UNIVERSIDAD NACIONAL DE COLOMBIA

PREGRADO EN ESTADISTICA

Departamento de Estadística
Facultad de Ciencias

— Actuaria de Contingencias de Vida —

Profesor:
Norman Diego Giraldo Gomez
Tercer Trabajo de Actuaria

Integrantes:

Ricardo William Salazar Espinal C.C. 1017219472

Medellín, Colombia

Medellin, septiembre 27 de 2024

Índice

Índice de figuras

Índice de cuadros

1 Contexto

Retomaremos la informacion del trabajo 1 por lo tanto durante la labor de este trabajo consideraremos los parametros que se usaron en este primer trabajo. Por lo tanto consideramos los siguientes parametros

tipo	$b =$ rate	$a =$ shape
1	0.0000281	0.0962189

como tenemos una distribucion Perks.1 tenemos su funcion definida de la siguiente manera.

$$tPx = e^{-a_1 t}\left(\frac{a_2 e^{a_3 x}+1}{a_2 e^{a_3(x+t)}+1}\right)^{\frac{1-a_1}{a_3}}$$

Notemos que esta funcion depende de 3 parametros los cuales nosotros tomaremos como $a_1 = 0.00025748, a_2 = 0.00002553, a_3 = 0.10128397$.

Por lo tanto definido los parametros que necesitaremos y puestos en contexto procederemos a la solucion de los puntos propuestos.

2 Punto a

Considere esta renta vitalicia: para una vida (x) empieza a pagar C = 2.5, mes vencido, desde el mes siguiente al mes $\lceil mT(x)^{ai}\rceil$ en el cual se presenta una enfermedad de alto riesgo. Defina $v = \frac{1}{(1+i)}$ y la variable aleatoria:

2.1 Solucion

Lo primero que realizamos es encontrar la identidad que nos permita resolver este problema. Para ello, consideraremos el ejercicio 6.7.1, pero ahora tomaremos en cuenta la variable $T(x)^{ai}$, para una vida x que está inicialmente sana y que luego recibe un diagnóstico de una enfermedad crónica.

$$Z^{ai} = a\frac{(m)}{\lceil mT(x)\rceil \wedge \lceil mT(x)^{ai}\rceil} = \frac{1}{m}\sum_{k=0}^{\lceil mT(x)\wedge\rceil \lceil mT(x)^{ai}\rceil} e^{-\delta_m k}$$

Entonces su costo neto es

$$E(Z^{ai}) = \frac{1}{m} \sum_{k=1}^{w-x} v^{\frac{k}{m}} \cdot {}_{\frac{k}{m}}P_x P(T(x)^{ai} \geq k)$$

Por el Teorema de Esperanza Iterada

$$E(Z) = E(E(Z \mid T)) = \sum_{k=1}^{\infty} E(Z \mid T = k) \times P(T(x)^{ai} = k)$$

De la definición de anualidad de vida temporal vencida.

$$(Z \mid T(x)^{ai} = k) = a\frac{(m)}{\lceil mT(x) \rceil \wedge k} \frac{1}{m} \sum_{k=1}^{\lceil mT(x) \rceil \wedge k} e^{-\delta_m k}$$

$$E(Z \mid T(x)^{ai} = k) = \frac{1}{m} \sum_{k=1}^{mk} v^{\frac{k}{m}} {}_{\frac{m}{k}}P_k$$

Sustituyendo en la expresión para $E(Z)$

$$E(Z) = \frac{1}{m} \sum_{k=1}^{m(w-x)} a\frac{(m)}{x:k|} \cdot P(T(x)^{ai} = k)$$

Para aplicar la fórmula de sumatoria por partes en el Lema (6.12), se definen:

$$f_k = \frac{1}{m} v^{\frac{k}{m}} {}_{\frac{k}{m}}P_x$$

$$g_k = P(T = k)$$

$$F_k = a\frac{(m)}{x:k|}$$

$$G_k = P(T(x)^{ai} \leq k)$$

Entonces con m=1 y n=w(w-x)

$$\sum_{k=1}^{m(w-x)} a\frac{(m)}{x:k|} \cdot P(T(x)^{ai} = k) = \sum_{k=m}^{n} g_k F_k = F_n G_n - F_0 G_0 - \sum_{k=m}^{n} f_k G_{k-1} = a\frac{(m)}{x:k|} - \sum_{k=1}^{m(w-x)} v^{\frac{k}{m}} {}_{\frac{k}{m}}P_x \cdot P(T(X)^{ai} \leq$$

Luego procedemos a realizar el calculo de esta esperanza usando la formula obtenida.

Al calcular la prima neta se tiene que toma un valor de $a_x^{(m,ai)} = E(Z^{ai}) = 18.2959241$ y tenemos el costo como: 548.8777219. En base a estos datos se podria realizar una mejor planificacion y evaluar el posible riesgo de este caso.

3 Punto b

Considere una renta vitalicia para una vida (x), que tiene costo neto dado por: $C \cdot m \cdot (a_x^{(m)} - a_x^{(m,ai)})$. Interprete este valor, que riesgo se esta cubriendo?.

3.1 Solucion

Esta diferencia mide el costo de la anualidad entre una persona sana y una persona con enfermedad crónica. En otras palabras, este valor refleja el impacto financiero del riesgo adicional que representa el estado de salud deteriorado de la persona (enfermedad crónica) sobre el valor de su renta vitalicia.

El riesgo que se está cubriendo aquí es el riesgo de aumento en los costos debido a la menor esperanza de vida de la persona enferma en comparación con una persona sana. Cuando una persona es diagnosticada con una enfermedad crónica, su probabilidad de supervivencia futura cambia, y esto afecta el valor presente de los pagos de una renta vitalicia. Si la esperanza de vida es menor, el costo de la anualidad (lo que la aseguradora espera pagar) también es menor.

Es decir, este valor representa el costo neto asociado a cubrir el riesgo de mortalidad adicional para una persona diagnosticada con una enfermedad crónica, comparado con una persona sana.

Al calcular entonces el valor de $C \cdot m \cdot (a_x^{(m)} - a_x^{(m,ai)}) = 1586.0632741$.

4 Punto c

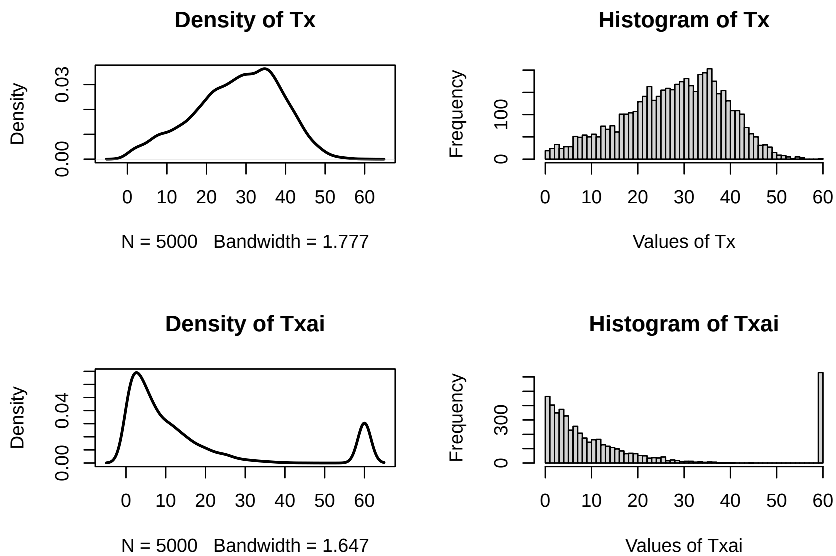
Simule $N = 5000$ valores de la variable aleatoria $V = Z - Z^{ai}$ con $Z = \frac{1}{m}\sum_{k=1}^{\lceil T(x) \rceil} v^{\frac{k}{m}}$. Reporte el histograma de V con el valor de $a_x^{(m)} - a_x^{(m,ai)}$ y su media, en el eje x.

4.1 Solucion

Para dar solución a este problema usando simulación, debemos seguir los siguientes pasos:

Lo primero es simular los tiempos de incidencia de la enfermedad y el tiempo de vida de la persona, que denotaremos por $TxaiyTx$,respectivamente. Debemos considerar los casos donde la enfermedad ocurre después de la muerte de la persona. Por lo tanto, el tiempo mínimo en Z^{ai} seria el mismo tiempo de Z lo que implicaría que la variable aleatoria $V = Z - Z^{ai}$ tomaria el valor de cero. Si la enfermedad ocurre antes de la muerte, procederemos a calcular las sumas para Z desde la edad de 50 años hasta el momento del deceso, mientras que para la enfermedad, los cálculos se realizarán desde la edad de 50 años hasta el momento de incidencia de la enfermedad.

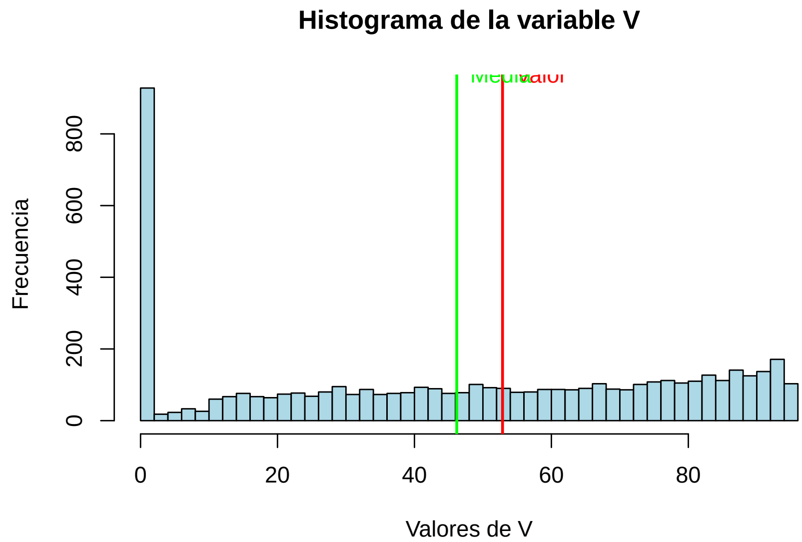
A continuación, debemos observar cómo sería el histograma y la dispersión para los tiempos $TxyTxai$, donde analizaremos la densidad para cada uno y un histograma con los datos.



Ahora generamos los valores de V como se indicó y procedemos a calcular la media de V. Adicionalmente, agregamos la etiqueta "valor", que contiene la diferencia $a_x^{(m)} - a_x^{(m,ai)}$, Observamos que ambos valores no son del todo iguales, mientras que la etiqueta "valor" toma el valor de 52.86, la media de la simulacion es de 46.18.La diferencia entre ambos valores es de 6.68. Esto sugiere que podría ser más adecuado realizar más simulaciones para obtener resultados más consistentes, o que la discrepancia observada se deba a la naturaleza aleatoria de la generación de valores.

Dada la complejidad del código, también es probable que haya algún error humano en alguna parte del mismo que esté provocando esta discrepancia. La programación, especialmente en modelos estadísticos, puede ser propensa a errores sutiles que afectan los resultados. Por lo tanto, una revisión exhaustiva del código es recomendable para identificar y corregir posibles fallos.Dado la falta de tiempo no fue posible identificar este tipo errores.

Recomendacion:Dado el costo computacional de realizar las 5000 simulaciones, se recomienda guardar los datos generados después de la simulación y cargarlos posteriormente. De esta manera, se evita tener que realizar la simulación cada vez, lo cual puede ser muy demorado.



5 Apendice de Codigos

5.1 Codigo punto a

```
muxt.pe1 <- function(t, x, pars) {
  a1 <- pars[1]
  a2 <- pars[2]
  a3 <- pars[3]
  m <- (a1 + a2 * exp(a3 * (x + t))) / (1 + a2 * exp(a3 * (x + t)))
  return(m)
}

tpx.pe1 = function(t, x, pars){
  a1 = pars[1]
  a2 = pars[2]
  a3 = pars[3]
  g = (1 - a1) / a3
  v = exp(-a1 * t) * ((a2 * exp(a3 * x) + 1) / (a2 * exp(a3 * (x + t)) + 1))^g
  return(v)
}

cociente <- function(t, x, a, b) {
  dgompertz(x + t, shape = a, rate = b) /
    pgompertz(x, shape = a, rate = b, lower.tail = FALSE)
}
```

```r
fTxai <- function(t, x, pars, a, b) {
  cociente(t, x, a, b) * tpx.pe1(t, x, pars)
}

# Definir parámetros
pars <- c(0.00025748, 0.00002553, 0.10128397)
x <- 50
a = 0.0000281
b = 0.0962189

library(rmutil)
tpxai<- function(t,x,pars,a,b){
  ft <- function(t) {
    fTxai(t, x, pars, a, b)}
p <-ifelse(t>0,1-int(ft, 0, t),0)
return(p)}

avxym = function(x, m, i, pars, a, b) {
  v = 1 / (1 + i)
  k = seq(0, m * (110-x))

  kmpx = sapply(k, function(k) tpx.pe1(k / m, x, pars))
  kmpy = sapply(k, function(k) tpxai(k/m, x, pars, a, b))

  vkm = v^(-k / m)
  suma = sum(vkm * kmpx * kmpy) / m
  return(suma)
}

# Parámetros del problema
x = 50
i = 0.05
m = 12
C = 2.5
a = 0.0000281  # Define a según sea necesario
b = 0.0962189  # Define b según sea necesario


zi<-avxym(x, m, i, pars, a, b)
Cpxy <- 12 * C * avxym(x, m, i, pars, a, b) # Costo de la anualidad
print(zi)
print(Cpxy)
```

5.2 Codigo punto b

```
avxym_no_enfermedad = function(x, m, i, pars) {
  v = 1 / (1 + i)

  k = seq(0, m *(110-x)-1)

  kmpx = sapply(k, function(k) tpx.pe1(k / m, x, pars))

  vkm = v^(-k / m)
  suma = sum(vkm * kmpx) / m
  return(suma)
}

# Parámetros del problema
x = 50
i = 0.05
m = 12
C = 2.5
a = 0.0000281  # Define a según sea necesario
b = 0.0962189  # Define b según sea necesario

# Calculo de a_x^m sin enfermedad
a_x_m = avxym_no_enfermedad(x, m, i, pars)

# Calculo de Cpxy con enfermedad
Cpxy_no_enfermo <- 12 * C * avxym(x, m, i, pars, a, b)

z<-avxym_no_enfermedad(x, m, i, pars)

valor<-(z-zi)

diferencia<- C*m*(z-zi)
```

5.3 Codigo punto C

```
#------------- Definiciones para el modelo Perks

#redefinimos la acomulada para fTxai aqui no necesitamos P(T>K)
library(rmutil)
tpxai<- function(t,x,pars,a,b){
  ft <- function(t) {
```

```r
    fTxai(t, x, pars, a, b)}
p <-ifelse(t>0,int(ft, 0, t),0)
return(p)}




# Definir la función de intensidad muxt para el modelo Perks
muxt.perks <- function(t, x, pars) {
  a1 <- pars[1]
  a2 <- pars[2]
  a3 <- pars[3]
  return(a1 + a2 * exp(a3 * (x + t)))  # Ejemplo de función
}

# Definir la función de supervivencia tpx para el modelo Perks
tpx.perks <- function(t, x, pars) {
  a1 <- pars[1]
  a2 <- pars[2]
  a3 <- pars[3]
  g <- (1 - a1) / a3
  p <- exp(-a1 * t) * ((a2 * exp(a3 * x) + 1) / (a2 * exp(a3 * (x + t)) + 1))^g
  return(p)
}

# Parámetros para el modelo Perks
pars <- c(0.00025748, 0.00002553, 0.10128397)
x <- 50  # Edad inicial
N <- 5000  # Número de muestras

# Definir la función de supervivencia acumulada
f <- function(t) {
  ifelse(t < 110 - x, 1 - tpx.perks(t, x, pars), 1)
}

# Definir la función random.function
random.function <- function(n, f, lower, upper) {
  u <- runif(n)  # Números aleatorios
  sapply(u, function(u_value) {
    target_function <- function(t) f(t) - u_value
    uniroot(target_function, lower = lower, upper = upper)$root  # Encuentra la raíz
  })
}
```

```
# Generar muestras Tx usando random.function
Tx <- random.function(N, f, lower = 0, upper = 60)


# Definición de fxai
fxai <- function(t) {
  if (t < 110 - x) {
    return(tpxai(t, x, pars, a, b))
  } else {
    return(1)
  }
}

# Generar Txai usando random.function
Txai <- random.function(N, fxai, lower = 0, upper = 60)

# Visualizar resultados
par(mfrow = c(2, 2))
plot(density(Tx), lwd = 2, main = "Density of Tx")
hist(Tx, 80, main = "Histogram of Tx", xlab = "Values of Tx")

plot(density(Txai), lwd = 2, main = "Density of Txai")
hist(Txai, 80, main = "Histogram of Txai", xlab = "Values of Txai")
```

```
# Función para calcular Z^{ai} cuando hay enfermedad, usando meses ajustados
avxym = function(x, m, i, pars, a, b, Txai) {
  v = 1 / (1 + i)

  # Convertimos Txai a meses vencidos
  Txai_ajustado = ceiling(Txai * 12)

  # Usamos el tiempo ajustado en vez de 110
  k = seq(1, Txai_ajustado)

  kmpx = sapply(k, function(k) tpx.pe1(k / m, x, pars))
  kmpy = sapply(k, function(k) tpxai(k / m, x, pars, a, b))  # Pasar los parámetros adi

  vkm = v^(-k / m)
  suma = sum(vkm * kmpx * kmpy) / m
  return(suma)
}

# Función para calcular Z cuando no hay enfermedad, usando meses ajustados
avxym_no_enfermedad = function(x, m, i, pars, Tx) {
```

```
  v = 1 / (1 + i)

  # Convertimos Tx a meses vencidos
  Tx_ajustado = ceiling(Tx * 12)

  # Usamos el tiempo ajustado en vez de 110
  k = seq(1, Tx_ajustado)

  kmpx = sapply(k, function(k) tpx.pe1(k / m, x, pars))

  vkm = v^(-k / m)
  suma = sum(vkm * kmpx) / m
  return(suma)
}

# Generación de la variable aleatoria V con el tiempo ajustado
generar_V = function(Tx, Txai, m, i, pars, a, b) {
  V = numeric(length(Tx))  # Vector para almacenar los resultados

  for (j in 1:length(Tx)) {
    if (Txai[j] > Tx[j]) {
      V[j] = 0  # Si Txai > Tx, el valor es cero
    } else {
      # Si Txai <= Tx, aplicamos la fórmula de meses vencidos y las funciones avxym y a
      Z_ai = avxym(Tx[j], m, i, pars, a, b, Txai[j])  # Aquí usamos Txai para Z^{ai}
      Z = avxym_no_enfermedad(Tx[j], m, i, pars, Tx[j])  # Aquí usamos Tx para Z
      V[j] = Z - Z_ai  # La diferencia entre Z y Z^{ai}
    }
  }
  return(V)  # Asegúrate de retornar V
}

# Parámetros del problema
x = 50
i = 0.05
m = 12
C = 2.5
a = 0.0000281  # Define a según sea necesario
b = 0.0962189  # Define b según sea necesario

# Generamos la variable V
V <- generar_V(Tx, Txai, m, i, pars, a, b)
```

```r
# Convertir el vector V en un data.frame
df_V <- data.frame(V = V)

# Guardar el data.frame en un archivo CSV
write.csv(df_V, "V_simulacion2.csv", row.names = FALSE)
```

```r
# Cargar el data.frame desde el archivo CSV
df_V_cargado <- read.csv("V_simulacion2.csv", header = TRUE)

# Extraer la columna 'V' como un vector
V_cargado <- df_V_cargado$V

# Verifica que sea un vector
print(is.vector(V_cargado))  # Debería devolver TRUE
```

```r
# Generar el histograma
media <- mean(V_cargado)
hist(V_cargado,
     main = "Histograma de la variable V",
     xlab = "Valores de V",
     ylab = "Frecuencia",
     col = "lightblue",
     border = "black",
     breaks = 50)

# Agregar línea vertical para el valor
abline(v = valor, col = "red", lwd = 2)  # Línea para el valor
text(valor, max(hist(V_cargado, plot = FALSE)$counts), labels = "Valor", pos = 4, col =

# Agregar línea vertical para la media
abline(v = media, col = "green", lwd = 2)  # Línea para la media
text(media, max(hist(V_cargado, plot = FALSE)$counts), labels = "Media", pos = 4, col =
```