line(mapping = aes(x = 0:(length(T_llegada)-1),y = T_llegada)) +
  labs(title = "Trayectoria de un proceso de renovación", x = "n", y = "T_n")
```




```r
Simulacion_2 = data.frame(simulacion_poisson_compuesto(10000, 1/100, 1, 10))
ggplot(data = Simulacion_2) +
  geom_line(mapping = aes(x = N_t,y = Y)) +
  labs(title = "Trayectoria de un proceso de Poisson compuesto", x = "n", y = "Y_n")
```



Entonces veamos que se cumple que $\lim_{t\to\infty} \frac{N_t}{t} = \frac{1}{\mu}$ donde μ es la tasa de llegada. Y se cumple $\lim_{t\to\infty} \frac{\sum_{i=1}^{n} y_i}{t} = \mu\lambda$

```
Simulacion_1$T_llegada[10000]/10000
```

```
## [1] 2.015182
```

```
Simulacion_2$Y[10000]/Simulacion_2$N_t[10000]
```

```
## [1] 0.0009956403
```

Así se verifica que se cumple el teorema elemental de renovación para ambos incisos.

Para el 3, suponga que las reclamaciones son exponenciales $Y_n = \exp(\alpha)$ y compruebe numericamente que la probabilidad de ruina se aproxima a:

$$\psi(u) = \frac{\lambda}{\alpha c} \exp(-(\alpha - \frac{\lambda}{c})u)$$

```
#Simulación
u = 0
c = 2
lambda = 1
alpha = 8
p = 0
for(i in 1:10000){
  if(all(simulacion_cramer_lundberg(t = 1000, u, c, lambda, alpha)$R_t >= 0))
    p = p + 1
```

```
}
1-p/10000
```

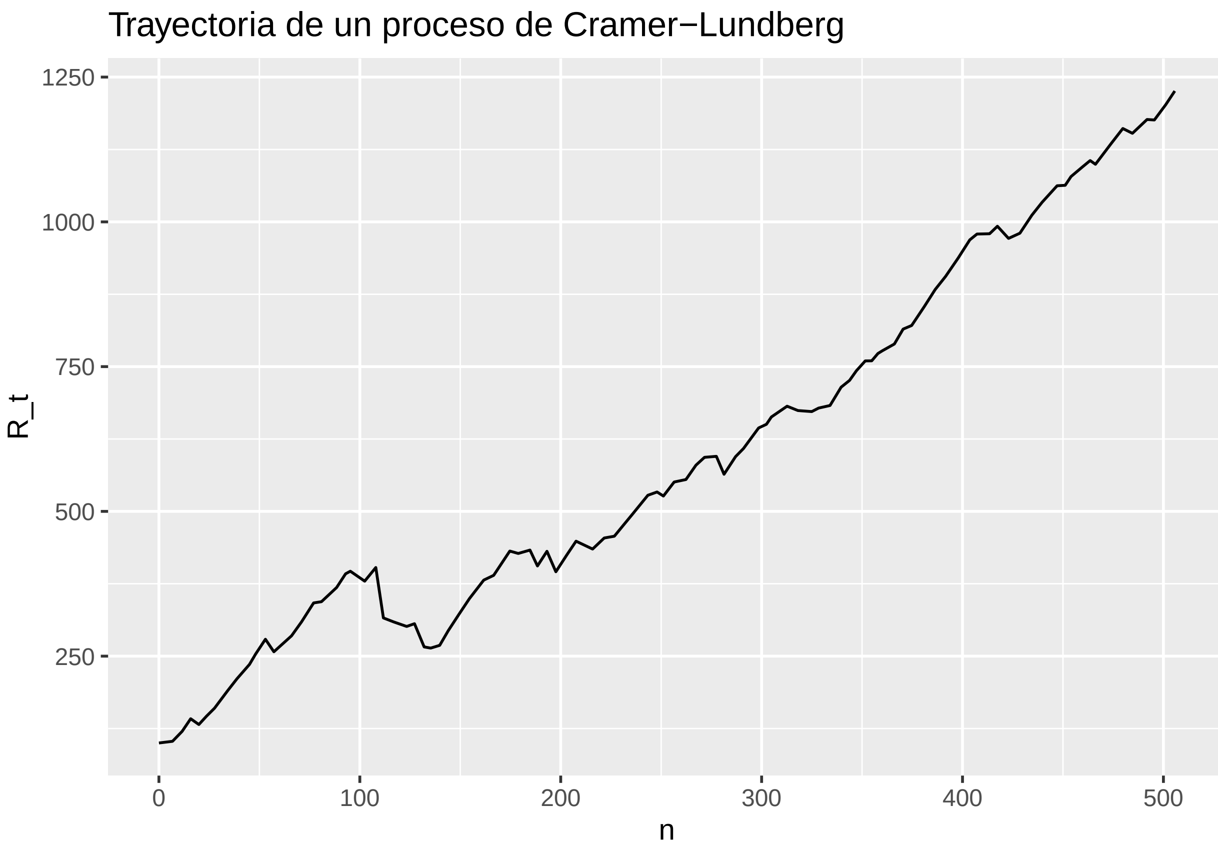
```
## [1] 0.0634
```

```
lambda/(alpha*c) * exp(-(alpha - lambda/c)*u)
```

```
## [1] 0.0625
```

Por ultimo, una vez verificado que todos los codigos anteriores dan buenos resultados. Sustituye el proceso de Poisson compuesto que aparece en el modelo de Cramer Lundberg, por un proceso de renovación general, donde $T_n \sim \Gamma(10, 2)$ y $Y_n \sim \exp(\frac{1}{20})$. Estima la probabilidad de ruina con $u = 100$ y $c = 5$. Concluye si los resultados que obtienes tienen sentido.

```
simulacion_cramer_lundberg <- function(t, u, c, alpha,beta){
  #Generamos los tiempos de salto
  N <- simulacion_renovacion(t,alpha,beta)
  #Generamos los saltos
  Y <- simulacion_renovacion(t,1,1/20)
  #Calculamos el proceso
  R <- u+c*N - Y

  return(list(N_t = N,R_t= R))
}
#Ejemplo
simulacion_3 = data.frame(simulacion_cramer_lundberg(100, 100, 6, 10, 2))
ggplot(data = simulacion_3) +
  geom_line(mapping = aes(x = N_t,y = R_t)) +
  labs(title = "Trayectoria de un proceso de Cramer-Lundberg", x = "n", y = "R_t")
```




```r
#Simulación
p = 0
for(i in 1:10000){
  if(all(simulacion_cramer_lundberg(t = 10000, u = 1000,
                                    c = 5, alpha = 10, beta = 2)$R_t >= 0))
    p = p + 1
}
1-p/10000
```

```
## [1] 0
```

Entonces tenemos una probabilidad de ruina igual a 0, lo cual tiene sentido, pues veamos que en cada tiempo en el que sucede un accidente perderemos en promedio 20, pero también como cada tiempo se distribuye como una $\Gamma(10, 2)$, entonces en promedio cada accidente ocurre cada 5 unidades de tiempo, y como la prima es de 5 por cada unidad de tiempo, terminamos ganando en promedio 25, así por cada suceso ganariamos en promedio 25-20 $=$ 5 unidades, lo cual no es significativamente grande para evitar la ruina, sin embargo el capital inicial nos protege de una mala racha.