

# Cálculo de la probabilidad de ruina de una aseguradora por medio de Cramer-Lundberg, Monte-Carlo y Pollaczek-Khinchin

Mauricio Hernández - Diego Pérez - Leyre Carpinteyro - Nemesio Navarro - Ricardo Díaz

**Abstract**—En este artículo se obtendrá el cálculo de la probabilidad ruina, a partir de la aproximación de Cramér-Lundberg, Monte Carlo crudo, Monte Carlo condicionado por varianza reducida y por la fórmula de Pollaczek-Khinchin. Con el fin estimar la probabilidad de ruina y tomar decisiones la prevengan, se realiza una comparación entre los métodos. Finalmente se llega a la conclusión de que el monto del capital inicial de la compañía influirá significativamente en la probabilidad de ruina. Así como al reconocimiento del método de Pollaczek-Khinchin como la aproximación con mayor verosimilitud con respecto a la probabilidad exacta.

**Index Terms**—Teoría de Riesgo, Modelo Monte Carlo, Modelo Clásico de Cramer Lundberg, Probabilidad de Ruina, Fórmula Pollaczek-Khinchin, Optimización Estocástica, Convolución.

## I. INTRODUCCIÓN

El sector de aseguradoras funciona a partir de un contrato donde, a cambio del cobro de una prima mensual, la aseguradora se compromete a indemnizar el daño o aportar capital en caso de algún siniestro [1]. De acuerdo con Marco Caparoso, las empresas que proveen el servicio financiero tienen muy poca probabilidad de quebrar. A lo largo de la historia de este sector en México, solamente cuatro empresas han sido reportadas en quiebra [2].

Debido al comportamiento estocástico de los siniestros, se ha desarrollado la Teoría de Riesgo dentro de las ciencias actuariales, estudiado por actuarios, economistas y matemáticos. La teoría, entre las diferentes situaciones que abarca, estudia la probabilidad del incumplimiento de la aseguradora en cubrir los siniestros [3].

La estimación de incumplimiento de una aseguradora depende de su riqueza inicial, la cantidad de reclamos y la distribución de estos. Para ello, el sueco Filip Lundberg propuso un modelo que posteriormente fue estudiado por Harald Cramer quien introdujo nuevos métodos, para generar el modelo conocido como *Modelo Clásico de Cramer-Lundberg*. Posteriormente, se estableció la *Fórmula de Pollaczek-Khinchin* para obtener la probabilidad teórica exacta de quiebra en una compañía.

En este trabajo se establecerá la comparación entre ambos métodos, además de con Monte Carlo, para la estimación de probabilidad de ruina de una compañía aseguradora, de acuerdo con parámetros de capital inicial y costo de la prima.

## II. DESCRIPCIÓN DE LA PROBLEMÁTICA

Ahora bien, en este caso específico, se considera una compañía aseguradora con un capital inicial  $u$  y una prima que pagan sus asegurados denotada por  $c$ , además, existe un monto total de pagos para un tiempo  $t$  que será denotado

como  $N(t)$ , para el cual, cada vez que ocurra un evento (como un accidente), la aseguradora deberá pagar una cantidad denominada  $Z$ , la cual se asume que sigue una distribución de probabilidad  $F$ .

La problemática a estudiar se basa en el escenario hipotético de que el monto total de los pagos, sea superior al capital de la empresa para cualquier tiempo determinado  $t$ , al cual se le denominará como el **momento de ruina de la compañía** y su existencia puede tener consecuencias graves, como la salida del mercado de la empresa.

Para el desarrollo de este trabajo se utilizará una base de datos de una compañía aseguradora con 27,121 registros de siniestros de automóviles, que también incluye el monto de este y la fecha cuando sucedió. A partir de los datos, se estimará la probabilidad de ruina, según un capital inicial y una prima mensual variable, por diversos métodos y se establecerá una comparación entre estos.

## III. PREGUNTAS DE INVESTIGACIÓN

Para la elaboración de este reporte, se plantearon las siguientes preguntas:

- 1) ¿Cuál es la probabilidad de ruina de la aseguradora?
- 2) ¿Qué acciones pueden ser tomadas para que el tiempo de ruina no sea alcanzado?
- 3) ¿Qué método hace la mejor estimación?

## IV. METODOLOGÍA UTILIZADA

La realización de la presente investigación se basó en distintas etapas que permitirían obtener las herramientas necesarias para poder calcular la probabilidad de que el tiempo ruina de la compañía aseguradora exista y sea alcanzado, asumiendo que su capital se comporta de acuerdo con el modelo clásico de riesgo, el cual fué mencionado con anterioridad pero será definido y descrito con formalidad en secciones posteriores de este documento.

El primer paso, fué la aplicación de un análisis descriptivo sobre la base de datos proporcionada para poder implementar técnicas de inferencia estadística, como la estimación de parámetros, los cuales serían útiles inmediatamente después para el ajuste de posibles distribuciones de probabilidad, tanto al tamaño del monto de los siniestros como a la magnitud de las ocurrencias de estos en un periodo de tiempo determinado.

Tomando en cuenta lo anterior, el segundo paso fué precisamente la ejecución de pruebas de bondad de ajuste de tipo  $\chi^2$  y Kolmogorov-Smirnoff, así como el cálculo de intervalos de confianza para los parámetros estimados en la etapa anterior. [3]

Una vez terminada la etapa de inferencia estadística, el tercer paso es una primera aproximación a la probabilidad de ruina de la compañía por medio de simulaciones de Monte Carlo, tanto en su forma estandar, así como su alternativa de varianza mínima.

Finalmente, se emplearon otras soluciones a la probabilidad de ruina. Primeramente, se utilizó la aproximación dada por la fórmula de Pollaczek-Khinchin [4], para lo cual se tuvieron que calcular las distintas convoluciones de la distribución exponencial. En segundo lugar, se implementó la llamada aproximación de Cramer-Lundberg [3], la cual podía ser calculada utilizando una expresión  $\rho$  conocida como la **carga de riesgo**, la cual será definida posteriormente de igual forma.

Para concluir el estudio, se realizó una comparación de todos los métodos utilizados para evaluar su desempeño y viabilidad en diferentes escenarios.

## V. MARCO TEÓRICO

### A. Modelo Clásico de Cramer-Lundberg

Describe una compañía aseguradora que experimenta dos flujos contrarios de dinero: el pago de primas por los usuarios a un intervalo específico  $t$  y reclamos de seguro  $N(t)$ . Se asume que el monto de los reclamos siguen una distribución exponencial, mientras que el total de incidentes por cada elemento de tiempo sigue una distribución de Poisson. El modelo matemático sigue la siguiente expresión:

$$X_t = u + ct - \sum_{i=1}^{N_t} Z_i, \forall t > 0 \quad (1)$$

Donde:

- $u$  es el capital inicial de la empresa
- $c$  es el monto fijo por pago de primas
- $t$  es el tiempo al cual se pagan las primas
- $N(t)$  es el número total de eventos para el tiempo  $t$
- $Z_i$  es el costo para la aseguradora del reclamo correspondiente al evento  $i$

El tiempo de Ruina está dado por:

$$T = \min\{t \geq 0 : X(t) < 0\}$$

Además, también se estableció que:

$$\begin{aligned} N(t) &\sim \text{Poisson}(\lambda) \\ Z_i &\sim \text{Exp}(\mu) \end{aligned}$$

Con ello, queda definido el modelo del capital de la compañía aseguradora, a partir del cual puede calcularse la probabilidad de la existencia del momento de ruina, la cual se representa como  $\varphi(t)$ :

$$\varphi(t) = P(X(t) < 0, t > 0 \mid X(0) = u)$$

Entonces, puede afirmarse que  $1 - \varphi$  es la probabilidad de *supervivencia* de la compañía. Todorova [5] menciona en su artículo que utilizando la independencia entre los conteos  $N(t)$  y los montos  $Z_1, Z_2 \dots$  se puede deducir que:

$$E[X(t)] = t(c - \lambda\mu) \geq 0$$

Donde  $\mu$  es la media de la distribución de  $Z_i$  y  $\lambda$  es el parámetro de la distribución de  $N(t)$ . De lo anterior, se deduce la *condición de capitalización*.

$$c > \lambda\mu$$

El capital  $c$  de la empresa, para el tiempo  $t$ , debe ser mayor que el egreso por el concepto de los reclams para el mismo período de tiempo. De no cumplirse esta condición, la probabilidad  $\varphi(t)$  será siempre igual a 1.

Algunos aspectos importantes sobre el modelo son la discretización del tiempo en todo momento, debido a que se utilizan intervalos definidos (días, meses años), por lo que el conteo de incidentes es siempre entero y correspondiente a la unidad de tiempo en cuestión.

Por otra parte, la gráfica de la función  $X(t)$  no será una curva suave debido a la discretización del tiempo. Las trayectorias serán saltos abruptos correspondientes a la entrada o salida de capital de las reservas de la empresa.

### B. Estimación de parámetros

En V-A se menciona que el modelo de Cramer-Lundberg requiere de dos parámetros,  $\lambda$  y  $\mu$ . Se asume que  $N(t) \sim \text{Poisson}(\lambda)$ , con  $\lambda$  siendo la cantidad de siniestros esperada por unidad de tiempo. Su función de probabilidad es

$$P[X = k] = \frac{e^{-\lambda} \lambda^k}{k!} \quad (2)$$

Con  $k = 1, 2, \dots$  como la cantidad de siniestros.

Por tanto,  $E[X] = \lambda$  debido a que

$$\begin{aligned} E[X] &= \sum_{k=0}^{\infty} k \left( \frac{e^{-\lambda} \lambda^k}{k!} \right) \\ &= e^{-\lambda} \sum_{k=1}^{\infty} \frac{k \lambda^k}{k!} \\ &= e^{-\lambda} \sum_{k=1}^{\infty} \frac{\lambda^k}{(k-1)!} \\ &= \lambda e^{-\lambda} \sum_{k=1}^{\infty} \frac{\lambda^{k-1}}{(k-1)!} \\ &= \lambda e^{-\lambda} \sum_{j=0}^{\infty} \frac{\lambda^j}{j!} \\ &= \lambda e^{-\lambda} e^{\lambda} \\ &= \lambda \end{aligned}$$

Por tanto, un estimador insesgado de  $\lambda$  es

$$\lambda = \frac{1}{n} \sum_{i=1}^n x_i = \bar{x} \quad (3)$$

Por otro lado, para  $Z_i$  se asume que  $Z_i \sim \text{Exp}(\mu)$ , con  $\mu$  generalmente interpretado como una tasa de ventos que suceden cada determinado periodo de tiempo. Su función de probabilidad es:

$$f_X(x) = \mu e^{-\mu x}$$

Ocasionalmente, la distribución exponencial se escribe en términos de una parametrización alternativa  $\beta = \frac{1}{\mu}$ . Por lo tanto, la función de densidad se reescribe como:

$$f_X(x) = \frac{1}{\beta} e^{-\frac{x}{\beta}}$$

Y será la utilizada en el presente experimento.

Debido a la naturaleza continua de la distribución exponencial, su media debe calcularse con el uso de una integral impropia sobre la función de densidad:

$$\begin{aligned} E[X] &= \int_0^{\infty} x \mu e^{-\mu x} dx \\ &= \mu \left[ \frac{-xe^{-\mu x}}{\mu} \Big|_0^{\infty} + \frac{1}{\mu} \int_0^{\infty} e^{-\mu x} dx \right] \\ &= \mu \left[ 0 + \frac{1}{\mu} \frac{-e^{-\mu x}}{\mu} \Big|_0^{\infty} \right] \\ &= \mu \frac{1}{\mu^2} = \frac{1}{\mu} \end{aligned}$$

Entonces si  $\beta = \frac{1}{\mu}$ , se afirma que  $E[X] = \beta$

El estimador de máxima verosimilitud está dado también por

$$\mu = \bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$$

Por lo tanto, considerando la parametrización, se reescribe:

$$\beta = \frac{1}{\mu} = \frac{1}{\bar{x}}$$

Dado lo anterior,  $\frac{1}{\bar{x}}$  será el estimador insesgado del parámetro de la distribución.

### C. Pruebas de bondad de ajuste e intervalos de confianza

En V-A y V-B, se mencionan los supuestos realizados para el desarrollo del modelo; sin embargo, se puede verificar el correcto ajuste de los datos a las distribuciones propuestas. Para ello, se aplicarán dos criterios:

- Cálculo del intervalo de confianza de 95% para  $\lambda$  y  $\mu$ , corroborando que el estimado a partir de los datos se ubique dentro de este.
- Realización una prueba de bondad de ajuste mediante una prueba de hipótesis de tipo Kolmogorov-Smirnov para el caso de la distribución exponencial, y de tipo  $\chi^2$  para el caso de la distribución de Poisson.

Guerriero [6] señala una forma simple de calcular el intervalo de confianza para una distribución de Poisson, por medio del uso del Teorema del Límite Central:

$$\begin{aligned} F_{low} &= \left( 1 - \frac{1.96}{\sqrt{k-1}} \right) \frac{k}{T} \\ F_{upp} &= \left( 1 + \frac{1.96}{\sqrt{k-1}} \right) \frac{k}{T} \end{aligned}$$

Dada una serie de eventos  $K$ , en un periodo de tiempo  $T$ . Zheng [7] muestra el intervalo de confianza resultante de nivel  $1 - \alpha$  para una distribución exponencial de parámetro  $\beta = \frac{1}{\mu}$ .

El cual es el resultado de una manipulación con una función pivote, y se expresa de la siguiente manera:

$$\left[ \frac{\chi_{2n}^2(\alpha/2)}{2 \sum_{i=1}^n X_i}, \frac{\chi_{2n}^2(1-\alpha/2)}{2 \sum_{i=1}^n X_i} \right]$$

Una aproximación simple puede ser dada utilizando una estimación normal de la distribución  $\chi^2$  que resulta en las siguientes expresiones:

$$\begin{aligned} \beta_{lower} &= \hat{\beta} \left( 1 - \frac{1.96}{\sqrt{n}} \right) \\ \beta_{upper} &= \hat{\beta} \left( 1 + \frac{1.96}{\sqrt{n}} \right) \end{aligned}$$

Por otro lado, para las pruebas de hipótesis se plantea que:

$H_0$ : Los datos siguen una distribución  $F$ .

$H_A$ : Los datos no siguen una distribución  $F$ .

Para aceptar o descartar  $H_0$ , se emplea el uso del valor crítico  $p$ , que varía dependiendo de la distribución  $F$ , comparándolo al nivel de significancia  $\alpha$  establecido ( $\alpha = 0.05$ ).

Si el estadístico es mayor que el valor crítico  $p > \alpha$ , se rechaza  $H_0$ ; en caso contrario, no hay evidencia estadística suficiente para rechazar  $H_0$ .

Cuando la prueba es para una distribución Chi-Cuadrada, el estadístico  $s$  se calcula como:

$$s = \sum_{i=1}^k \frac{(O_i - E_i)^2}{E_i}$$

Donde  $i$  denota al número de clase y:

$$E_i = n(F(X_u) - F(X_l))$$

Es la frecuencia esperada para la clase  $i$ , donde  $X_u$  denota al límite superior de la clase y  $X_l$  al límite inferior y  $n$  al número total de datos.

Sabiendo que  $s$  sigue aproximadamente una distribución Chi-cuadrada con  $k - c$  grados de libertad,  $k$  corresponde al número de clases y  $c$  a la cantidad de parámetros estimados.

Para la prueba de Kolmogorov-Smirnov, el estadístico para evaluar la prueba depende de la similitud entre la función de distribución acumulada empírica  $\tilde{F}$  y la teórica  $F$ .

Donde  $\tilde{F}$  puede calcularse como:

$$\tilde{F}(x_i) = N(x < x_i) / n \text{ para } i \in 1, 2, \dots, n$$

Siendo  $N(x < x_i)$  la cardinalidad del conjunto de valores de  $x < x_i$ .

El estadístico se encarga de identificar la máxima diferencia que existe entre estas distribuciones, este está definido como:

$$s = \max_{1 \leq i \leq n} |F(x_i) - \tilde{F}(x_i)|$$

El criterio para determinar el resultado de la prueba se hace mediante valores críticos  $v_{crit}$  estimados para niveles de significancia y tamaños de muestra dados, estos son obtenidos frecuentemente mediante simulaciones. Los valores críticos que se utilizarán en este trabajo son los expuestos por Lilliefors en [8] para un nivel de significancia de 5% y muestra grande.

#### D. Expresión exacta de la probabilidad de ruina

García [3], muestra una expresión que sirve para calcular la probabilidad de ruina exacta, bajo el supuesto de que el tamaño de los montos de los reclamos sigue una distribución exponencial, por lo que, dado el hecho de que se utilizará el exacto mismo criterio para los reclamos anexados en la base de datos sobre la cual se basa el presente documento, puede considerarse como válida. De requerirse su demostración, uno puede dirigirse a [3].

Así pues, la expresión de la probabilidad de ruina exacta está definida como:

$$\psi(u) = \frac{1}{1+\theta} \exp \left\{ \frac{-\theta u}{(1+\theta)E[x]} \right\} \quad (4)$$

Donde  $\theta = \frac{c}{\lambda\mu} - 1 > 0$

#### E. Método de Monte Carlo

Es una forma en la que se puede estimar probabilidades que en la práctica, es difícil de conocer. Por la Ley de los Grandes Números, siguiendo a Maya [3], se puede estimar para propósitos de este trabajo, la probabilidad de ruina de la aseguradora a partir de una muestra de variables aleatorias, que sean independientes entre sí y se distribuyan por medio de la distribución de Poisson.

$$E[G(Y)] \approx \frac{1}{n} \sum_{i=1}^n G(Y_i) \quad (5)$$

Donde  $Y$  es la variable aleatoria que se desea estimar,  $Y_i$  una muestra,  $G$  una función que indica cuando el capital de la empresa  $U(t)$  es inferior a 0, de acuerdo a  $Y$ .

$$G(Y) = \begin{cases} 1 & \text{si } U(t) < 0 \text{ para } t > 0 \\ 0 & \text{o.c.} \end{cases} \quad (6)$$

Con este método se obtiene la probabilidad de ruina de acuerdo con la cantidad de simulaciones realizadas, por esto mismo se le conoce como *Método de Monte Carlo crudo*.

Con Monte Carlo se puede realizar la estimación del error cometido. De acuerdo con Maya en [3], este error es aproximadamente  $\frac{\sigma}{\sqrt{n}}$ . Sin embargo, para este resultado se debe de conocer la desviación estándar poblacional de  $Y$ . Por ello, se realiza una estimación a partir de la ecuación:

$$\hat{\sigma}^2 = \frac{1}{n-1} \sum_{i=1}^n (G(Y_i) - \hat{I}_n)^2$$

Donde  $\hat{I}_n = \frac{1}{n} \sum_{i=1}^n G(Y_i)$ .

Para la reducción de la varianza de una determinada iteración  $i$ , dependerá de la varianza de  $i-1$ , además de la media acumulada hasta  $i$  e  $i-1$  como se muestra en ecuación siguiente:

$$Var_i = (1 - \frac{1}{i-1})Var_{i-1} + i(\hat{I}_i - \hat{I}_{i-1})^2$$

Así como de la definición de la varianza:

$$Var[X] = E[X^2] - E[X]^2$$

Donde la varianza obtenida a partir de ambas ecuaciones debe ser menor al error  $\frac{\sigma}{\sqrt{n}}$ .

#### F. Fórmula de Pollaczek-Khinchin

De acuerdo con Maldonado [4], describe la probabilidad de no ruina, donde el número de reclamos sigue una distribución de Poisson homogénea, por medio de la teoría de colas. La cola integrada de una variables aleatoria  $Z$  de una distribución acumulada  $F$ , denotada como  $F_Z^s(y)$  es definida por

$$F_Z^s(y) = \begin{cases} \frac{1}{\mu} \int_0^x [1 - F_Z(y)] dy & \text{para } x \geq 0 \\ 0, & \text{o.c.} \end{cases}$$

La convolución de una distribución  $F$  es igual a la distribución de la suma de variables aleatorias  $F$  distribuidas.

$$\sum_{n=1}^N X_i \sim F^{*(n)} \quad \text{si } X_i \sim F$$

Dada la fórmula de Pollaczek-Khinchin:

$$\phi(u) = \left(1 - \frac{\lambda\mu}{c}\right) \sum_{n=1}^{\infty} \left(\frac{\lambda\mu}{c}\right)^n (F_Z^s)^{*n}(u) \quad (7)$$

Se necesita determinar la  $n$ -ésima convolución de  $F_Z^s$ , la cual es la distribución de cola de  $F_Z$ :

$$F_Z^s(u) = 1 - e^{-\frac{u}{\mu}}$$

Se tiene entonces que la convolución que se busca es la distribución de la suma de  $n$  variables exponencialmente distribuidas:

Para

$$\sum_{n=1}^N X_i \sim F_Z^{*(n)} \quad \text{con } X_i \sim \exp\left(\frac{1}{\mu}\right)$$

se debe determinar  $F_Z^{*(n)}$ .

Es un resultado conocido que la función  $F_Z^{*(n)}$  se trata de la función Erlang, la cual es:

$$F_Z^{*(n)}(u) = 1 - \sum_{k=0}^{n-1} \frac{\left(\frac{1}{\mu}u\right)^k e^{-\frac{1}{\mu}u}}{k!}$$

Donde la distribución de cola  $F_Z^{s*(n)}(u)$  es:

$$F_Z^{s*(n)} = 1 - F_Z^{*(n)} = \sum_{k=0}^{n-1} \frac{\left(\frac{1}{\mu}u\right)^k e^{-\frac{1}{\mu}u}}{k!}$$

Sustituyendo en la formula de Pollaczek-Khinchin:

$$\phi(u) = \left(1 - \frac{\lambda\mu}{c}\right) \sum_{n=1}^{\infty} \left(\frac{\lambda\mu}{c}\right)^n \sum_{k=0}^{n-1} \frac{\left(\frac{1}{\mu}u\right)^k e^{-\frac{1}{\mu}u}}{k!}$$

Notemos que la suma indexada con  $k$  tiene un límite superior que depende del índice de la suma previa, por lo tanto, es posible cambiar el orden de de dichas sumas:

$$\phi(u) = \left(1 - \frac{\lambda\mu}{c}\right) e^{-\frac{1}{\mu}u} \sum_{k=0}^{\infty} \sum_{n=k+1}^{\infty} \left(\frac{\lambda\mu}{c}\right)^n \frac{\left(\frac{1}{\mu}u\right)^k}{k!}$$

Notemos que la suma de índice  $n$  es geométrica y converge, ya que  $c > \lambda\mu$ , por lo tanto:

$$\phi(u) = \left(1 - \frac{\lambda\mu}{c}\right) e^{-\frac{1}{\mu}u} \sum_{k=0}^{\infty} \frac{\left(\frac{1}{\mu}u\right)^k}{k!} \frac{\left(\frac{\lambda\mu}{c}\right)^{k+1} \left(1 - \left(\frac{\lambda\mu}{c}\right)^{\infty}\right)}{1 - \frac{\lambda\mu}{c}}$$

Notemos que  $\left(\frac{\lambda\mu}{c}\right)^{\infty}$  converge a 0, ya que  $\left|\frac{\lambda\mu}{c}\right| < 1$ , entonces:

$$\phi(u) = \left(1 - \frac{\lambda\mu}{c}\right) e^{-\frac{1}{\mu}u} \sum_{k=0}^{\infty} \frac{\left(\frac{\lambda\mu}{c} \cdot \frac{1}{\mu}u\right)^k \frac{\lambda\mu}{c}}{k! \left(1 - \frac{\lambda\mu}{c}\right)}$$

Lo cual puede escribirse como:

$$\phi(u) = \frac{\left(1 - \frac{\lambda\mu}{c}\right) e^{-\frac{1}{\mu}u}}{\left(1 - \frac{\lambda\mu}{c}\right)} \frac{\lambda\mu}{c} \sum_{k=0}^{\infty} \frac{\left(\frac{\lambda\mu}{c}\right)^k}{k!}$$

Se sabe que  $\sum_{k=0}^{\infty} \frac{x^k}{k!} = e^x$ , entonces finalmente:

$$\begin{aligned} \phi(u) &= e^{-\frac{1}{\mu}u} \frac{\lambda\mu}{c} e^{\frac{\lambda\mu}{c}} \\ \phi(u) &= \frac{\lambda\mu}{c} e^{\left(\frac{\lambda\mu}{\mu c} - \frac{1}{\mu}\right)u} \end{aligned}$$

Sin embargo, como el propósito de este trabajo es obtener la probabilidad de ruina  $\psi(u)$ , entonces se tiene que:

$$\psi(u) = 1 - \phi(u)$$

Por lo tanto:

$$\psi(u) = 1 - \frac{\lambda\mu}{c} e^{\left(\frac{\lambda\mu}{\mu c} - \frac{1}{\mu}\right)u} \quad (8)$$

### G. Aproximación de Cramer-Lundberg

La aproximación de Cramer-Lundberg requiere del cálculo de un límite que requiere la definición de 3 objetos nuevos.

Primero, se define lo que se conoce como la **carga de seguridad de la compañía**, denotada como  $\rho$  y representada por:

$$\rho = \frac{c - \lambda\mu}{\lambda\mu} > 0 \quad (9)$$

La condición es vital para la supervivencia de la compañía, ya que si por alguna razón  $\rho \leq 0$ , implicaría que  $c \leq \lambda\mu$ , lo cual violaría la condición de capitalización de la compañía, definida previamente.

La segunda definición importante para esta aproximación es la función  $h : \mathbb{R} \rightarrow \mathbb{R}$

$$h(r) = \int_0^{\infty} \exp(rz) dF(z) - 1 \quad (10)$$

Se asume que existe  $0 < r_{\infty} \leq \infty$  tal que  $h(r) \uparrow +\infty$  cuando  $r \uparrow r_{\infty}$ . Esto significa que para algún  $r > 0$ ,  $m_Z(r) < \infty$ , y por lo tanto  $F$  es de cola ligera. Observe que  $h(0) = 0$ ,  $h$  es creciente, convexa y continua. [4]

El tercer constructo necesario es un coeficiente al que se le denomina *de ajuste*. Y está expresado por:

$$R = \frac{\rho}{\mu(1 + \rho)} \quad (11)$$

Con las definiciones recién dadas, Maldonado [4] plantea la aproximación de Cramer-Lundberg para una distribución exponencial del tamaño de las reclamaciones, partiendo de:

$$\lim_{u \rightarrow \infty} (\exp(Ru) \psi(u)) = \frac{1}{1 + \rho}$$

Para llegar a:

$$\psi(u) = \frac{1}{1 + \rho} \exp\left(-\frac{\rho u}{\mu(1 + \rho)}\right) \quad (12)$$

Donde  $\psi(u)$  es la probabilidad de ruina.

El desarrollo de la evaluación del límite se muestra en su totalidad en [4], y no se incluirá en el presente documento debido a que su extensión y dificultad están por fuera del rango de cobertura de este curso, y por lo tanto también de este documento. Sin embargo, en caso de que el lector desee consultar la derivación completa, se anexa el documento digital en las referencias.

## VI. ANÁLISIS DE DATOS

Después de ordenar la base de datos por fechas, se obtuvieron los parámetros correspondientes al promedio de siniestros por día con  $\bar{\lambda} = 74.30$  y el promedio de pagos por siniestro de  $\bar{\mu} = 30771.37$ .

Por medio del Teorema Central del Límite, se estimaron los intervalos de confianza para  $\hat{\lambda}$  y  $\hat{\mu}$  poblacionales, obteniendo con un 95% de confianza que  $\hat{\lambda} \in [73.42, 75.19]$  y  $\hat{\mu} \in [30405.155, 31137.6]$ .

Para la cantidad de siniestros por día, se realizó la prueba de bondad de ajuste Chi-Cuadrada con un 95% de confianza y en la cual se obtuvo  $p_{value} = 0.1654$ . Por tanto, como  $p_{value} > \alpha$ , no hay evidencia estadística suficiente para rechazar que los datos poseen una distribución de Poisson, como se puede ver en la Figura 1 y Figura 2.

Con respecto al monto de los siniestros, la hipótesis que proviene de una distribución exponencial tampoco existe evidencia estadística suficiente para ser rechazada, debido a que su  $p_{value} > \alpha$ , con  $p_{value} = 0.2467$ . Por tanto, el monto

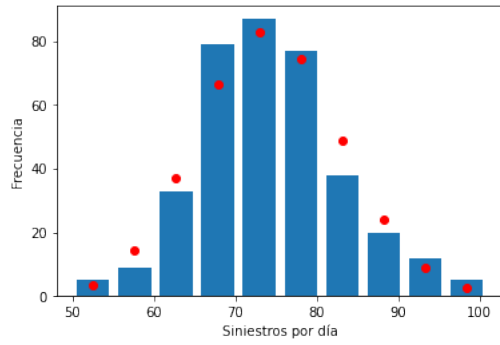


Fig. 1. Prueba de bondad de ajuste Poisson para la cantidad de siniestros por día

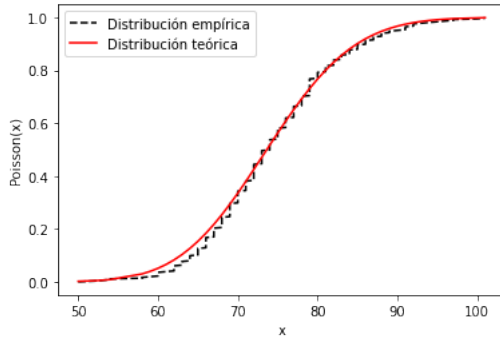


Fig. 2. Curvas de distribución teórica y empírica para Poisson

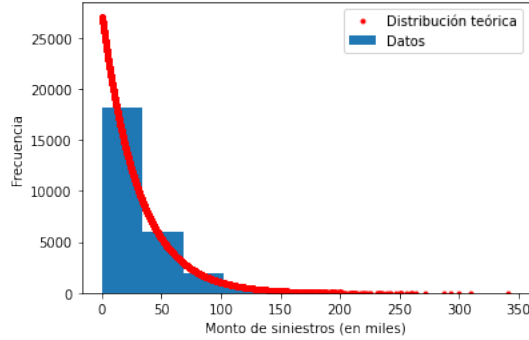


Fig. 3. Prueba de bondad de ajuste Exponencial para el monto de los siniestros

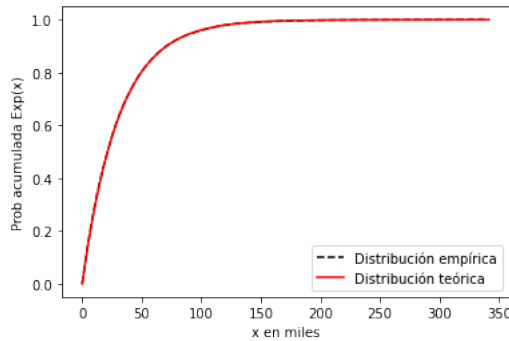


Fig. 4. Curvas de distribución teórica y empírica para exponencial

de los siniestros sigue una distribución exponencial, como se muestra en la Figura 3 y Figura 4.

Estableciendo el Modelo Clásico de Cramer-Lundberg en la ecuación 1, de acuerdo con los parámetros estimados, se puede ver un comportamiento de una simulación del capital de la compañía  $X_t$  como el mostrado en Figura 5, donde el costo de la prima total estuvo determinado por  $c = 1.1(\lambda\mu) = 2,515,083.64$ , es decir, un costo aproximado de 33,850.38mxn por cliente al mes; y el capital inicial es  $u = 0$ .

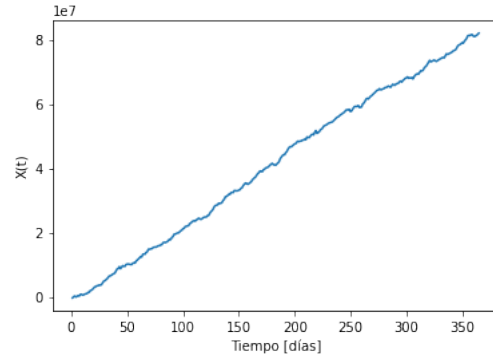


Fig. 5. Gráfica de capital de la compañía X al tiempo t, con  $u = 0$  y  $c = 2,515,083.64$

Para 50 simulaciones de los datos, con mismos parámetros de  $\lambda, \mu, c$  y  $u$ , la Figura 6 muestra un comportamiento hacia la alza a lo largo de un año.

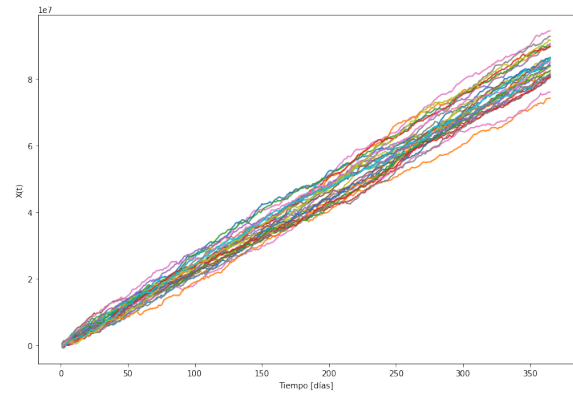


Fig. 6. 50 simulaciones, a partir de  $\lambda$  y  $\mu$  obtenidas,  $u = 0$  y  $c = 2,515,083.64$

Para la obtención de la probabilidad de ruina primero se calculó de manera exacta por la aproximación de Cramer-Lundberg en ecuación 12, que resulta en  $\psi(u = 0) = 0.9090$ . A partir de  $\psi(u)$  se puede establecer una comparación con las probabilidades obtenidas por el método de Monte Carlo crudo y Monte Carlo Condicionado por reducción de varianza

#### A. Método de Monte Carlo

En la Figura 7 se muestra el resultado de la probabilidad de ruina por el método de Monte Carlo tras 1000 simulaciones, el cual arrojó un resultado de 0.272 con un capital inicial  $u = 0$ .

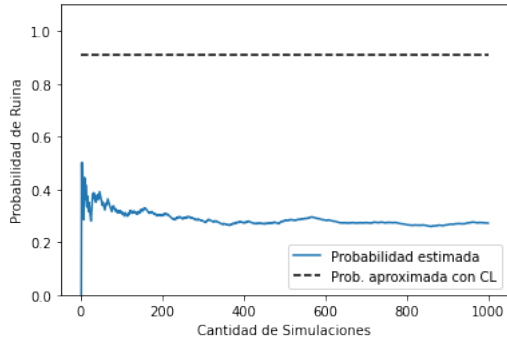


Fig. 7. Probabilidad de ruina con Monte Carlo crudo para 1,000 simulaciones y  $u = 0$

### B. Método de Monte Carlo con varianza reducida

Con este método, la probabilidad de ruina de resultó 0.254. Siguiendo a la Figura 8, se puede apreciar la estabilización de la estimación de  $\psi(u = 0)$  poco después de la simulación 200.

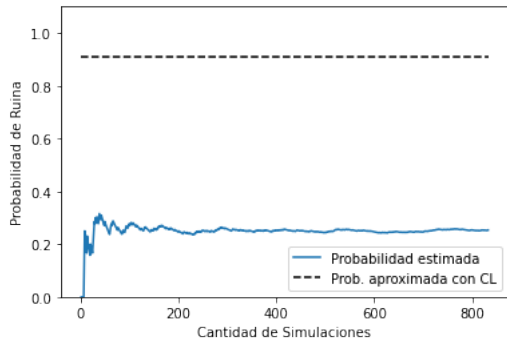


Fig. 8. Probabilidad de ruina - Monte Carlo Condicionado por reducción de varianza para 1,000 simulaciones y  $u = 0$

### C. Aproximación de Cramer-Lundberg

Por medio de la aproximación del método de Cramer-Lundberg se observa una reducción exponencial de la probabilidad de ruina al aumentar la capital inicial como se observa representado graficamente en la Figura 9.

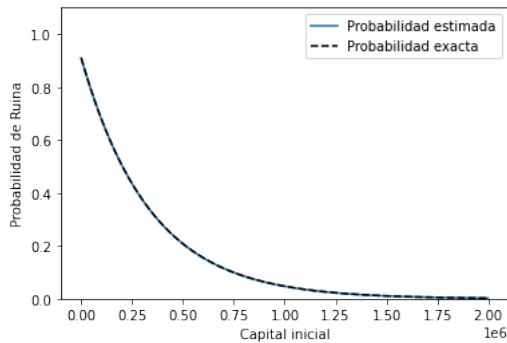


Fig. 9. Probabilidad de ruina - Aproximación de Cramer-Lundberg

### D. Pollaczek-Khinchin

1) *Aproximado*: En la Figura 10 se muestra una comparación entre la probabilidad exacta calculada con la ecuación 4 y la probabilidad estimada por la aproximación computacional de Pollaczek-Khinchin. Se puede apreciar que para  $u < 1000$ , ambos métodos muestran una probabilidad muy similar de ruina para la compañía; sin embargo, para  $u > 1000$  comienza un decrecimiento para la probabilidad.

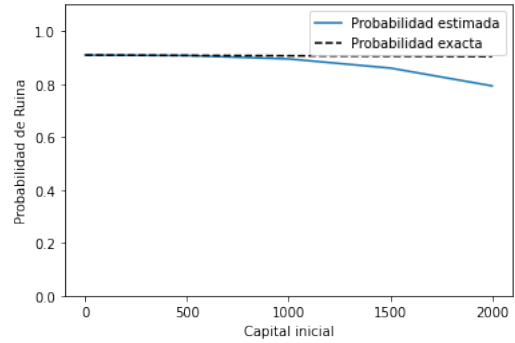


Fig. 10. Probabilidad de ruina - Pollaczek-Khinchin aproximado

2) *Analítico*: Implementando la ecuación 8 de Pollaczek Khinchin, con los parámetros obtenidos de  $\lambda$  y  $\mu$ ; al compararla con la fórmula del cálculo de la probabilidad exacta en ecuación 4, se muestra un mismo comportamiento según varía el monto del capital inicial. Por tanto, la probabilidad de ruina reduce conforme aumenta el capital inicial  $u$ .

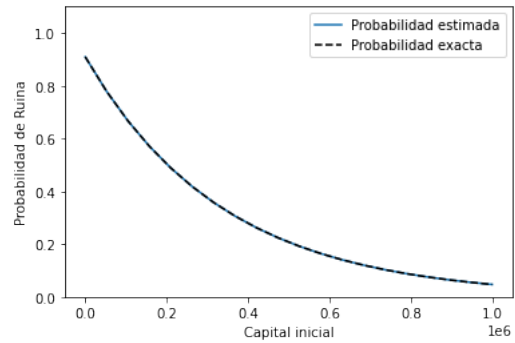


Fig. 11. Probabilidad de ruina - Pollaczek-Khinchin analítica

## VII. DISCUSIÓN DE RESULTADOS

Con base en los resultados obtenidos podemos ver que la probabilidad de ruina obtenida mediante simulaciones de Monte Carlo está muy alejada de la probabilidad exacta, siendo además muy costosos computacionalmente, ya que para obtener resultados más precisos se requiere la generación de numerosas réplicas, por lo cual, es evidente que estos métodos fueron los que tuvieron un peor desempeño.

Capital Inicial	Monte Carlo	Monte Carlo Var. Reducida
1	0.251	0.264
10	0.251	0.273
100	0.254	0.269
1000	0.264	0.248

TABLE I

COMPARACIÓN DE LA PROBABILIDAD DE RUINA ENTRE MÉTODO MONTE CARLO Y MONTE CARLO CON VARIANZA REDUCIDA PARA DIFENTES CAPITALES INICIALES

Mediante la aproximación computacional de Pollaczek-Khinchin se obtienen resultados muy cercanos a la probabilidad exacta cuando el capital inicial es cercano a 0, lo cual limita mucho al método, sin embargo, para casos específicos en donde se considere un capital pequeño o nulo es un buen método, ya que su implementación es sencilla y se obtiene directamente de la fórmula de Pollaczek-Khinchin.

Capital Inicial	Pollaczek Khinchin	Pollaczek Khinchin Analítico
1	0.909091	0.909088
10	0.909091	0.909064
100	0.909076	0.908822
1000	0.894543	0.906409

TABLE II

COMPARACIÓN DE LA PROBABILIDAD DE RUINA CON LA POLLACZEK KHINCHIN Y POLLACZEK KHINCHIN ANALÍTICO PARA DIFENTES CAPITALES INICIALES

Finalmente, la implementación de la aproximación de Cramer-Lundberg y de la expresión exacta de la fórmula de Pollaczek-Khinchin fueron realizadas tras obtenerse de manera analítica, es decir, en ambas se obtiene la probabilidad de ruina exacta para el modelo, por lo cual se compararán de acuerdo con el nivel de exigencia conceptual y de cálculo detrás de sus desarrollos.

Capital Inicial	Cramer Lundberg	Exacta
1	0.909088	0.909088
10	0.909064	0.909064
100	0.908822	0.908822
1000	0.906409	0.906409

TABLE III

COMPARACIÓN DE LA PROBABILIDAD DE RUINA OBTENIDA CON CRAMER LUNDBERG Y LA PROBABILIDAD EXACTA PARA DIFENTES CAPITALES INICIALES

Se concluyó que para el caso analizado, de reclamos exponencialmente distribuidos, obtener una expresión reducida de la fórmula de Pollaczek-Khinchin resulta más simple que hacerlo mediante Cramer-Lundberg, lo cual se debe a que para el caso exponencial la enésima convolución simplifica significativamente la sumatoria infinita a calcular, es por ello que se concluye que este método es el mejor para obtener la probabilidad de ruina para el modelo de riesgo estudiado.

## VIII. CONCLUSIONES

Retomando los análisis de los resultados obtenidos, se tiene que la probabilidad de ruina obtenida puede ser calculada con cierto grado de precisión, sin embargo, la magnitud exacta para el modelo varía dependiendo del capital inicial con el que se cuente. Distintas instancias fueron generadas, a las cuales se

les asignaron distintos valores de  $u$ , más específicamente, se consideró  $u = 1$ ,  $u = 10$ ,  $u = 100$  y  $u = 1000$ . Utilizando los resultados obtenidos, es posible calcular la probabilidad  $\psi$  para cualquier valor de  $u$  deseado. Al aumentar el capital inicial del modelo, se reduce la probabilidad de que el tiempo de ruina sea alcanzado.

De todos los métodos implementados, se concluyó que el más apropiado es el método de Pollaczek-Khinchin analítico, para el cual se sugiere buscar extender la fórmula para trabajar con capital inicial de mayor tamaño. Además, implementar el método de Taylor igualmente en Pollaczek, y extenderlo para poder trabajar otro tipo de modelos diferente al exponencial. Por otro lado, lograr implementar una alternativa del Método de Montecarlo para mejores aproximaciones.

## REFERENCES

- [1] BBVA, "Los seguros: cuántos tipos existen y cuánto cuestan," 2021, last accessed Octubre 16 2021. [Online]. Available: <https://www.bbva.com/es/los-seguros-cuantos-tipos-existen-y-cuanto-cuestan/>
- [2] Y. Torres, "Difícil que ocurra una quiebra en el sector asegurador," 2021, last accessed Octubre 16 2021. [Online]. Available: <https://www.elfinanciero.com.mx/economia/difícil-que-ocurra-una-quiebra-en-el-sector-asegurador/>
- [3] B. I. G. Maya, "Estimación numérica de la probabilidad de ruina: Caso subexponencial," Ph.D. dissertation, Universidad Autónoma Metropolitana-Iztapalapa, División de Ciencias Básicas e Ingeniería, Febrero 2016.
- [4] A. D. Maldonado, "Probabilidad de ruina con el modelo clásico de Cramer-Lundberg para distribuciones de Cola Ligera," Ph.D. dissertation, Universidad Tecnológico Mixteca, Julio 2011.
- [5] E. T. et. al., "Probabilidad de ruina en el modelo clásico de Cramer-Lundberg," *Temas de Ciencia y Tecnología*, vol. 15, no. 45, pp. 9–18, 2011.
- [6] V. Guerriero, "Power law distribution: Method of multi-scale inferential statistics," *Journal of Modern Mathematics Frontier*, vol. 1, no. 1, pp. 21–28, 2012.
- [7] S. Zheng, "Exact confidence intervals," 2017, last accessed Octubre 05 2021. [Online]. Available: <https://people.missouristate.edu/songfengzheng/Teaching/MTH541/Lecture%20notes/CI.pdf>
- [8] H. W. Lilliefors, "On the kolmogorov-smirnov test for the exponential distribution with mean unknown," *Journal of the American Statistical Association*, vol. 64, no. 325, pp. 387–389, 1969. [Online]. Available: <https://www.tandfonline.com/doi/abs/10.1080/01621459.1969.10500983>
- [9] F. Dufresne and H. U. Gerber, "Three methods to calculate the probability of ruin," *ASTIN Bulletin*, vol. 19, no. 1, p. 71–90, 1989.
- [10] J. M. van Zyl, "Application of the kolmogorov-smirnov test to estimate the threshold when estimating the extreme value index," *Communications in Statistics - Simulation and Computation*, vol. 40, no. 2, pp. 199–207, 2011. [Online]. Available: <https://doi.org/10.1080/03610918.2010.533227>



## IX. APÉNDICE

El código entero que fué usado para realizar todo el análisis se incluye a continuación

```
1 from google.colab import drive
2 drive.mount('/content/drive/')
3
4 import numpy as np
5 import pandas as pd
6 import scipy.stats as ss
7 from scipy.optimize import curve_fit
8 from scipy.special import factorial
9 from scipy.stats import poisson
10 import math as m
11 import matplotlib.pyplot as plt
12 %matplotlib inline
13
14 df = pd.read_csv('/content/drive/Shared drives/Optimizacion Estocastica/Optimizacion Estocastica/Reto/
15                 Datos_aseguradora.csv', encoding='latin-1')
16
17 # Estadística descriptiva
18 df.describe()
19
20 # Valores faltantes en la base de datos por columna
21 df.info()
22
23 # Ordenar por fecha
24 df = df.set_index(['Fecha del Siniestro'])
25 df = df.sort_index()
26 df
27
28 # Obtencion de promedio de pagos de siniestros
29 cost_siniestros = df['Monto del siniestro']
30 cost_siniestros = pd.DataFrame(cost_siniestros)
31 cost_siniestros.columns = ['Promedio de pago']
32
33 mu = cost_siniestros.mean()[0]
34 mu
35
36 # Promedio de siniestros por dia
37
38 num_siniestros = df.groupby(df.index).count()['Monto del siniestro']
39 num_siniestros = pd.DataFrame(num_siniestros)
40 num_siniestros.columns = ['Promedio de siniestros por dia']
41 lambd = num_siniestros.sum()[0]/365
42 lambd
43
44 # Promedio de pago por dia
45
46 #  $c > \mu * \lambda$ 
47  $\mu * \lambda$ 
48
49 # Comportamiento del capital total
50
51 # Parametros para la estimacion de la probabilidad de ruina
52 U=0 # Capital inicial
53 theta=0.1 # Ganancia neta
54  $c = \lambda * \mu * (1 + \theta)$  # Prima por unidad de tiempo
55
56 # Monto de siniestros por dia
57 cost_siniestros_por_dia = df.groupby(df.index).sum()['Monto del siniestro']
58 cost_siniestros_por_dia
59
60 # Acumulado de montos de siniestros por dia
61 sum_Z = [cost_siniestros_por_dia[0:i+1].sum() for i in range(len(cost_siniestros_por_dia))]
62
63 # Grafica de capital de la compania (X) al tiempo t
64 t = np.array(list(range(len(cost_siniestros_por_dia))))
65  $X = U + c * t - \text{sum\_Z}$ 
66
67 plt.figure(figsize=(15,10))
68 plt.plot(t,X)
69
70 # Distribucion de la tasa de siniestro por dia
71 # Estimacion por intervalo del parametros de la distribucion Poisson
72
73 # Datos
74 z_95 = ss.norm.ppf(.975)
```

```

75 N = df.shape[0]
76 lambd
77
78 # Calculo de intervalo
79 lim_inf = lambd - z_95*m.sqrt(lambd/N)
80 lim_sup = lambd + z_95*m.sqrt(lambd/N)
81 intervalo = [lim_inf,lim_sup]
82 intervalo
83
84 #Prueba de bondad de ajuste (Poisson)
85 datos_poisson = df.groupby(df.index).count()['Monto del siniestro']
86 datos_poisson=np.array(datos_poisson.sort_values(axis=0))
87
88 entries, bin_edges, patches = plt.hist(datos_poisson, bins=10, label='Conteo')
89
90 # Funcion de distribucion (Funcion de densidad acumulada) empirica y teorica
91 empiric_cdf=[i/len(datos_poisson) for i in range(len(datos_poisson))]
92 exact_cdf=poisson.cdf(datos_poisson,lambd)
93
94 plt.plot(datos_poisson,empiric_cdf,'--k',label='Distribucion empirica')
95 plt.plot(datos_poisson,exact_cdf,'r',label='Distribucion teorica')
96 plt.legend()
97
98 # Funcion de distribucion de probabilidad empirica y teorica
99 x=np.array([(bin_edges[i]+bin_edges[i+1])/2 for i in range(len(bin_edges)-1)])
100 estimadas_poiss=[(poisson.cdf(bin_edges[i+1],lambd)-poisson.cdf(bin_edges[i],lambd))*len(datos_poisson) for
101 i in range(len(bin_edges)-1)]
102 reales=entries
103
104 plt.bar(x,reales,width=4)
105 plt.plot(x,estimadas_poiss,marker='o', linestyle='',color='r')
106
107 # Prueba de bondad Chi-cuadrada
108 ss.chisquare(f_obs = reales, f_exp = estimadas_poiss, ddof=1)
109
110 # Prueba de hipotesis
111 if ss.chisquare(f_obs = reales, f_exp = estimadas_poiss, ddof=1).pvalue<0.05:
112     print('Se rechaza la Ho, los datos no siguen una distribucion Poisson.')
113 else:
114     print('No hay suficiente evidencia para rechazar la Ho.')
115
116 # Distribucion de pagos por siniestro
117 # Estimacion por intervalo del parametro de la distribucion exponencial
118
119 # Datos
120 z_95 = ss.norm.ppf(0.95+(1-.95)/2)
121 N = df.shape[0]
122 mu
123
124 # Calculo de intervalo
125 lim_inf = mu - mu*z_95/np.sqrt(N)
126 lim_sup = mu + mu*z_95/np.sqrt(N)
127 intervalo = [lim_inf,lim_sup]
128 intervalo
129
130 # Prueba de bondad de ajuste (Exponencial)
131 # Kolmogorov-Smirnov
132
133 def expon_pdf(x,mu):
134     return 1/mu*np.exp(-1/mu*x)
135
136 def expon_cdf(x,mu):
137     return 1-np.exp(-1/mu*x)
138
139 # Funcion de distribucion de probabilidad empirica y teorica
140 datos_expon=df['Monto del siniestro']
141
142 datos_expon=np.array(datos_expon.sort_values(axis=0))
143 mu=datos_expon.mean()
144 entries, bin_edges, patches = plt.hist(datos_expon, label='Datos')
145 plt.plot(datos_expon,expon_pdf(datos_expon,mu)*sum(datos_expon),'.r', label='Distribucion teorica')
146 plt.legend()
147
148 # Funcion de distribucion (Funcion de densidad acumulada) empirica y teorica
149 empiric_cdf=[i/len(datos_expon) for i in range(len(datos_expon))]
150 exact_cdf=expon_cdf(datos_expon,mu)

```

```

151
152 plt.plot(datos_expon,empiric_cdf,'--k',label='Distribucion empirica')
153 plt.plot(datos_expon,exact_cdf,'r',label='Distribucion teorica')
154 plt.legend()
155
156 # Prueba de hipotesis
157 D=np.max(np.abs(exact_cdf-empiric_cdf))
158 statistic=1.06/np.sqrt(len(datos_expon))
159
160 if D>statistic:
161     print('Se rechaza la Ho, los datos no siguen una distribucion exponencial.')
162 else:
163     print('No hay suficiente evidencia para rechazar la Ho.')
164
165
166 # Chi cuadrada
167
168 # Prueba de bondad Chi-cuadrada
169 entries, bin_edges, patches = plt.hist(datos_expon, label='Datos', bins=12)
170 plt.plot(datos_expon,expon_pdf(datos_expon,mu)*sum(datos_expon),'.r', label='Distribucion teorica')
171 plt.legend()
172
173 # Frecuencias por clase empirica y teorica
174 x=np.array([(bin_edges[i]+bin_edges[i+1])/2 for i in range(len(bin_edges)-1)])
175
176 estimadas_expon = [(expon_cdf(bin_edges[i+1],mu)-expon_cdf(bin_edges[i],mu))*len(datos_expon) for i in
177                     range(len(bin_edges)-1)]
178 reales=entries
179
180 plt.bar([str(np.round(num,2)) for num in x],reales,label='Frecuencia real')
181 plt.plot([str(np.round(num,2)) for num in x],estimadas_expon,marker='o', linestyle='',color='r',label='
182 Frecuencia esperada de clase')
183
184
185 plt.xticks(rotation=45)
186 plt.legend()
187
188 # Prueba de bondad Chi-cuadrada
189 degrees_of_freedom=1
190 ss.power_divergence(f_obs = reales, f_exp = estimadas_expon, ddof=degrees_of_freedom)
191
192 # Prueba de hipotesis
193 if ss.chisquare(f_obs = reales, f_exp = estimadas_expon, ddof=1).pvalue<0.05:
194     print('Se rechaza la Ho, los datos no siguen una distribucion exponencial.')
195 else:
196     print('No hay suficiente evidencia para rechazar la Ho.')
197
198 # Simulaciones
199
200 # Parametros para la estimacion de la probabilidad de ruina
201 U=0 # Capital inicial
202 theta=0.1 # Ganancia neta
203 c=lambd*mu*(1+theta) # Prima por unidad de tiempo
204
205 # Monte Carlo
206
207 def MonteCarloSimulation(U,c,lambd,mu,time):
208     N=ss.poisson.rvs(lambd,size=time) # Numero de siniestros por dia
209     Z=[]
210     for i in range(time):
211         Z.append(np.sum(ss.expon.rvs(mu,size=N[i]))) # Montos de siniestros por dia
212     sum_Z=[np.sum(Z[0:i+1]) for i in range(len(Z))] # Suma acumulada de montos de siniestros al dia t
213     t=np.array(range(1,time+1))
214
215     X=np.ones(len(t))*U+c*t-sum_Z # Capital de la compania en el tiempo t
216     return t,X
217
218 t,X = MonteCarloSimulation(0,c,lambd,mu,365)
219 plt.plot(t,X)
220 print(np.mean(X<0))
221
222 # Probabilidad Exacta
223
224 prob_exacta = (mu*lambd/c)*(np.exp(-U/mu + U*lambd/c))
225 prob_exacta
226
227 # Probabilidad de Ruina Frecuentista
228

```

```

226 def getFrequentistRuinProbability(Model,U,c,lambd,mu,time,simulations_number,method=None,significance_level
    =0.05,epsilon=0.01):
227     ensurance_ruin=[]
228     ruin_probabilities=[]
229     s=[]
230     z_alpha_2=ss.norm.ppf(1-significance_level/2)
231     std_var_2=0
232     for simulation in range(simulations_number):
233         t,X=Model(U,c,lambd,mu,time)
234         ensurance_ruin.append(int(np.mean(X<0)!=0))
235         ruin_prob=np.mean(ensurance_ruin) # Probabilidad de ruina
236         ruin_probabilities.append(ruin_prob) # Probabilidades de ruina por simulacion
237         std_dev = np.sqrt(np.mean(np.array(ruin_probabilities)**2)-np.mean(np.array(ruin_probabilities)
    **2)) # Calculo desviacion estandar
238
239         try:
240             std_var_2=(1-1/(len(ruin_probabilities)-1))*std_var_2+len(ruin_probabilities)*(ruin_prob-
    passed_ruin_prob)**2 # Calculo de varianza
241         except:
242             pass
243         passed_ruin_prob=ruin_prob
244         if (method=='var_min_profe') and (std_dev<epsilon/z_alpha_2) and (simulation>50): # Monte Carlo
    condicionando desviacion estandar
245             break
246         if (method=='var_min_brenda') and (np.sqrt(std_var_2)<epsilon/z_alpha_2) and (simulation>50): #
    Monte Carlo minima varianza
247             break
248
249     print('Probability ruin for',simulation+1,'simulations.')
250     # Grafica de simulaciones vs probabilidad de ruina
251     x=range(1,len(ruin_probabilities)+1)
252     plt.plot(x,ruin_probabilities,label='Probabilidad estimada')
253     plt.plot(x,np.ones(len(x))*prob_exacta,'--k',label='Probabilidad exacta')
254     plt.xlabel('Simulaciones')
255     plt.ylabel('Probabilidad de Ruina')
256     plt.ylim([0,1.1])
257     plt.legend()
258     plt.show()
259     return ruin_probabilities, ruin_probabilities[-1]
260
261 # Grafica de probabilidad de ruina (Monte Carlo)
262 ruin_probabilities,ruin_prob=getFrequentistRuinProbability(MonteCarloSimulation,0,c,lambd,mu,365,1000)
263 print(ruin_prob)
264
265 # Grafica de probabilidad de ruina (Monte Carlo condicionado)
266 ruin_probabilities,ruin_prob=getFrequentistRuinProbability(MonteCarloSimulation,0,c,lambd,mu,365,1000,
    method='var_min_profe',significance_level=0.05,epsilon=0.05)
267 print(ruin_prob)
268
269 # Grafica de varias simulaciones
270 plt.figure(figsize=(12,8))
271 simulations_graphs=30
272 for sim in range(simulations_graphs):
273     t,X = MonteCarloSimulation(0,c,lambd,mu,365)
274     plt.plot(t,X)
275     plt.xlabel('Tiempo [dias]')
276     plt.ylabel('X(t)')
277
278 # Funcion para el calculo de la probabilidad de ruina con aproximacion de Cramer-Lundberg
279 def ruinProbabilityCramerLundberg(u,lambd,mu,c):
280     rho = (c-lambd*mu)/(lambd*mu)
281     temp = np.exp(-(rho*u)/(mu*(1+rho)))
282     return (1/(1+rho))*temp
283
284 # Probabilidad de ruina con aproximacion de Cramer-Lundberg (aproximado) para u = 0
285 ruinProbabilityCramerLundberg(U,lambd,mu,c)
286
287 # Implementacion de aproximacion de Cramer-Lundberg (capital inicial vs probabilidad de ruina)
288 ux=np.array(range(0,2000000,100))
289
290 y=[ruinProbabilityCramerLundberg(ux_i,lambd,mu,c) for ux_i in ux]
291 y_exact=[ruinProbabilityExact(ux_i) for ux_i in ux]
292
293 plt.plot(ux,y,label='Probabilidad estimada')
294 plt.plot(ux,y_exact,'--k',label='Probabilidad exacta')
295
296 plt.xlabel('Capital inicial')

```

```

297 plt.ylabel('Probabilidad de Ruina')
298 plt.ylim([0,1.1])
299 plt.legend()
300 plt.show()
301
302 import sympy as sym
303 from sympy.abc import x as xsym
304 from sympy.abc import u as usym
305 from sympy.abc import b as bsym
306
307 #Enesima potencia de convolucion.
308 def nth_Convolution_sym(u,mu,lambd,c,n):
309
310     F_tail=(1-sym.exp(-bsym*usym))
311     conv_F_past=F_tail
312
313     for power in range(0,n):
314         conv_F=sym.integrate(conv_F_past.subs(usym,xsym)*F_tail.subs(usym,xsym-usym),(xsym,0,usym))
315         conv_F_past=conv_F
316
317     return c/(lambd*mu)-float(conv_F.subs(usym,u).subs(bsym,1/mu))
318
319 # Funcion para el calculo de la probabilidad de ruina aproximada con Pollaczek-Khinchin
320 def ruinProbabilityPollaczek(N,u,lambd,mu,c):
321
322     first_term=(1-lambd*mu/c)
323     n=range(1,N+1)
324     second_term=[(lambd*mu/c)**n_i for n_i in n]
325
326     convolution=np.array([nth_Convolution_sym(u,mu,lambd,c,1)**n_i for n_i in n]) #Enesima potencia de
        convolucion.
327
328     solvence_prob=first_term*np.sum(second_term*convolution/N)
329
330     return 1-solvence_prob
331
332 # Probabilidad de ruina con Pollaczek-Khinchin (aproximado) para u = 0
333 ruinProbabilityPollaczek(1,0,lambd,mu,c)
334
335 # Implementacion de formula de Pollaczek-Khinchin (aproximado) para u = 1000
336 Ux=1000
337 n=np.linspace(1,20,5).astype(int)
338
339 y=[ruinProbabilityPollaczek(n_i,Ux,lambd,mu,c) for n_i in n]
340
341 y_exact=ruinProbabilityExact(Ux)*np.ones(len(n))
342
343 plt.plot(np.flipud(n),y,label='Probabilidad estimada')
344 plt.plot(n,y_exact,'--k',label='Probabilidad exacta')
345
346 print(ruinProbabilityExact(Ux))
347
348 plt.xlabel('Iteraciones')
349 plt.ylabel('Probabilidad de Ruina')
350 plt.xticks(ticks=n)
351 plt.ylim([-0.1,1.1])
352 plt.legend()
353 plt.show()
354
355 # Implementacion de formula de Pollaczek-Khinchin aproximado (capital inicial vs probabilidad de ruina)
356 Ux=2000
357 x=np.linspace(0,Ux,5)
358 y=[ruinProbabilityPollaczek(1,Ux_i,lambd,mu,c) for Ux_i in x]
359
360 y_exact=[ruinProbabilityExact(Ux_i) for Ux_i in x]
361
362 plt.plot(x,y,label='Probabilidad estimada')
363 plt.plot(x,y_exact,'--k',label='Probabilidad exacta')
364
365 print(ruinProbabilityExact(U))
366
367 plt.xlabel('Capital inicial')
368 plt.xticks(ticks=x)
369 plt.ylabel('Probabilidad de Ruina')
370 plt.ylim([0,1.1])
371 plt.legend()
372 plt.show()

```

```

373
374
375 # Funcion para el calculo de la probabilidad de ruina (analitico) con Pollaczek-Khinchin
376 def ruinProbabilityPollaczekExact(u, lambd, mu, c):
377     return lambd*mu/c*np.exp((lambd*mu-c)/(mu*c)*u)
378
379 # Implementacion de formula de Pollaczek-Khinchin (analitico) para u = 1000
380 Ux=1000
381 n=np.linspace(1,20,5).astype(int)
382
383 y=[ruinProbabilityPollaczekExact(Ux,lambd,mu,c) for n_i in n]
384
385 y_exact=ruinProbabilityExact(Ux)*np.ones(len(n))
386
387 plt.plot(np.flipud(n),y,label='Probabilidad estimada')
388 plt.plot(n,y_exact,'--k',label='Probabilidad exacta')
389
390 print(ruinProbabilityExact(Ux))
391
392 plt.xlabel('Iteraciones')
393 plt.ylabel('Probabilidad de Ruina')
394 plt.xticks(ticks=n)
395 plt.ylim([-0.1,1.1])
396 plt.legend()
397 plt.show()
398
399 # Implementacion de formula de Pollaczek-Khinchin analitico (capital inicial vs probabilidad de ruina)
400 Ux=1000000
401 x=np.linspace(0,Ux,20)
402 y=[ruinProbabilityPollaczekExact(Ux_i,lambd,mu,c) for Ux_i in x]
403
404 y_exact=[ruinProbabilityExact(Ux_i) for Ux_i in x]
405
406 plt.plot(x,y,label='Probabilidad estimada')
407 plt.plot(x,y_exact,'--k',label='Probabilidad exacta')
408
409 print(ruinProbabilityExact(U))
410
411 plt.xlabel('Capital inicial')
412 plt.ylabel('Probabilidad de Ruina')
413 plt.ylim([0,1.1])
414 plt.legend()
415 plt.show()
416
417 # Obtencion de probabilidades de ruina para cada metodo
418 U=[10*n for n in range(0,4)]
419 methods=['Exacta','Monte Carlo','Monte Carlo varianza reducida','Aproximacion Pollaczek Khintchine','Pollaczek Khintchine Analitico','Cramer Lundberg']
420 prob_exact=[ruinProbabilityExact(U_i) for U_i in U]
421 prob_MonteCarlo=[getFrequentistRuinProbability(MonteCarloSimulation,U_i,c,lambd,mu,365,1000) for U_i in U]
422 prob_MonteCarloVarMin=[getFrequentistRuinProbability(MonteCarloSimulation,U_i,c,lambd,mu,365,1000,method='var_min_profe',significance_level=0.05,epsilon=0.05) for U_i in U]
423 prob_AproxPollaczek=[ruinProbabilityPollaczek(1,U_i,lambd,mu,c) for U_i in U]
424 prob_PollaczekAnalit=[ruinProbabilityPollaczekExact(U_i,lambd,mu,c) for U_i in U]
425 prob_CramerLundberg=[ruinProbabilityCramerLundberg(U_i,lambd,mu,c) for U_i in U]
426
427 # Preparacion de datos para la tabla comparativa
428 probs=[prob_exact,[prob_MonteCarlo[i][1] for i in range(np.array(prob_MonteCarlo).shape[0])],[
429     prob_MonteCarloVarMin[i][1] for i in range(np.array(prob_MonteCarloVarMin).shape[0])],
430     prob_AproxPollaczek,prob_PollaczekAnalit,prob_CramerLundberg]
431
432 table={}
433
434 for method in methods:
435     table[method]=probs[methods.index(method)]
436
437 table['Capital inicial']=U
438
439 # Creacion de tabla comparativa
440 Table=pd.DataFrame(table)
441 Table.index=U
442 Table=Table.set_index('Capital inicial')
443
444 # Tabla comparativa
445 Table

```