Tarea 2 Estadística Actuarial II

Maria Carolina Navarro Monge C05513 Tábata Picado Carmona C05961
Jose Pablo Trejos Conejo C07862

```
#Se cargan las librerías necesarias
library(readxl)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
```

Ejercicio 2

Usando la metodología de Muestreo por Importancia, Si X~N(0.5,0.5) estime:

a. P(X<-5)

Mediante el metódo de integración por Montecarlo se obtiene el siguiente resultado:

```
#----------------------------Ejercicio 2---------------------------------------

#---Estimación de función de distribución mediante Muestreo por Importancia----

set.seed(2901)
n <- 10^4 #tamaño de la muestra

X <- rnorm(n, 0.5, sqrt(0.5))

f <- dnorm(X, 0.5, sqrt(0.5))

valor_estimado_1 <- mean(f)

valor_real <- pnorm(-5, 0.5, sqrt(0.5))
```

```r
comparación <- data.frame("Estimación" = valor_estimado_1, "Valor real" = valor_real)

print(comparación)
```

```
##   Estimación   Valor.real
## 1  0.3991698 3.678924e-15
```

Como se puede observar, la estimación resultante converge lento al valor real. Por lo tanto, mediante Muestreo por Importancia se puede acelelar la convergencia empleando una densidad auxiliar. Para este caso, la densidad auxiliar a utilizar es una exponencial truncada de la forma $\lambda e^{-\lambda(x-t)}$, con x>t.

La probabilidad a estimar es equivalente a P(X>6). Nos basaremos en esta para aproximarla mediante el Muestro por Importancia por medio de una exponencial truncada en 6 con $\lambda = 1$. El procedimiento se muestra en el siguiente algoritmo:

```r
A <- rexp(n)+6 #datos aleatorios mayores a 6 con distribución exponencial

w <- dnorm(A, 0.5, sqrt(0.5)) / dexp(A-6)

valor_estimado <- mean(w)

resumen <- data.frame("Estimación" = valor_estimado, "Valor real" = valor_real)

print(resumen)
```

```
##     Estimación   Valor.real
## 1 3.669418e-15 3.678924e-15
```

De tal manera, se obtiene una mejor aproximación de la probabilidad P(X<-5), pues, es muy similar al valor real.

b. Estime el error absoluto de la estimación del punto a.

El error absoluto de la estimación es:

```r
error_absoluto <- abs(valor_estimado-valor_real)
```

#Ejercicio 5

Una aseguradora tiene un producto llamado Doble Seguro de Vida (DSV) el cual paga 2 veces la suma asegurada si la persona fallece antes de los 60 años, paga 1 suma asegurada cuando la persona cumple los 60 años (si no ha fallecido) y paga 1 suma asegurada si fallece después de los 60 años. Considerando:

a. Las tablas de vida dinámicas de la SUPEN (https://webapps.supen.fi.cr/tablasVida/Recursos/documentos/tavid2000-2150.xls)

b. Un cliente de 30 años, hombre con una suma asegurable de 1 000 000 colones.

Construya con la ayuda de un MCMC la distribución de los pagos por año de que se espera de este seguro. Use al menos 10 000 iteraciones. Y muestre Histograma.

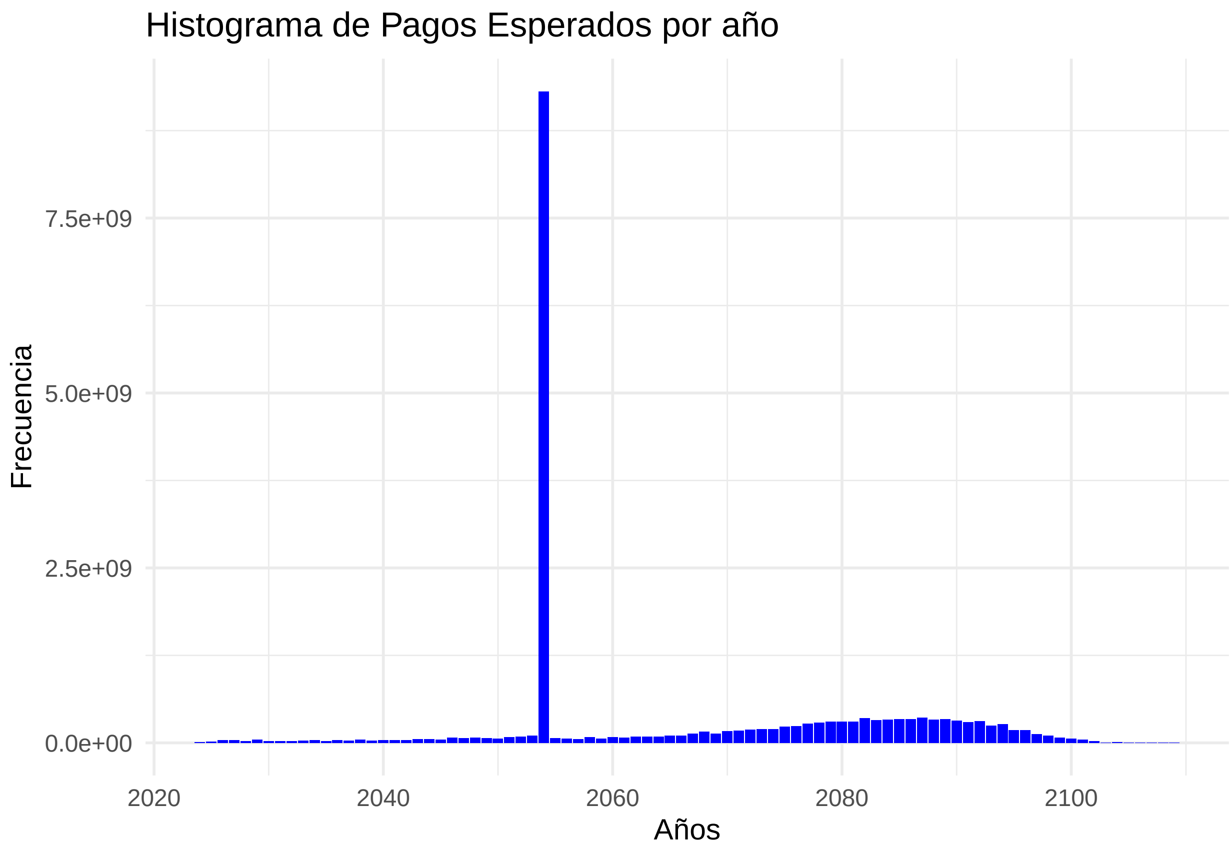
La construcción de la distribución de los pagos por año mediante MCMC se muestran en el siguiente código:

```r
#Se carga la base de datos
tabla_vida <- read_excel("tavid2000-2150.xls",
                         col_types = "numeric")

#Se filtra la base de datos para obtener los datos de un hombre nacido en 1994
#con edades mayor o igual a 30

datos <- subset(tabla_vida, sex == 1 & ynac == 1994 & edad >=30, select = c(edad,qx, year))

#Se obtienen las probabilidades de sobrevivencia

px <- 1- datos$qx

#Se añaden las probabilidades de sobrevivencia a la base datos
datos$px <- px

qx<- datos$qx
suma_asegurada_1 <- 10^6
suma_asegurada_2 <- 2*10^6

#-------------------MCMC--------------------------------/

#Se simulan diversas trayectorias de vida de la persona
set.seed(2901)
iteraciones=10^4
n=length(px)
pago <- rep(0, 86)

for (i in 1:iteraciones) {
  U <- runif(n)  # Se toman como probabilidades de muerte
  t <- 1
  cont <- 1

  #Determinación del año de fallecimiento
  while (t == 1) {
    if (U[cont] < px[cont]) {
      cont <- cont + 1
    } else {
      t <- 0
    }
  }
  año_fallecimiento <- cont - 1

  #Asignar los pagos correspondientes al año de fallecimiento
  if (año_fallecimiento < 30) {
    pago[año_fallecimiento + 1] <- pago[año_fallecimiento + 1] + suma_asegurada_2
  } else if (año_fallecimiento ==30) {
    pago[31] <- pago[31]+ suma_asegurada_1
  }else {
    pago[31] <- pago[31] + suma_asegurada_1
    pago[año_fallecimiento + 1] <- pago[año_fallecimiento + 1] + suma_asegurada_1
  }
}
```

```
resultado <- data.frame("Años pago"= datos$year, "Pago" = pago)
```

El histograma de los pagos esperados por año es el siguiente:

```
ggplot(data = resultado, aes(x = Años.pago, y = Pago)) +
  geom_bar(stat = "identity", fill = "blue") +
  labs(title = "Histograma de Pagos Esperados por año", x = "Años", y = "Frecuencia")+
  theme_minimal()
```

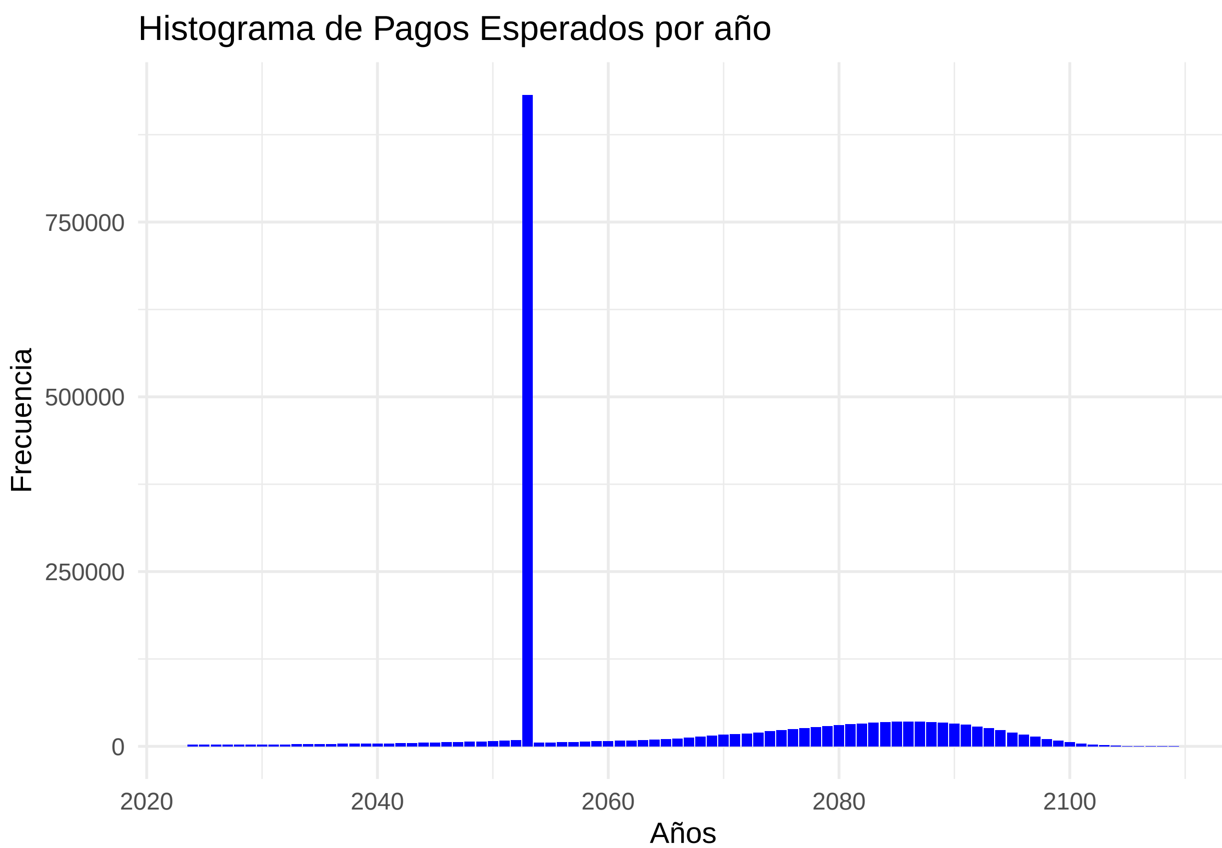


Es evidente que la mayor cantidad de pagos se sitúan a los 60 años y posteriomente después de los 60, lo que indica que es más probable que el cliente fallezca después de los 60 años.

Se puede verificar el resultado obtenido por MCMC si lo comparamos con un histograma obtenido mediante un método determinista como el que se muestra a continuación:

```
#----- Método determinista----------------------------------------------------/

#Se crea una función que obtiene n_p_30
n_p_30 <- c(0)

n_p_30_function <- function(px) {

  for (i in 1:length(px)) {
    resultado <-1
    for(j in 1: i){
```

```r
      resultado <- resultado*px[j]
    }

    n_p_30[i] <- resultado
  }

  return(n_p_30)
}

n_p_30 <- n_p_30_function(px)

datos$n_p_30 <- n_p_30


#se calculan los pagos esperados para cada año
pago_esperado <- c(0)
pago_esperado[1] <- suma_asegurada_2*qx[1]

#caso fallecimiento antes de los 60 años
for (i in 2: 29 ) {
    pago_esperado[i] <- suma_asegurada_2*n_p_30[i-1]*qx[i]
}

#caso sobrevive a los 60 años

pago_esperado[30] <-suma_asegurada_1*n_p_30[30]

#caso fallecimiento después de los 60 años

for (i in 1: (length(px)-30)) {
  pago_esperado[30+i] <- suma_asegurada_1*n_p_30[30+i-1]*qx[30+i]
}

resultado_determinista <- data.frame("Años pago"= datos$year, "Pago" = pago_esperado)

ggplot(data = resultado_determinista, aes(x = Años.pago, y = Pago)) +
  geom_bar(stat = "identity", fill = "blue") +
  labs(title = "Histograma de Pagos Esperados por año", x = "Años", y = "Frecuencia")+
  theme_minimal()
```



Como se puede observar, los pagos esperados mediante el método determinista y el MCMC muestran distribuciones muy similares. Por tanto, se verifica que el resultado que se obtuvo por el método MCMC es aceptable.

```
fnormal <- function(x,mu1,mu2,sigma1, sigma2) {
  fx= exp(-((x-mu1)^2/(2*(sigma1)))) - exp(-((x-mu2)^2/(2*(sigma2))))
  return(fx)
}

mu1 <- 4
mu2 <- 2
sigma1 <- 4
sigma2 <- 1

fZ <- function(x){return(fnormal(x,mu1,mu2,sigma1,sigma2))}
```

```
# Valores para el rango de la gráfica
x_values <- seq(0, 16, length.out = 1000)

par(mfrow = c(1, 2))

# Gráfico de la distribución de Z y las medias de X1 y X2
plot(x_values, fZ(x_values), type = "l", col = "blue", lwd = 2,
     xlab = "Z", ylab = "Densidad", main = "Distribución de Z = X1 - X2")

# Líneas verticales para las medias de X1 y X2
```

```
abline(v = c(mu1, mu2), col = c("red", "green"), lty = c(2, 2), lwd = 2)

# Etiquetas para las medias
text(mu1, 0.20, "Media X1", pos = 1, col = "red")
text(mu2, 0.10, "Media X2", pos = 1, col = "green")

# Gráfico de la distribución en valor absoluto de Z y las medias de X1 y X2
plot(x_values, abs(fZ(x_values)), type = "l", col = "blue", lwd = 2, xlab = "Z",
     ylab = "Densidad (Valor Absoluto)", main = "Distribución de Z = X1 - X2")

abline(v = c(mu1, mu2), col = c("red", "green"), lty = c(2, 2), lwd = 2)

text(mu1, 0.20, "Media X1", pos = 1, col = "red")
text(mu2, 0.10, "Media X2", pos = 1, col = "green")
```