Instituto Tecnológico y de Estudios Superiores de Monterrey

Escuela de Ingeniería y Ciencias

Ingeniería en Ciencia de Datos y Matemáticas

Análisis de Riesgo de una Compañía Aseguradora

Optimización Estocástica

Cantú Rodríguez Pamela A01285128

Ferreira Guadarrama Emiliano A01654418

Núñez López Daniel I. A01654137

Ruiz Alvaro Ana Paula A01367467

Ugalde Jiménez Ana Sofía A01702639

Supervisado por
Dr. Jaime Eduardo Martínez Sánchez
Dr. Fernando Elizalde Ramírez

Monterrey, Nuevo León. Fecha, 11 de septiembre de 2022

***Abstract* — Las aseguradoras deben ser capaces de solventar los gastos de sus clientes y de la empresa para evitar ir a la ruina. Sin embargo, es imposible saber con certeza cuándo ocurrirá un accidente. En este artículo se presentan dos modelos, la Aproximación de Cramer-Lundberg y la Transformada de Laplace, con el objetivo de calcular la probabilidad de ruina a partir de una base de datos de una aseguradora. En la primera fase, los reclamos se modelaron con una distribución exponencial y se utilizó la Aproximación de Cramer-Lundberg. Se realizaron dos simulaciones: Monte Carlo y Monte Carlo con reducción de varianza. La primera arrojó que para una aseguradora con un capital inicial de $14,000,000 y una prima de $2,290,000 la probabilidad de ruina al 90 % de confianza es de [15.75 %, 22.65 %]. Mientras que con reducción de varianza, la probabilidad de ruina al 95 % de confianza es de [17.54 %, 25.99 %]. Por otro lado, se calculó dicha probabilidad vía la transformada de Laplace, obteniendo como resultado una probabilidad de 0.0116 para una mezcla de uniformes y de 0.11582 para una distribución gamma de los reclamos. Se concluye que la probabilidad de ruina tiene una mayor sensibilidad al valor de la prima que el del capital inicial. Asimismo, se mostró la importancia de utilizar diversos método para predecir la probabilidad de ruina de la empresa.**

***Palabras clave* — Probabilidad de ruina, Aproximación de Cramer Lundberg, Transformada de Laplace, Método de Monte Carlo, Monte Carlo con reducción de varianza, distribución exponencial, distribución Gamma, distribución uniforme**

1. Introducción

Las compañías de seguros cumplen con un papel importante en la vida de sus clientes al proveer respaldo en eventos inesperados. Uno de los principales retos de las compañías aseguradoras es contar con el suficiente capital para cubrir las necesidades de sus clientes y, a su vez, solventar los gastos de la empresa y contar con utilidades.

A partir de esta necesidad, se calcula matemáticamente la probabilidad de riesgo de una compañía, en la que se observa si en algún momento dado, la compañía pudiera llegar a estar en capitales negativos. Aunque esto no significa que la empresa estará en quiebra, sirve como estimador para saber cómo se podría comportar en un futuro.

A lo largo del presente artículo se expone una estimación de la probabilidad de ruina por medio de una simulación de Monte Carlo clásica y una simulación de Monte Carlo con reducción de varianza utilizando el modelo clásico de Cramer-Lundberg y modelando la frecuencia de los montos de los reclamos como una distribución exponencial (Jiménez Hernández et al., 2011). Asimismo, se explora el cálculo exacto de la probabilidad de ruina a través de la Transformada de Laplace modelando la frecuencia de los montos de los reclamos como una mezcla de uniformes y como una distribución gamma.

Los datos que se analizarán en este artículo pertenecen al segmento de seguros de autos.

2. Descripción de la problemática

El servicio que ofrecen las aseguradoras de autos es la gestión de riesgos a través de contratos de seguro. La compañía se hace responsable del pago en caso de que ocurra cierto evento en el futuro, mientras que el usuario paga una prima (anualidad, mensualidad, etc.) menor por la protección en ese evento en el futuro incierto (Beers, 2021). Las personas están constantemente expuestas a accidentes como choques automovilísticos, incendios, terremotos, robos, etc. Sin embargo, es imposible saber con certeza cuándo ocurrirán dichos siniestros, así como la cantidad de dinero necesaria para solventar los gastos.

Existen modelos matemáticos, como la aproximación de Cramer-Lundberg y la Transformada de Laplace, capaces de calcular las probabilidades de que cierto evento azaroso ocurra.

2.1. Descripción del modelo de Cramer-Lundberg

El modelo clásico de riesgo de Cramer-Lundberg es de gran utilidad para calcular la probabilidad de ruina. En dicho modelo se toman en cuenta el capital inicial de la compañía, el monto que se recibe por unidad de tiempo (prima), tamaño y número de reclamos que llegan a la compañía en determinado momento. (Jiménez Hernández et al., 2011)

El capital inicial de una compañía de seguros es:

$$\mu >= 0 \tag{1}$$

En donde el número promedio de eventos por unidad de tiempo aleatorio sigue un proceso de Poisson homogéneo:

$$N(t) : t > 0 \tag{2}$$

con una intensidad de

$$\lambda > 0 \tag{3}$$

Mientras que los tiempos de llegada entre los eventos siguen una distribución exponencial y son variables aleatoriamente independientes. Los asegurados que recibe la compañía son una prima constante por unidad de tiempo:

$$c > 0 \tag{4}$$

Por lo tanto el capital de la compañía al tiempo t es:

$$X(t) = u + ct - \sum_{k=1}^{N(t)} Z_k \tag{5}$$

N(t) es el número de reclamaciones que llegan a la compañía en un intervalo *(0, t]*. Y como se puede ver en la fórmula anterior, *X(t)* es el modelo clásico de riesgo, o bien el modelo clásico de Cramer-Lundberg.

En cuanto a la trayectoria que sigue el modelo de riesgo, se tiene un capital inicial u. Al iniciar, *X(t)* crece al recibir las primas de sus asegurados, pero decrece al momento de recibir las primeras reclamaciones. Esto hasta que termina un periodo, y después vuelve a crecer y se repite el proceso como se puede ver en la siguiente tabla:

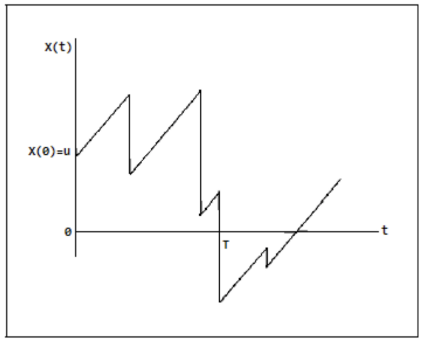

Figura 1: Trayectoria del Modelo de Riesgo (Jiménez Hernández et al., 2011)

Para determinar si la compañía estará con un capital menor a cero, se utiliza el cálculo de la probabilidad de ruina, en el que, el momento de ruina se da por:

$$T = min\{t >= 0 : X(t) < 0\} \tag{6}$$

Y se denota como:

$$\phi(u) = P(X(t) < 0, \text{para algún } t > 0|X(0) = u) \tag{7}$$

En donde $\phi(u)$ es la probabilidad de ruina dependiente del capital inicial u.

2.2. Aproximación de Cramer-Lundberg

Esta aproximación es útil cuando Z_j se distribuye de manera exponencial. En una fila de espera este escenario corresponde a una clasificación con notación de Kendall a un sistema *(M/M/1)*;es decir, que solamente se cuenta con un servidor y que las llegadas son determinadas por un proceso de Poisson, mientras que los servicios del trabajo tienen una distribución exponencial. (Grandell, 2000)

Dentro de las aproximaciones propuestas para calcular $\phi(u)$, y que se pueden resolver mediante el teorema del límite se encuentra la aproximación de Cramer-Lundberg:

$$\phi(u) \sim \phi_{CL}(u) \stackrel{\text{def}}{=} \frac{\rho\mu}{h'(R) - c/\alpha} e^{-Ru}, u \to \infty, \tag{8}$$

En donde resuelta utilizando límites tenemos que

$$\lim_{u\to\infty} e^{Ru}\phi(u) = \frac{\rho\mu}{h'(R) - c/\alpha}, \tag{9}$$

En donde el coeficiente de ajuste R es la solución positiva de

$$h(r) \stackrel{\text{def}}{=} \int_0^\infty (e^{rz} - 1)dF(z) = cr/\alpha \tag{10}$$

Cabe mencionar que esta aproximación requiere de que las colas de F disminuyan exponencialmente rápido. Esta aproximación es muy precisa para valores grandes del capital inicial en donde de (9) tenemos que

$$\phi(u) = \frac{\rho\mu}{h'(R) - c/\alpha} e^{-Ru} \tag{11}$$

En donde podemos controlar la probabilidad de ruina a través de las constantes c, R y u. (Jiménez Hernández et al., 2011)

2.3. Probabilidad de Ruina vía la transformada de Laplace

Con el objetivo de calcular las probabilidades de ruina utilizando las transformadas de Laplace, primero se realizó la transformada de los montos de los reclamos dada por F. En este caso F representa una distribución de mezcla de uniformes, esto debido a que los datos no se comportan de forma exponencial, que era el caso asignado, pero sí se comportan de forma uniforme, por el otro lado, en el caso de una distribución gamma, se tomó en cuenta el parámetro de la forma como 1 y esto hace que la distribución gamma se comporte de de forma exponencial, ya que solamente se utiliza el parámetro de la tasa.

Para este cálculo, se utilizó la siguiente función:

$$l_F(s) = \int_0^\infty e^{-sy} dF(y) \tag{12}$$

Luego de esto, se introdujo el resultado en la siguiente ecuación:

$$L_{\varphi(u)}(s) = \frac{1}{s} - \frac{c - \lambda\mu}{cs - \lambda(1 - l_F(s))}, s > 0 \tag{13}$$

Para obtener los resultados y llegar a una probabilidad exacta de ruina, basta con hacer los cálculos pertinentes y resolver ambas ecuaciones utilizando los parámetros que se obtendrán para el promedio de llegadas de los reclamos, en donde se obtendrá λ y el promedio de los montos de reclamaciones para distintos tipos de autos, que será dada por μ. Como se cuenta con una distribución F uniforme, en la ecuación (12) la derivada de F será cero, por lo que los cálculos podrán ser simplificados en la modelación.

3. Preguntas de investigación

A partir de esta problemática, el objetivo principal es determinar si la empresa llegará en algún punto a alcanzar la ruina, y en caso de que esta aseveración sea verdadera, en qué periodo es que la empresa se encontrará en ese estado. La interrogante que se busca resolver es ¿cuál es la probabilidad de que la compañía se vaya a la ruina en tiempo finito, o en un intervalo dado de tiempo?

La pregunta se responderá a partir de un modelo de simulación de riesgo para un horizonte finito de 10,000 días en que se asume que la distribución de los reclamos es una exponencial. Asimismo, se utilizará la Transformada de Laplace para el cálculo exacto de la probabilidad asumiendo una distribución gamma y una mezcla de uniformes para los reclamos.

4. Metodología utilizada

4.1. Simulación de Monte Carlo

En este proyecto se utilizó el método Monte Carlo (MMC), también conocido como la simulación de Monte Carlo. El MMC es una táctica matemática para estimar las probabilidades de un evento incierto, en esta ocasión, la probabilidad de ruina de la aseguradora (IBM, s.f.). El método Monte Carlo consiste en cuatro pasos:

Paso 1: Identificar la función que modele el proceso

La base de cualquier simulación de Monte Carlo es aquella función que describa el rango de probabilidades de que ocurra cierto evento, en otras palabras, la función de densidad de probabilidad (Harrison, 2010).

Paso 2: Definir los parámetros de entrada

Después de haber establecido el modelo, se definieron los parámetros de entrada, así como las distribuciones que éstos siguen. Como se estableció en la sección 2.1, en el modelo de Cramer-Lundberg, los siniestros por unidad de tiempo debe seguir una distribución de Poisson (Aurzada y Buck, 2020). Asimismo, se estableció que el monto de los reclamos se distribuyen de manera exponencial (5.2).

Paso 3: Configurar y preparar las simulaciones

Para implementar el método Monte Carlo, se hizo uso del lenguaje de programación R junto con la librería *ruin* para un horizonte de tiempo de 10,000 días y con 10,000 simulaciones en total (8).

Paso 4: Analizar los resultado obtenidos

El método Monte Carlo simula la llegada de reclamos, así como los montos de éstos. Posteriormente, revisa si las pérdidas totales superan la suma de la prima y el capital inicial para cada unidad de tiempo. Si los reclamos superan la suma de la prima y el capital, el algoritmo se detiene y cuenta una ruina. Por otro lado, si los reclamos no superan dicha cantidad, la simulación continúa hasta concluir el periodo de tiempo establecido y cuenta una no ruina (Brown et al., 2018). El proceso se repite y finalmente se estimó la probabilidad de ruina con una proporción:

4.2. Simulación de Monte Carlo con reducción de varianza

También se implementará una simulación de Monte Carlo con reducción de varianza mediante el método de variantes antitéticas. Este método intenta reducir la varianza mediante la introducción de correlación negativa entre pares de observaciones.

Para utilizar este método de reducción, se sigue una serie de pasos:

1. Se genera $(U_1, U_2, ..., U_k)$. Se construye un primer par: $X_1 = h(U_1, U_2, ..., U_k)$ y $X_2 = h(1 - U_1, 1 - U_2, ..., 1 - U_k)$
2. Se generan independientemente k iid uniformes nuevas para construir un nuevo par X_3, X_4 y se emparejan par por par hasta llegar a m-pares.
3. Se utiliza el estimador
$$\overline{Y}(m) = \sum_{j=1}^{m} Y_j, \tag{14}$$
en donde
$$\begin{aligned} &Y_1 = \tfrac{X_1+X_2}{2} \\ &Y_2 = \tfrac{X_3+X_4}{2} \\ &\vdots \\ &Y_m = \tfrac{X_{2m-1}+X_{2m}}{2}. \end{aligned} \tag{15}$$
4. Para construir nuestro nuevo intervalo de confianza se define la varianza muestral como
$$s^2(m) = \frac{1}{m-1} \sum_{j=1}^{m} (Y_j - \overline{Y}_m)^2. \tag{16}$$

Por lo que el intervalo queda como $\overline{Y}(m) \pm z_{\alpha/2}\frac{s(m)}{\sqrt{n}}$ y es un intervalo de $100(1-\alpha)\,\%$ para la media μ. (Sigman, 2007)

$$\frac{\text{cantidad de ruinas}}{\text{cantidad de simulaciones}} \tag{17}$$

5. Análisis de datos

Con el objetivo de ajustar la estimación de la probabilidad de ruina a los datos obtenidos por la aseguradora, los parámetros usados se obtuvieron de la base de datos estipulada.

5.1. Descripción de los datos

La base de datos originalmente contaba con 27,122 observaciones, cada una de ella con 8 propiedades:

1. Fecha del siniestro: fecha con el formato de día/mes/año (XX/XX/XXXX) en la que ocurrió el siniestro
2. Tipo de Auto: tipo de auto de acuerdo a la aseguradora (austero, camioneta, compacto, de lujo, deportivo y subcompacto)
3. Modelo: año modelo del vehículo
4. Monto del siniestro: cantidad de dinero que ha costado el siniestro reportado
5. Aplica cobertura: indica si la asegurado tiene la responsabilidad de pagar los daños del siniestro en caso de que el cliente lo desee
6. Deducible: cantidad de dinero pagada por el cliente si este decide hacer uso del seguro
7. Reclamo de no Cobertura: petición del cliente a la aseguradora para no hacer uso de su seguro
8. Pérdida total: si los daños del vehículo superar el 50 % del valor comercial del vehículo

Se limpió la base de datos para obtener un total de 27,102 observaciones:

- Se eliminaron aquellas filas que no contenían tipo de auto, deducible ni reclamo
- El montó se cambió a 0 en las observaciones en las que se aplicó el reclamo de no cobertura
- Las columnas se convirtieron a variables categóricas, enteros y fechas

5.2. Inferencia estadística

Para poder utilizar el modelo Cramer-Lundberg la cantidad de reclamos deben seguir una distribución de Poisson. Por lo tanto, tenemos:

H_0: la cantidad de reclamos por unidad de tiempo no siguen una distribución de Poisson

H_A: la cantidad de reclamos por unidad de tiempo siguen una distribución de Poisson

Se realizó la prueba de χ^2, obteniendo un $valor - p = 1$ por lo que se acepta la hipótesis alternativa y se infiere que la cantidad de reclamos por unidad de tiempo siguen una distribución de Poisson.

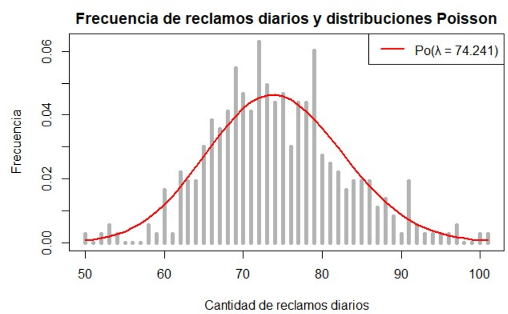

Figura 2: Frecuencia de reclamos diarios

De la misma manera, se establecieron las siguiente hipótesis sobre la distribución de los montos de los siniestros.

Para obtener la estimación de la probabilidad de ruina mediante la aproximación de Cramer-Lundberg:

H_0: los montos de los siniestros no siguen una distribución de exponencial

H_A: los montos de los siniestros siguen una distribución exponencial

Después de haber realizado la prueba de χ^2, se obtuvo un $valor - p = 0.3064$ por lo que se acepta la hipótesis alternativa y se infiere que los montos de los siniestros siguen una distribución exponencial.

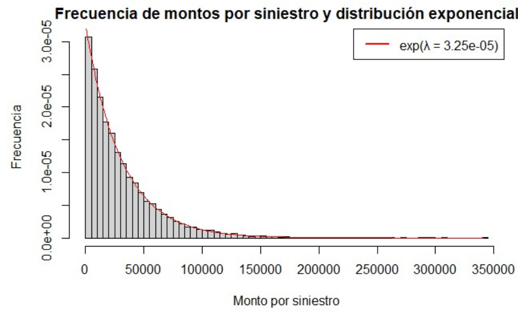

Figura 3: Frecuencia de montos por siniestro

Para calcular la probabilidad de ruina vía la transformada de Laplace, como se mencionó anteriormente, se utilizaron dos casos.

Caso 1:

H_0: los montos de los siniestros no siguen una distribución de gamma

H_A: los montos de los siniestros siguen una distribución gamma

Después de haber realizado la prueba de χ^2, se obtuvo un $valor - p = 0.3064$ por lo que se acepta la hipótesis alternativa y se infiere que los montos de los siniestros siguen una distribución gamma.

Caso 2:

H_0: los montos de los siniestros siguen una distribución de mezcla de uniformes

H_A: los montos de los siniestros no siguen una distribución de mezcla de uniformes

Se realizó una prueba de K-S, obteniendo un $valor - p = 5.762e^{-14}$. Así se acepta la hipótesis nula y se admite que los montos de los siniestros siguen una distribución de mezcla de uniformes.

El modelo de Cramer Lundberg requiere que las observaciones sean independientes, por lo tanto, se definieron las siguientes hipótesis:

H_0: las observaciones son independientes

H_A: las observaciones no son independientes

A través de una prueba de Fisher se obtuvo un $valor - p = 1$, por lo tanto, se acepta la hipótesis nula y se confirma que las observaciones se comportan de manera independiente.

Para poder calcular la probabilidad de ruina, también se hicieron asunciones sobre ambas medias: la cantidad de reclamos y el monto de los reclamos.

$$H_0 : \mu \neq \bar{x} \tag{18}$$

$$H_A : \mu = \bar{x} \tag{19}$$

Al ser una muestra de gran tamaño y contar con observaciones independientes y aleatorias, se acepta la hipótesis alternativa y se dice que la media poblacional es igual a la media muestral para ambos casos respectivamente.

Así, se obtuvo que la cantidad de reclamos sigue una distribución de Poisson con $\lambda = 74.241$. Mientras que los montos de dichos reclamos se distribuyen de manera exponencial con $\lambda = 3.25^{e-05}$ y Gamma con $\lambda = 3.25^{e-05}$.

5.3. Resultados obtenidos

5.3.1. Estimación de la probabilidad de ruina vía MMC

Se realizaron simulaciones para determinar los parámetros a largo plazo de la aseguradora: c y u, o capital inicial y pago de primas por parte de los usuarios. Se realizaron simulaciones con diferentes valores de dichos parámetros para así predecir el resultado en diferentes escenarios.

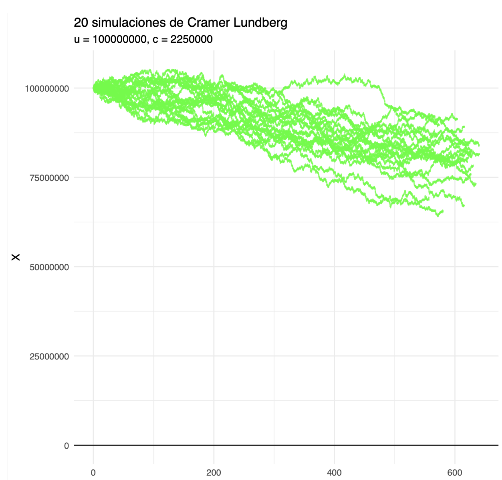

Figura 4: Simulaciones con $u = 100,000,000$ y $c = 2,250,000$

Como se puede observar en la Figura 4, si se invierte inicialmente una cantidad de $100,000,000 y se tiene una prima de $2,250,000 existe una tendencia negativa en el capital total de la aseguradora a través del tiempo y la empresa estará en ruina eventualmente.

Por otro lado, como se observa en la figura 5, si se invierten inicialmente $100,000,000, pero se tiene una prima de $2,300,000, hay una tendencia positiva en las ganancias de la aseguradora a través del tiempo.

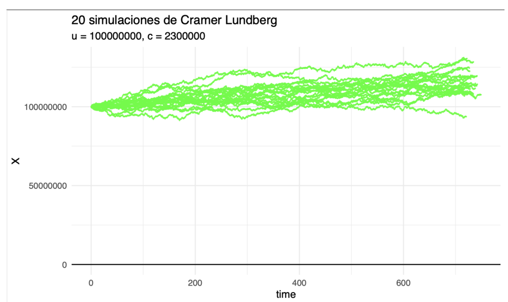

Figura 5: Simulaciones con $u = 100,000,000$ y $c = 2,300,000$

Con un capital inicial de $25,000,000 y una prima de $2,290,000 se observan probabilidades tanto de éxito como de ruina, como se ilustra en la figura 6.

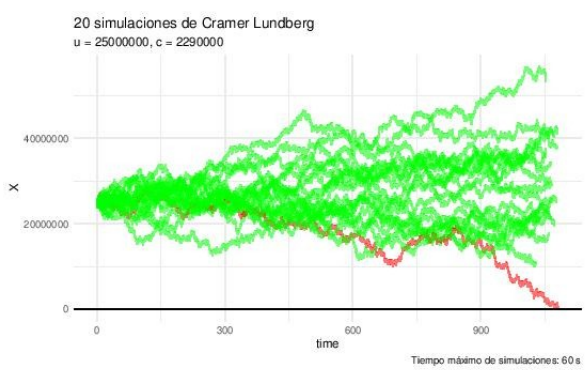

Figura 6: Simulaciones con $u = 25,000,000$ y $c = 2,290,000$

Por último, se obtuvo que para una aseguradora con un capital inicial de $14,000,000 y una prima de $2,290,000 la probabilidad de ruina al 90 % de confianza es de [15.75 %, 22.65 %] en una simulación con horizonte de tiempo de un año a *10000* iteraciones.

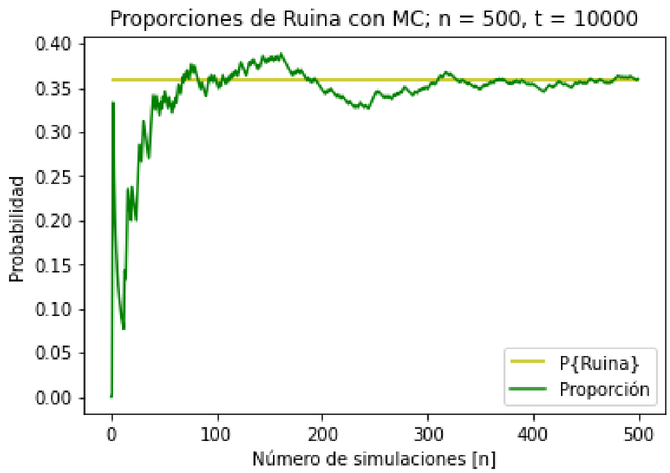

Figura 7: Proporción de Simulaciones Monte Carlo

5.3.2. Estimación de la probabilidad de ruina vía MMC con reducción de varianza

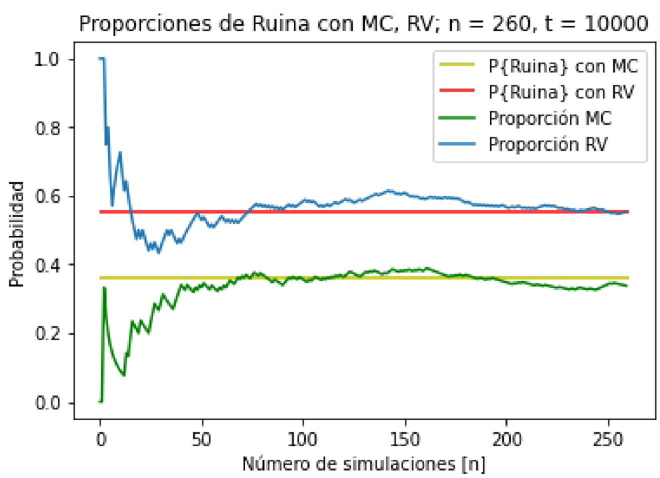

Figura 8: Proporción de Simulaciones Monte Carlo con Reducción de Varianza

Después de haber realizado 10,000 iteraciones con una reducción de varianza, se obtuvo que para una aseguradora con un capital inicial de \$14,000,000 y una prima de \$2,290,000 la probabilidad de ruina al 95 % de confianza es de [17.54 %, 25.99 %] en una simulación con horizonte de tiempo de un año.

5.4. Cálculo de la probabilidad de ruina vía transformada de Laplace

En este apartado se obtiene la probabilidad de ruina tanto de la distribución Gamma como la de mezcla de exponenciales. De cierta forma, se realiza una comparación entre ambas, con base en el resultado concreto que se obtiene de cada una.

5.4.1. Cálculo exacto de la probabilidad de ruina con mezcla de uniformes

$$\phi(u) = \mathscr{L}_f^{-1} = 1 + \frac{(\lambda\mu - c)e^{\lambda u/c}}{c} \tag{20}$$

$$\phi(u) = 1 + \frac{((74.241)(3.07 \times 10^4) - 2.29 \times 10^6)e^{(74.241)(1.4\times10^7)/2.29\times10^6}}{2.29 \times 10^6} \tag{21}$$

$$\phi(u) = 0.0116 \tag{22}$$

5.4.2. Cálculo exacto de la probabilidad de ruina con distribución gamma

$$\phi(u) = \mathscr{L}_f^{-1} = \frac{\lambda\mu}{c} e^{u(\frac{\lambda\mu - c}{c\mu})} \tag{23}$$

$$\phi(u) = \frac{(74.241)(3.07 \times 10^4)}{2.29 \times 10^6} e^{1.4\times10^7 \frac{(74.241)(3.07\times10^4)-2.29\times10^6}{(2.29\times10^6)(3.07\times10^4)}} \tag{24}$$

$$\phi(u) = 0.11582 \tag{25}$$

6. Discusión de resultados

En las figuras 4 y 5 se puede observar que, teniendo el mismo capital inicial, una diferencia de \$50,000 es capaz de cambiar la tendencia de una negativa a una positiva en el saldo de la empresa. Sin embargo, si se analiza la figura 6, donde la prima es casi idéntica pero el capital inicial es tan solo el 25 % de lo estipulado en las figuras anteriores, hay tendencias tanto negativas como positivas en el capital de la aseguradora. Si se compara esto con las dos figuras anteriores, se puede inferir que la probabilidad de ruina es más sensible hacia una variación en la prima que hacia una en el capital inicial. Es decir, en el caso de una aseguradora que recién comienza, tiene mayor importancia la cantidad de clientes y la cuota que estos paguen, que la inversión inicial.

El resultado obtenido del cálculo de la probabilidad de ruina con mezcla de uniformes fue de 1.16 %, mientras que el de distribución Gamma fue de 11.582 %. Con base en dichos resultados, se puede inferir que existe una mayor probabilidad de ruina cuando se trata de una distribución Gamma. En el caso de este reto, el momento de ruina es el primer momento en el cual una aseguradora tiene un capital menor o igual a cero.

Aunque se realizaron diversas simulaciones, para alcanzar una mejor propuesta, lo más recomendable sería desarrollar simulaciones con aún más métodos y también utilizar un tiempo más largo, así como variaciones en los parámetros iniciales. Esto es de utilidad por la variedad de casos siendo considerados, brindando así más confianza con los resultados, pues aunque todos los resultados obtenidos se podrían considerar bastante buenos, estos serían significativamente más verídicos con simulaciones más dispares.

7. Conclusiones

Para finalizar, a partir de las simulaciones realizadas se obtuvo un promedio de porcentaje de ruina de todos los modelos utilizados en la investigación es de 13.43 %, por lo que es una probabilidad de ruina bastante baja y favorecedora. Por ello, se puede asumir que los parámetros establecidos al comienzo, como la capital inicial y la constante prima, son bastante adecuados para llevar a cabo esta operación.

El hecho de realizar diversas simulaciones y aproximaciones con distintos métodos, ha sido de bastante utilidad para tener un mejor entendimiento de procesos estocásticos. También, se han logrado mejorar ciertas habilidades como el pensamiento analítico y la resolución de problemas por medio del mismo.

Referencias

Aurzada, F., & Buck, M. (2020). Ruin probabilities in the Cramér–Lundberg model with temporarily negative capital. *European Actuarial Journal 2020 10:1*, *10*(1), 261-269. https://doi.org/10.1007/S13385-020-00223-4

Beers, B. (2021). A Brief Overview of the Insurance Sector. Consultado el 25 de agosto de 2022, desde https://www.investopedia.com/ask/answers/051915/how-does-insurance-sector-work.asp

Brown, M., Rondon, D., Goffard, P.-O., & Sarantsev, A. (2018). EVALUATION OF THE RUIN PROBABILITY IN ORDERED RISK MODELS.

Grandell, J. (2000). Simple approximations of ruin probabilities. *Insurance: Mathematics and Economics*, *26*(2), 157-173. https://doi.org/https://doi.org/10.1016/S0167-6687(99)00050-5

Harrison, R. L. (2010). Introduction To Monte Carlo Simulation. *AIP conference proceedings*, *1204*, 17. https://doi.org/10.1063/1.3295638

IBM. (s.f.). ¿Qué es la simulación de Monte Carlo? - México — IBM. Consultado el 24 de agosto de 2022, desde https://www.ibm.com/mx-es/cloud/learn/monte-carlo-simulation

Jiménez Hernández, J., Todorova Kolkovska, E., & Maldonado Santiago, A. D. (2011). Probabilidad de ruina en el modelo clásico de Cramer-Lundberg. *Temas de Ciencia y Tecnología*, *15*, 9-18. https://www.utm.mx/edi_anteriores/temas45/1ENSAYO_45_2.pdf

Sigman, K. (2007). Introduction to reducing variance in Monte Carlo simulations. http://www.columbia.edu/~ks20/4703-Sigman/4703-07-Notes-ATV.pdf

8. Apéndice

8.1. Apéndice A: Descripción de los datos

```
```{r import_dataset}
ds = read.csv("Datos_aseguradora.csv")
```

Limpieza inicial de la base de datos

```{r limpieza_inicial}
# eliminamos ~20 observacionse que no contienen tipo de auto, deducible
ni reclamo de no cobertura
df <- subset(ds, ds$Tipo.de.auto != "")

# convertimos las columnas del df a fechas, factors o enteros
df$Fecha.del.Siniestro <- as.Date(df$Fecha.del.Siniestro, format="%d/%m/%y")
df$Tipo.de.auto <- factor(df$Tipo.de.auto)
df$Modelo <- factor(df$Modelo, ordered = TRUE)
df$Aplica.cobertura <- factor(df$Aplica.cobertura)
df$Deducible[is.na(df$Deducible)] <- 0
df$Deducible <- as.integer(df$Deducible)
df$Reclamo.de.no.Cobertura <- factor(df$Reclamo.de.no.Cobertura)
df$Pérdida.total <- factor(df$Pérdida.total)

summary(df)
```
```

8.2. Apéndice B: Inferencia estadística

```
# Número de reclamos (N) \~ Po(lambda)

```{r preparacion_N}
# generamos todos los días en el rango de dates
all_days <- as.Date(seq(
  min(df$Fecha.del.Siniestro),
  by = "day",
  length.out = as.integer(max(df$Fecha.del.Siniestro)+1 -
  min(df$Fecha.del.Siniestro))
))

# calculamos reclamos diarios con table
right = data.frame(table(df$Fecha.del.Siniestro))
right$Var1 <- as.Date(right$Var1)

# si nuestro dataset tuviera días sin reclamos estos no aparecerían en la
#tabla de frecuencias
left = data.frame(all_days)

# utilizamos merge lo cual nos da un df con NA en las fechas que no hubo reclamos
reclamos_diarios <- merge(left, right, by.x = 'all_days', by.y = "Var1", all.x = TRUE)

# cambiamos el index a la columna de all_days
```
```

```
rownames(reclamos_diarios) <- reclamos_diarios$all_days
reclamos_diarios$all_days <- NULL

# cambiamos los NAs por 0
reclamos_diarios[is.na(reclamos_diarios)] <- 0

# calculamos la proporción en la que sucede que haya x reclamos en un día
proporciones = table(reclamos_diarios)/dim(reclamos_diarios)[1]

x.ax = as.integer(names(proporciones))
x.range = min(x.ax):max(x.ax)

# también nos aseguramos de que las proporciones incluyan todos los
# valores dentro del rango de N
proporciones = merge(
  data.frame(reclamos_diarios = x.range),
  data.frame(proporciones),
  by = "reclamos_diarios",
  all.x = TRUE
  )

# cambiamos el index a la columna de reclamos_diarios
rownames(proporciones) <- x.range
proporciones$reclamos_diarios <- NULL

proporciones[is.na(proporciones)] <- 0


plot(
  x = x.range,
  y = proporciones$Freq,
  main="Proporción de número de reclamos diarios",
  xlab = "# de reclamos diarios",
  ylab = "Proporción",
  type = 'h',
  lwd = 5
  )
```

H_{0}: Sample is drawn from *Po(*$\lambda=\bar{N}$*)*

H_{1}: Sample is **NOT** drawn from *Po(*$\lambda=\bar{N}$*)*

```{r bondad_de_ajuste_Poisson}

n_rate = mean(reclamos_diarios$Freq)

candidate = dpois(x.range, lambda = n_rate)

chisq.test(
  as.vector(proporciones),
  p=candidate,
  rescale.p = TRUE,
  simulate.p.value=TRUE)
```

```
```{r N-Po_visualizacion}
```
```

```
plot(
  x = x.range,
  y = proporciones$Freq,
  main = "Frecuencia de reclamos diarios y distribuciones Poisson",
  xlab = "Cantidad de reclamos diarios",
  ylab = "Frecuencia",
  type = 'h',
  lwd = 5,
  col = 'gray70'
  )

lines(x.range,
  candidate,
  col='red',
  lwd = 2
  )


legend("topright",
  legend = paste0("Po(lambda = ", round(n_rate, 3), ")"),
  col = "red",
  lwd = 2
  )
```

Monto de siniestro (Z) ~ Exp(lambda)

H_{0}: Sample is drawn from *exp(* $\lambda=\frac{1}{\overline{Z}}$*)*
H_{1}: Sample is not from *exp(* $\lambda=\frac{1}{\overline{Z}}$*)*

```{r}
Z = df$Monto.del.siniestro
ks.test(Z, pexp, rate = 1/mean(Z))
```

```{r Z-exp_visualizacion}
xrange = seq(1, max(Z), 1000)
z_rate = 1/mean(Z)


hist(
  Z,
  breaks = 100,
  freq = FALSE,
  main = "Frecuencia de montos por siniestro y distribución exponencial",
  xlab = "Monto por siniestro",
  ylab = "Frecuencia"
  )

lines(
  xrange,
  dexp(xrange, rate = z_rate),
  col = 'red'
  )

legend("topright",
  legend = paste0(
    'exp(lambda = ',
```

```
    round(z_rate, 7),
    ')'
    ),
  lwd = 2,
  col = 'red'
  )
```

```{r}
x.range = seq(0, 250000, 1000)

hist(df$Monto.del.siniestro, breaks = 1000, freq = F)
lines(x.range, dgamma(x.range, shape = 1, scale = mean(df$Monto.del.siniestro)), col = 'red')
```

```{r}

groups = split(df, df$Tipo.de.auto)
montos = c()


for (group in groups){
  tab = table(group$Monto.del.siniestro)
  tab = tab/sum(tab)
  tab.min = as.integer(names(tab[1]))
  tab.max = as.integer(names(tail(tab,1)))
  mu = mean(group$Monto.del.siniestro)
  montos = c(montos, mu)

  #plot(tab)
  plot(tab, xlim = c(0, 342000), ylim = c(0, 0.004))
}

montos

# plot(tab, xlim = c(10000, 20000))

names(groups)
for (group in groups){
  print(dim(group))
}

aust = groups$Austero$Monto.del.siniestro[-c(1, 2, 3)]
tab = table(aust)

x = as.integer(names(tab))
y = as.vector(tab)

plot(x, y, type = 'h')
lines(rep(mean(y), 10000), col='red')



f = aust/sum(aust)

a = min(aust)
b = max(aust)
```

```
1/12 * (b-a)^2
var(aust)

(a+b)/2
mean(aust)

1 / (b-a)
mean(f)

chisq.test(runif(aust, a, b), aust)


plot(ecdf(aust))
curve(punif(x, a, b), add = T, col = 'red')

ks.test(aust, punif, a, b)
```


```{r preparacion_N}
# generamos todos los días en el rango de dates
all_days <- as.Date(seq(
  min(df$Fecha.del.Siniestro),
  by = "day",
  length.out = as.integer(max(df$Fecha.del.Siniestro)+1 - min(df$Fecha.del.Siniestro))
))

# calculamos reclamos diarios con table
right = data.frame(table(df$Fecha.del.Siniestro))
right$Var1 <- as.Date(right$Var1)

# si nuestro dataset tuviera días sin reclamos estos no aparecerían en la tabla de frecuencias
left = data.frame(all_days)

# utilizamos merge lo cual nos da un df con NA en las fechas que no hubo reclamos
reclamos_diarios <- merge(left, right, by.x = 'all_days', by.y = "Var1", all.x = TRUE)

# cambiamos el index a la columna de all_days
rownames(reclamos_diarios) <- reclamos_diarios$all_days
reclamos_diarios$all_days <- NULL

# cambiamos los NAs por 0
reclamos_diarios[is.na(reclamos_diarios)] <- 0

# calculamos la proporción en la que sucede que haya x reclamos en un día
proporciones = table(reclamos_diarios)/dim(reclamos_diarios)[1]

x.ax = as.integer(names(proporciones))
x.range = min(x.ax):max(x.ax)

# también nos aseguramos de que las proporciones incluyan todos los valores dentro del rango de N
proporciones = merge(
  data.frame(reclamos_diarios = x.range),
  data.frame(proporciones),
  by = "reclamos_diarios",
  all.x = TRUE
  )
```

```
# cambiamos el index a la columna de reclamos_diarios
rownames(proporciones) <- x.range
proporciones$reclamos_diarios <- NULL

proporciones[is.na(proporciones)] <- 0
```

```{r}

fisher.test(data.frame(proporciones, table(cut(df$Monto.del.siniestro, breaks=52)))[, c(1, 3)])

```

8.3. Apéndice C: Estimación de la probabilidad de ruina vía MMC y MMC con reducción de varianza

```
from random import random, setstate, getstate
from math import e, log as ln
from functools import reduce
import pandas as pd
import matplotlib.pyplot as plt

def rpois(n: int = 1, param: float = 1, rng=random):
    """Poisson generator based upon the inversion by sequential search from
    (https://en.wikipedia.org/wiki/Poisson_distribution#Random_variate_generation)"""
    r = []
    for _ in range(n):
        x = 0
        p = e**-param
        s = p
        u = rng()
        while u > s:
            x += 1
            p *= param/x
            s += p
        r.append(x)
    return r


def rexp(n: int = 1, param: float = 1, rng=random):
    return [-ln(u)/param for u in (rng() for _ in range(n))]

class Insurance():

    __ruin_warning = False

    def __init__(self, u: float, c: float, arr_rate: float, amt_rate:
    float, inv_rng: bool = False, initial_state=None) -> None:
        self.u = u
        self.c = c
        self.arr_rate = arr_rate
        self.amt_rate = amt_rate
        self.x = u
        self.claims_received = 0
        self.max_capital = u
```

```
        self.min_capital = u
        self._time = 0
        self._ruined = False
        self.traj = []
        if initial_state is None:
            self._initial_state = getstate()
        else:
            self._initial_state = initial_state
            setstate(initial_state)
        if inv_rng == False:
            self.rng = random
        else:
            self.rng = lambda: 1 - random()

    def __iadd__(self, delta_t: int) -> None:
        # if the company is ruined don't do anything
        if self._ruined:
            return self

        for _ in range(delta_t):
        # debug
        #   print(f"{self._time:>5}\t+{self.c:>15,.2f}")
        #   last_x = self.x
        #
            # claims
            claim_amt = rpois(param=self.arr_rate, rng=self.rng)[0]
            dt = 1/(claim_amt+1)
            for _ in range(claim_amt):
                claim = rexp(param=self.amt_rate, rng=self.rng)[0]
                self.x -= claim
                self._time += dt
                self.traj.append((self._time, self.x))

            # debug
            #   print(f"{self._time:>5}\t-{(last_x-self.x)/claim_amt:>15,.2f}")
            #

                # updating internal vars
                self.max_capital = max(self.x, self.max_capital)
                self.min_capital = min(self.x, self.min_capital)
                if self.x <= 0:
                    self.ruin()
                    break



            self._time = round(self._time)

            # premiums
            self.x += self.c
            self.traj.append((self._time, self.x))

        # debug
        #   print(f"{self._time:>5}\t-{(last_x-self.x)/claim_amt:>15,.2f}")
        #

        return self
```

```
    def ruin(self):
        self._ruined = True
        if Insurance.__ruin_warning:
            print(f"""\
    The company is ruined!!
    It survived {self._time} days and ended up with a capital of ${self.x:,.2f}.
    The highest capital achieved was ${self.max_capital:,.2f},
    The lowest capital recorded was ${self.min_capital:,.2f}""")

    def __str__(self) -> str:
        return f"""\
The company has survived for {self._time} days and has a current capital of ${self.x:,.2f}.
The highest capital achieved yet is ${self.max_capital:,.2f},
The lowest capital recorded yet is ${self.min_capital:,.2f}"""


proportions_over_time = []
failures = 0
n = 500
time = 10000

for i in range(1, n+1):
    I = Insurance(14_000_000, 2_290_000, 74.241, 3.25e-5)
    for _ in range(time):
        I += 1

    failures += I._ruined
    proportions_over_time.append(failures/i)

    #traj = pd.DataFrame(I.traj)

    #plt.plot(traj[0], traj[1])
    #plt.ylim(0, I.max_capital*1.03)
    #plt.xlim(0, I._time*1.01)
    #print(f"{I.max_capital=}\n{I.min_capital=}")

    print('\r',i, end='')

p = failures/n
print(f"Crude Monte Carlo with {n} samples until time = {time}")
print(f'\rfailure rate: {p:.2%}')
print(f'10% confidence interval: [{p - 1.96*(p*(1-p)/n)**.5:.2%}, {p + 1.96*(p*(1-p)/n)**.5:.2%}]')

# antithetic vars
import math
# estimation of number of trials at 5% of +-5%
z = 1.65
E = 0.05

n_ = z**2*p*(1-p)/E**2

# round to the nearest 10th
n_ = math.ceil(n_/10)*10

failures_ = 0
proportions_over_time_ = []

for i in range(1, n_+1, 2):
```

```
    # first run
    I = Insurance(14_000_000, 2_290_000, 74.241, 3.25e-5)
    for _ in range(time):
        I += 1

    failures_ += I._ruined
    proportions_over_time_.append(failures_/i)

    # second run
    i += 1
    I_ = Insurance(14_000_000, 2_285_016, 74.241, 3.25e-5, inv_rng = True,
    initial_state = I._initial_state)
    for _ in range(time):
        I_ += 1

    failures_ += I_._ruined
    proportions_over_time_.append(failures_/i)

    print('\r',i, end='')

p_ = failures_/n_
print(f"Variance Reduction Monte Carlo with {n_} samples until time = {time}")
print(f'\rfailure rate: {p_:.2%}')
print(f'5% confidence interval:
[{p_ - 1.65*(p_*(1-p_)/n_)**.5:.2%}, {p_ + 1.65*(p_*(1-p_)/n_)**.5:.2%}]')

plt.hlines(p, 0, n_, colors='y')
plt.hlines(p_, 0, n_, colors='red')

plt.plot(proportions_over_time[:n_], color = 'g')

plt.plot(proportions_over_time_)


plt.title(f"Proporciones de Ruina con MC, RV; n = {n_}, t = {time}")
plt.legend(['P{Ruina} con MC', 'P{Ruina} con RV', 'Proporción MC', 'Proporción RV'])
plt.xlabel('Número de simulaciones [n]')
plt.ylabel('Probabilidad')

plt.hlines(p, 0, n, colors='y')
plt.plot(proportions_over_time, color = 'g')
plt.title(f"Proporciones de Ruina con MC; n = {n}, t = {time}")
plt.legend(['P{Ruina}', 'Proporción'], loc = 'lower right')
plt.xlabel('Número de simulaciones [n]')
plt.ylabel('Probabilidad')
```