Estimaciones de la probabilidad de ruina de una compañía aseguradora a través de métodos estocásticos a tiempo discreto

Samuel Méndez Salette Noemi Ethan Verduzco Brenda Martínez Jesús Marroquín

***Abstract*— En este artículo se presentan diferentes métodos de aproximación basados en el modelo de Cramer-Lundberg para calcular la probabilidad de ruina de una compañía aseguradora. Entre las estimaciones obtenidas se encuentran el método de Monte-Carlo frecuentista; de varianza reducida, la fórmula de Pollaczek-Khinchin y la aproximación De Vylder. Variando el capital inicial de la compañía, como resultados se obtuvieron las probabilidades de que la empresa quiebre a largo plazo.**

***Index Terms*— análisis de riesgo, modelo Cramer-Lundberg, modelo estocástico, tiempo discreto, aproximación De Vylder, fórmula Pollaczek-Khinchin.**

I. INTRODUCCIÓN

Día a día las empresas se enfrentan a grandes retos que requieren de un análisis cuidadoso y detallado para tomar las mejores decisiones que beneficien a la empresa. En las compañías aseguradoras, uno de los principales desafíos, o temores debido a que está basado en la incertidumbre, es conocer si en algún horizonte de tiempo finito, la empresa no podrá solventarse y por lo tanto, caerá en bancarrota. Es justo esta problemática la que se estudia en la teoría de riesgo, la cual tiene el objetivo de analizar y estimar las posibilidades de que una empresa caiga en ruina. Pero ¿por qué es importante conocer la probabilidad de ruina de una compañía? Por una razón muy simple, para implementar las estrategias y tomar las decisiones necesarias para no quebrar. Con el paso del tiempo, se han generado varios modelos que representan este escenario, sin embargo, el más conocido es el Modelo Clásico de Cramer-Lundberg, en el cual se toman en cuenta varios aspectos como el capital inicial de la empresa, el número y monto de los reclamos que se reciben por unidad de tiempo, entre otros, tal como se explica en [1]. Además, en la literatura, se puede encontrar diferentes maneras de obtener la probabilidad de ruina [2]. Estos métodos parten del modelo clásico y algunos cuentan con una solución exacta, mientras que otros son aproximaciones.

II. MARCO CONCEPTUAL

La teoría del riesgo es un caso especial en los procesos estocásticos que ha crecido de manera acelerada durante los últimos años, unos ejemplos de estos procesos son la espera en una fila y el número de llamadas que se puedan recibir en cierto intervalo de tiempo [3]. En este artículo se concentrará en el caso de una compañía aseguradora que trata con accidentes de automóviles.

Se cuentan con una base de datos de una aseguradora. En ella se tienen variables como el día del siniestro, el modelo y el año del auto, el monto del siniestro junto con su deducible, y si es pérdida total.

Por medio del presente proyecto, se pretende construir un algoritmo, utilizando la base de datos mencionada con anterioridad, que sea capaz de simular la situación de la compañía aseguradora para poder predecir el tiempo y la probabilidad de que esta vaya a la ruina.

Existe un gran número de investigaciones entorno a las compañías aseguradoras modelando la optimización de sus recursos y sobre distintas manera de medir su rentabilidad.

Se realizó una investigación en Pakistán sobre las variables determinantes para la rentabilidad de una aseguradora, donde la conclusión fue que, en cuanto a la devolución de activos, existe una relación positiva entre la rentabilidad, el tamaño de la compañía y el volumen del capital, mientras que hay una relación negativa entre la rentabilidad y la proporción de pérdida [4].

En cuanto al cálculo para el tiempo de ruina de la compañía, se utilizó el modelo de Cramer-Lundberg (Ecuación 1):

II-A. *Modelo de Cramer-Lundberg*

Este es un modelo originalmente creado por Lundberg y retomado en 1993 por Cramer quien se encarga de ponerlo en un contexto estocástico. [5]. El modelo está representado por la siguiente expresión.

$$U(t) = u + ct - \sum_{i=1}^{N(t)} X_i \tag{1}$$

Donde:

- $U(t)$ representa el capital contable de la aseguradora en el tiempo t.
- u es el capital inicial.
- c representa la tasa de primas; es decir, lo que recibe la aseguradora de sus clientes.

Además, el modelo se rige bajo los siguientes supuestos:

- El número de reclamos N que ocurren en un período es aleatorio, satisfaciendo que el tiempo $T_i \geq 0$ [6].
- El i-ésimo reclamo T_i ocasiona un costo de X_i. $\{X_i\}$ se asume que es una secuencia independiente e idénticamente distribuida (i.i.d.) compuesta de variables no negativas [6].
- $\{X_i\}$ y $\{T_j\}$ son independientes; por lo tanto, $\{X_i\}$ y N son independientes [6].
- Las reclamaciones llegan en tiempos aleatorios de acuerdo a un proceso de Poisson homogéneo $N(t)$: $t > 0$ de intensidad $\lambda > 0$ [1].

El desarrollo de tecnologías en la ciencia computacional ha dado paso al desarrollo de modelos con variables estocásticas, y también existen numerosos artículos que combinan la teoría de los modelos estocásticos con la modelación de un algoritmo para poder darle solución al problema de la aseguradora. Por ejemplo, se realizó un algoritmo usando *deep learning* y cadenas de Markov [7] para darle solución a la problemática.

Incluso se han creado modelos un poco más “reales” donde se toman en cuenta situaciones de préstamos, inversiones e inflaciones [8].

En el caso en que los reclamos son una mezcla de exponenciales, se compararon las probabilidades de ruina obtenidas utilizando la transformada de Laplace, el algoritmo de Panjer y la aproximación Cramer-Lundberg; en donde Panjer proporcionó las mejores aproximaciones [1].

Cuando los reclamos siguen una distribución Gamma, en este caso no se puede calcular directamente la inversa de la transformada de Laplace, entonces se utilizará Panjer y Cramer-Lundberg. Se obtuvo que la aproximación utilizando Cramer-Lundberg decrece más rápido que con Panjer [1].

En general, se tiene la conclusión de que a mayor capital invertido inicialmente (u), menor será la probabilidad de ruina.

II-B. *Preguntas de Investigación*

Este artículo tiene el objetivo de obtener distintas aproximaciones de la probabilidad de ruina de una empresa aseguradora a través de diversos métodos como el método de Monte Carlo Frecuentista y de varianza reducida, ambos basados en el Modelo Clásico de Cramer-Lundberg. Además, también se tiene el objetivo de encontrar las probabilidades de quiebra a través de otros métodos como la fórmula de Pollaczek y del método De Vylder.

Para lo anterior, además de los cálculos matemáticos correspondientes, se realizarán simulaciones en el lenguaje computacional *Python* para visualizar diferentes trayectorias del comportamiento del superávit $(U(t))$ de la empresa, así como obtener la probabilidad de ruina a través de los métodos antes mencionados.

Las preguntas que se buscan responder son:

1. ¿Existe algún horizonte de tiempo finito en donde la empresa aseguradora no pueda solventarse?
2. ¿Cuál es la prima ideal para disminuir la probabilidad de ruina en un horizonte finito de 365 días?
3. ¿Cuál la probabilidad de que la compañía no quiebre con una determinada inversión inicial *u*?
4. ¿Los diferentes métodos de aproximación convergen al mismo valor de probabilidad de ruina?

III. METODOLOGÍA

A continuación se presenta la metodología seguida para obtener la probabilidad de ruina a través de distintos métodos como los basados en Monte Carlo (MMC) y las aproximaciones de Pollaczek y De Vylder. Sin embargo, antes se da a conocer un poco acerca de cada método.

III-A. *Monte Carlo frecuentista*

El método de Monte Carlo, se refiere a una simulación la cual permite calcular estadísticamente el valor de una secuencia de sucesos estocásticos, es decir no deterministas, o sujetos a la variabilidad. En pocas palabras, el método consiste en ejecutar varias simulaciones pero en cada una de ellas se cambia aleatoriamente su valor con base en la función estadística que los define. Este método es uno de los más utilizados para calcular probabilidades, pues trabaja con secuencias de números aleatorios y su importancia recae en que permite ver los escenarios posibles de las decisiones que se toman y evaluar el impacto de riesgo.

En este proyecto se utilizará el MMC para calcular la probabilidad de ruina, es decir se realizarán 10,000 simulaciones o trayectorias en donde cada una tomará un grupo diferente de valores aleatorios de las funciones de probabilidad. En este caso, de la distribución de Poisson para el número de siniestros por unidad de tiempo, y de la distribución exponencial para el tamaño de los montos o reclamaciones. De dicha manera, se podrá saber cuál es el riesgo de que un escenario que dañe a la compañía aseguradora ocurra.

Cabe mencionar que se asumirá la probabilidad de ruina vía frecuentista, es decir que ésta se calculará a través del número de trayectorias en las que se obtuvo un tiempo de ruina entre el número de trayectorias totales hasta el momento. Esto quiere decir que se obtendrán en total 10,000 probabilidades de ruina y el punto importante, es ver si éstas con el paso de las simulaciones convergen a un cierto valor, el cual se acerque a la probabilidad de ruina original. Dicho lo anterior, ahora se pasará a describir cómo será la metodología para la simulación.

En primer lugar, se debe de definir las características de las simulaciones. Como ya se mencionó, en total se realizarán 10,000 trayectorias, en donde en cada una de ellas se tomará un horizonte de tiempo finito de 365 días. El código implementado, también tiene la característica de estar parametrizado, para de esta manera, poder probar las simulaciones con distintos valores. Los parámetros que el usuario tiene que brindar al programa son $\hat{\lambda}$ que se refiere al parámetro estimado de la distribución de Poisson (promedio de siniestros por día); $\hat{\mu}$ que es el promedio del tamaño de las reclamaciones; el valor u que se refiere al capital inicial de la compañía aseguradora; y c que representa la tasa de primas que recibe la aseguradora por unidad de tiempo.

Ahora si, se mencionará brevemente el proceso que se hace en cada simulación, ya que más adelante se ejemplificará más a fondo.

1. Ingresar parámetros y valores correspondientes a $\hat{\lambda}$, $\hat{\mu}$, u y c.
2. Generar un número aleatorio que siga una distribución de Poisson.
3. Generar la cantidad de valores aleatorios (cantidad igual al número anteriormente generado) que siguen una distribución exponencial.
4. Calcular el superávit de la compañía para un horizonte finito de 365 días.
5. Obtener la probabilidad de ruina vía frecuentista.

Algunos aspectos importantes a considerar, es que en cada día, se evalúa si el superávit $U(t)$ se encuentra por debajo de 0, pues si es así, se dice que la compañía quebró. Al tiempo mínimo en que la función $U(t)$ es negativo se le conoce como **tiempo de ruina**.

$$\tau = min\{t \geq 0 : U(t) < 0\} \tag{2}$$

Por lo tanto, en caso de tener un tiempo de ruina, se dice que en dicha simulación la compañía quebró. Finalmente, se pasa a calcular la probabilidad de ruina, la cual, como ya se mencionó se asume vía frecuentista. Por lo tanto, si en la primera simulación se llega a la ruina, entonces se dice que la probabilidad de ruina es 1. Sin embargo, si en la segunda simulación no se llega a ruina, entonces, la probabilidad de ruina hasta la segunda trayectoria es de 0.5. Este proceso se repite para cada una de las 10,000 simulaciones.

III-A.1. Error estimación: Gracias al teorema del límite central, es posible calcular el error de estimación del método Monte Carlo. Sean $I = E(g(x))$ e $\tilde{I}_M = \frac{1}{M}\Sigma_{i=1}^{M} g(X_i)$ entonces siendo σ la desviación estándar de $g(X)$ se tiene que $\frac{\sigma}{\sqrt{M}}$ es la desviación estándar de $\tilde{I}_M$, por lo que:

$$P(|I - \tilde{I}_M| < \frac{c\sigma}{\sqrt{M}}) \approx P(|Z_M| < c) = 2\phi(c) \tag{3}$$

[9]

Por ello, el error cometido en la estimación es aproximadamente $\frac{\sigma}{M}$. Con un intervalo de confianza de $1-\alpha\,\%$. Si seleccionamos los parámetros de tal forma que $\phi(c) = \frac{\alpha}{2}$, de tal forma que con probabilidad de $1-\alpha$ se puede asegurar que el valor exacto está en el intervalo:

$$\left[\tilde{I} - \frac{c\sigma}{\sqrt{M}}, \tilde{I} + \frac{c\sigma}{\sqrt{M}}\right] \tag{4}$$

[9]

III-B. Monte Carlo varianza mínima

Una de las principales problemáticas que afectan al método de Monte Carlo es su lentitud para converger. La rapidez está en función del número de trayectorias generadas y de la varianza de estas. Por ello, existe un método que se enfoca en solucionarlo disminuyendo la varianza. Se plantea el problema de una forma que permite reducir el tiempo pero donde se obtiene la estimación de la esperanza que es de interés. [9]

Uno de los métodos se denomina *variable de control* y se enfoca en encontrar una variable correlacionada con aquella que sea de interés. Se puede definir una tercera variable aleatoria de la siguiente forma:

$$Z = X + \beta\,[Y - E(Y)] \tag{5}$$

Por ello se puede deducir que:

$$E(Z) = E(X) \tag{6}$$

$$Var(Z) = Var(X) + \beta^2 Var(Y) + 2\beta cov(X,Y) \tag{7}$$

A partir de ello, se puede deducir que la reducción de la varianza está dado por:

$$1 - [\rho(X,Y)]^2 \tag{8}$$

donde $\rho(X,Y)$ es el coeficiente de correlación. [9]

III-C. Fórmula de Pollaczek

La fórmula de Pollaczek–Khinchin acierta la relación entre la longitud de la cola y la transformada de Laplace, en este caso se utilizará para computar la probabilidad de ruina final.

La función de ruina $\psi(u)$ satisface la ecuación integral:

$$\beta\psi(u) = \lambda\left(\int_0^\infty \bar{F}_u(x)dx + \int_0^u \psi(u-x)\bar{F}_u(x)dx\right). \tag{9}$$

En donde la ecuación 9 lleva a una formula de $\psi(u)$ en la forma de convoluciones infinitas; y para esto, se necesita la distribución de colas integrada F_U^s de F_U dadas por:

$$F_U^s(x) = \frac{1}{\mu}\int_0^x \bar{F}_U(y)dy,\ x \geq 0. \tag{10}$$

Por lo tanto, la representación de la formula para $\psi(u)$ se deriva en el teorema de la fórmula de Pollaczek-Khinchin: [10]

Para todo $u \geq 0$

$$\psi(u) = \left(1 - \frac{\lambda\mu}{\beta}\right)\sum_{n=1}^{\infty}\left(\frac{\lambda\mu}{\beta}\right)(F_U^s)^{*n}(u). \tag{11}$$

III-D. Aproximación De Vylder

La aproximación de De Vylder, se basa en estimar la probabilidad de ruina a través de un modelo de con reclamaciones exponenciales. Por lo tanto, se puede definir el siguiente proceso [2].

$$\tilde{C}_t = u + \tilde{c}t - \sum_{j=1}^{\tilde{N}} \tilde{Y}_j \tag{12}$$

En donde:

- $\tilde{c}$ es una nueva tasa de ingresos por primas.
- $\{\tilde{N}_t : t \geq t\}$ es un proceso de Poisson de parámetro $\tilde{\lambda}$.
- $\tilde{Y}_j$ son variables aleatorias con distribución $\tilde{\alpha}$.

Para estimar la probabilidad de ruina con este nuevo modelo, se debe de igualar los tres primeros momentos de los procesos (Cramer-Lundberg y De Vylder) y así, obtener los parámetros $\tilde{\alpha}, \tilde{\lambda}, \tilde{c}$ en términos del riesgo original [2]. Entonces, la probabilidad de ruina de dicho modelo queda como:

$$\psi(u) \approx \frac{\tilde{\lambda}}{\tilde{c}\tilde{\alpha}} e^{-(\tilde{\alpha} - \tilde{\lambda}/\tilde{c})u} \tag{13}$$

En donde:

$$\tilde{\alpha} = 3\frac{\mu_2}{\mu_3} \tag{14}$$

$$\tilde{\lambda} = \frac{9}{2}\frac{\mu_2^3}{\mu_3^2}\lambda \tag{15}$$

$$\tilde{c} = c - \lambda\mu + \frac{3}{2}\frac{\mu_2^2}{\mu_3}\lambda \tag{16}$$

[2].

IV. ANÁLISIS DE DATOS

En esta sección se presentan paso por paso el procedimiento que se siguió para obtener la probabilidad de ruina de la empresa, a través de un enfoque de Monte Carlo-Frecuentista, Monte-Carlo de varianza mínima, fórmula de Pollaczek y aproximación De Vylder.

IV-A. *Exploración de los datos*

Antes de realizar cualquier cálculo, se debe de analizar la base de datos que se brindó por el socio formador para empezar a plantear y comprobar que los supuestos necesarios para aplicar el modelo clásico se cumplan.

La base de datos que se brindó, contiene un total de 27121 registros, los cuales representan accidentes que han ocurrido a lo largo del año 2020, más específicamente en cada uno de los días de dicho año. En cuanto a las variables, se cuenta tanto con variables numéricas como categóricas, siendo esta la distribución:

- **Variables categóricas:** fecha del siniestro, tipo de auto, aplica cobertura, reclamo de cobertura, pérdida total.
- **Variables numéricas:** modelo, monto del siniestro y deducible.

Otra de las observaciones que se realizó en esta exploración, fue ver si existían valores nulos en las diferentes variables. En total se encontraron 62 datos faltantes, de los cuales 39 pertenecen a la columna de deducible, mientras que los restantes al tipo de auto. Dichos valores nulos se sustituyeron por 0 para de esta manera no borrar el registro completo. También, otro dato curioso que se encontró fue que en la columna perteneciente al reclamo de cobertura, pues habían registros que aparecían como una fórmula de Excel sin completar. Lo que se hizo con estos datos fue sustituirnos con un "No" debido a que era los resultados correspondientes a la condición de su fórmula de Excel original.

Una vez realizada esta exploración, se seleccionó el conjunto de datos con los que se trabajará. Para el proyecto, se utilizará todos los datos disponibles en la base, es decir los 27121 registros. Ya definidos los datos, se ordenaron los registros por fecha y, de esta manera, se comprendieron los datos con los que se trabajó.

IV-B. *Análisis inferencial de los datos*

En esta subsección se hablará sobre el proceso inferencial que se realizó con el objetivo de visualizar si la base de datos cumplía con los supuestos establecidos por el método de Cramer-Lundberg.

IV-B.1. Intervalos de confianza: En primer lugar, se realizaron los intervalos de confianza para estimar los parámetros λ y μ, en donde de momento se supone que siguen una distribución de Poisson y una distribución exponencial respectivamente. Este proceso inferencial para los parámetros de la distribución, no se está asumiendo ninguna hipótesis de la distribución, únicamente de los parámetros.

Comencemos hablando sobre el estimador $\hat{\lambda}$, el cual representa el promedio del número de siniestros o reclamaciones ($N(t)$) que llegan en cierto intervalo de tiempo, en este caso, de un día. Para obtener su valor estimado, se calculó un promedio ($\bar{x}$). Para esto, se agruparon los datos por fecha, es decir que se trabajó cada uno de los 365 días por separado. De cada día, se contaba el número total de reclamaciones y se dividió la suma de dichos valores entre 365. De esta forma, se obtuvo un estimador del parámetro λ poblacional cuyo valor resultó se de: $\hat{\lambda} = \bar{x} = 74.3041$. Con este valor del estimador obtenido, se pasó a realizar el intervalo de confianza del 95 %. Para esto, se utilizó la siguiente fórmula:

$$\bar{x} - z_{\alpha/2}\sqrt{\frac{\bar{x}}{n}} \leq \lambda \leq \bar{x} + z_{\alpha/2}\sqrt{\frac{\bar{x}}{n}} \qquad (17)$$

En donde n es el número total de observaciones y $z_{\alpha/2}$ es el valor de z basado en alfa y en la distribución normal estándar. Sustituyendo en la fórmula anterior los valores correspondientes, se obtienen los siguientes números para el límite inferior y superior del intervalo de confianza:

Límite inferior = 74.3041 - 1.96 $\cdot\sqrt{\frac{74.3041}{365}} = 73.4197$

Límite superior = 74.3041 + 1.96 $\cdot\sqrt{\frac{74.3041}{365}} = 75.1884$

Por los resultados anteriores, se infiere con un 95 % de confianza que el valor del parámetro λ poblacional, se encuentra entre:

$$73.4197 \leq \lambda \leq 75.1884 \qquad (18)$$

Pasando hablar ahora sobre los intervalos de confianza para el estimador μ, de igual forma primero se calculó $\hat{\mu}$. Para esto, simplemente se calculó el promedio de los montos de los siniestros de la base de datos. Es decir, se obtuvo el promedio de la columna "Monto del siniestro". Con esto, se obtuvo el siguiente valor de $\hat{\mu} = 30771.3756$. En [11] se demuestra que el intervalo de confianza para la media de una distribución exponencial, está dada por la siguiente fórmula.

$$\frac{2n}{\hat{\lambda}\chi^2_{1-\frac{\alpha}{2},2n}} < \frac{1}{\lambda} < \frac{2n}{\hat{\lambda}\chi^2_{\frac{\alpha}{2},2n}} \qquad (19)$$

En donde n es el número de observaciones y χ se refiere al valor de la distribución chi cuadrada con cierta probabilidad y grados de libertad, cuyo valor se obtiene de la tabla de distribución correspondiente. Antes de continuar con el cálculo del intervalo, se debe de mencionar que el valor de $\hat{\mu}$ es equivalente a $\hat{\mu} = 1/\lambda$ correspondiente al parámetro de la distribución exponencial. Dicho lo anterior, se calculan los limites del intervalo con un 95 % de confianza:

- Límite inferior:

$$\frac{2(27121)}{3.2497e{-}05 \cdot 54889.44377} = 30408.4145 \qquad (20)$$

- Límite superior:

$$\frac{2(27121)}{3.2497e{-}05 \cdot 53598.34483} = 31140.9049 \qquad (21)$$

De las ecuaciones anteriores, se obtiene tanto el límite superior como el inferior que conforman el intervalo de confianza para el parámetro de la distribución exponencial $\mu = \frac{1}{\lambda}$. Por lo tanto, se infiere con un $95\,\%$ de confianza que dicho parámetro se encuentra entre los siguientes valores:

$$30408.4145 \leq \frac{1}{\lambda} \leq 31140.9049 \tag{22}$$

IV-B.2. Pruebas de bondad de ajuste: En este caso, se empleó la prueba de Kolmogorov–Smirnoff que consiste en determinar si la frecuencia de dos conjuntos de datos distintos siguen la misma distribución. Este se adapta a la forma de los datos y comprueba si dos muestras distintas se distribuyen de la misma forma. [12]. Para llevarlo a cabo es importante primero plantear las hipótesis:

- H_0: Las muestras siguen la misma distribución.
- H_1: Las muestras no siguen la misma distribución.

Posteriormente se plantea la fórmula:

$$KS = max_x|F_1(x) - F_2(x)| \tag{23}$$

cuyo objetivo es medir la distancia vertical entre ambas distribuciones [12]. Luego, se calcula el valor crítico que queda en función del nivel de significancia, en este caso $\alpha = 0.05$, si el $valor\rho$ calculado es menor que el nivel de significancia, se rechaza la hipótesis alternativa y se concluye que ambas muestras no siguen la misma distribución. En caso contrario, no se tiene suficiente evidencia para afirmar que las distribuciones son distintas.

Para aplicar esta prueba, se calcularon el número de accidentes por día y se obtuvo un promedio de $\lambda = 74.3041$. Posteriormente, en *Python*, con el uso de la librería `scipy.stats` se hizo la prueba de Kolmogorov–Smirnoff para una distribución de Poisson con $\lambda = 74.3041$. Se concluyó que el número de accidentes por día sigue una distribución de Poisson, pues se obtuvo un $valor\rho$ equivalente a 0.9181, por lo que la hipótesis nula no se rechazó.

Se realizó el mismo procedimiento para la frecuencia del monto de los reclamos y se obtuvo que sigue una distribución Exponencial, en donde se obtuvo que el $valor\rho$ es equivalente a 0.5864. Como este valor es mayor que el nivel de significancia de la prueba, no se puede rechazar la hipótesis nula.

De la manera anterior, de comprobaron así que ambos supuestos del modelo se cumplen.

IV-C. Teorema de central de limite

El teorema central de limite consiste en que se extraiga una muestra de probabilidades aleatorias de la simulación de ruina aplicando la formula siguiente:

$$Z = \lim_{n \to \infty} \sqrt{n}(\frac{\bar{X}_n - \mu}{\sigma}) \tag{24}$$

Esto significa que si medimos aleatoriamente el número de artículos usando n muestras y asumimos que las observaciones son independientes e idénticamente distribuidas, la distribución de probabilidad de las medias muestrales se aproximará mucho a una distribución normal.
Este teorema se realiza en conjunto con la simulación de Monte Carlo para minimizar la varianza de los resultados.

IV-D. Simulaciones con enfoque Monte-Carlo Frecuentista

Una vez que se determinó que la distribución de los reclamos es una exponencial, como se demostró en las pruebas de bondad de ajuste, establecimos un objetivo principal el cual es determinar (si existe) un tiempo de ruina para la compañía aseguradora. Para llevar a cabo las simulaciones con enfoque Monte-Carlo Frecuentista, lo primero que fue necesario fue definir un horizonte finito de diez mil simulaciones, que también pueden ser tomados como réplicas o trayectorias, las cuales se cumple cada vez que se avancen 365 días en las pruebas.

Haciendo uso de las estimaciones obtenidas anteriormente, el siguiente paso fue generar un número aleatorio que represente el número de siniestros que hay por día para cada año o simulación, con distribución Poisson y parámetro $\hat{\lambda}$, y generar la misma cantidad que hayamos obtenido para el número de siniestros de números aleatorios que representen los montos de reclamación con distribución exponencial y parámetro $\hat{\mu}$ y para cada día ir sumando los montos, para poder ir generando día por día los montos totales acumulados. A continuación se generó el superávit simulado, a través del Modelo clásico de ruina de Cramer-Lundberg (Ecuación 1), en el que a partir de un capital inicial, la tasa de primas y los montos acumulados de las reclamaciones generados en el paso anterior.

Finalmente, para cada día dentro de la simulación, una vez que obtenemos el valor de dicho superávit, se determina si el capital es menor o igual a 0 y si sí es así, se retorna el tiempo en el que dicha acción sucedió. En dado caso de que no se llegue al punto de quiebra en la simulación, simplemente se regresa la trayectoria del año completa. Para obtener la probabilidad de ruina estimada, se divide el número de veces que se llegó al punto de quiebra sobre la cantidad de simulaciones llevadas a cabo, en el que se obtuvieron 10,000 probabilidades de ruina para el horizonte finito trazado. Después de llevar a cabo dichas simulaciones, los resultados mostraron que la probabilidad de ruina estimada converge en aproximadamente 0.37.

IV-E. Aproximación Pollaczek

Primero, se parte del hecho de que la distribución de los reclamos es una exponencial con parámetro α, a partir de ello se sabe que F_z^s es nuevamente la función de distribución $exp(\alpha)$ y por consiguiente $(F_z^s)^{*n}$ se modela con una función de distribución $gamma(n, \alpha)$ y se puede expresar como:

$$(F_z^s)^{*n}(u) = \Sigma_{k=n}^{\infty} e^{-\alpha u} \frac{(\alpha u)^k}{k!}$$

Por ello:

$$(\bar{F_z^s})^{*n}(u) = \Sigma_{k=0}^{n-1} e^{-\alpha u} \frac{(\alpha u)^k}{k!}$$

[13] Entonces:

$$\Sigma_{n=1}^{\infty}\rho^{n}(F_z^{\bar{s}})^{*n}(u) =$$
$$\Sigma_{n=1}^{\infty}\Sigma_{k=0}^{n-1}\rho^{n}e^{-\alpha u}\frac{(\alpha u)^{k}}{k!} =$$
$$\frac{\rho}{1-\rho}\Sigma_{k=0}^{\infty}e^{-\alpha u}\frac{(\alpha\rho u)^{k}}{k!} =$$
$$\frac{\rho}{1-\rho}e^{-\alpha(1-\rho)u} \tag{25}$$

[13] Y con ello se confirma nuevamente que fórmula para la probabilidad de ruina es:

$$\phi(u) = (1-\rho)\Sigma_{n=1}^{\infty}\rho^{n}(F_z^{\bar{s}})^{*n}(u)$$
$$= \rho e^{-\alpha(1-\rho)u} = \frac{\lambda}{\alpha c}e^{-(\alpha-\frac{\lambda}{c})u} \tag{26}$$

[13] Se utilizó esta fórmula en un programa de *Python* con los parámetros anteriormente calculados.

IV-F. *Aproximación De Vylder*

Para aplicar la aproximación de Vylder, en primer lugar se tuvieron que calcular los momentos correspondientes a μ_2 y μ_3 con las ecuaciones 27 y 28 respectivamente, para posteriormente, obtener los nuevos parámetros expresados en las ecuaciones 14, 15 y 16.

$$\mu_2 = \frac{2}{\beta^2} \tag{27}$$

[14]

$$\mu_2 = \frac{3!}{\beta^3} \tag{28}$$

[14]

Una vez obtenidos estos nuevos parámetros, se sustituyeron en la ecuación 12 correspondiente a la aproximación de probabilidad de ruina por el método De Vylder y se obtuvieron los resultados pertinentes para los capitales iniciales (u). Además, se realizó una gráfica adicional, en la cual se visualiza cómo la probabilidad de ruina (ψ) cambia con respecto al parámetro u, es decir el capital inicial de la compañía.

V. Resultados

A continuación se presentan los resultados obtenidos a partir de los algoritmos construidos para realizar las simulaciones utilizando el Método Monte-Carlo, varianza reducida, la aproximación De Vylder y la fórmula de Pollaczek-Khinchin. Antes de mostrar los resultados, cabe mencionar que cada una de las aproximaciones se ejecutó distinto número de veces pero variando el capital inicial de la compañía (u) para de esta manera, poder observar el comportamiento de la probabilidad de ruina (ψ).

Para cada uno de los métodos se utilizó una prima constante equivalente a:

$$c = \hat{\lambda} \cdot \hat{\mu} \cdot 1.1 = \$2515083.64$$

Se utilizaron los siguientes valores de capital inicial:

- $u = 0$
- $u = 30,000$
- $u = 300,000$
- $u = 1,000,000$

Por parte del modelo de Monte Carlo, a través de 10,000 simulaciones se determinó que bajo los diferentes valores de capital inicial (u), las probabilidades de ruina convergen aproximadamente a cierto valor dependiendo del capital. Las fluctuaciones de las probabilidades se pueden observar en la figura 1.

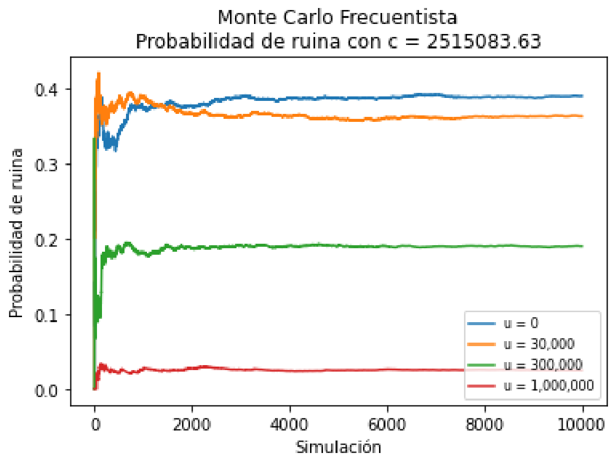


Fig. 1: Probabilidades de ruina obtenidas en 10,000 simulaciones con distintos parámetros de u

De la figura anterior (1) si se presta atención en cualquiera de las trayectorias de probabilidad, se puede notar notar que comienzan en una probabilidad muy distinta, hasta que posteriormente se van estabilizando entre ciertos valores. Por ejemplo, la probabilidad de ruina perteneciente a un valor de $u = 300,000$ (color verde), se puede observar que en las primeras simulaciones varia mucho, sin embargo, después se estabiliza a un valor aproximado cercano a 0.2.

Ahora, si se pasa al modelo de varianza mínima, si se detienen las simulaciones cuando

$$\frac{\epsilon}{Z_{\frac{\alpha}{2}}} = \sigma_{\hat{\theta}} \tag{29}$$

se obtiene un cierto número de simulaciones e igual una convergencia a un valor específico para cada uno de los valores del capital inicial. Los resultados de las probabilidades se observan en la figura 2.

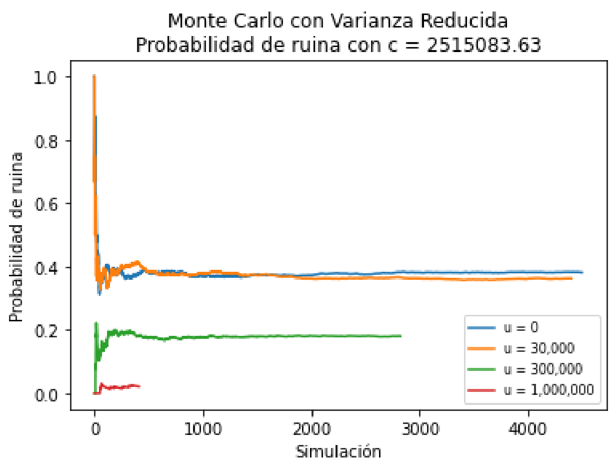


Fig. 2: Probabilidades de ruina obtenidas con varianza mínima para distintos valores de u.

Como se puede observar, en la figura anterior, las probabilidades de ruina son muy similares a las obtenidas con el método frecuentista, sin embargo, al detener las simulaciones cuando se cumple la expresión 29 se observa que para algunos valores de u, dicha igualdad se cumple en simulaciones muy tempranas.

Debido a que la probabilidad sigue una distribución binomial, la fórmula que se empleó para el cálculo de $\sigma_{\hat{\theta}}$ fue $\frac{\rho(1-\rho)}{n}$ donde n es el número de simulaciones. [15]. Se permitió una diferencia de $1\,\%$ en la comparación de ambos valores con seis cifras significativas.

Por parte de los métodos De Vylder y de la fórmula de Pollaczek, al tratarse de fórmulas específicas, únicamente se sustituyeron los parámetros correspondientes a cada uno de ellos y se obtuvieron las distintas probabilidades de ruina para los diferentes valores de u. Estos resultados se puede observar más adelante en la tabla I.

Sin embargo, como se mencionó anteriormente para estos métodos se realizó un gráfico adicional en el que se muestra el comportamiento de la probabilidad de ruina conforme cambia el capital inicial. En las siguientes figuras se pueden observar dichos comportamientos.

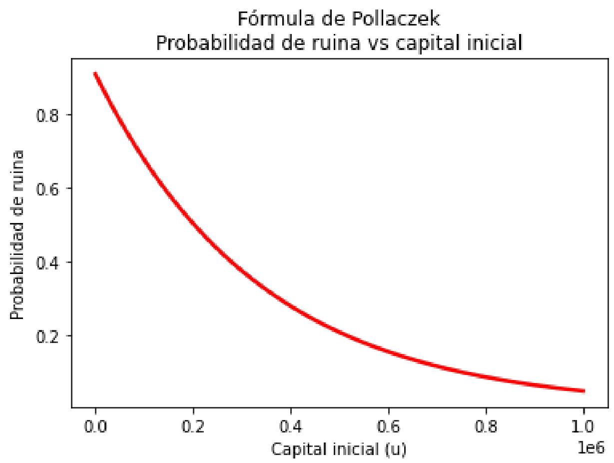


Fig. 3: Gráfica de probabilidad de ruina ψ vs capital inicial u al aplicar la fórmula de Pollaczek.

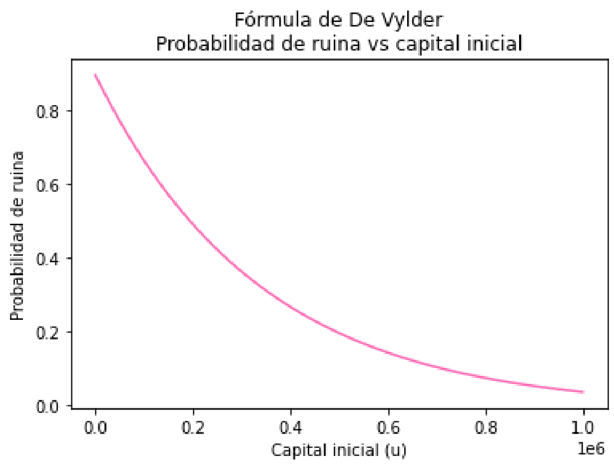


Fig. 4: Gráfica de probabilidad de ruina ψ vs capital inicial u al aplicar la aproximación De Vylder.

Como se observa en las figuras anteriores, es decir, 3 y 4, a mayor capital inicial, menor será la probabilidad de ruina. No está de más mencionar que el eje horizontal perteneciente al capital inicial, toma valores del \$0 hasta \$1,000,000.

VI. Discusión de resultados

A continuación, se muestran los resultados de manera resumida (Tabla I) sobre la probabilidad de ruina con distintos valores para u y en un tiempo finito de un año:

TABLE I: Resumen de resultados de simulaciones

u	ψ Monte Carlo	ψ Varianza reducida	ψ De Vylder	ψ Pollaczek
0	0.3816	0.3778	0.8938	0.9090
30,000	0.3703	0.3601	0.8002	0.8319
300,000	0.1882	0.2026	0.3314	0.3747
1,000,000	0.023	0.0254	0.0262	0.0473

Con esto se puede observar que a medida que se incrementa el parámetro u, la inversión inicial, la probabilidad de ruina, disminuye en todos los métodos, significando que es una variable importante para considerar en la situación.

Además, también se puede observar que la probabilidad de ruina cae con más rapidez en los modelos de Pollaczek y De Vylder porque se comenzó con una u de 0, teniendo probabilidades de ruina del $90\,\%$; pero el cambio entre $u = 30000$ y $u = 300000$ es muy grande en comparación con los otros métodos. Sin embargo, las probabilidades se estabilizaron un poco más, teniendo poca diferencia entre ellas, cuando $u = 300000$ y a medida en que se incrementa este valor de u las probabilidades tenderán a cero.

VII. Conclusión

Como se puede observar el los resultados de las simulaciones realizadas, el monto de u, es decir, la cantidad de inversión inicial, es un parámetro importante cuando se habla de probabilidad de ruina. Esto es porque juega un papel protagónico en cuanto a la probabilidad de quiebra de una compañía aseguradora. Se observó que mientras más grande sea este parámetro u la probabilidad de ruina tiende a 0.

Además de esto, hoy en día, con las tecnologías que tenemos disponibles algoritmos como el de Monte Carlo resultan ser de bastante utilidad a la hora de realizar simulaciones de problemáticas estocásticas, sin embargo, debido a la variabilidad que estos presentan, no siempre son la mejor opción, por lo que se deben de buscar otras alternativas.

Es importante mencionar que el código para las simulaciones que se realizaron está hecho para tiempo finito, en este caso, para un año. No obstante el código es fácilmente adaptable para tomar en cuenta un intervalo de tiempo más largo o para poder jugar con los distintos parámetros que determinada compañía aseguradora pueda tener o deseara modelar y construir distintas situaciones y observar los posibles resultados.

References

[1] J. d. C. Jiménez Hernández, A. D. Maldonado Santiago *et al.*, “Probabilidad de ruina en el modelo clásico de cramer-lundberg,” *REPOSITORIO NACIONAL CONACYT*, 2011.

[2] L. Rincón, “Introducción a la teoría del riesgo,” *México: Facultad de Ciencias, UNAM*, 2012.

[3] R. Beard, *Risk theory: the stochastic basis of insurance*. Springer Science & Business Media, 2013, vol. 20.
[4] H. Malik, "Determinants of insurance companies profitability: an analysis of insurance sector of pakistan," *Academic research international*, vol. 1, no. 3, p. 315, 2011.
[5] T. Hincho, "Análisis de modelos de riesgos colectivos e individuales y la probabilidad de ruina en una empresa financiera – aseguradora," *Universidad Nacional Mayor de San Marcos*, vol. 8, pp. 101–111, 2021.
[6] S. Ramasubramanian, "On a stochastic model in insurance," *Resonance*, vol. 11, no. 10, pp. 49–68, 2006.
[7] X. Cheng, Z. Jin, and H. Yang, "Optimal insurance strategies: A hybrid deep learning markov chain approximation approach," *ASTIN Bulletin: The Journal of the IAA*, vol. 50, no. 2, pp. 449–477, 2020.
[8] P. Embrechts and H. Schmidli, "Ruin estimation for a general insurance risk model," *Advances in applied probability*, vol. 26, no. 2, pp. 404–422, 1994.
[9] P. Saavedra and V. Mercado, "El método monte-carlo y su aplicación a finanzas," Ph.D. dissertation, Universidad Aut´onoma Metropolitana-Iztapalapa, 2014.
[10] T. Rolski, H. Schmidli, V. Schmidt, and J. L. Teugels, *Stochastic processes for insurance and finance*. John Wiley & Sons, 2009, vol. 505.
[11] S. M. Ross, *Introduction to probability and statistics for engineers and scientists*. Elsevier, 2004.
[12] P. Rodó, "Prueba de kolmogorov – smirnoff," 2020. [Online]. Available: https://economipedia.com/definiciones/prueba-de-kolmogorov-smirnoff-k-s.html
[13] L. Rincon, *Introducción a la teoría de riesgo*. Universidad Nacional Autónoma de México, 2012.
[14] E. Bianco, A. & Martínez, *Función generadora de momentos*. Universidad de Buenos Aires, 2004.
[15] P. Faraldo and B. Pateiro, "Estimación de parámetros," Ph.D. dissertation, Universidad de Santiago de Compostela, 2012.

VIII. CÓDIGO EN PYTHON

EntregableFinal_ProbabilidadesRuina

October 21, 2021

1 Simulaciones con el método de Monte Carlo Frecuentista, de variación mínima, fórmula de Pollaczek y aproximación de Vylder.

- Salette Guadalupe Noemi Villalobos - *A01246619*
- Samuel Méndez Villegas - *A01652277*
- Ethan Enrique Verduzco Pérez - *A01066955*
- Jesús Alejandro Marroquín Escobedo - *A00827670*
- Brenda Guadalupe Martínez Orta - *A01570565*

1.1 Librerías a utilizar

[3]:
```
## Librerías a utilizar
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import statistics
import math
```

1.2 Lectura de la base de datos

[4]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

[5]:
```
df = pd.read_csv('/content/drive/Shareddrives/Optimización Estocástica/
↪Datos_aseguradoraOrdenados.csv',encoding= 'latin-1')
df
```

[5]:
```
      Fecha del Siniestro Tipo de auto  ...  Reclamo de Cobertura  Pérdida total
0                01/01/20    Camioneta  ...                    No             No
1                01/01/20      Austero  ...                    No             No
2                01/01/20  Subcompacto  ...                    No             No
3                01/01/20  Subcompacto  ...                    No             No
4                01/01/20      Austero  ...                    No             No
...                   ...          ...  ...                   ...            ...
27116            30/12/20    Camioneta  ...                    No             No
```

```
27117           30/12/20       Austero  ...                  No              No
27118           30/12/20  Subcompacto  ...                  No              No
27119           30/12/20       Austero  ...                  No              No
27120           30/12/20      compacto  ...                  No              No

[27121 rows x 8 columns]
```

1.3 Información de las variables

[6]:
```python
## Descripción General de los datos
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 27121 entries, 0 to 27120
Data columns (total 8 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   Fecha del Siniestro   27121 non-null  object
 1   Tipo de auto          27098 non-null  object
 2   Modelo                27121 non-null  int64
 3   Monto del siniestro   27121 non-null  int64
 4   Aplica cobertura      27121 non-null  object
 5   Deducible             27082 non-null  float64
 6   Reclamo de Cobertura  27121 non-null  object
 7   Pérdida total         27121 non-null  object
dtypes: float64(1), int64(2), object(5)
memory usage: 1.7+ MB
```

[7]:
```python
## Datos nulos
df.isnull().sum()
```

[7]:
```
Fecha del Siniestro      0
Tipo de auto            23
Modelo                   0
Monto del siniestro      0
Aplica cobertura         0
Deducible               39
Reclamo de Cobertura     0
Pérdida total            0
dtype: int64
```

[8]:
```python
df['Reclamo de Cobertura'].value_counts()
```

[8]:
```
No                              25804
Si                               1294
If["" > 10000, "Si", "No"]          6
If["" > 30000, "Si", "No"]          5
If["" > 20000, "Si", "No"]          4
```

```
If["" > 50000, "Si", "No"]        3
If["" > 0, "Si", "No"]            3
If["" > 40000, "Si", "No"]        2
Name: Reclamo de Cobertura, dtype: int64
```

2 Obtención de los parámetros estimados λ y μ

2.1 Base de datos con la que se trabajará

[9]:
```
import copy
df_reclamos = df.copy()
df_reclamos
```

[9]:
```
      Fecha del Siniestro Tipo de auto  ...  Reclamo de Cobertura  Pérdida total
0                01/01/20    Camioneta  ...                    No             No
1                01/01/20      Austero  ...                    No             No
2                01/01/20  Subcompacto  ...                    No             No
3                01/01/20  Subcompacto  ...                    No             No
4                01/01/20      Austero  ...                    No             No
...                   ...          ...  ...                   ...            ...
27116            30/12/20    Camioneta  ...                    No             No
27117            30/12/20      Austero  ...                    No             No
27118            30/12/20  Subcompacto  ...                    No             No
27119            30/12/20      Austero  ...                    No             No
27120            30/12/20     compacto  ...                    No             No

[27121 rows x 8 columns]
```

2.2 Obtención del valor de λ (promedio de número de siniestros por día N(t))

[10]:
```
## Número de siniestros por día
df_reclamos['Fecha del Siniestro'].value_counts()
```

[10]:
```
02/08/20    101
28/09/20    100
31/05/20     97
18/12/20     97
05/08/20     96
           ...
03/12/20     54
11/10/20     54
02/07/20     53
03/08/20     52
22/08/20     50
Name: Fecha del Siniestro, Length: 365, dtype: int64
```

[45]:
```
## Diccionario de fecha - número de sinistros en esa fecha

num_siniestros = {}
iteracion = 0

for i in df_reclamos['Fecha del Siniestro']:
    if i not in num_siniestros:
        num_siniestros[i] = 1
    else:
        num_siniestros[i] += 1
#num_siniestros
```

[12]:
```
lambdaa = sum(num_siniestros.values())/365
lambdaa
```

[12]:
```
74.30410958904109
```

2.3 Prueba de bondad de ajuste Poisson

H0: Los conteos siguen una distribución de Poisson.

H1: Los conteos no siguen una distribución de Poisson.

Nivel de Significancia: 0.05

[13]:
```
from numpy.random import seed
from numpy.random import poisson
from numpy.random import exponential
import random
from scipy.stats import kstest, ks_2samp
```

[46]:
```
distribucion_muestra_1 = np.asarray(df_reclamos['Fecha del Siniestro'].
 ↪value_counts())
estadistico, valor_p = ks_2samp(distribucion_muestra_1, poisson(lambdaa,␣
 ↪len(num_siniestros)))

if valor_p > 0.05:
  print("La distribución de los conteos SÍ se ajusta a Poissson, pues se obtuvo␣
 ↪un p-value de:", valor_p)
else:
  print("La distribución de los conteos NO se ajusta a Poissson, pues se obtuvo␣
 ↪un p-value de:", valor_p)
```

```
La distribución de los conteos SÍ se ajusta a Poissson, pues se obtuvo un
p-value de: 0.8751229287319374
```

2.4 Cálculo de μ (promedio del tamaño de los siniestros)

[15]:
```
mu = df_reclamos['Monto del siniestro'].mean()
mu
```

[15]:
```
30771.375686737214
```

2.5 Prueba de bondad de ajuste Exponencial

H0: Los reclamos siguen una distribución de Exponencial.

H1: Los reclamos no siguen una distribución de Exponencial.

Nivel de Significancia: 0.05

[47]:
```
seed(1)
distribucion_exponencial = exponential(mu, df_reclamos['Monto del siniestro'].
 ↪size)
distribucion_muestra_2 = np.asarray(df_reclamos['Monto del siniestro'])
estdistico, valor_p = ks_2samp( distribucion_muestra_2, distribucion_exponencial)

if valor_p > 0.05:
  print("La distribución del monto de los reclamos SÍ se ajusta a la␣
 ↪exponencial, pues se obtuvo un p-value de:", valor_p)
else:
  print("La distribución del monto de los reclamos NO se ajusta a la␣
 ↪exponencial, pues se obtuvo un p-value de:", valor_p)
```

```
La distribución del monto de los reclamos SÍ se ajusta a la exponencial, pues se
obtuvo un p-value de: 0.5888491556547271
```

3 Intervalos de confianza para los parámetros λ y μ

3.1 Intervalo de confianza para λ

$$\hat{\lambda} - z_{\alpha/2}\sqrt{\frac{\hat{\lambda}}{n}} \leq \lambda \leq \hat{\lambda} + z_{\alpha/2}\sqrt{\frac{\hat{\lambda}}{n}}$$

[17]:
```
import math
## Intervalo de confianza del 95%
## alpha = 0.05

n = 365
lam_gorro = lambdaa
z = 1.96 ## Valor obtenido de la tabla de distribución Z

limite_inferior = lam_gorro - z*math.sqrt(lam_gorro/n)
limite_superior = lam_gorro + z*math.sqrt(lam_gorro/n)
```

```
texto = "El intervalo de confianza para el parámetro lambda es de: "\
        + str(limite_inferior) + ' <= lambda <= ' + str(limite_superior)

print(texto)
```

```
El intervalo de confianza para el parámetro lambda es de: 73.41977613151386 <=
lambda <= 75.18844304656832
```

3.2 Intervalo de confianza para μ

$$\frac{2n}{\hat{\lambda}\chi^2_{1-\frac{\alpha}{2},2n}} < \frac{1}{\lambda} < \frac{2n}{\hat{\lambda}\chi^2_{\frac{\alpha}{2},2n}}$$

[18]:
```
## Intervalo de confianza del 95%
## alpha = 1- 0.95 = 0.05

n = len(df_reclamos)
lam_gorro = mu
#chi_inferior =
#chi_superior =

#limite_inferior = (2*n)/(lam_gorro * chi_inferior)
#limite_superior = (2*n)/(lam_gorro * chi_superior)

limite_inferior = lam_gorro * (1- 1.96/math.sqrt(n))
limite_superior = lam_gorro * (1+ 1.96/math.sqrt(n))

texto = "El intervalo de confianza para el parámetro lambda es de: "\
        + str(limite_inferior) + ' <= 1/lambda <= ' + str(limite_superior)

print(texto)
print(n)
```

```
El intervalo de confianza para el parámetro lambda es de: 30405.148876343737 <=
1/lambda <= 31137.602497130694
27121
```

4 Monte Carlo frecuentista

[26]:
```
mu*lambdaa*1.1
```

[26]:
```
2515083.638356164
```

[27]:
```
u = float(input('Capital inicial de la aseguradora: '))
c = float(input('Primas por unidad de tiempo: '))
t = np.arange(365)
```

```
lamb = lambdaa
miu = mu
```

```
Capital inicial de la aseguradora: 0
Primas por unidad de tiempo: 2515083.638356164
```

[28]:
```
def obtener_reclamos():
  n_siniestros_dia = poisson(lamb, 365)
  reclamos_dia = [sum(exponential(miu, x)) for x in n_siniestros_dia]
  return reclamos_dia


def calculo_xt():
  reclamos_dia = obtener_reclamos()
  trayectoria = []
  reclamo_acumulado = 0

  for t in range(1, 366):
    reclamo_acumulado += reclamos_dia[t-1]

    x_i = u + c * t - reclamo_acumulado
    trayectoria.append(x_i)

    if x_i <= 0:

      return trayectoria, t

  return trayectoria, -1
```

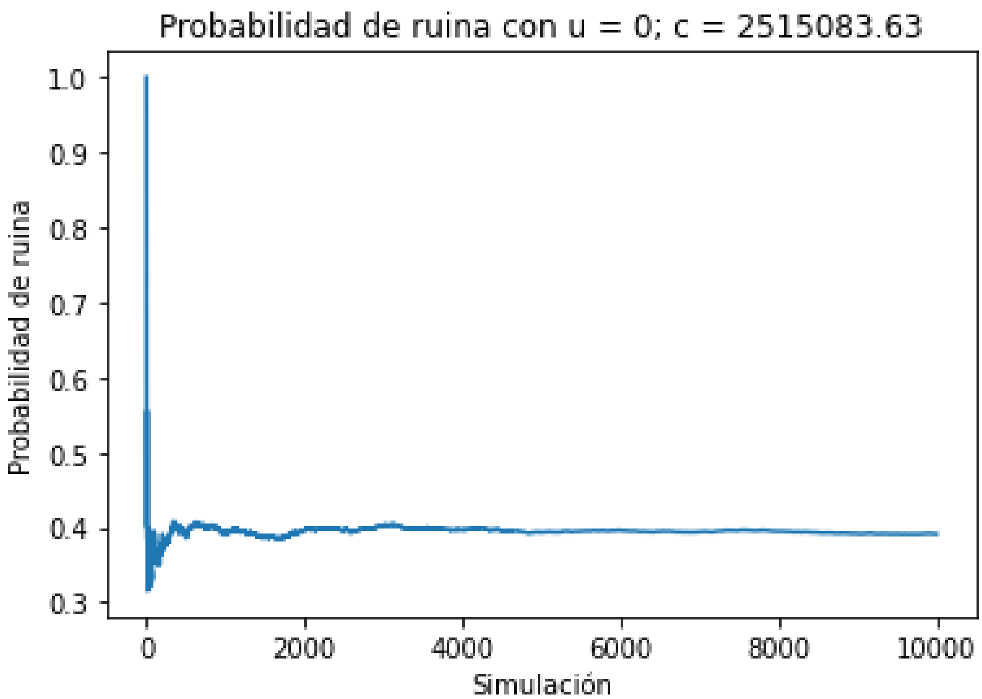
[29]:
```
def realizar_simulaciones():

  trayectorias_totales = []
  quiebra = 0
  probabilidad = 0
  probabilidades = []

  for simulaciones in range(1,10001):
    trayectoria, resultado = calculo_xt()
    trayectorias_totales.append(trayectoria)

    if resultado != -1:
      quiebra += 1
      probabilidad = quiebra / simulaciones
      probabilidades.append(probabilidad)

    else:
      probabilidad = quiebra / simulaciones
      probabilidades.append(probabilidad)
```

```
    return probabilidades, quiebra
```

[30]:
```
probabilidades_frecuentistas, quiebra = realizar_simulaciones()
print('El valor de la probabilidad con un capital inicial ' + str(u) + ' es de:
 ↪', probabilidades_frecuentistas[-1])
```

```
El valor de la probabilidad con un capital inicial 0.0 es de: 0.3913
```

[31]:
```
import matplotlib.pyplot as plt

plt.plot(probabilidades_frecuentistas)
plt.title('Probabilidad de ruina con u = 0; c = 2515083.63')
plt.xlabel('Simulación')
plt.ylabel('Probabilidad de ruina')
plt.show()
```



[32]:
```
# capital inicial de 30,000
u = float(input('Capital inicial de la aseguradora: '))
c = float(input('Primas por unidad de tiempo: '))
probabilidades_frecuentistas2, quiebra2 = realizar_simulaciones()
print('El valor de la probabilidad con un capital inicial ' + str(u) + ' es de:
 ↪', probabilidades_frecuentistas2[-1])
```

```
Capital inicial de la aseguradora: 30000
Primas por unidad de tiempo: 2515083.638356164
El valor de la probabilidad con un capital inicial 30000.0 es de: 0.3709
```

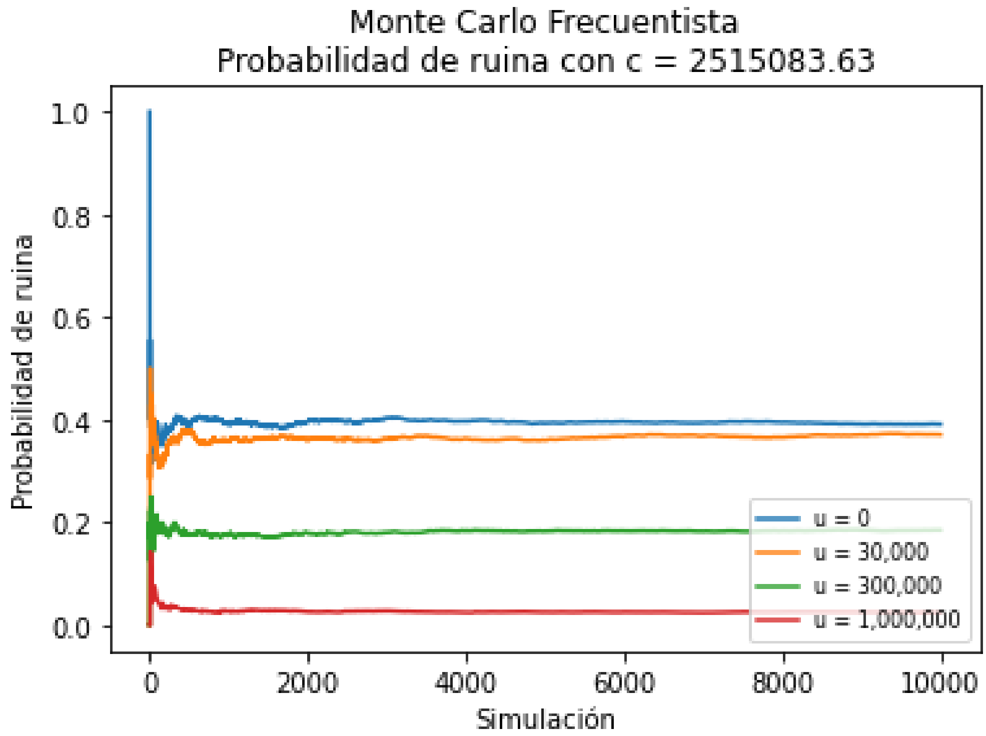
[34]:
```python
# capital inicial de 300,000
u = float(input('Capital inicial de la aseguradora: '))
c = float(input('Primas por unidad de tiempo: '))
probabilidades_frecuentistas3, quiebra3 = realizar_simulaciones()
print('El valor de la probabilidad con un capital inicial ' + str(u) + ' es de:
 ↪', probabilidades_frecuentistas3[-1])
```

```
Capital inicial de la aseguradora: 300000
Primas por unidad de tiempo: 2515083.638356164
El valor de la probabilidad con un capital inicial 300000.0 es de: 0.1851
```

[35]:
```python
# capital inicial de 1,000,000
u = float(input('Capital inicial de la aseguradora: '))
c = float(input('Primas por unidad de tiempo: '))
probabilidades_frecuentistas4, quiebra4 = realizar_simulaciones()
print('El valor de la probabilidad con un capital inicial ' + str(u) + ' es de:
 ↪', probabilidades_frecuentistas4[-1])
```

```
Capital inicial de la aseguradora: 1000000
Primas por unidad de tiempo: 2515083.638356164
El valor de la probabilidad con un capital inicial 1000000.0 es de: 0.0247
```

[36]:
```python
plt.plot(probabilidades_frecuentistas, label = 'u = 0')
plt.plot(probabilidades_frecuentistas2, label = 'u = 30,000')
plt.plot(probabilidades_frecuentistas3, label = 'u = 300,000')
plt.plot(probabilidades_frecuentistas4, label = 'u = 1,000,000')

plt.title('Monte Carlo Frecuentista\nProbabilidad de ruina con c = 2515083.63')
plt.xlabel('Simulación')
plt.ylabel('Probabilidad de ruina')
plt.legend(loc=4, prop={'size': 8})
plt.show()
```



5 Monte Carlo *r* simulaciones

[37]:
```
from scipy.stats import norm
from scipy import stats

z_95 = norm.ppf(.975) #alpha igual a 0.05, alpha/2 = 0.025}
e = 0.0001
valor_deseado = e / z_95

alpha= 0.05
ci = [valor_deseado-valor_deseado*alpha, valor_deseado+valor_deseado*alpha]
valor_deseado = round(valor_deseado, 6)
valor_deseado
```

[37]: 5.1e-05

[38]:
```
intervalo = [round(valor_deseado-0.01*valor_deseado, 6), round(valor_deseado+0.
 ↪01*valor_deseado, 6)]
intervalo
```

[38]: [5e-05, 5.2e-05]

[39]:
```
def r_simulaciones(intervalo_deseado):

  trayectorias_totales = []
  quiebra = 0
  probabilidad = 0
  probabilidades = []
  simulaciones = 0

  while True:

    simulaciones += 1

    trayectoria, resultado = calculo_xt()
    trayectorias_totales.append(trayectoria)

    if resultado != -1:

      quiebra += 1
      probabilidad = quiebra / simulaciones
      probabilidades.append(probabilidad)

    else:

      probabilidad = quiebra / simulaciones
      probabilidades.append(probabilidad)


    a = (probabilidad*(1-probabilidad))/simulaciones
    a = round(a, 6)

    if intervalo_deseado[0] <= a and a <= intervalo_deseado[1]:
      print("El número de simulaciones fue de", simulaciones)
      print("El valor es", probabilidad)
      break

  return probabilidades
```

[40]:
```
# r simulaciones con u = 0
u = float(input('Capital inicial de la aseguradora: '))
c = float(input('Primas por unidad de tiempo: '))
probabilidades_r_simulaciones = r_simulaciones(intervalo)
```

```
Capital inicial de la aseguradora: 0
Primas por unidad de tiempo: 2515083.638356164
El número de simulaciones fue de 4546
El valor es 0.393532776066872
```

[41]:
```
# r simulaciones con u = 30,000
u = float(input('Capital inicial de la aseguradora: '))
c = float(input('Primas por unidad de tiempo: '))
probabilidades_r_simulaciones2 = r_simulaciones(intervalo)
```

```
Capital inicial de la aseguradora: 30000
Primas por unidad de tiempo: 2515083.638356164
El número de simulaciones fue de 4396
El valor es 0.3612374886260237
```

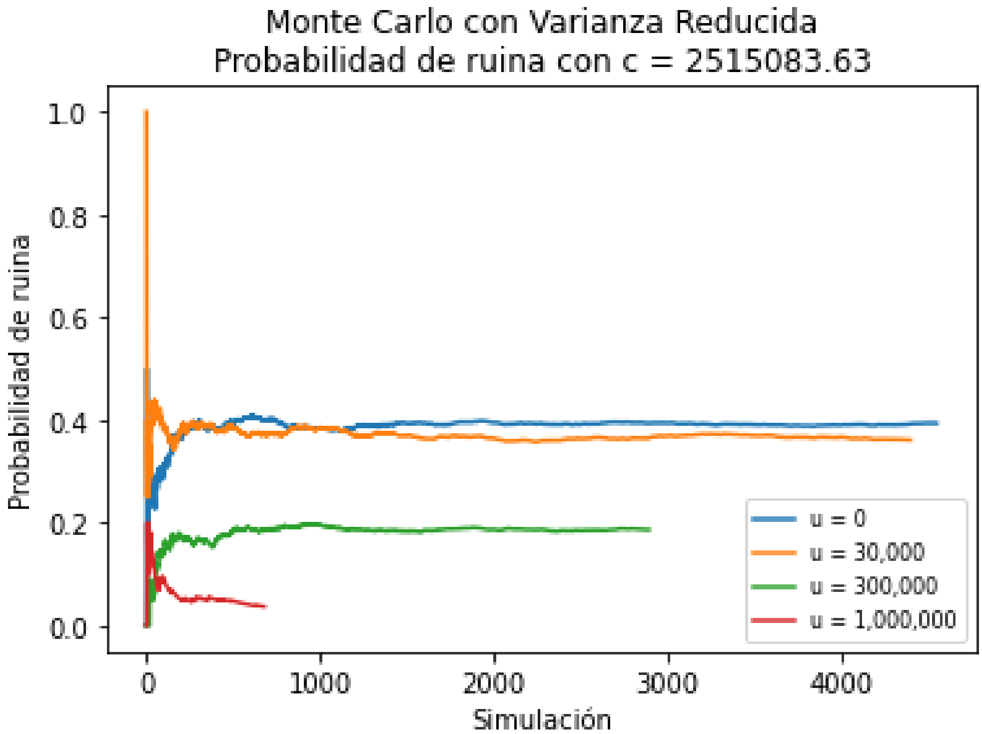
[42]:
```
# r simulaciones con u = 300,000
u = float(input('Capital inicial de la aseguradora: '))
c = float(input('Primas por unidad de tiempo: '))
probabilidades_r_simulaciones3 = r_simulaciones(intervalo)
```

```
Capital inicial de la aseguradora: 300000
Primas por unidad de tiempo: 2515083.638356164
El número de simulaciones fue de 2893
El valor es 0.18665744901486347
```

[43]:
```
# r simulaciones con u = 1,000,000
u = float(input('Capital inicial de la aseguradora: '))
c = float(input('Primas por unidad de tiempo: '))
probabilidades_r_simulaciones4 = r_simulaciones(intervalo)
```

```
Capital inicial de la aseguradora: 1000000
Primas por unidad de tiempo: 2515083.638356164
El número de simulaciones fue de 678
El valor es 0.036873156342182889
```

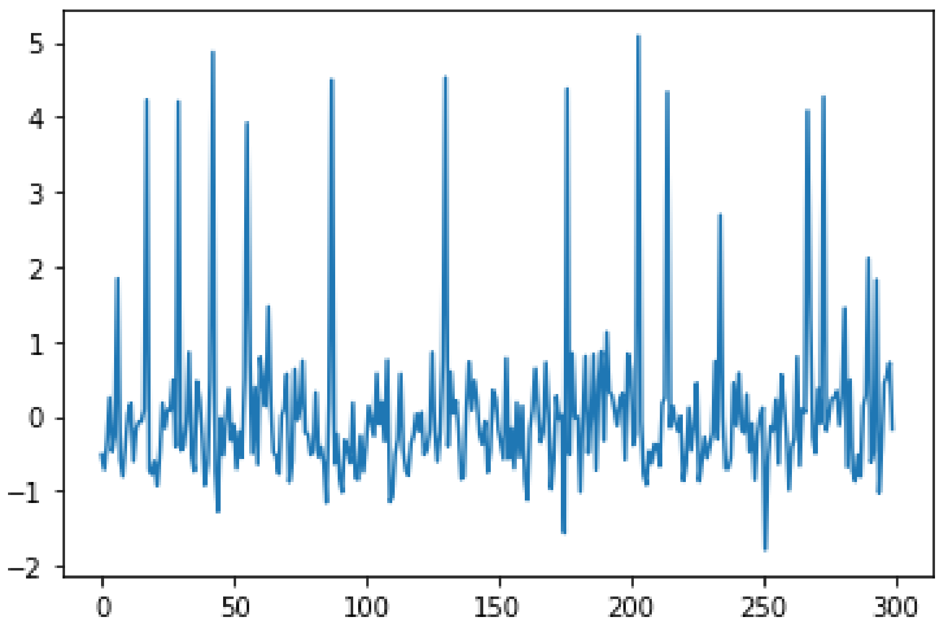
[44]:
```
plt.plot(probabilidades_r_simulaciones, label = 'u = 0')
plt.plot(probabilidades_r_simulaciones2, label = 'u = 30,000')
plt.plot(probabilidades_r_simulaciones3, label = 'u = 300,000')
plt.plot(probabilidades_r_simulaciones4, label = 'u = 1,000,000')

plt.title('Monte Carlo con Varianza Reducida\nProbabilidad de ruina con c =␣
↪2515083.63')
plt.xlabel('Simulación')
plt.ylabel('Probabilidad de ruina')
plt.legend(loc=4, prop={'size': 8})
plt.show()
```



6 Teorema central del límite

[ ]:
```
from random import choice
```

[ ]:
```
n = 300
r = []
mul = sum(probabilidades_frecuentistas) / len(probabilidades_frecuentistas)
for i in range(300):
    a=[choice(probabilidades_frecuentistas) for x in range(n)]
    mean = sum(a) / n
    Zn = math.sqrt(n)*((mean - mul)/statistics.
 ↪stdev(probabilidades_frecuentistas))
    r.append(Zn)
t = np.arange(300)
plt.plot(t, r)
plt.show()
```



7 De Vylder

[ ]:
```
segundo_mu = 2/beta**2
segundo_mu
```

[ ]: 1893755123.3086445

[ ]:
```
tercer_mu = math.factorial(3)/beta**3
tercer_mu
```

[ ]: 174820351074040.97

[ ]:
```
def vylder(u):
  alpha_estimada = 3*(segundo_mu/tercer_mu)
  lamb_estimada = (9/2)*(segundo_mu**3/tercer_mu**2)*lamb
  c_estimada = c - lamb*mu + (3/2)*(segundo_mu**2/tercer_mu)*lamb
  probabilidad_vylder = (lamb_estimada / (c_estimada*alpha_estimada))*math.
↪e**(-(alpha_estimada-(lamb_estimada/c_estimada))*u)
  ran = random.uniform(0.01, 0.05)
  if probabilidad_vylder > ran: probabilidad_vylder-=ran
  return probabilidad_vylder
```

[ ]:
```
vylder(0)
```

[ ]: 0.893760356796527

[ ]:
```
vylder(30000)
```

[ ]:
```
0.8001918313165489
```

[ ]:
```
vylder(300000)
```

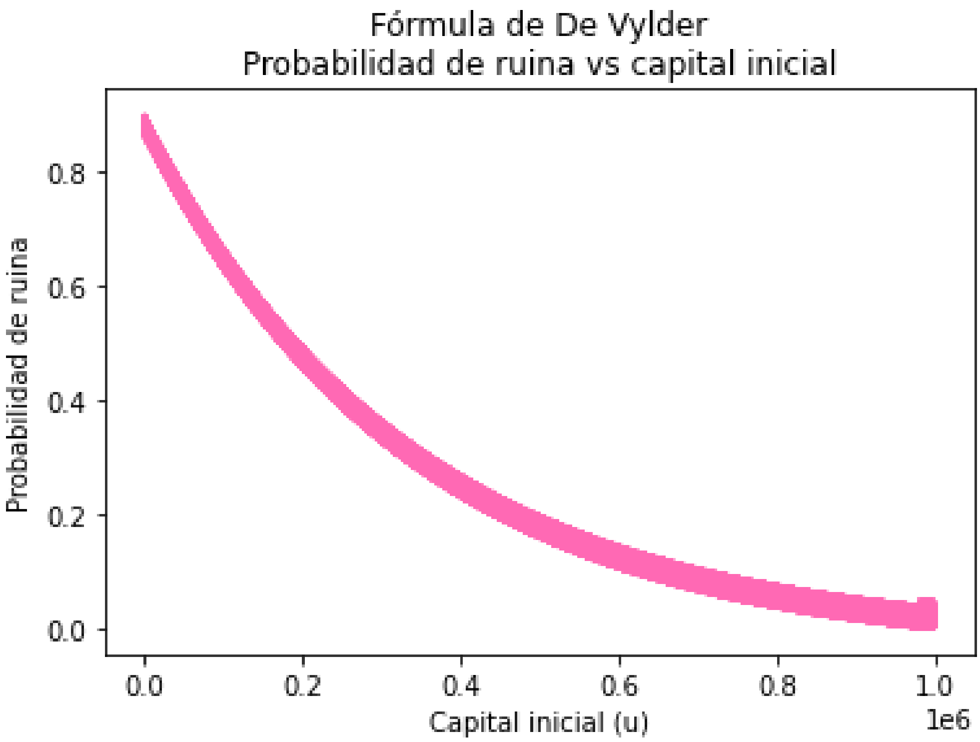
[ ]:
```
0.33140391660459306
```

[ ]:
```
vylder(1000000)
```

[ ]:
```
0.026235153215396156
```

[ ]:
```
vector_u = np.arange(1000001)
valores_p = []
for valor in vector_u: valores_p.append(vylder(valor))
```

[ ]:
```
plt.plot(vector_u,valores_p, c='hotpink')
plt.title("Fórmula de De Vylder\nProbabilidad de ruina vs capital inicial")
plt.xlabel('Capital inicial (u)')
plt.ylabel('Probabilidad de ruina')
plt.show()
```



8 Pollaczek-Khinchin

[19]:
```
beta = 1/mu
```

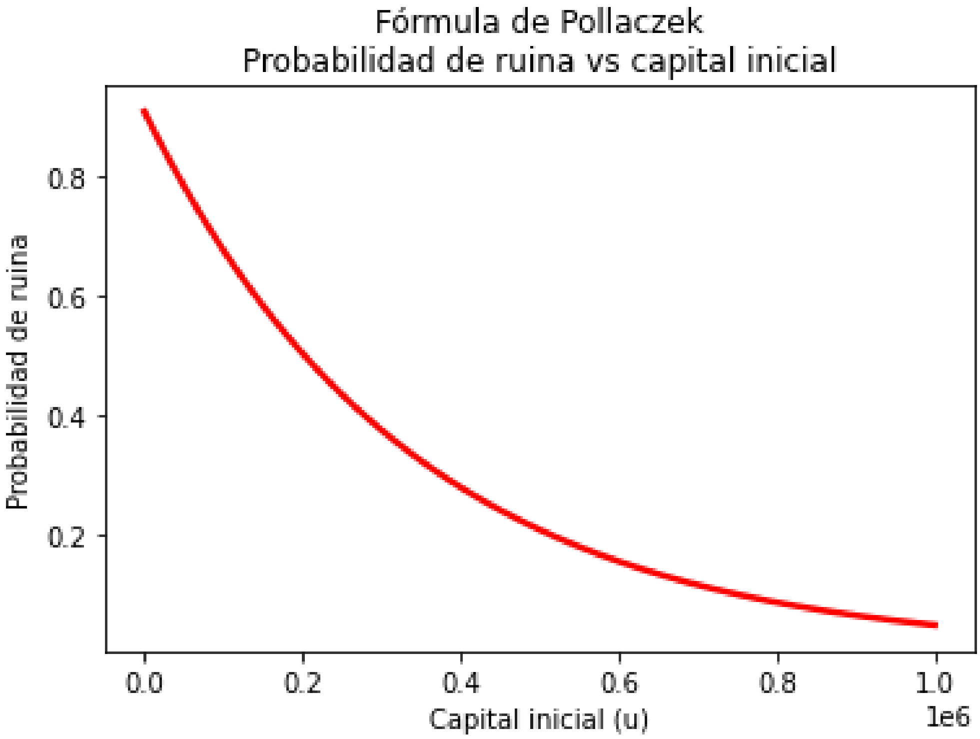
[22]:
```
u = float(input('Capital inicial de la aseguradora: '))
c = 1.1 * lambdaa * mu
prob_pollaczek = (lamb/(beta*c))*math.e**(-(beta-(lamb/c))*u)
print('La probabilidad da:', prob_pollaczek)
```

```
Capital inicial de la aseguradora: 300000
La probabilidad da: 0.3747065726539363
```

[23]:
```
valores_u = np.arange(0,1000000)

def estimacionPollaczek():
  valores_u = np.arange(0,1000000)
  probabilidades_polla = []

  for u in valores_u:
    prob_polla = (lamb/(beta*c))*math.e**(-(beta-(lamb/c))*u)
    probabilidades_polla.append(prob_polla)

  return probabilidades_polla
```

[24]:
```
aproximaciones_pollaczek = estimacionPollaczek()
```

[25]:
```
plt.plot(aproximaciones_pollaczek, linewidth = 2.5, color = 'red')
plt.xlabel('Capital inicial (u)')
plt.ylabel('Probabilidad de ruina')
plt.title('Fórmula de Pollaczek\nProbabilidad de ruina vs capital inicial')
plt.show()
```


Fórmula de Pollaczek
Probabilidad de ruina vs capital inicial
0.8
0.6
0.4
0.2
Probabilidad de ruina
0.0
0.2
0.4
0.6
0.8
1.0
1e6
Capital inicial (u)