Universidad Autónoma de Chiapas

Facultad de Ciencias en Física y Matemáticas

Estudio del impacto de la pandemia en un establecimiento de comida rápida: Análisis y estimación de parámetros para el modelo modificado de Cramer-Lundberg

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

LICENCIADA EN MATEMÁTICAS APLICADAS

PRESENTA:

MARA DOMINGUEZ LIMAS

Director:

Dr. Yofre Hernán García Gómez

Tuxtla Gutiérrez, Chiapas a - de - del 2024.

Dedicatoria

Agradecimientos

Tabla de contenidos

1. Introducción

Parte I.

Preliminares

2. Probabilidad

La probabilidad es el lenguaje necesario para representar eventos aleatorios. Se utiliza para la toma de decisiones basadas en la incertidumbre y para predecir la frecuencia con la que ocurrirán ciertos eventos en condiciones especificas. Permite cuantificar las posibilidades de que ocurra un evento y predecir resultados. La probabilidad se expresa como un número entre 0 y 1.

El origen de la teoría de la probabilidad se remonta a los siglos XVI y XVII, cuando surgió como respuesta a problemas prácticos relacionados con juegos de azar.

La teoría de la probabilidad ha evolucionado desde problemas prácticos en juegos de azar hasta convertirse en una disciplina matemática rigurosa con aplicaciones en numerosos campos como la estadística, la física, la economía, y la ingeniería. Los esfuerzos de muchos matemáticos a lo largo de los siglos han contribuido a su desarrollo y formalización.

La probabilidad tiene muchas aplicaciones prácticas en la vida cotidiana. Por ejemplo la probabilidad puede determinar las posibilidades de ganar en juegos de azar como la lotería, el póker, la ruleta y otros juegos de casino, hasta ser usada para las compañías de seguros para calcular las primas de los seguros de vida, salud, automóviles, entre otros. Esto se basa en el análisis de datos históricos y la evaluación de riesgos.

Para brindar mayor conocimiento e indagar una idea mas concreta de lo que es la probabilidad se presentan las siguientes definiciones: (Castañeda, Arunachalam, y Dharmaraja 2012)

Definición 2.1 (Probabilidad clásica)**.** Si un experimento aleatorio puede resultar en n resultados mutuamente excluyentes e igualmente probables y si s de estos resultados tienen un atributo A, entonces la probabilidad de A es la fracción s/n.

Definición 2.2 (Probabilidad frecuentista)**.** Suponiendo que después de n repeticiones, para valores muy grandes de n, un evento A puede ocurrir s veces. Entonces $p = s/n$.

Definición 2.3 (σ-álgebra)**.** Sea $\Omega \neq \Phi$. Una colección $\mathfrak{F}$ de subconjuntos de Ω es una σ-álgebra sobre Ω, si:

i. $\Omega \in \mathfrak{F}$.

ii. Si $A \in \mathfrak{F}$ entonces $A^c \in \mathfrak{F}$.

iii. Si $A_1, A_2, ... \in \mathfrak{F}$ entonces $\bigcup_{i=1}^{\infty} A_i \in \mathfrak{F}$.

Los elementos de $\mathfrak{F}$ se llaman eventos.

Definición 2.4 (Espacio medible)**.** Sean $\Omega \neq \Phi$ y $\mathfrak{F}$ una $\sigma - lgebra$ sobre Ω. La pareja $(\Omega, \mathfrak{F})$ se llama espacio medible.

Definición 2.5 (Eventos mutuamente excluyentes)**.** Dos eventos A y B se dicen mutuamente excluyentes si $A \cap B = \Phi$

Definición 2.6 (Espacio de probabilidad)**.** Sea $(\Omega, \mathfrak{F})$ un espacio medible (Definición 2.4). Una función P definida sobre $\mathfrak{F}$ y de valor real que satisface las siguientes condiciones:

i. $P(A) \geq 0$ para todo $A \in \mathfrak{F}$.

ii. $P(\Omega) = 1$.

iii. Si $A_1, A_2, ...$ son elementos de $\mathfrak{F}$ mutuamente excluyentes, esto es

$$A_j \ \cap \ A_j = \Phi \ para \ todo \ i \neq j$$

$$entonces \ P\left(\bigcup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty} P(A_i)$$

Se llama medida de probabilidad sobre $(\Omega, \mathfrak{F})$.

La tripla $(\Omega, \mathfrak{F}, P)$ se llama espacio de probabilidad.

2.1. Variables aleatorias

En estadística y probabilidad, una variable aleatoria es una función que asigna un valor numérico a cada posible resultado de un experimento aleatorio. Las variables aleatorias son

Definición 2.7 (Variable aleatoria)**.** Sean $(\Omega, \mathfrak{F}, P)$ un espacio de probabilidad (Definición 2.6) y $\tilde{\Omega}, \tilde{\mathfrak{F}}$ un espacio medible (Definición 2.4). Una $\mathfrak{F} - \mathfrak{F}$- variable aleatoria es una aplicación $X : \Omega \to \tilde{\Omega}$ tal que, para todo $A \in \mathfrak{F}$ se tiene que $X^{-1}(A) \in \mathfrak{F}$. Si $(\tilde{\Omega}, \tilde{\mathfrak{F}}) = (\mathbb{R}, \mathcal{B})$, entonces, se dice que X es una variable aleatoria real

Ejemplo (***Variable aleatoria***)

Sean

2.2. Variables aleatorias discretas

Definición 2.8 (Variable aleatoria discreta)**.** Sean X una variable aleatoria real (Definición 2.7) y F_X su función de distribución. Se dice que F_X presenta un salto en el punto $a \in \mathbb{R}$ si

$$F_X(a) - F_X(a^-) \neq 0$$

La diferencia $F_X(a) - F_X(a^-)$ se llama magnitud del salto y por las propiedades desarrolladas anteriormente se tiene que es igual a $P(X = a)$

Ejemplo (***Variable aleatoria discreta***)

Sean

2.3. Variables aleatorias continuas

Definición 2.9 (Variable aleatoria continua)**.** Sean X una variable aleatoria real (Definición 2.7) definida sobre el espacio de probabilidad $(\Omega, \mathfrak{F}, P)$. Se dice que X es absolutamente continua, si y solo si, existe una función real no negativa e integrable f_X tal que, para todo $x \in \mathbb{R}$, se satisface:

$$F_X(x) = \int_{-\infty}^{x} f_X(t)dt. \tag{2.1}$$

La función f_X recibe el nombre de función de densidad (fdd) de la variable aleatoria X.

Ejemplo (***Variable aleatoria continua***)

Sean

3. Estadística

4. Ciencia de Datos

4.1. Data Frame

Los dataframes son un objeto que se muestran en un formato tabla, los cuales pueden tener diferentes tipos de datos en su interior.

Una cuestión habitual es preguntarse en qué casos debes usar un data frame o una matriz en R. Los data frames son estructuras de datos muy similares a las matrices, pero en el caso de los data frames puedes tener diferentes tipos de datos dentro de las columnas. En consecuencia, la diferencia es que las matrices almacenan tipos de datos homogéneos mientras que los data frames almacenan tipos de datos heterogéneos.

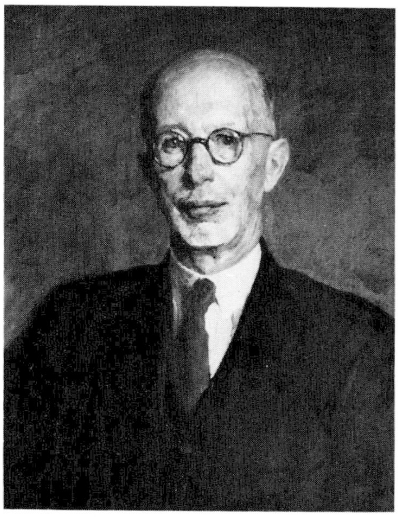
Los Dataframes se representan como un tipo especial de lista en la que cada elemento de la lista debe tener el atributo misma longitud. Cada elemento de la lista se puede considerar como una columna y la longitud de cada elemento de la lista es el número de filas. A diferencia de las matrices, los dataframes pueden almacenar diferentes clases de objetos en cada columna. Las matrices deben Hacer que cada elemento sea de la misma clase (por ejemplo, todos los enteros o todos los números).

(Wickham, Çetinkaya-Rundel, y Grolemund 2023)

5. Modelo Clásico de Cramér-Lundberg

5.1. Introducción

El **modelo clásico de Cramér-Lundberg**, también conocido como **modelo de riesgo clásico o modelo de ruina teórica**, fue descrito por **Filip Lundberg** en **1903** y enriquecido con los aportes de **Harald Cramér** en **1930**. Este modelo describe el **superávit de una aseguradora** que recibe ingresos por primas y, a su vez, paga los montos de las reclamaciones o siniestros que ocurren.



FILIP LUNDBERG
Målning av D Tägtström

En términos más técnicos, el **modelo de Cramér-Lundberg** se utiliza para estudiar la **probabilidad de ruina** de una compañía de seguros. Aquí están algunos conceptos clave:

1. **Proceso de Riesgo Clásico**: El capital de la compañía se modela como un **proceso de riesgo**, donde las primas ingresan y las reclamaciones salen. La probabilidad de ruina se refiere al riesgo de que la compañía no pueda cumplir con sus obligaciones debido a insuficiencia de capital.

2. **Distribuciones de Reclamaciones**: Se supone que las **distribuciones del tamaño de los reclamos** son de **cola ligera**, lo que significa que las reclamaciones extremadamente grandes son poco probables pero pueden tener un impacto significativo en la solvencia de la compañía.

3. **Fórmulas Exactas y Aproximadas**: Se han desarrollado **métodos** para obtener **fórmulas exactas y aproximadas** para la probabilidad de ruina en este modelo. Estas fórmulas permiten estimar la posibilidad de que la compañía quiebre debido a pérdidas inesperadas.

4. **Simulaciones y Estimaciones**: Además de las fórmulas, se realizan **simulaciones** de las trayectorias del proceso de riesgo para diferentes niveles de capital inicial. Esto proporciona una perspectiva más completa de las características operativas de la aseguradora .

En resumen, el **modelo de Cramér-Lundberg** es una herramienta valiosa para evaluar el riesgo financiero en el ámbito de las compañías de seguros y comprender las posibilidades de ruina en función de las fluctuaciones en el capital disponible.

5.2. Descripción del modelo

El modelo de Cramér Lundberg se define enseguida como un cierto tipo de proceso estocástico a tiempo discreto.

Se modela el numero de reclamaciones como un proceso de Poisson $(N(t))$ con parámetro λt donde $t \geq 0$, las pérdidas individuales X_i son variables aleatorias independientes e idénticamente distribuidas (iii)

Definición 5.1.

1. El proceso estocástico a tiempo discreto del monto de reclamaciones se define como $\{(X_k) : k \in \mathbb{N}\}$, donde las X_k para $k = 1, 2, ...,$ son variables aleatorias (Definición 2.7) no negativas, independientes e idénticamente distribuidas y tienen función de distribución común F, media finita denotada por $E[X_k] = \mu$ y varianza $Var(X_k) = \sigma^2 \leq \infty$.

2. Los tiempos de reclamación ocurren en instantes aleatorios de tiempo denotados por T_i tales que
$$0 < T_1 < T_2 < \cdots$$

3. El proceso de ocurrencia de las reclamaciones es el número de reclamaciones contingentes en el intervalo $[0, t]$, y se define como sigue:

$$N(t) = \sup\{n \geq 1 : T_n \leq t \leq 0\}$$

y por convención se usará $\sup\ \emptyset = 0$.

4. Los tiempos entre llegadas son los tiempos que transcurren entre dos reclamaciones sucesivas, los cuales se denotan por

$$Y_i = T_i \quad Y_k = T_k - T_{k-1}, \quad k = 2, 3, \cdots ,$$

   donde las variables aleatorias Y_k son i.i.d. con distribución exponencial y media finita $E[Y_1] = \frac{1}{\lambda}$.

5. La sucesión $\{(X_k) : k \in \mathbb{N}\}$ es independiente de la sucesión $\{(Y_k) : k \in \mathbb{N}\}$.

Definición 5.2. El proceso de agregado de siniestros $(S(t))_{t \geq 0}$ de un portafolio se define como:

$$S(t) := \begin{cases} \sum_{i=1}^{N(t)} X_i, & si \quad N(t) > 0, \\ 0, & si \quad N(t) = 0. \end{cases} \tag{5.1}$$

El proceso $(S(t))_{t\geq 0}$ es un proceso de suma parcial aleatoria, que se obtiene al sustituir el índice determinado n en la suma $S_n = X_1 + \cdots + X_n$ por la variable aleatoria $N(t)$:

$$S(t) = X_1 + \cdots + X_{N(t)} = S_{N(t)}, \quad t \geq 0,$$

que se le conoce como proceso compuesto. Nótese que cuando $Var(X_k) = \sigma^2 < \infty$, el proceso del agregado de siniestros $(S(t))t \geq 0$ comparte varias propiedades con el proceso de una suma parcial, como las propiedades asintóticas del teorema del límite central y la ley fuerte de los grandes números.

Lema 5.1. *El agregado de siniestros $(S(t))$ con $t \geq 0$ tiene función de distribución*

$$G_s(x) = P[S(t) \leq x] = \sum_{n=0}^{\infty} e^{-\lambda t} \frac{(\lambda t)^n}{n!} F**(x) \quad t \leq 0, x \leq 0, \tag{5.2}$$

donde $F^{n*}(x) = P[\sum_{i=q}^{n} X_i \leq x]$ es la n-ésima convolución de F.

Prueba. By induction. □

Lema 5.2. *Para cada $r \geq 0$, el proceso del agregado de siniestros $(S(t))_{t\geq 0}$ tiene función generadora de momentos*

$$M_{S(t)}(r) = E[e^{rS(t))}] = M_{N(t)}(lnM_{x_i}(r)). \tag{5.3}$$

Prueba. By induction. □

El proceso de riesgo $((U(t))_{t\geq 0}$ se define como sigue:

$$U(t) = u + ct - S(t) \quad t \geq 0, \tag{5.4}$$

en donde $u \geq 0$ denota el capital inicial, ct el ingreso vía prima durante el periodo $[0, t]$ a una tasa constante $c > 0$ y $S(t)$ es el agregado de siniestros en el momento t.

Proposición 5.1. *Para cada $r \geq 0$, el proceso de riesgo $(U(t))_{t\geq 0}$ tiene función generadora de momentos*

$$M_{U(t)}(r) = e^{r(u+ct)} M_{S(t)}(-r). \tag{5.5}$$

Sea $(U(t))_{t\geq 0}$ un proceso de riesgo, entonces para el modelo de renovación se tiene que:

$$E[U(t)] = u + ct - \mu E[N(t)],$$

y para el modelo clásico de Crámer-Lundberg es:

$$E[U(t)] = u + ct - \lambda \mu t.$$

Prueba. By induction. □

Lema 5.3. *Sea* $(U(t))_{t\geq 0}$ *un proceso de riesgo, entonces para el modelo de renovación se tiene que :*

$$Var(U(t)) = \mu^2 Var(N(t)) + \sigma^2 E[N(t)],$$

y para el modelo clásico Cramér-Lundberg es:

$$Var(U(t)) = \lambda t(\sigma^2 + \mu^2).$$

Prueba. By induction. □

6. Estudio de caso: Análisis de ventas de una empresa de comida rápida

6.1. Descripción del problema

Se presentan datos que son parte de las ventas de una empresa de comida rápida. Para brindar una mejor idea del estudio caso se presentará una extensión de lo que es una empresa de comida rápida y una franquicia .

Una empresa de comida rápida es un tipo de negocio dentro del sector alimenticio que se especializa en la preparación y venta de alimentos y bebidas que pueden ser servidos y consumidos rápidamente. Estas empresas suelen ofrecer menús estandarizados y utilizan métodos de producción en masa para asegurar la rapidez y la eficiencia en el servicio. Algunas características típicas de una empresa de comida rápida incluyen:

1-Rapidez en el Servicio: La principal característica es la rapidez con la que los clientes pueden recibir sus pedidos, generalmente en pocos minutos.

2-Menú Limitado y Estandarizado: Los menús suelen ser simples y consistentes en todas las ubicaciones de la cadena, permitiendo una preparación rápida y eficiente.

3-Precios Asequibles: Los precios suelen ser más bajos en comparación con los restaurantes tradicionales, lo que los hace accesibles a una amplia gama de clientes.

4-Métodos de Preparación Eficientes: Utilizan técnicas de cocción rápida, como freidoras y parrillas de alta eficiencia, así como la pre-preparación de ingredientes.

5-Autoservicio y Comida para Llevar: Muchas de estas empresas ofrecen opciones de autoservicio y comida para llevar, facilitando el consumo rápido y conveniente.

6-Ambiente Informal: El ambiente es generalmente informal y está diseñado para facilitar el flujo rápido de clientes, con áreas de autoservicio y estaciones de recolección de pedidos.

Ejemplos de empresas de comida rápida incluyen grandes cadenas internacionales como McDonald's, Burger King, KFC, Subway, Carl's Jr y Taco Bell. Estas empresas han expandido su presencia globalmente, adaptándose a diferentes mercados y culturas mientras mantienen su enfoque en la rapidez y eficiencia del servicio.

Muchas de estas empresas otorga parte del derecho a operar en diferentes franquicias.

Una franquicia es un modelo de negocio en el cual una empresa (el franquiciador) otorga a otra parte (el franquiciado) el derecho a operar un negocio utilizando su marca, productos, servicios y modelo operativo a cambio de una tarifa o regalías. Este modelo permite a los franquiciadores expandir su marca y presencia en el mercado sin tener que invertir en nuevas ubicaciones directamente. Al mismo tiempo, ofrece a los franquiciados la oportunidad de operar un negocio con una marca establecida y un sistema probado. Algunas características clave de una franquicia incluyen:

1-Marca y Sistema Operativo: El franquiciado utiliza la marca comercial, logotipos, productos y servicios del franquiciador, así como su sistema operativo, que puede incluir recetas, métodos de producción, estrategias de marketing, entre otros.

2-Pago de Tarifas: El franquiciado paga al franquiciador una tarifa inicial y, en muchos casos, regalías continuas basadas en un porcentaje de las ventas. Estas tarifas cubren el uso de la marca y el soporte continuo del franquiciador.

3-Capacitación y Soporte: El franquiciador proporciona capacitación inicial y soporte continuo al franquiciado, lo cual puede incluir formación en gestión empresarial, operaciones, marketing y servicio al cliente.

4-Estándares y Control de Calidad: Para mantener la consistencia y calidad de la marca, el franquiciador establece estándares y procedimientos que los franquiciados deben seguir. Esto puede incluir auditorias y evaluaciones periódicas.

5-Contrato de Franquicia: La relación entre el franquiciador y el franquiciado está regulada por un contrato de franquicia que detalla los derechos y obligaciones de ambas partes, incluyendo la duración del acuerdo y las condiciones de renovación.

6-Territorio Exclusivo: A menudo, el contrato otorga al franquiciado un territorio exclusivo en el cual puede operar, evitando la competencia directa con otras franquicias de la misma marca en esa área.

Ejemplos de negocios que utilizan el modelo de franquicia incluyen cadenas de comida rápida (como McDonald's, Subway y KFC), Carl's Jr, hoteles (como Marriott y Hilton), servicios de limpieza, tiendas de conveniencia y muchos otros sectores. Este modelo permite una expansión rápida y eficiente del negocio mientras ofrece a los emprendedores la oportunidad de operar con el respaldo de una marca establecida.

Esta empresa ofrece en sus productos dos tipos de hamburguesas, bebidas como sodas, cervezas y cafés. En la información de las ventas incluye registros obtenidos por una Terminal Punto de Venta (TPV), el sistema de pago usado en establecimientos comerciales para procesar transacciones de venta. Permite a los comerciantes aceptar el uso de tarjetas de crédito y débito. Se tiene registro de las ventas por cada día desde el año 2018 al noviembre del 2020, haciendo un cierre pandémico, se registran en dos etapas, antes de pandemia los datos a partir del 25 de julio del 2018 al 10 de abril del 2020. Los datos después de pandemia se registraron a partir del 3 de agosto del 2020 y cerrando finalmente el 10 de noviembre del 2020, los cuales se les aplicó una homogenización para un buen manejo de los datos y así poder presentar un mejor análisis.

6.2. Elementos de Ciencia de Datos

6.2.1. Importación de los datos originales a R

Se cargan los datos del archivo excel para ser leído y almacenarlo en un dataframe (Sección 4.1) para poder manipular cada uno de los elementos de la base de datos.

```
Datos <- read_excel("~/TM/Datos.xlsx", col_types =
                  c("numeric","numeric","numeric","numeric","numeric",
                    "numeric","numeric","numeric","numeric","numeric",
                    "numeric","numeric","numeric","numeric","numeric",
                    "numeric","numeric","numeric","numeric","numeric",
                    "numeric","numeric","numeric","numeric","numeric",
                    "numeric","numeric","numeric","numeric","numeric",
                    "numeric","numeric","numeric","numeric","numeric",
                    "numeric","numeric","numeric","numeric","numeric",
                    "numeric","numeric","numeric","numeric","numeric",
                    "numeric","numeric","numeric","numeric","numeric",
                    "numeric","numeric","numeric","numeric","numeric",
                    "numeric","numeric","numeric","numeric","numeric",
                    "numeric","numeric","numeric","numeric","numeric",
                    "numeric","numeric","numeric","numeric","numeric",
                    "numeric","numeric","numeric","numeric","numeric"
                        ))

Datos
```

```
# A tibble: 2,416 x 75
    ...1  ...2  ...3  ...4  ...5  ...6 `240519` `82253.5` `247172.9`  ...10 `973`
   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>    <dbl>     <dbl>       <dbl> <dbl> <dbl>
 1    NA    NA    NA    NA    NA    NA       NA        NA          NA    NA    NA
 2    NA    NA    NA    NA    NA    NA    9382.     9441.          NA    NA 9382.
 3  2018     7    30     4 43306    NA  188038.        NA     188038.    NA   682
 4  2018     7    30     5 43307    NA  220827         NA     220827     NA   694
 5  2018     7    30     6 43308    NA  240519         NA     240519     NA   762
 6  2018     7    30     7 43309    NA  229542         NA     229542     NA   704
 7  2018     7    30     1 43310    NA  239776.        NA     239776.    NA   706
 8  2018     7    31     2 43311    NA  172946         NA     172946     NA   602
 9  2018     7    31     3 43312    NA  210671         NA     210671     NA   709
10  2018     8    31     4 43313    NA  195662         NA     195662     NA   649
# i 2,406 more rows
# i 64 more variables: `384` <dbl>, `1136` <dbl>, ...14 <dbl>,
#   `371.753114754098` <dbl>, `2428.16666666666` <dbl>, ...17 <dbl>,
```

```
#   ...18 <dbl>, `2099` <dbl>, `222` <dbl>, `2162` <dbl>, ...22 <dbl>,
#   `1384...23` <dbl>, `559` <dbl>, `1384...25` <dbl>, ...26 <dbl>,
#   `2402` <dbl>, `658` <dbl>, `2525` <dbl>, ...30 <dbl>, `1556...31` <dbl>,
#   `462` <dbl>, `1556...33` <dbl>, ...34 <dbl>, `61` <dbl>, `57` <dbl>, ...
```

Como se puede apreciar, es necesario eliminar columnas, reducido a las columnas de las ventas de los diferentes productos que oferta la empresa para así obtener las ventas totales.

6.2.2. Homogenización del total de ventas por día

Mediante este proceso se eliminan y se corrigen valores atípicos, valores faltantes o alguna inconsistencia que se presentan en los datos. Esto implica la eliminación de datos incompletos, conocidos como datos faltantes o NA. También se renombran cabeceras de cada columna del dataframe para facilitar la lectura del archivo.

```
#Eliminamos las columnas con datos faltantes
datos <- Datos[, -c(6,8,10,12,14,16,18,20,22,24, 26,28, 30,32, 34, 36,
                    38,39, 40,  42, 44, 46, 48, 50,51,52, 53, 54,55, 56,
57,58,59,60,61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 73,74,72, 75)]
#Cambiar las cabeceras del dataframe
datos <- data.frame(datos%>% rename(Año = ...1 ))
datos <- data.frame(datos%>% rename(Mes = ...2, Semana = ...3, Dia= ...4,
                  Fecha = ...5, Ventas= X240519,
                  TotalVentas = X247172.9,Customers = X973,
                  Totalcustomers = X1136,
                  CH_Promedio = X371.753114754098,
                  TotalCH_Promedio = ...17, Burgers = X2099,
                  TotalBurgers = X2162,
                  Thickurgers = X1384...23,
                  TotalThickburgers = X1384...25,
                  Total_Burgers = X2402,
                  T_Total_Burgers= X2525,Bebidas = X1556...31,
                  Totalbebidas = X1556...33, Cervezas = X61,
                  Totalcervezas = X83, Cafes = X83,
                  Totalcafes = X54, Ventas_TPV = X158904,
                  Total_Ventas_TPV = X173412.5,
                  HR_Laborales = ...47,
                  Total_HR_Laborales = ...49 ))
```

Por último, se realizó la separación de la base de datos antes de la pandemia y después de la pandemia, presentando así los dataframes resultantes en el siguiente código y como se ilustra posteriormente:

```r
#Analisis de los datos:
#Filtros por días:
#___________________________Antes de la pandemia___________________________

fil_before1<- data.frame(subset(datos, Año <= 2019))
fil_before2<- data.frame(subset(datos, Año == 2020 & Mes <= 4))
fil_before <- rbind(fil_before1, fil_before2)

#________________________Después de la pandemia____________________________

fil_dat_after <- data.frame(subset(datos, Año == 2020 & Mes>=8))
```

6.2.3. DataFrame Resultante: Antes de la pandemia

```
    Año Mes Dia   Ventas TotalVentas
3  2018   7   4 188037.5    188037.5
4  2018   7   5 220827.0    220827.0
5  2018   7   6 240519.0    240519.0
6  2018   7   7 229542.0    229542.0
7  2018   7   1 239775.6    239775.6
8  2018   7   2 172946.0    172946.0
9  2018   7   3 210671.0    210671.0
10 2018   8   4 195662.0    195662.0
11 2018   8   5 193051.0    193051.0
12 2018   8   6 206803.0    206803.0
13 2018   8   7 219399.0    219399.0
14 2018   8   1 224106.5    224106.5
15 2018   8   2 152557.0    152557.0
16 2018   8   3 148675.5    148675.5
17 2018   8   4 142344.0    142344.0
18 2018   8   5 152189.5    152189.5
19 2018   8   6 171891.0    171891.0
20 2018   8   7 194642.5    194642.5
21 2018   8   1 210023.5    210023.5
22 2018   8   2 129751.5    129751.5
23 2018   8   3 125867.5    125867.5
24 2018   8   4 133380.0    133380.0
25 2018   8   5 135070.5    135070.5
26 2018   8   6 158927.0    158927.0
27 2018   8   7 175382.0    175382.0
28 2018   8   1 187502.0    187502.0
29 2018   8   2  78116.0     78116.0
```

30	2018	8	3	71871.0	71871.0
31	2018	8	4	82471.5	82471.5
32	2018	8	5	98739.5	98739.5
33	2018	8	6	131185.5	131185.5
34	2018	8	7	157909.5	157909.5
35	2018	8	1	186582.0	186582.0
36	2018	8	2	68030.0	68030.0
37	2018	8	3	71887.0	71887.0
38	2018	8	4	74843.5	74843.5
39	2018	8	5	77880.5	77880.5
40	2018	8	6	128575.0	128575.0

6.2.4. DataFrame Resultante: Después de la pandemia

	Año	Mes	Dia	Ventas	TotalVentas
743	2020	8	2	44908.70	53887.70
744	2020	8	3	51733.00	69538.00
745	2020	8	4	57876.95	57876.95
746	2020	8	5	47554.60	57540.10
747	2020	8	6	67138.20	70025.20
748	2020	8	7	89172.70	99932.70
749	2020	8	1	113384.70	130996.70
750	2020	8	2	40974.60	47464.60
751	2020	8	3	48520.10	57037.10
752	2020	8	4	46649.15	52529.65
753	2020	8	5	40556.90	51963.40
754	2020	8	6	70159.10	81004.10
755	2020	8	7	77736.80	89482.30
756	2020	8	1	107918.30	120988.30
757	2020	8	2	48885.80	59032.80
758	2020	8	3	48311.20	57724.70
759	2020	8	4	44658.00	46319.00
760	2020	8	5	63104.30	69302.80
761	2020	8	6	67481.30	76436.30
762	2020	8	7	89795.30	102998.80
763	2020	8	1	102744.90	118060.90
764	2020	8	2	34713.50	42455.00
765	2020	8	3	40403.40	53956.40
766	2020	8	4	47028.10	56991.10
767	2020	8	5	31703.55	45095.55
768	2020	8	6	59907.40	75201.90
769	2020	8	7	89533.40	98554.40
770	2020	8	1	103271.20	115973.20

```
771 2020    8    2  47309.40     59826.40
```

Para realizar la identificación de las diferentes distribuciones e implementar los gráficos como los histogramas y los boxplot, se elaboran filtros correspondientes para las ventas por cada día antes y después de pandemia

Los comandos para realizar los filtros es el siguiente:

```
#filtra los datos del día Domingo antes de la pandemia
fil_dat_1_before<- data.frame(subset(fil_before, Dia == 1))
#___________________________________________________________
#filtra los datos del día Lunes antes de la pandemia
fil_dat_2_before<- data.frame(subset(fil_before, Dia == 2))
#___________________________________________________________
#filtra los datos del día martes antes de la pandemia
fil_dat_3_before<- data.frame(subset(fil_before, Dia == 3))
#___________________________________________________________
#filtra los datos del día miércoles antes de la pandemia
fil_dat_4_before<- data.frame(subset(fil_before, Dia == 4))
#___________________________________________________________
#filtra los datos del día Jueves antes de la pandemia
fil_dat_5_before<- data.frame(subset(fil_before, Dia == 5))
#___________________________________________________________
#filtra los datos del día Viernes antes de la pandemia
fil_dat_6_before<- data.frame(subset(fil_before, Dia == 6))
#___________________________________________________________
#filtra los datos del día Sábado antes de la pandemia
fil_dat_7_before<- data.frame(subset(fil_before, Dia == 7))
```

```
#____________________________________________________________________
fil_dat_after <- data.frame(subset(datos, Año == 2020 & Mes>=8))
#filtra los datos del día domingo despues de la pandemia
fil_dat_1_after<- data.frame(subset(datos,
                                    Año == 2020 & Mes >= 8 & Dia==1))
#____________________________________________________________________
#filtra los datos del día lunes despues de la pandemia
fil_dat_2_after<- data.frame(subset(datos,
                                    Año == 2020 & Mes >= 8 & Dia ==2))
#____________________________________________________________________
#filtra los datos del día martes despues de la pandemia
fil_dat_3_after<- data.frame(subset(datos,
                                    Año == 2020 & Mes >= 8 & Dia ==3))
#____________________________________________________________________
#filtra los datos del día miércoles despues de la pandemia
```

```
fil_dat_4_after<- data.frame(subset(datos,
                                    Año == 2020 & Mes >= 8 & Dia ==4))
#_____________________________________________________________________
#filtra los datos del día jueves despues de la pandemia
fil_dat_5_after<- data.frame(subset(datos,
                                    Año == 2020 & Mes >= 8 & Dia ==5))
#_____________________________________________________________________
#filtra los datos del día viernes despues de la pandemia
fil_dat_6_after<- data.frame(subset(datos,
                                    Año == 2020 & Mes >= 8 & Dia ==6))
#_____________________________________________________________________
#filtra los datos del día sábado despues de la pandemia
fil_dat_7_after<- data.frame(subset(datos,
                                    Año == 2020 & Mes >= 8 & Dia ==7))
```

Para visualizar gráficamente el comportamiento de las ventas por día, a continuación se construyen los histogramas por día, en los periodos antes y después de la pandemia.

6.2.5. Análisis gráfico de las ventas por día:

Para la generación de los gráficos se empleó los filtros anteriores, con el fin de obtener un análisis descriptivo de cada uno de ellos.

6.2.5.1. Histogramas de los datos de ventas por día antes de la pandemia

El código que genera estos histogramas en R es el siguiente:

```
# muestra el gráfico de todos los histogramas de ventas por día
# antes de la pandemia
par(mar=c(2,2,2,2))
par(mfrow =c(3,3))
hist(fil_dat_1_before$Ventas,
     main = "Ventas día Domingo", xlab = "Ventas")
hist(fil_dat_2_before$Ventas,
     main = "Ventas día Lunes", xlab = "Ventas")
hist(fil_dat_3_before$Ventas,
     main = "Ventas día Martes", xlab = "Ventas")
hist(fil_dat_4_before$Ventas,
     main = "Ventas día Miércoles", xlab = "Ventas")
hist(fil_dat_5_before$Ventas,
     main = "Ventas día Jueves", xlab = "Ventas")
```

```
hist(fil_dat_6_before$Ventas,
     main = "Ventas día Viernes", xlab = "Ventas")
hist(fil_dat_7_before$Ventas,
     main = "Ventas día Sábado", xlab = "Ventas")
```

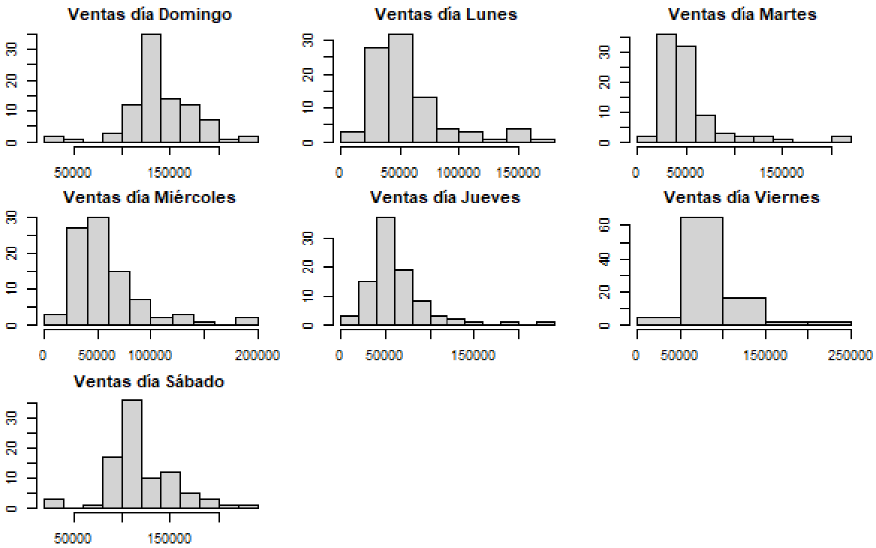


Figura 6.1.: Histogramas de las ventas por día antes de la pandemia.

Para profundizar en el análisis descriptivo de las ventas periódicas se presenta ahora los boxplot de las ventas por cada día de la semana.

6.2.5.2. Boxplot de datos de ventas por días antes de la pandemia

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  30977  122963  135498  138498  158347  239776 
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   7430   35919   48676   57388   68030  172946 
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  12950   35210   44398   53367   56624  219702 
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  10832   35847   44213   56365   63585  195662 
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  13547   42368   49479   61068   70047  220827 

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  10331   65730   79464   86514   96522  240519 

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  24219  100751  110603  119332  135176  229542 
```

El código que genera en R los boxplot es el siguiente:

```r
# muestra el gráfico de todos los boxplot de ventas por día
# antes de la pandemia
fig <- plot_ly(y =fil_dat_1_before$Ventas, name = "Ventas día Domingo",
               boxpoints = "all",type = "box", )
fig <- fig %>% add_trace(y = fil_dat_2_before$Ventas,
                         name = "Ventas día Lunes", boxpoints = "all")
fig <- fig %>% add_trace(y = fil_dat_3_before$Ventas,
                         name = "Ventas día Martes", boxpoints = "all")
fig <- fig %>% add_trace(y = fil_dat_4_before$Ventas,
                         name = "Ventas día Miércoles", boxpoints = "all")
fig <- fig %>% add_trace(y = fil_dat_5_before$Ventas,
                         name = "Ventas día Jueves", boxpoints = "all")
fig <- fig %>% add_trace(y = fil_dat_6_before$Ventas,
                         name = "Ventas día Viernes", boxpoints = "all")
fig <- fig %>% add_trace(y = fil_dat_7_before$Ventas,
                         name = "Ventas día Sábado", boxpoints = "all")

fig
```

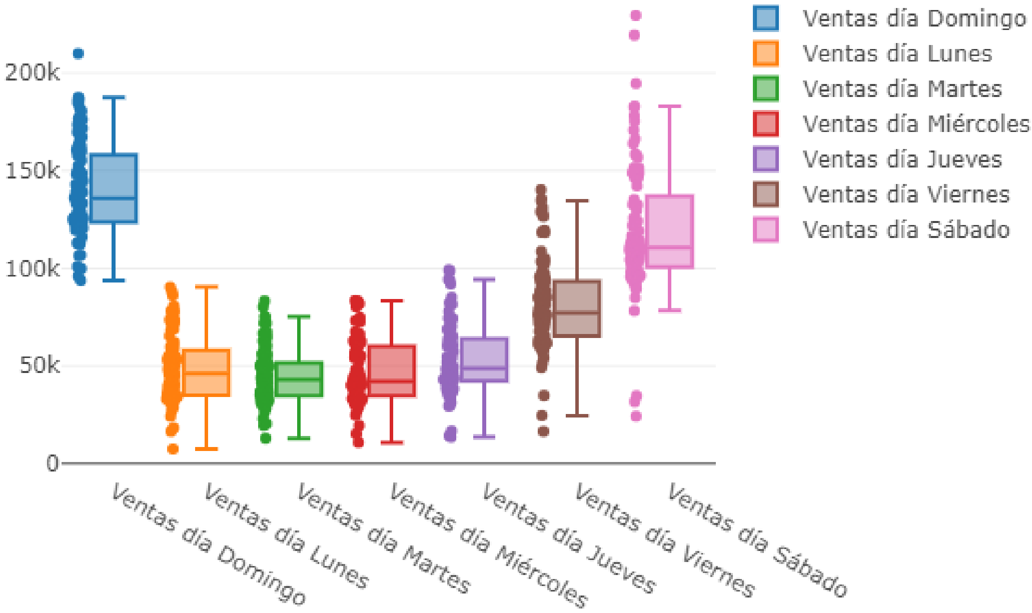


Figura 6.2.: Boxplot de las ventas por día antes de la pandemia.

6.2.5.3. Histogramas de los datos de ventas por día después de la pandemia

El código que genera estos histogramas en R es el siguiente:

```
# muestra el gráfico de todos los histogramas de ventas por día
#despues de la pandemia
par(mar=c(2,2,2,2))
par(mfrow =c(3,3))
hist(fil_dat_1_after$Ventas,
     main = "Ventas día Domingo", xlab = "Ventas")
hist(fil_dat_2_after$Ventas,
     main = "Ventas día Lunes", xlab = "Ventas")
hist(fil_dat_3_after$Ventas,
     main = "Ventas día Martes", xlab = "Ventas")
hist(fil_dat_4_after$Ventas,
     main = "Ventas día Miércoles", xlab = "Ventas")
hist(fil_dat_5_after$Ventas,
     main = "Ventas día Jueves", xlab = "Ventas")
hist(fil_dat_6_after$Ventas,
     main = "Ventas día Viernes", xlab = "Ventas")
```

```
hist(fil_dat_7_after$Ventas,
     main = "Ventas día Sábado", xlab = "Ventas")
```

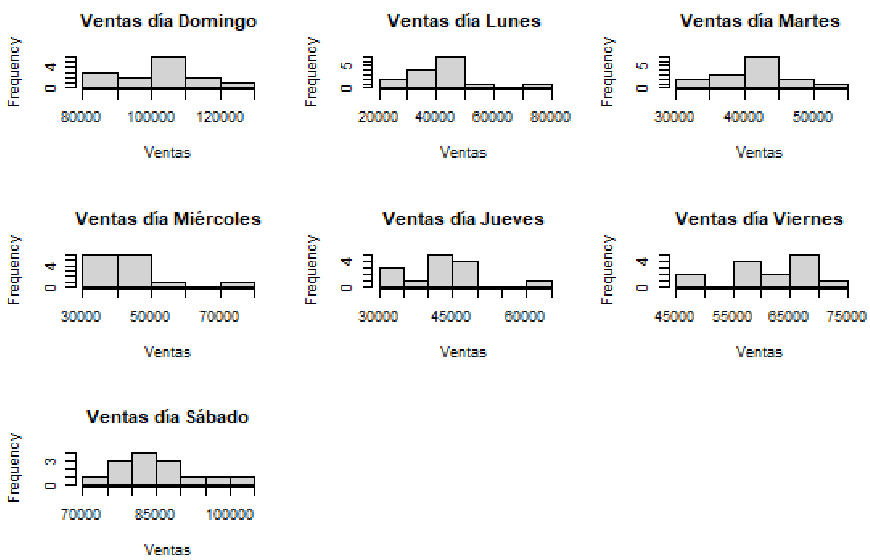


Figura 6.3.: Histogramas de las ventas por día después de la pandemia.

El comportamiento periódico de las ventas, se observa en los boxplot de las ventas por cada día de la semana.

6.2.5.4. Boxplot de datos de ventas por días limpios después de la pandemia

```
summary(fil_dat_1_after$Ventas)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  80431   95346  103691  101919  107639  122397
```

```
summary(fil_dat_2_after$Ventas)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  27106   33867   40804   42339   47566   77299
```

```r
summary(fil_dat_3_after$Ventas)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  30514   38860   41889   41752   44732   51733 
```

```r
summary(fil_dat_4_after$Ventas)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  30335   34697   43071   43243   46480   70983 
```

```r
summary(fil_dat_5_after$Ventas)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  31704   39901   43144   43010   47048   63104 
```

```r
summary(fil_dat_6_after$Ventas)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  45908   56956   63361   61191   66915   70159 
```

```r
summary(fil_dat_7_after$Ventas)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  73606   78646   84130   85684   89730  101467 
```

El código que genera estos los boxplot en R es el siguiente:

```r
# muestra el gráfico de todos los boxplot de ventas por día antes de la pandemia
fig <- plot_ly(y =fil_dat_1_after$Ventas, name = "Ventas día Domingo",
               boxpoints = "all",type = "box")

fig <- fig %>% add_trace(y = fil_dat_2_after$Ventas,
                         name = "Ventas día Lunes", boxpoints = "all")
fig <- fig %>% add_trace(y = fil_dat_3_after$Ventas,
                         name = "Ventas día Martes", boxpoints = "all")
fig <- fig %>% add_trace(y = fil_dat_4_after$Ventas,
                         name = "Ventas día Miércoles", boxpoints = "all")
fig <- fig %>% add_trace(y = fil_dat_5_after$Ventas,
```

```
                         name = "Ventas día Jueves", boxpoints = "all")
fig <- fig %>% add_trace(y = fil_dat_6_after$Ventas,
                         name = "Ventas día Viernes", boxpoints="all")
fig <- fig %>% add_trace(y = fil_dat_7_after$Ventas,
                         name = "Ventas día Sábado", boxpoints = "all")
fig
```

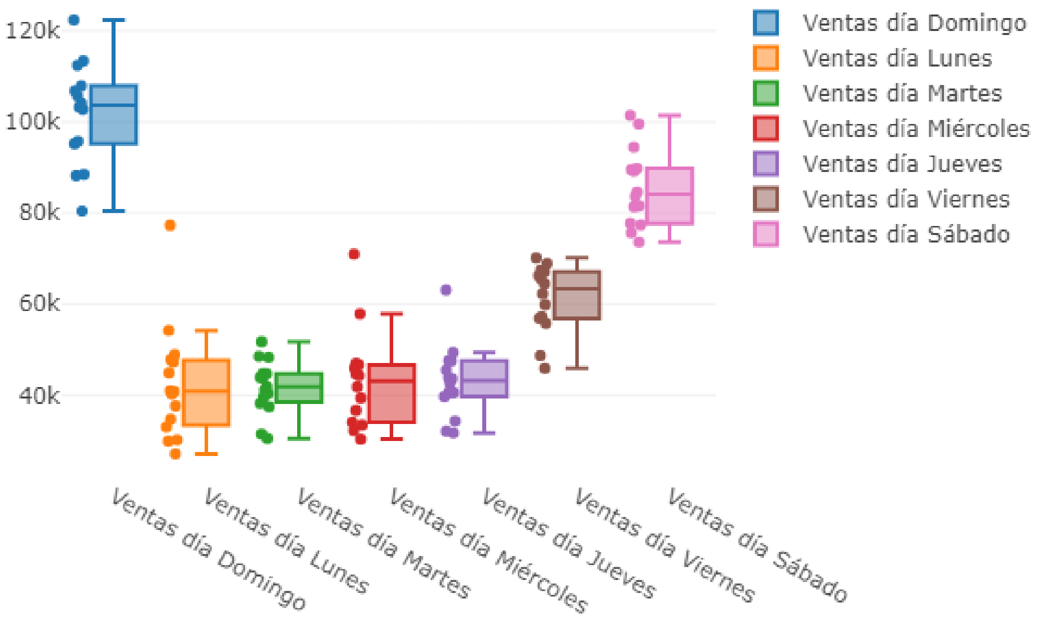


Figura 6.4.: Boxplot de las ventas por día después de la pandemia.

6.3. Identificación de distribuciones de probabilidad de las ventas por día

Con el fin de adaptar el modelo de Cramér Lundberg es necesario ajustar distribuciones de probabilidad de las ventas para cada día de la semana. Haciendo uso del test de Jarque Bera que ayuda a identificar cuales días siguen una distribución normal y el test de Kolmogorov de Smirnov.

El siguiente código aplica los diferentes test de normalidad para las ventas por día antes de la pandemia:(Citar Jarque Bera)

```r
#Calcularemos los test correspondientes a los filtros
#de ventas por día:

#JARQUE BERA
jarque.bera.test(fil_dat_1_before$Ventas)
```

```
    Jarque Bera Test

data:  fil_dat_1_before$Ventas
X-squared = 4.3177, df = 2, p-value = 0.1155
```

```r
#El valor p no es menor que   = .05,
#entonces hay evidencia suficiente para decir
#por lo tanto los datos se distribuyen normalmente
#Se ajustara una distribuciión con el comando siguiente:

fitdistr(fil_dat_1_before$Ventas, "normal")
```

```
      mean           sd
  139929.468    24521.524
 (  2675.518) (  1891.877)
```

```r
#JARQUE BERA
jarque.bera.test(fil_dat_2_before$Ventas)
```

```
    Jarque Bera Test

data:  fil_dat_2_before$Ventas
X-squared = 2.7606, df = 2, p-value = 0.2515
```

```r
#El valor p no es menor que   = .05,
#entonces hay evidencia suficiente para decir
#por lo tanto los datos se distribuyen normalmente
#Se ajustara una distribuciión con el comando siguiente:
#ajustamos una distribución (normal)
fitdistr(fil_dat_2_before$Ventas, "normal")
```

```
     mean          sd
  48125.734   17150.338
 ( 1917.466) ( 1355.853)
```

```r
#JARQUE BERA
jarque.bera.test(fil_dat_3_before$Ventas)
```

```
	Jarque Bera Test

data:  fil_dat_3_before$Ventas
X-squared = 4.6048, df = 2, p-value = 0.1
```

```r
#El valor p no es menor que   = .05,
#entonces hay evidencia suficiente para decir
#por lo tanto los datos se distribuyen normalmente
#Se ajustara una distribuciión con el comando siguiente:
#ajustamos una distribución (normal)
fitdistr(fil_dat_3_before$Ventas, "normal")
```

```
     mean          sd    
  44509.755   14312.338 
 ( 1590.260) ( 1124.483)
```

```r
#JARQUE BERA
jarque.bera.test(fil_dat_4_before$Ventas)
```

```
	Jarque Bera Test

data:  fil_dat_4_before$Ventas
X-squared = 3.3204, df = 2, p-value = 0.1901
```

```r
#El valor p no es menor que   = .05,
#entonces hay evidencia suficiente para decir
#por lo tanto los datos se distribuyen normalmente
#Se ajustara una distribuciión con el comando siguiente:
#ajustamos una distribución (normal)
fitdistr(fil_dat_4_before$Ventas, "normal")
```

```
     mean          sd    
  46904.516   16238.151 
 ( 1815.480) ( 1283.739)
```

```r
#JARQUE BERA
jarque.bera.test(fil_dat_5_before$Ventas)
```

```
    Jarque Bera Test

data:  fil_dat_5_before$Ventas
X-squared = 4.553, df = 2, p-value = 0.1026
```

```r
#El valor p no es menor que   = .05,
#entonces hay evidencia suficiente para decir
#por lo tanto los datos se distribuyen normalmente
#Se ajustara una distribuciión con el comando siguiente:
#ajustamos una distribución (normal)
fitdistr(fil_dat_5_before$Ventas, "normal")
```

```
     mean          sd
  52786.734   18403.175
 ( 2032.291) ( 1437.047)
```

```r
#JARQUE BERA
jarque.bera.test(fil_dat_6_before$Ventas)
```

```
    Jarque Bera Test

data:  fil_dat_6_before$Ventas
X-squared = 2.5406, df = 2, p-value = 0.2808
```

```r
#El valor p no es menor que   = .05,
#entonces hay evidencia suficiente para decir
#por lo tanto los datos se distribuyen normalmente
#Se ajustara una distribuciión con el comando siguiente:
#ajustamos una distribución (normal)
fitdistr(fil_dat_6_before$Ventas, "normal")
```

```
     mean          sd
  81601.876   23756.037
 ( 2591.996) ( 1832.818)
```

Para el caso del día sábado, denotado como séptimo día, no se obtuvo una distribución normal por lo que se utilizo la paqueteria de R fitdistrplus para probar el ajuste de otras distribuciones clásicas de probabilidad. Las distribuciones probadas se presentan en el código siguiente:

```
fitdistr(fil_dat_7_before$Ventas, "weibull")
```

```
      shape          scale
  3.646021e+00   1.335448e+05
 (2.784858e-01) (4.365588e+03)
```

```
fitdistr(fil_dat_7_before$Ventas, "exponential")
```

```
      rate
  8.379989e-06
 (8.882771e-07)
```

```
fitdistr(fil_dat_7_before$Ventas, "lognormal")
```

```
     meanlog        sdlog
  11.64220349    0.33374967
 ( 0.03537739) ( 0.02501560)
```

De lo anterior podemos decir que el mejor ajuste se obtuvo al usar la distribución Weibull.

6.3.1. Prueba de la distribución Weibull:

Una vez ajustada la distribución Weibull, se prueban sus parametros con la aplicación del test de Kolmogorov de Smirnov. El siguiente código implementa ambos procedimientos:

```
#Calcularemos los test correspondientes a los filtros de de ventas por día:
#JARQUE BERA
jarque.bera.test(fil_dat_7_before$Ventas)
```

```

    Jarque Bera Test

data:  fil_dat_7_before$Ventas
X-squared = 12.707, df = 2, p-value = 0.001741
```

```r
#Fitting de de una distri bución weibull
#fil_dat_7_before$Ventas
fit.weibull <- fitdist(fil_dat_7_before$Ventas, "weibull")
fit.weibull
```

```
Fitting of the distribution ' weibull ' by maximum likelihood
Parameters:
          estimate   Std. Error
shape 4.784508e+00    0.3789532
scale 1.295070e+05 3155.3275585
```

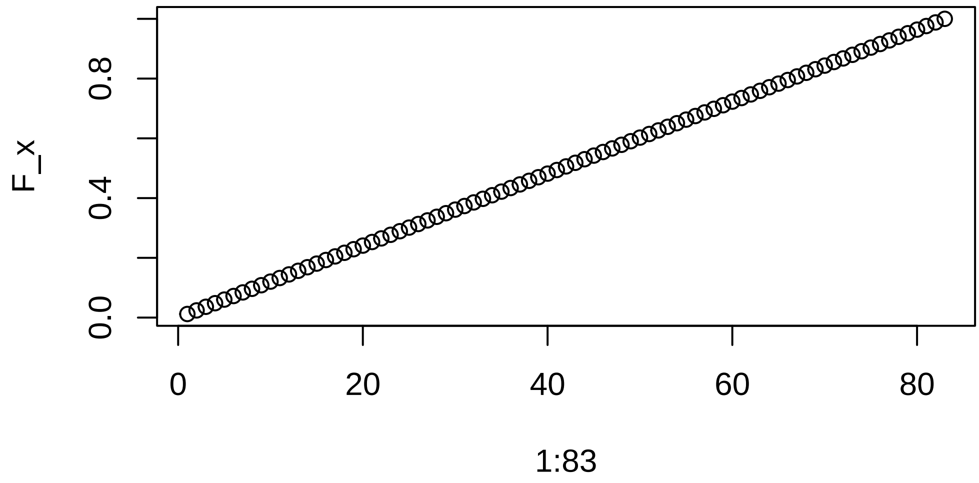
```r
vector_ventas7_ordenados <- fil_dat_7_before$Ventas[order(fil_dat_7_before$Ventas)]
n = 83
F_x = numeric(n)
for(i in 1:n){
  F_x[i] <- (i/ n)
}
#F_x
f_gorro_x <-  pweibull(vector_ventas7_ordenados,
                       shape = 4.784508e+00, scale =1.295070e+05,
                       lower.tail = TRUE, log.p = FALSE)
ks <- abs(F_x - f_gorro_x)
```

6.3.1.1. Representación gráfica al aplicar el test de Kolmogorov de Smirnov

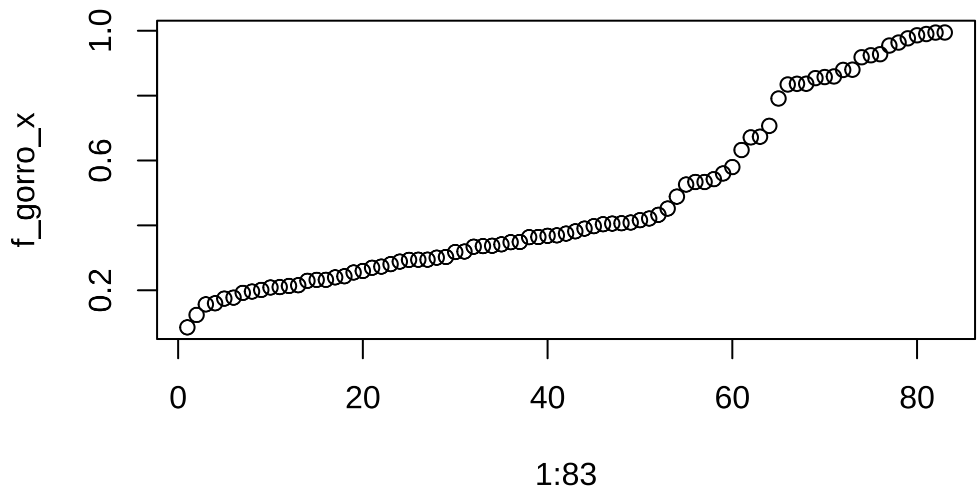
```
[1] 0.1936014
```

```
[1] 0.1492794
```

```r
x <- 1:83
g1 <- plot(1:83, F_x)
```

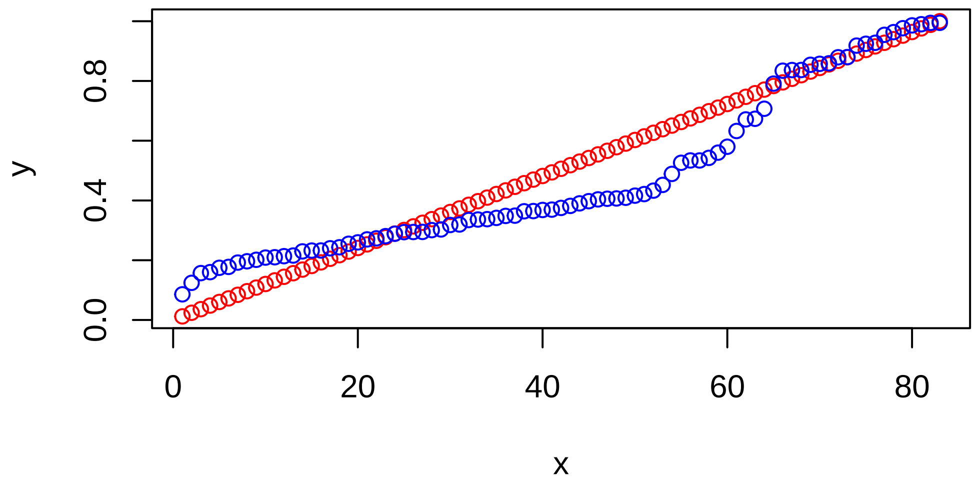



```
g2 <- plot(1:83, f_gorro_x)
```



```r
plot(x, F_x, col='red', xlab='x', ylab='y')

#add second line to plot
points(x, f_gorro_x, col='blue')
```



Para el periodo despues de pandemia se repite la misma metodología. Aplicamos el test de Jarque Bera a los datos para cada día de la semana después de pandemia.

```
    Jarque Bera Test

data:  fil_dat_1_after$Ventas
X-squared = 0.24366, df = 2, p-value = 0.8853

      mean          sd
  101919.050    10877.075
 (  2907.021) (  2055.574)


    Jarque Bera Test

data:  fil_dat_2_after$Ventas
X-squared = 0.65149, df = 2, p-value = 0.722
```

```
     mean         sd
  39841.721    7873.446
 ( 2104.267) ( 1487.941)


    Jarque Bera Test

data:  fil_dat_3_after$Ventas
X-squared = 0.39157, df = 2, p-value = 0.8222

     mean         sd
  41751.747    5687.030
 ( 1468.385) ( 1038.305)


    Jarque Bera Test

data:  fil_dat_4_after$Ventas
X-squared = 3.8727, df = 2, p-value = 0.1442

     mean         sd
  43243.143   10517.841
 ( 2811.011) ( 1987.685)


    Jarque Bera Test

data:  fil_dat_5_after$Ventas
X-squared = 2.0444, df = 2, p-value = 0.3598

     mean         sd
  43010.307    7741.889
 ( 2069.107) ( 1463.080)


    Jarque Bera Test

data:  fil_dat_6_after$Ventas
X-squared = 1.4585, df = 2, p-value = 0.4823
```

```
     mean          sd    
  61191.300    7202.989 
 ( 1925.080) ( 1361.237)
```

```

    Jarque Bera Test

data:  fil_dat_7_after$Ventas
X-squared = 0.85511, df = 2, p-value = 0.6521
```

```
     mean          sd    
  85684.058    8371.359 
 ( 2237.340) ( 1582.038)
```

```
#_______Después de la pandemia____________________________________
#El valor p no es menor que   = .05,
#entonces hay evidencia suficiente para decir
#que la muestra  proviene de una población con distribución normal.
#JARQUE BERA
jarque.bera.test(fil_dat_1_after$Ventas)
```

```

    Jarque Bera Test

data:  fil_dat_1_after$Ventas
X-squared = 0.24366, df = 2, p-value = 0.8853
```

```
fitdistr(fil_dat_1_after$Ventas, "normal")
```

```
      mean           sd     
  101919.050    10877.075 
 (   2907.021) (   2055.574)
```

```
#________________________________________________________________
#Calcularemos los test correspondientes
#a los filtros de de ventas por día:
#El valor p no es menor que   = .05,
#entonces hay evidencia suficiente para decir
#que la muestra proviene de una población con distribución normal.
#JARQUE BERA
jarque.bera.test(fil_dat_2_after$Ventas)
```

```
    Jarque Bera Test

data:  fil_dat_2_after$Ventas
X-squared = 0.65149, df = 2, p-value = 0.722
```

```r
fitdistr(fil_dat_2_after$Ventas, "normal")
```

```
     mean          sd    
  39841.721    7873.446 
 ( 2104.267) ( 1487.941)
```

```r
#_________________________________________________________________
#Calcularemos los test correspondientes
# a los filtros de de ventas por día:
#El valor p no es menor que   = .05,
#entonces hay evidencia suficiente para decir
#que la muestra proviene de una población con distribución normal.
#JARQUE BERA
jarque.bera.test(fil_dat_3_after$Ventas)
```

```
    Jarque Bera Test

data:  fil_dat_3_after$Ventas
X-squared = 0.39157, df = 2, p-value = 0.8222
```

```r
fitdistr(fil_dat_3_after$Ventas, "normal")
```

```
     mean          sd    
  41751.747    5687.030 
 ( 1468.385) ( 1038.305)
```

```r
#_________________________________________________________________
#Calcularemos los test correspondientes
# a los filtros de de ventas por día:
#El valor p no es menor que   = .05,
#entonces hay evidencia suficiente para decir
#que la muestra proviene de una población con distribución normal.
#JARQUE BERA
jarque.bera.test(fil_dat_4_after$Ventas)
```

```
	Jarque Bera Test

data:  fil_dat_4_after$Ventas
X-squared = 3.8727, df = 2, p-value = 0.1442
```

```
fitdistr(fil_dat_4_after$Ventas, "normal")
```

```
     mean          sd     
  43243.143   10517.841 
 ( 2811.011) ( 1987.685)
```

```
#_________________________________________________________________
#Calcularemos los test correspondientes
# a los filtros de de ventas por día:
#El valor p no es menor que   = .05,
#entonces hay evidencia suficiente para decir
#que la muestra proviene de una población con distribución normal.
#JARQUE BERA
jarque.bera.test(fil_dat_5_after$Ventas)
```

```
	Jarque Bera Test

data:  fil_dat_5_after$Ventas
X-squared = 2.0444, df = 2, p-value = 0.3598
```

```
fitdistr(fil_dat_5_after$Ventas, "normal")
```

```
     mean          sd     
  43010.307    7741.889 
 ( 2069.107) ( 1463.080)
```

```
#_________________________________________________________________
#Calcularemos los test correspondientes
# a los filtros de de ventas por día:
# El valor p no es menor que   = .05,
#entonces hay evidencia suficiente para decir
#que la muestra proviene de una población con distribución normal.
#JARQUE BERA
jarque.bera.test(fil_dat_6_after$Ventas)
```

```
    Jarque Bera Test

data:  fil_dat_6_after$Ventas
X-squared = 1.4585, df = 2, p-value = 0.4823
```

```
fitdistr(fil_dat_6_after$Ventas, "normal")
```

```
      mean          sd
  61191.300    7202.989
 ( 1925.080) ( 1361.237)
```

```
#_______________________________________________________________
#Calcularemos los test correspondientes
# a los filtros de de ventas por día:
#El valor p no es menor que  = .05,
#entonces hay evidencia suficiente para decir
#que la muestra proviene de una población con distribución normal.
jarque.bera.test(fil_dat_7_after$Ventas)
```

```
    Jarque Bera Test

data:  fil_dat_7_after$Ventas
X-squared = 0.85511, df = 2, p-value = 0.6521
```

```
#Se ajustara una distribución con el comando siguiente:
fitdistr(fil_dat_7_after$Ventas, "normal")
```

```
      mean          sd
  85684.058    8371.359
 ( 2237.340) ( 1582.038)
```

Con las distribuciones identificadas de las ventas por día en los dos periodos, ahora es posible simular las variables aleatorias de las ventas de cada día de la semana y por lo tanto simular las ganancias de toda la semana.

6.4. Modelo Modificado de Cramér-Lundberg para identificar la probabilidad de las ganancias insuficientes en una empresa de comida rápida

Como ya se vio en la sección (Capítulo 5) El Modelo de Cramér-Lundberg (MCL) es una herramienta utilizada en el campo de la gestión de riesgos y las finanzas para evaluar la probabilidad de incumplimiento o pérdidas insuficientes en una empresa. Aunque no es específicamente diseñado para analizar la variación de las ganancias de una empresa, en particular una empresa de comida rápida, puede adaptarse para evaluar la probabilidad de ganancias insuficientes en este tipo de negocio, que puede llevar a la decisión de cierre de las actividades de la empresa.

Para ello la ecuación general del modelo modificado que representa la evolución de las ganancias en el tiempo hasta la semana t, denotada por G queda definida como:

$$G(t) = u - ct + V(t), \quad t = 1, 2, 3, ... \tag{6.1}$$

la cual se estudiara en los dos periodo de tiempo (pre-pandemia, pos-pandemia).

- $V(t)$: son los ingresos de las ganancias por día para cada semana
- c : Son los costos de las obligaciones financieras del inversionista, como prestamos, pagos por adquisición de la franquicia, compras de equipos inmobiliarios, pagos de empleados, etc.
- $u \geq 0$: Es el capital inicial con el cual inicia la empresa a trabajar.

6.4.1. Caso pre-pandemia

Para la implementación del modelo modificado Cramér-Lundberg con los datos antes de la pandemia se hizo la simulación de las variables aleatorias mediante las distribuciones de las ventas por día tomado como las ganancias acumuladas, y se estimó los parámetros de u y c, por lo que el modelo es determinista y se incrementa por semana.

6.4.1.1. Modelo de Cramér-Lundberg simulado en R con datos reales de las ventas por día de la empresa.

En la (Figura 6.5) se muestra la trayectoria del modelo bajo los supuestos del modelo modificado, el proceso comienza con $t = 90$, que son las 90 semanas de de las ventas registradas por día antes del cierre pandemico, con un capital inicial de $u = 1759629$, una tasa de $c = 0.29$, los montos de las ganancias tienen distribución normal y weibull.

El código que genera la trayectoria es el siguiente:

```r
#Simulación Modelo Clásico de Cramér-Lundberg
#para tres meses: Antes de la pandemia
#Con los datos reales de la empresa
library(dplyr)#librería para poder renombrar las cabeceras
# de los dataframes
#Parámetros
set.seed(13) #semilla fija
u = 1759629 #surplus(capital inicial de salvamento)
#Es un estimado a partir de la media de las ganancias por semana,
#multiplicado por 10/3,
#siendo una proporción para evitar la ruina
# u (sum(medias))*(10/3)
c =0.29*u #prima de pago cada timepo t. c=0.5*u
t_final = 90
# S(t) = \sum_{i=1}^{N(t)}X_i
#donde N(t)~ Poisson (lambda*t)
# X_i ~ exponencial (lambda_Xi)
#CL = REPRESENTA EL MODELO DE CRAMER LUNDBERG
#Simulación de trayectoria de CL_t, cuando t < t_final.
trayectoria_CLt <- function(u, c, t_final)
{
  tiempo <- c(0)
  Cramer_trayectoria <- c(u)
  while(tiempo[length(tiempo)] < t_final)
  {
    tiempo_llegada <- (1)
  Y_i <-  (rnorm(1, mean = 139929.468, sd = 24521.524 )  +
            rnorm(1, mean = 48125.734 , sd=17150.338 )   +
            rnorm(1, mean =  44509.755, sd = 14312.338)  +
            rnorm(1, mean =   46904.516, sd = 16238.151) +
            rnorm(1, mean = 52786.734  , sd = 18403.17)  +
            rnorm(1, mean = 118645.9   , sd = 36530.24)   +
    rweibull(1,  shape = 4.784508e+00, scale = 1.295070e+05 ) )
    tiempo <- c(tiempo, tiempo[length(tiempo)] +
                 tiempo_llegada,tiempo[length(tiempo)] +
                 tiempo_llegada )
    Cramer_trayectoria <- c(Cramer_trayectoria,
           Cramer_trayectoria[length(Cramer_trayectoria)]-
                               c*tiempo_llegada,
          Cramer_trayectoria[length(Cramer_trayectoria)]-
                               c*tiempo_llegada +  Y_i )
    if(Cramer_trayectoria[length(Cramer_trayectoria)] < 0){
      ruina = 1
```

```
    }
    else{
      ruina = 0
    }
  }
  df <- data.frame(tiempo, Cramer_trayectoria)
  df_trayectoria <- data.frame(df%>% rename
                               (Tiempo = tiempo,
                                 Ct = Cramer_trayectoria))
  return(df_trayectoria)

}
trayectoria <- trayectoria_CLt(u,c, t_final )

#Intalación de plotly usando github
#devtools::install_github("ropensci/plotly")

fig_tr<- plot_ly(trayectoria, x = ~Tiempo, y = ~Ct,
name = "Ganancias: G(t)", type = "scatter", mode = "lines")
fig_tr <- fig_tr %>% add_trace(x = ~Tiempo, y = u,
name = "Capital inicial: u", type = "scatter", mode = "lines")
fig_tr
```

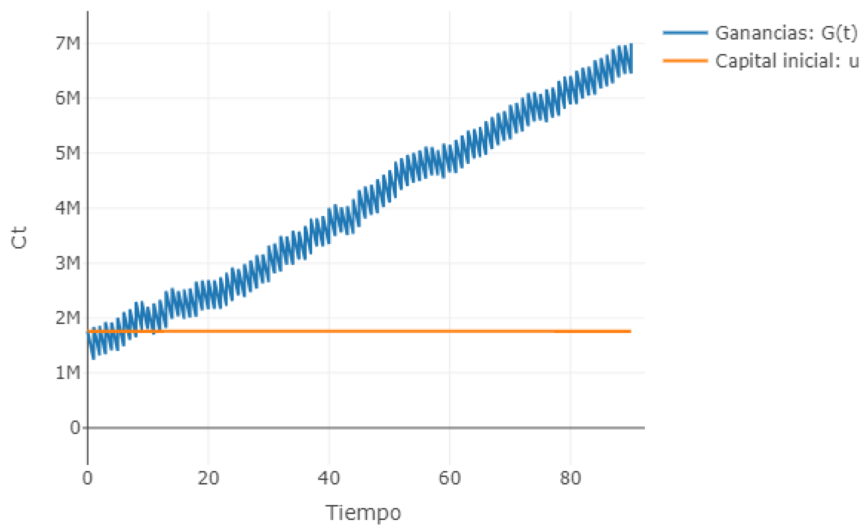


Figura 6.5.: Trayectoria del proceso de evolución del modelo modificado de Cramér Lundberg antes de la pandemia.

6.4.1.2. Probabilidad de ruina

Con la simulación de la variable aleatoria $G(t)$, es posible ahora calcular la probabilidad de ruina, utilizando el método de Monte Carlo(citar).

Para ello, se quiere analizar el momento en el que $G(t) \leq u$, es decir cuando la ganancia al final de la semana t es menor al capital inicial. Cuando $G(t) \leq 0$, se alcanza como la ruina de la empresa, para ambos casos se utiliza la comparación con la tasa de interés para préstamos bancarios de 0.8 anual.

El código para simular la probabilidad de las ganancias finales bajas es el siguiente:

```
#Simulación de la probabilidad de ruina: Antes de la pandemia
#Con los datos reales de la empresa
library(dplyr) # librería para poder renombrar
#las cabeceras de los dataframes

#Parámetros
set.seed(13) #semilla fija
u = 1759629 #surplus(capital inicial de salvamento)
#Es un estimado a partir de la media de las ganancias por semana,
#multiplicado por 10/3,
#siendo una proporción para evitar la ruina
# u (sum(medias))*(10/3)
c = 0.29*u #prima de pago cada tiempo t. c=0.5*u
lambda_Nt = 0.5
#lambda_Xi = 3
t_final = 48
mu = 1 #tiempos entrellegadas constantes
trayectoria_CLt <- function(u, c, lambda_Nt, t_final)
{
  tiempo <- c(0)
  Cramer_trayectoria <- c(u)
  while(tiempo[length(tiempo)] < t_final)
  {
    tiempo_llegada <- (1)
    #Suponiendo los tiempos entrellegadas constantes \mu = 1
    Y_i <-  (rnorm(1, mean = 139929.468, sd = 24521.524 ) +
             rnorm(1, mean = 48125.734 , sd=17150.338 ) +
             rnorm(1, mean =  44509.755, sd = 14312.338 ) +
             rnorm(1, mean =   46904.516, sd = 16238.151 ) +
             rnorm(1, mean = 52786.734   , sd = 18403.17 ) +
             rnorm(1, mean = 118645.9   , sd = 36530.24) +
             rweibull(1,  shape = 4.784508e+00, scale = 1.295070e+05))
```

```
    tiempo <- c(tiempo, tiempo[length(tiempo)] +
                  tiempo_llegada,tiempo[length(tiempo)] +
                  tiempo_llegada )
    Cramer_trayectoria <-c(Cramer_trayectoria,
                Cramer_trayectoria[length(Cramer_trayectoria)]-
                  c*tiempo_llegada,
                Cramer_trayectoria[length(Cramer_trayectoria)]-
                  c*tiempo_llegada +  Y_i )
    if(Cramer_trayectoria[length(Cramer_trayectoria)] < 0){
      ruina = 1
    }
    else{
      ruina = 0
    }
  }
# 1.08*u es la ganancia inferior a la de un tasa de
  #un título financiero
#para el año 2018
  if(Cramer_trayectoria[length(Cramer_trayectoria)] < 1.08*u) {
    ganancia_no_deseada = 1

  }
  else{
    ganancia_no_deseada = 0
  }
  df <- data.frame(tiempo, Cramer_trayectoria)
  df_trayectoria <- data.frame(df%>%rename
                              (Tiempo = tiempo,
                               Ct = Cramer_trayectoria))
  salida <- c(ruina, ganancia_no_deseada)
  return(salida)

}
trayectoria <- trayectoria_CLt(u,c,lambda_Nt, t_final )
#Método de monte carlo para estimar
#la probabilidad de las ganancias bajas
n_replicaciones = 100
r_baja_ganancia <- replicate(n_replicaciones,
trayectoria_CLt(u, c, lambda_Nt, t_final)
[2])
#r_baja_ganancia
prob_baja_ganancia <- sum(r_baja_ganancia >0)/n_replicaciones
```

Es claro que la evolución de $G(t)$ antes de la pandemia genera una probabilidad de ganancias bajas cero.

```
[1] 0
```

El código para calcular la probabilidad de ruina es el siguiente:

```
#Simulación de la probabilidad de ruina: Antes de la pandemia
#Con los datos reales de la empresa
library(dplyr) # libreria para poder renombrar
#las cabeceras de los dataframes

#Parametros
set.seed(13) #semilla fija
u = 1759629 #surplus(capital inicial de salvamento)
#Es un estimado a partir de la media de las ganancias por semana,
#multiplicado por 10/3,
#siendo una proporción para evitar la ruina
# u (sum(medias))*(10/3)
c = 0.29*u #prima de pago cada tiempo t. c=0.5*u
lambda_Nt = 0.5
#lambda_Xi = 3
t_final = 48
mu = 1 #tiempos entrellegadas constantes
trayectoria_CLt <- function(u, c, lambda_Nt, t_final)
{
  tiempo <- c(0)
  Cramer_trayectoria <- c(u)
  while(tiempo[length(tiempo)] < t_final)
  {
    tiempo_llegada <- (1)
#Suponiendo los tiempos entrellegadas constantes \mu = 1
    Y_i <-  (rnorm(1, mean = 139929.468, sd = 24521.524 ) +
             rnorm(1, mean = 48125.734 , sd=17150.338 )   +
             rnorm(1, mean =  44509.755, sd = 14312.338 ) +
             rnorm(1, mean =   46904.516, sd = 16238.151) +
             rnorm(1, mean = 52786.734  , sd = 18403.17 ) +
             rnorm(1, mean = 118645.9   , sd = 36530.24)   +
             rweibull(1,  shape = 4.784508e+00, scale = 1.295070e+05))
    tiempo <- c(tiempo, tiempo[length(tiempo)] +
                  tiempo_llegada,tiempo[length(tiempo)] +
                  tiempo_llegada )
    Cramer_trayectoria <-c(Cramer_trayectoria,
```

```
                Cramer_trayectoria[length(Cramer_trayectoria)]-
                             c*tiempo_llegada,
                Cramer_trayectoria[length(Cramer_trayectoria)]-
                             c*tiempo_llegada +  Y_i )
    if(Cramer_trayectoria[length(Cramer_trayectoria)] < 0){
      ruina = 1
    }
    else{
      ruina = 0
    }
  }
# 1.08*u es la ganancia inferior a la de un tasa de
# un título financiero para el año 2018
  if(Cramer_trayectoria[length(Cramer_trayectoria)] < 1.08*u) {
    ganancia_no_deseada = 1

  }
  else{
    ganancia_no_deseada = 0
  }
  df <- data.frame(tiempo, Cramer_trayectoria)
  df_trayectoria <- data.frame(df%>% rename
                              (Tiempo = tiempo,
                               Ct = Cramer_trayectoria))
  salida <- c(ruina, ganancia_no_deseada)
  return(salida)

}
trayectoria <- trayectoria_CLt(u,c,lambda_Nt, t_final )
#Método de monte carlo para estimar la probabilidad de ruina
n_replicaciones = 100
r_ruina <- replicate(n_replicaciones,
            trayectoria_CLt(u, c, lambda_Nt, t_final)[1])
#r_ruina
prob_ruin <- sum(r_ruina>0)/n_replicaciones
```

En efecto la evolución de $G(t)$ antes de la pandemia genera una probabilidad ruina cero.

```
[1] 0
```

6.4.1.3. Análisis de sensibilidad

Se desea ahora analizar los cambios del resultado del modelo cuando se modifican sus entradas o parámetros, por lo cual se realiza un análisis de sensibilidad de algunos parámetros.

Para ello del modelo modificado de Cramér Lundberg se toma los parámetros u y c del modelo modificado y se establece los rangos o valores posibles para cada uno de los ciertos parámetros, también se toma las medianas de las ganancia en la semana final de 100 simulaciones de las trayectorias del modelo, los rangos se determinan hasta que el modelo presente un comportamiento diferente.

```r
#Simulación Modelo Clásico de Cramer-Lundberg para tres meses:
#Antes de la pandemia Con los datos reales de la empresa

library(dplyr) # libreria para poder renombrar
#las cabeceras de los dataframes
#Parametros
set.seed(13) #semilla fija
u = 1759629 #surplus(capital inicial de salvamento)
#Es un estimado a partir de la media de las ganancias por semana,
#multiplicado por 10/3,
#siendo una proporción para evitar la ruina
# u (sum(medias))*(10/3)
decimal_c = 0.29
c =decimal_c*u #prima de pago cada timepo t. c=0.5*u
lambda_Nt = 0.5
#lambda_Xi = 3
t_final = 90
# S(t) = \sum_{i=1}^{N(t)}X_i
#donde N(t)~ Poisson (lambda*t)
# X_i ~ exponencial (lambda_Xi)
#CL = REPRESENTA EL MODELO DE CRAMER LUNDBERG
#Simulación de trayectoria de CL_t, cuando t < t_final.
trayectoria_CLt <- function(u, c, lambda_Nt, t_final)
{
  tiempo <- c(0)
  Cramer_trayectoria <- c(u)
  while(tiempo[length(tiempo)] < t_final)
  {
    #tiempo_llegada <- rexp(1, rate = lambda_Nt)
    tiempo_llegada <- (1)
    Y_i <-  (rnorm(1, mean = 139929.468, sd = 24521.524 ) +
            rnorm(1, mean = 48125.734 , sd=17150.338 )    +
```

```
            rnorm(1, mean =  44509.755, sd = 14312.338 )  +
            rnorm(1, mean =   46904.516, sd = 16238.151 ) +
            rnorm(1, mean = 52786.734  , sd = 18403.17 )  +
            rnorm(1, mean = 81601.876  , sd = 23756.037)  +
            rweibull(1,  shape = 4.784508e+00, scale = 1.295070e+05 ))
    tiempo <- c(tiempo, tiempo[length(tiempo)] +
                  tiempo_llegada,
                tiempo[length(tiempo)] +
                  tiempo_llegada )
    Cramer_trayectoria <- c(Cramer_trayectoria,
            Cramer_trayectoria[length(Cramer_trayectoria)]-
                          c*tiempo_llegada,
            Cramer_trayectoria[length(Cramer_trayectoria)]-
                          c*tiempo_llegada +  Y_i )
    if(Cramer_trayectoria[length(Cramer_trayectoria)] < 0)
      {
      ruina = 1
      }
    else
      {
      ruina = 0
      }
  }
  df <- data.frame(tiempo, Cramer_trayectoria)
  df_trayectoria <- data.frame(df%>% rename
                               (Tiempo = tiempo,
                                Ct = Cramer_trayectoria))
  return(df_trayectoria$Ct[length(df_trayectoria$Ct)])

}

# La función generador_mediana nos calcula
#la mediana de las ganancias finales
#de 100 trayectorias, fijando el u= surplus y el c
generador_mediana<- function(ui,cj)
  {
  ganancia_final_replicas <- replicate(100,
                                       trayectoria_CLt(u, c,
                                        lambda_Nt, t_final))
  return( median(ganancia_final_replicas))
  }
#Se crea la rejilla donde se hace el analisis de sensibilidad
#para diferentes valores de u y c
```

```r
grid_u <- seq(from = (u-100000*4), to = (u+100000*4), by = 100000)
grid_c <- seq(from = (decimal_c-0.01*4), to = (decimal_c+0.01*4),
              by = 0.01)
u_t <- grid_u

matriz_mediana <- matrix(rep(0, length(grid_u)*length(grid_c)),
nrow= length(grid_u), ncol= length(grid_c))
G_t <- matriz_mediana
for (i in 1:length(grid_u))
  {
    for (j in 1:length(grid_c))
      {
        G_t[i,j] <- generador_mediana(u_t[i],
                              grid_c[j]*u_t[i])
      }
  }
#matriz_mediana

#Grafica del ánalisis de sensibilidad
library(plotly)
library(ggplot2)
c_t <- grid_u*grid_c
fig <- plot_ly(
  type = 'surface',
  contours = list(
    x = list(show = TRUE, start = u_t[1],
             end = u_t[length(u_t)],
             size =100000 , color = 'red'),
    z = list(show = TRUE, start = G_t[1],
             end = G_t[length(G_t)],
             size = 0.01*100000)),
  x = ~u_t,
  y = ~c_t,
  z = ~G_t)
fig <- fig %>% layout(
  scene = list(autosize = F, width = 500, height = 500,
    xaxis = list(nticks = 20),
    zaxis = list(nticks = 8),
    camera = list(eye = list(x = 0,
                             y = -1, z = 1)),
    aspectratio = list(x = .9, y = .8, z = 0.2)))

fig
```

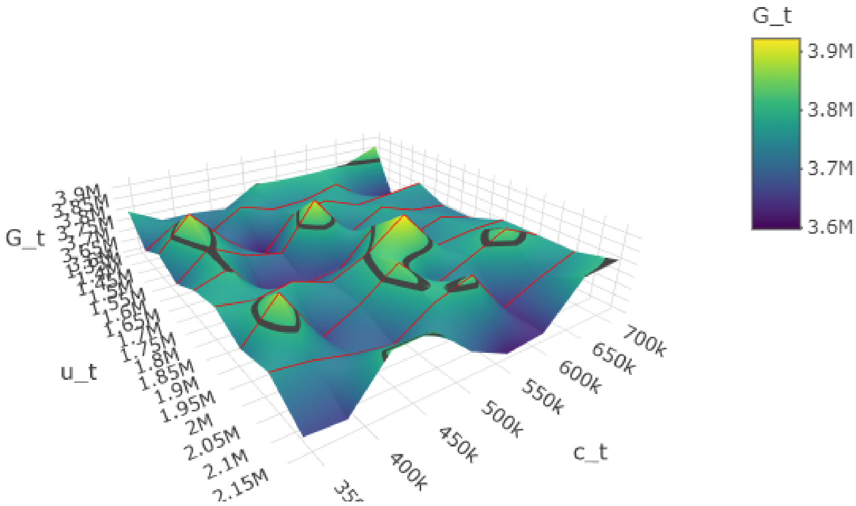


Figura 6.6.: Análisis de sensibilidad pre-pandemia para 90 semanas

6.4.2. Caso pos-pandemia

De manera análoga a la metodología empleada con los datos pre-pandemicos, para la simulación del modelo Crámer-Lundberg con los datos después de la pandemia se hizo uso de las medias de las distribuciones de las ventas por día tomado como las ganancias acumuladas, y se estimó los parámetros de u y c.

6.4.2.1. Modelo de Cramér-Lundberg simulado en R con datos reales de las ventas por día de la empresa

En la (Figura 6.7) se muestra la trayectoria del modelo bajo los supuestos del modelo modificado, el proceso comienza con $t = 14$, que son las 14 semanas de de las ventas registradas por día después del cierre pandemico, con un capital inicial de $u = 1759629$, una tasa de $c = 0.29$, los montos de las ganancias tienen todas distribución normal (N).

El código que genera la trayectoria es el siguiente:

```
library(dplyr) # libreria para poder renombrar
#las cabeceras de los dataframes
#Parametros
set.seed(13) #semilla fija
u = 1759629 #surplus(capital inicial de salvamento)
```

```
#Es un estimado a partir de la media de las ganancias por semana,
#multiplicado por 10/3,
#siendo una proporción para evitar la ruina
# u (sum(medias))*(10/3)
c = (0.29*u) #prima de pago cada timepo t. c=0.5*u
lambda_Nt = 0.5
#lambda_Xi = 3
t_final = 14
# S(t) = \sum_{i=1}^{N(t)}X_i
#donde N(t)~ Poisson (lambda*t)
# X_i ~ exponencial (lambda_Xi)
#CL = REPRESENTA EL MODELO DE CRAMER LUNDBERG
#Simulación de trayectoria de CL_t, cuando t < t_final.
trayectoria_CLt_post_pandemia <- function(u, c, lambda_Nt, t_final)
{
  tiempo <- c(0)
  Cramer_trayectoria <- c(u)
  while(tiempo[length(tiempo)] < t_final)
  {
    tiempo_llegada <- (1)#rexp(1, rate = lambda_Nt)
    Y_i <-  (rnorm(1, mean = 101919.050 , sd = 10877.075)+
             rnorm(1, mean =  39841.721 , sd= 7873.446)  +
             rnorm(1, mean =   41751.747, sd = 5687.030) +
             rnorm(1, mean =   43243.143, sd = 10517.841)+
             rnorm(1, mean = 43010.307  , sd = 7741.889 )+
             rnorm(1, mean = 61191.300  , sd = 7202.989) +
             rnorm(1, mean =  85684.058 , sd = 8371.359 ) )
tiempo <- c(tiempo, tiempo[length(tiempo)] +
              tiempo_llegada,tiempo[length(tiempo)] +
              tiempo_llegada )
Cramer_trayectoria <- c(Cramer_trayectoria,
Cramer_trayectoria[length(Cramer_trayectoria)] -
  c*tiempo_llegada,
Cramer_trayectoria[length(Cramer_trayectoria)] -
  c*tiempo_llegada +  Y_i )
  }
  df_post_pandemia <- data.frame(tiempo, Cramer_trayectoria)
df_trayectoria_post_pandemia <- data.frame(df_post_pandemia%>%
                                        rename(Tiempo = tiempo,
                                        Ct = Cramer_trayectoria))
  return(df_trayectoria_post_pandemia)
}
trayectoria_post_pandemia <- trayectoria_CLt_post_pandemia(u,c,
```

```
                                                lambda_Nt,t_final)


library(plotly)
fig_tr2 <- plot_ly(trayectoria_post_pandemia, x = ~Tiempo,
                                                y = ~Ct,
          name = "Ganancias: G(t)",
          type = "scatter", mode = "lines")

fig_tr2 <- fig_tr2 %>% add_trace(x = ~Tiempo, y = u,
        name = "Capital inicial: u",
        type = "scatter", mode = "lines")
fig_tr2
```

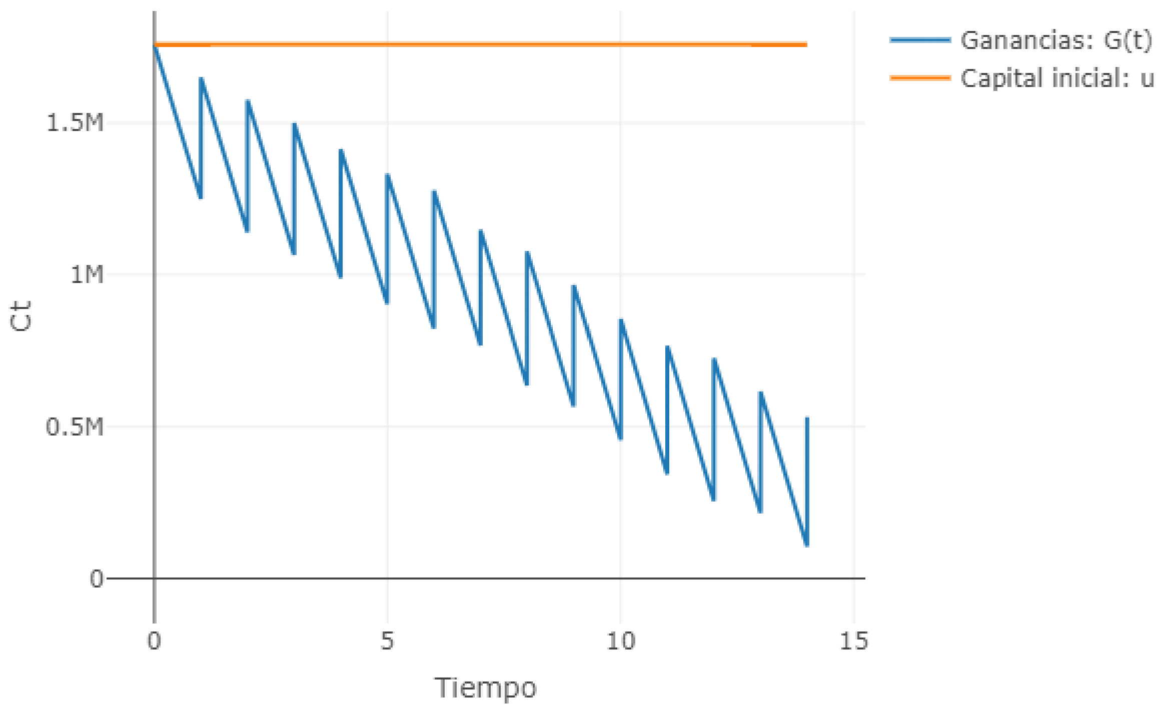


Figura 6.7.: Trayectoria de pérdidas del proceso de evolución modificado de Cramér Lundberg después de la pandemia

Se emplean los mismos parámetros obtenidos en el periodo de tiempo pre pandemia del modelo modificado donde se observa una completa ruina al alcanzar 17 semanas.

```r
library(dplyr) # libreria para poder renombrar
# las cabeceras de los dataframes

#Parametros
set.seed(13) #semilla fija
u = 1759629 #surplus(capital inicial de salvamento)
#Es un estimado a partir de la media de las ganancias por semana,
#multiplicado por 10/3,
#siendo una proporción para evitar la ruina
# u (sum(medias))*(10/3)
c = 0.29*u #prima de pago cada timepo t. c=0.5*u
lambda_Nt = 0.5
#lambda_Xi = 3
t_final = 17
# S(t) = \sum_{i=1}^{N(t)}X_i
#donde N(t)~ Poisson (lambda*t)
# X_i ~ exponencial (lambda_Xi)
#CL = REPRESENTA EL MODELO DE CRAMER LUNDBERG
#Simulación de trayectoria de CL_t, cuando t < t_final.
trayectoria_CLt_post_pandemia <- function(u, c, lambda_Nt, t_final)
{
  tiempo <- c(0)
  Cramer_trayectoria <- c(u)
  while(tiempo[length(tiempo)] < t_final)
  {
    tiempo_llegada <- (1)#rexp(1, rate = lambda_Nt)
    Y_i <-  (rnorm(1, mean = 101919.050 , sd = 10877.075  )
+ rnorm(1, mean =  39841.721 , sd= 7873.446
    ) +
rnorm(1, mean =   41751.747, sd = 5687.030  ) +
rnorm(1, mean =   43243.143, sd = 10517.841 ) +
rnorm(1, mean = 43010.307  , sd = 7741.889 ) +
rnorm(1, mean = 61191.300  , sd = 7202.989) +
rnorm(1, mean =  85684.058 , sd = 8371.359 ) )
    tiempo <- c(tiempo, tiempo[length(tiempo)]
    + tiempo_llegada,tiempo[length(tiempo)]
    + tiempo_llegada )
    Cramer_trayectoria <- c(Cramer_trayectoria,
    Cramer_trayectoria[length(Cramer_trayectoria)]
    - c*tiempo_llegada,
    Cramer_trayectoria[length(Cramer_trayectoria)]-
    c*tiempo_llegada +  Y_i )
  }
```

```
  df_post_pandemia <- data.frame(tiempo, Cramer_trayectoria)
df_trayectoria_post_pandemia <- data.frame(df_post_pandemia%>%
                                    rename(Tiempo = tiempo,
    Ct = Cramer_trayectoria))
  return(df_trayectoria_post_pandemia)
}
trayectoria_post_pandemia <- trayectoria_CLt_post_pandemia(u,c,
                                          lambda_Nt, t_final )

library(plotly)
#plot(trayectoria$Tiempo, trayectoria$Ct, type= "l")
fig_tr3 <- plot_ly(trayectoria_post_pandemia, x = ~Tiempo,
                                             y = ~Ct,
            name = "Ganancias:G(t)",
            type = "scatter", mode = "lines")

fig_tr3 <- fig_tr3 %>% add_trace(x = ~Tiempo, y = u,
           name = "Capital inicial:u",
           type = "scatter", mode = "lines")
fig_tr3
```

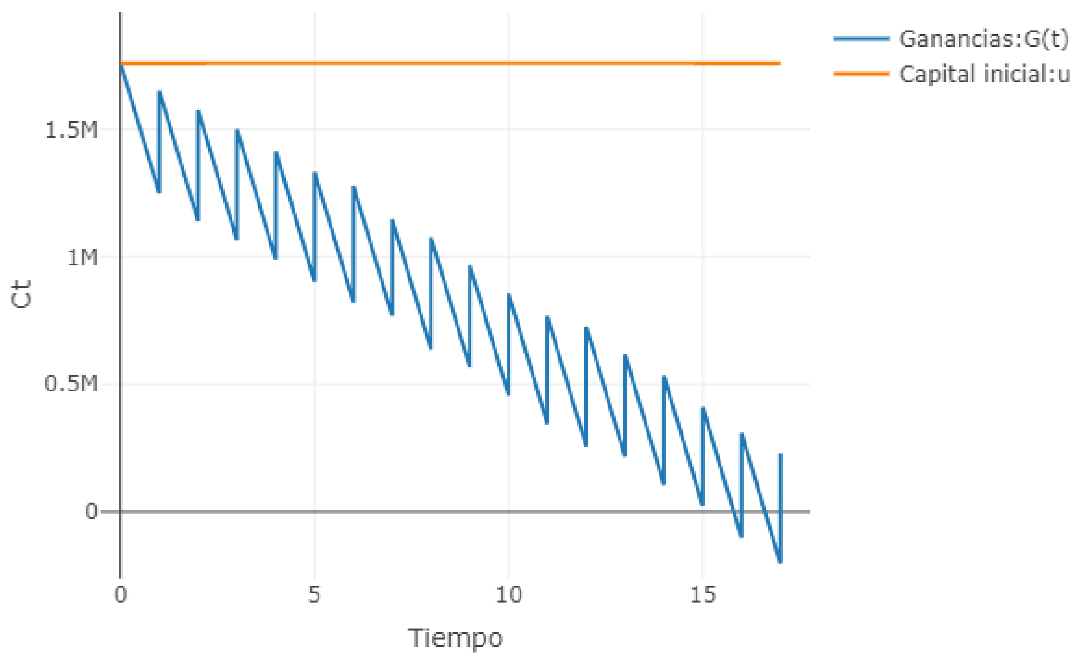


Figura 6.8.

La evolución del proceso de pérdidas en el modelo modificado de Cramér-Lundberg proporciona una herramienta más precisa y adaptable para predecir y contrarrestar los impactos financieros negativos de la empresa.

Se incorpora un factor adicional y ajuste que mejoran la simulación del modelo, mediante la modificación del parámetro c, reduciendo un 17% presentándose así los resultados siguientes mediante el siguiente código.

```
library(dplyr) # libreria para poder renombrar
# las cabeceras de los dataframes

#Parametros
set.seed(13) #semilla fija
u = 1759629 #surplus(capital inicial de salvamento)
#Es un estimado a partir de la media de las ganancias por semana,
#multiplicado por 10/3,
#siendo una proporción para evitar la ruina
# u (sum(medias))*(10/3)
c = (0.29*u)*(0.83) #prima de pago cada timepo t. c=0.5*u
lambda_Nt = 0.5
#lambda_Xi = 3
t_final = 14
# S(t) = \sum_{i=1}^{N(t)}X_i
#donde N(t)~ Poisson (lambda*t)
# X_i ~ exponencial (lambda_Xi)
#CL = REPRESENTA EL MODELO DE CRAMER LUNDBERG
#Simulación de trayectoria de CL_t, cuando t < t_final.
trayectoria_CLt_post_pandemia <- function(u, c, lambda_Nt, t_final)
{
  tiempo <- c(0)
  Cramer_trayectoria <- c(u)
  while(tiempo[length(tiempo)] < t_final)
  {
    tiempo_llegada <- (1)#rexp(1, rate = lambda_Nt)

    Y_i <-  (rnorm(1, mean = 101919.050 , sd = 10877.075) +
            rnorm(1, mean =  39841.721 , sd= 7873.446) +
            rnorm(1, mean =   41751.747, sd = 5687.030) +
            rnorm(1, mean =   43243.143, sd = 10517.841) +
            rnorm(1, mean = 43010.307  , sd = 7741.889) +
            rnorm(1, mean = 61191.300  , sd = 7202.989) +
            rnorm(1, mean =  85684.058 , sd = 8371.359 ))
    tiempo <- c(tiempo, tiempo[length(tiempo)] +
                tiempo_llegada,tiempo[length(tiempo)] +
```

```r
                tiempo_llegada )
    Cramer_trayectoria <- c(Cramer_trayectoria,
                    Cramer_trayectoria[length(Cramer_trayectoria)]-
                          c*tiempo_llegada,
                    Cramer_trayectoria[length(Cramer_trayectoria)]-
                            c*tiempo_llegada + Y_i )
  }
  df_post_pandemia <- data.frame(tiempo, Cramer_trayectoria)
  df_trayectoria_post_pandemia <- data.frame(df_post_pandemia%>%
                                          rename(Tiempo = tiempo,
                                          Ct = Cramer_trayectoria))
  return(df_trayectoria_post_pandemia)
}
trayectoria_post_pandemia <- trayectoria_CLt_post_pandemia(u,c,
                                            lambda_Nt, t_final )


library(plotly)
#plot(trayectoria$Tiempo, trayectoria$Ct, type= "l")
fig_tr4 <- plot_ly(trayectoria_post_pandemia, x = ~Tiempo,
                   y = ~Ct,
         name = "Ganancias: G(t)",
         type = "scatter", mode = "lines")

fig_tr4 <- fig_tr4 %>% add_trace(x = ~Tiempo, y = u,
        name = "Capital inicial: u",
        type = "scatter", mode = "lines")
fig_tr4
```

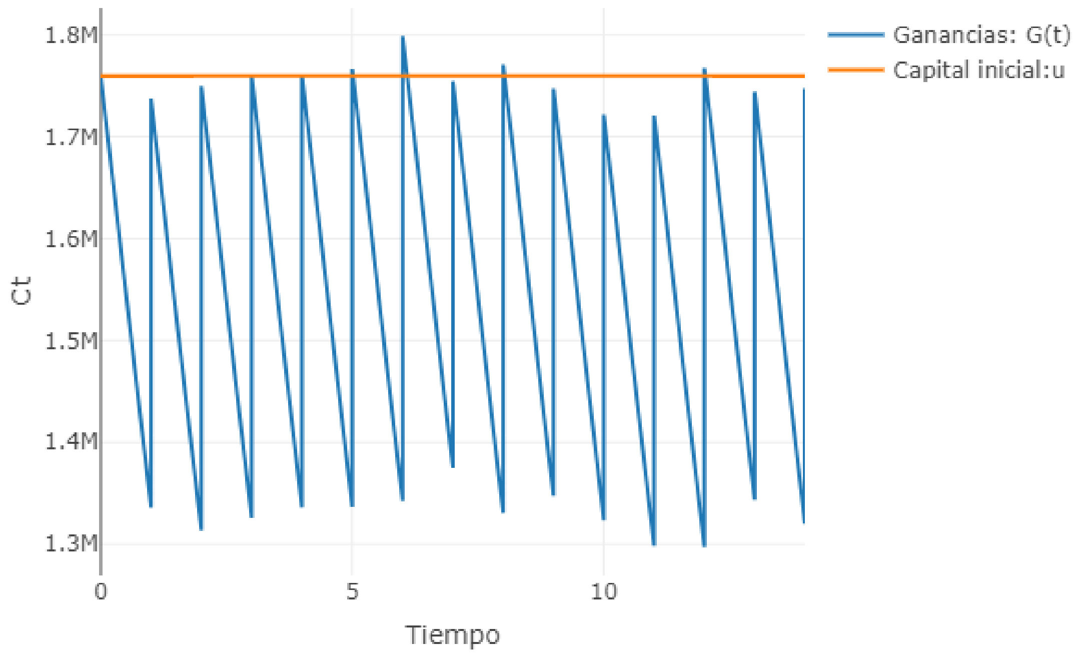


Figura 6.9.: Trayectoria del proceso de pérdidas generadas por el modelo de Cramér Lundberg después de la pandemia con una reducción de los costos al 17%.

6.4.2.2. Probabilidad de ruina

Al igual que para el caso con los datos antes de la pandemia se analizó la semana en el que $G(t) \leq u$ y cuando $G(t) \leq 0$, haciendo la comparación con la tasa de interés de 0.8 anual en una cuenta de ahorro para una inversión del capital inicial u, para este periodo tomamos $t = 14$, es decir los datos registrados hasta las 14 semanas después de pandemia donde la tasa es de 0.0233

El código para calcular la probabilidad de las ganancias finales bajas es el siguiente:

```
library(dplyr) # librería para poder renombrar las
# cabeceras de los dataframes

#Parámetros
set.seed(13) #semilla fija
u = 1759629 #surplus(capital inicial de salvamento)
#Es un estimado a partir de la media de las ganancias por semana,
#multiplicado por 10/3,
#siendo una proporción para evitar la ruina
# u (sum(medias))*(10/3)
```

```r
c = 0.29*u #prima de pago cada timepo t. c=0.5*u
lambda_Nt = 0.5
#lambda_Xi = 3
t_final = 14
mu = 1 #tiempos entrellegadas constantes
# S(t) = \sum_{i=1}^{N(t)}X_i
#donde N(t)~ Poisson (lambda*t)
# X_i ~ exponencial (lambda_Xi)
#CL = REPRESENTA EL MODELO DE CRAMER LUNDBERG
#Simulación de trayectoria de CL_t, cuando t < t_final.
trayectoria_CLt_post_pandemia <- function(u, c, lambda_Nt, t_final)
{
  tiempo <- c(0)
  Cramer_trayectoria <- c(u)
  while(tiempo[length(tiempo)] < t_final)
  {
    #tiempo_llegada <- rexp(1, rate = lambda_Nt)
    tiempo_llegada <- (1)
#Suponiendo los tiempos entrellegadas constantes \mu = 1
    Y_i <-  (rnorm(1, mean = 101919.050 , sd = 10877.075   )
              + rnorm(1, mean =  39841.721 , sd= 7873.446  )
              + rnorm(1, mean =   41751.747, sd = 5687.030 )
              + rnorm(1, mean =   43243.143, sd = 10517.841)
              + rnorm(1, mean = 43010.307  , sd = 7741.889 )
              + rnorm(1, mean = 61191.300  , sd = 7202.989 )
              + rnorm(1, mean =  85684.058 , sd = 8371.359))
    tiempo <- c(tiempo, tiempo[length(tiempo)] +
                  tiempo_llegada,tiempo[length(tiempo)]+
                  tiempo_llegada)
    Cramer_trayectoria <- c(Cramer_trayectoria,
    Cramer_trayectoria[length(Cramer_trayectoria)] -
      c*tiempo_llegada,
    Cramer_trayectoria[length(Cramer_trayectoria)]-
      c*tiempo_llegada +  Y_i )
    if(Cramer_trayectoria[length(Cramer_trayectoria)] < 0){
      ruina = 1
    }
    else{
      ruina = 0
    }
  }
# 1.08*u es la ganancia inferior a la de un tasa de
# un título financiero  para el año 2018
```

```
  if(Cramer_trayectoria[length(Cramer_trayectoria)] < (1 + (14*0.08)/48)*u) {
    ganancia_no_deseada = 1

  }
  else{
    ganancia_no_deseada = 0
  }
  df_post_pandemia <- data.frame(tiempo, Cramer_trayectoria)
  df_trayectoria_post_pandemia <- data.frame(df_post_pandemia%>%
                                  rename(Tiempo = tiempo,
                                         Ct = Cramer_trayectoria))
  salida <- c(ruina, ganancia_no_deseada)
  return(salida)

}
trayectoria_post_pandemia <- trayectoria_CLt_post_pandemia(u,c,
                                            lambda_Nt, t_final )
#Método de monte carlo para estimar la probabilidad de ruina
n_replicaciones = 100
r_baja_ganancia_post_pandemia <- replicate(n_replicaciones,
                          trayectoria_CLt_post_pandemia(u, c,
                                         lambda_Nt, t_final)[2])
r_baja_ganancia_post_pandemia
prob_baja_ganancia_post_pandemia <- sum(r_baja_ganancia_post_pandemia >0)/
n_replicaciones
prob_baja_ganancia_post_pandemia
```

Claramente la probabilidad de las ganancias bajas de la evolución de $G(t)$ después de la pandemia sea 1, pues como se muestra en la Figura 6.7 todas las ventas registradas están por debajo del capital inicial

```
[1] 1
```

El código para calcular la probabilidad de las ganancias finales bajas durante las 17 semanas es el siguiente:

```
library(dplyr) # libreria para poder renombrar
#las cabeceras de los dataframes

#Parametros
set.seed(13) #semilla fija
u = 1759629 #surplus(capital inicial de salvamento)
```

```
#Es un estimado a partir de la media de las ganancias por semana,
#multiplicado por 10/3,
#siendo una proporción para evitar la ruina
# u (sum(medias))*(10/3)
c = 0.29*u #prima de pago cada timepo t. c=0.5*u
lambda_Nt = 0.5
#lambda_Xi = 3
t_final = 17
mu = 1 #tiempos entrellegadas constantes
#df_media <- data.frame(df_media%>% rename(tiempos = Tiempo ,
#media_trayectoria = cramer_media)
# S(t) = \sum_{i=1}^{N(t)}X_i
#donde N(t)~ Poisson (lambda*t)
# X_i ~ exponencial (lambda_Xi)
#CL = REPRESENTA EL MODELO DE CRAMER LUNDBERG
#Simulación de trayectoria de CL_t, cuando t < t_final.
trayectoria_CLt_post_pandemia <- function(u, c, lambda_Nt, t_final)
{
  tiempo <- c(0)
  Cramer_trayectoria <- c(u)
  while(tiempo[length(tiempo)] < t_final)
  {
    #tiempo_llegada <- rexp(1, rate = lambda_Nt)
    tiempo_llegada <- (1)
#Suponiendo los tiempos entrellegadas constantes \mu = 1
    Y_i <-  (rnorm(1, mean = 101919.050 , sd = 10877.075)+
              rnorm(1, mean =  39841.721 , sd= 7873.446) +
              rnorm(1, mean =   41751.747, sd = 5687.030)+
              rnorm(1, mean =   43243.143, sd = 10517.841)+
              rnorm(1, mean = 43010.307   , sd = 7741.889)+
              rnorm(1, mean = 61191.300   , sd = 7202.989)+
              rnorm(1, mean =  85684.058 , sd = 8371.359 ))
    tiempo <- c(tiempo, tiempo[length(tiempo)]+
                   tiempo_llegada,tiempo[length(tiempo)]+
                   tiempo_llegada )
    Cramer_trayectoria <- c(Cramer_trayectoria,
                     Cramer_trayectoria[length(Cramer_trayectoria)]-
                     c*tiempo_llegada,
                     Cramer_trayectoria[length(Cramer_trayectoria)]-
                     c*tiempo_llegada +  Y_i )
    if(Cramer_trayectoria[length(Cramer_trayectoria)] < 0){
      ruina = 1
    }
```

```
    else{
      ruina = 0
    }
  }
# 1.08*u es la ganancia inferior a la de un tasa de
# un título financieropara el año 2018
  if(Cramer_trayectoria[length(Cramer_trayectoria)] < 1.02833*u) {
    ganancia_no_deseada = 1

  }
  else{
    ganancia_no_deseada = 0
  }
  df_post_pandemia <- data.frame(tiempo, Cramer_trayectoria)
  df_trayectoria_post_pandemia <- data.frame(df_post_pandemia%>%
                                             rename(Tiempo = tiempo,
                                             Ct = Cramer_trayectoria))
  salida <- c(ruina, ganancia_no_deseada)
  return(salida)

}
trayectoria_post_pandemia <- trayectoria_CLt_post_pandemia(u,c,
                                             lambda_Nt, t_final )
#Método de monte carlo para estimar la probabilidad de ruina
n_replicaciones = 100
r_baja_ganancia_post_pandemia <- replicate(n_replicaciones,
          trayectoria_CLt_post_pandemia(u, c, lambda_Nt, t_final)[2])
#r_baja_ganancia_post_pandemia
prob_baja_ganancia_post_pandemia <- sum(r_baja_ganancia_post_pandemia >0)/
n_replicaciones
prob_baja_ganancia_post_pandemia
```

```
[1] 1
```

El código para calcular la probabilidad de ruina es el siguiente:

```
library(dplyr) # libreria para poder renombrar
#las cabeceras de los dataframes

#Parametros
set.seed(13) #semilla fija
u = 1759629 #surplus(capital inicial de salvamento)
```

```r
#Es un estimado a partir de la media de las ganancias por semana,
#multiplicado por 10/3,
#siendo una proporción para evitar la ruina
# u (sum(medias))*(10/3)
c = 0.29*u #prima de pago cada timepo t. c=0.5*u
lambda_Nt = 0.5
#lambda_Xi = 3
t_final = 14
mu = 1 #tiempos entrellegadas constantes
#df_media <- data.frame(df_media%>% rename(tiempos = Tiempo ,
#media_trayectoria = cramer_media)
# S(t) = \sum_{i=1}^{N(t)}X_i
#donde N(t)~ Poisson (lambda*t)
# X_i ~ exponencial (lambda_Xi)
#CL = REPRESENTA EL MODELO DE CRAMER LUNDBERG
#Simulación de trayectoria de CL_t, cuando t < t_final.
trayectoria_CLt_post_pandemia <- function(u, c, lambda_Nt, t_final)
{
  tiempo <- c(0)
  Cramer_trayectoria <- c(u)
  while(tiempo[length(tiempo)] < t_final)
  {
    #tiempo_llegada <- rexp(1, rate = lambda_Nt)
    tiempo_llegada <- (1)
#Suponiendo los tiempos entrellegadas constantes \mu = 1
    Y_i <-  (rnorm(1, mean = 101919.050 , sd = 10877.075)  +
               rnorm(1, mean =  39841.721 , sd= 7873.446 ) +
               rnorm(1, mean =   41751.747, sd = 5687.030) +
               rnorm(1, mean =   43243.143, sd = 10517.841)+
               rnorm(1, mean = 43010.307   , sd = 7741.889 )+
               rnorm(1, mean = 61191.300   , sd = 7202.989) +
               rnorm(1, mean =  85684.058 , sd = 8371.359 ))
    tiempo <- c(tiempo, tiempo[length(tiempo)] +
                  tiempo_llegada,tiempo[length(tiempo)] +
                  tiempo_llegada )
    Cramer_trayectoria <- c(Cramer_trayectoria,
                       Cramer_trayectoria[length(Cramer_trayectoria)]-
                          c*tiempo_llegada,
                       Cramer_trayectoria[length(Cramer_trayectoria)]-
                         c*tiempo_llegada +  Y_i )
    if(Cramer_trayectoria[length(Cramer_trayectoria)] < 0){
      ruina = 1
    }
```

```
    else{
      ruina = 0
    }
  }
# 1.08*u es la ganancia inferior a la de un tasa de
# un título financiero para el año 2018
  if(Cramer_trayectoria[length(Cramer_trayectoria)] < 1.02333*u) {
    ganancia_no_deseada = 1

  }
  else{
    ganancia_no_deseada = 0
  }
  df_post_pandemia <- data.frame(tiempo, Cramer_trayectoria)
  df_trayectoria_post_pandemia <- data.frame(df_post_pandemia%>%
                                    rename(Tiempo = tiempo,
                                    Ct = Cramer_trayectoria))
  salida <- c(ruina, ganancia_no_deseada)
  return(salida)

}
trayectoria_post_pandemia <- trayectoria_CLt_post_pandemia(u,c,
                                                lambda_Nt, t_final )
#Método de monte carlo para estimar la probabilidad de ruina
n_replicaciones = 100
r_ruina_post_pandemia <- replicate(n_replicaciones,
          trayectoria_CLt_post_pandemia(u, c, lambda_Nt, t_final)[1])
#r_ruina_post_pandemia
prob_ruin_post_pandemia <- sum(r_ruina_post_pandemia>0)/
n_replicaciones
prob_ruin_post_pandemia
```

Lo que significa que a pesar de que las ganancias sean todas menores al capital inicial, en todas la simulaciones, nunca se presenta una que sea menor que cero.

```
[1] 0
```

El código para calcular la probabilidad de ruina durante las 17 semanas es el siguiente:

```
library(dplyr) # libreria para poder renombrar
#las cabeceras de los dataframes
```

```r
#Parametros
set.seed(13) #semilla fija
u = 1759629 #surplus(capital inicial de salvamento)
#Es un estimado a partir de la media de las ganancias por semana,
#multiplicado por 10/3,
#siendo una proporción para evitar la ruina
# u (sum(medias))*(10/3)
c = 0.29*u #prima de pago cada timepo t. c=0.5*u
lambda_Nt = 0.5
#lambda_Xi = 3
t_final = 17
mu = 1 #tiempos entrellegadas constantes
#df_media <- data.frame(df_media%>% rename(tiempos = Tiempo ,
#media_trayectoria = cramer_media)
# S(t) = \sum_{i=1}^{N(t)}X_i
#donde N(t)~ Poisson (lambda*t)
# X_i ~ exponencial (lambda_Xi)
#CL = REPRESENTA EL MODELO DE CRAMER LUNDBERG
#Simulación de trayectoria de CL_t, cuando t < t_final.
trayectoria_CLt_post_pandemia <- function(u, c, lambda_Nt, t_final)
{
  tiempo <- c(0)
  Cramer_trayectoria <- c(u)
  while(tiempo[length(tiempo)] < t_final)
  {
    #tiempo_llegada <- rexp(1, rate = lambda_Nt)
    tiempo_llegada <- (1)
# Suponiendo los tiempos entrellegadas constantes \mu = 1
    Y_i <-  (rnorm(1, mean = 101919.050 , sd = 10877.075)  +
               rnorm(1, mean =  39841.721 , sd= 7873.446)  +
               rnorm(1, mean =   41751.747, sd = 5687.030) +
               rnorm(1, mean =   43243.143, sd = 10517.841)+
               rnorm(1, mean = 43010.307  , sd = 7741.889) +
               rnorm(1, mean =  85684.058 , sd = 8371.359))
    tiempo <- c(tiempo, tiempo[length(tiempo)] +
                  tiempo_llegada,tiempo[length(tiempo)] +
                  tiempo_llegada )
    Cramer_trayectoria <- c(Cramer_trayectoria,
                      Cramer_trayectoria[length(Cramer_trayectoria)]-
                      c*tiempo_llegada,
                      Cramer_trayectoria[length(Cramer_trayectoria)]-
                      c*tiempo_llegada +  Y_i )
    if(Cramer_trayectoria[length(Cramer_trayectoria)] < 0){
```

```r
      ruina = 1
    }
    else{
      ruina = 0
    }
  }
# 1.08*u es la ganancia inferior a la de
# un tasa de un título financiero para el año 2018
  if(Cramer_trayectoria[length(Cramer_trayectoria)] < 1.02833*u) {
    ganancia_no_deseada = 1

  }
  else{
    ganancia_no_deseada = 0
  }
  df_post_pandemia <- data.frame(tiempo, Cramer_trayectoria)
  df_trayectoria_post_pandemia <- data.frame(df_post_pandemia%>%
                                   rename(Tiempo = tiempo,
                                          Ct = Cramer_trayectoria))
  salida <- c(ruina, ganancia_no_deseada)
  return(salida)

}
trayectoria_post_pandemia <- trayectoria_CLt_post_pandemia(u,c,
                                          lambda_Nt, t_final )
#Método de monte carlo para estimar la probabilidad de ruina
n_replicaciones = 100
r_ruina_post_pandemia <- replicate(n_replicaciones,
                          trayectoria_CLt_post_pandemia(u, c,
                                         lambda_Nt, t_final)[1])
#r_ruina_post_pandemia
prob_ruin_post_pandemia <- sum(r_ruina_post_pandemia>0)/
n_replicaciones
prob_ruin_post_pandemia
```

```
[1] 1
```

6.4.2.3. Análisis de sensibilidad

Modificando los parámetros u y c para erl periodo pso-pandemia del modelo modificado y estableciendo los rangos o valores posibles para cada uno de los ciertos parámetros, se presenta el análisis de sensibilidad durante las semanas registradas después de la pandemia que fueron 14,. Se toman la mediana de las 100 simulaciones de la ganancia final de la semana obtenida por el modelo, es decir la venta registrada en la última semana de los datos registrados.

```
library(dplyr) # libreria para poder renombrar
# las cabeceras de los dataframes

#Parametros
set.seed(13) #semilla fija
u = 1759629 #surplus(capital inicial de salvamento)
#Es un estimado a partir de la media de las ganancias por semana,
#multiplicado por 10/3,
#siendo una proporción para evitar la ruina
# u (sum(medias))*(10/3)
decimal_c = 0.29
c =decimal_c*u #prima de pago cada timepo t. c=0.5*u
lambda_Nt = 0.5
#lambda_Xi = 3
t_final = 14
# S(t) = \sum_{i=1}^{N(t)}X_i
#donde N(t)~ Poisson (lambda*t)
# X_i ~ exponencial (lambda_Xi)
#CL = REPRESENTA EL MODELO DE CRAMER LUNDBERG
#Simulación de trayectoria de CL_t, cuando t < t_final.
trayectoria_CLt_post_pandemia <- function(u, c, lambda_Nt, t_final)
{
  tiempo <- c(0)
  Cramer_trayectoria <- c(u)
  while(tiempo[length(tiempo)] < t_final)
  {
    #tiempo_llegada <- rexp(1, rate = lambda_Nt)
    tiempo_llegada <- (1)
    Y_i <-  (rnorm(1, mean = 101919.050 , sd = 10877.075) +
              rnorm(1, mean =  39841.721 , sd= 7873.446)  +
              rnorm(1, mean =   41751.747, sd = 5687.030) +
              rnorm(1, mean =   43243.143, sd = 10517.841)+
              rnorm(1, mean = 43010.307   , sd = 7741.889) +
              rnorm(1, mean = 61191.300   , sd = 7202.989) +
              rnorm(1, mean =  85684.058 , sd = 8371.359))
    tiempo <- c(tiempo, tiempo[length(tiempo)]
                + tiempo_llegada,tiempo[length(tiempo)]
                + tiempo_llegada )
    Cramer_trayectoria <- c(Cramer_trayectoria,
                      Cramer_trayectoria[length(Cramer_trayectoria)]-
```

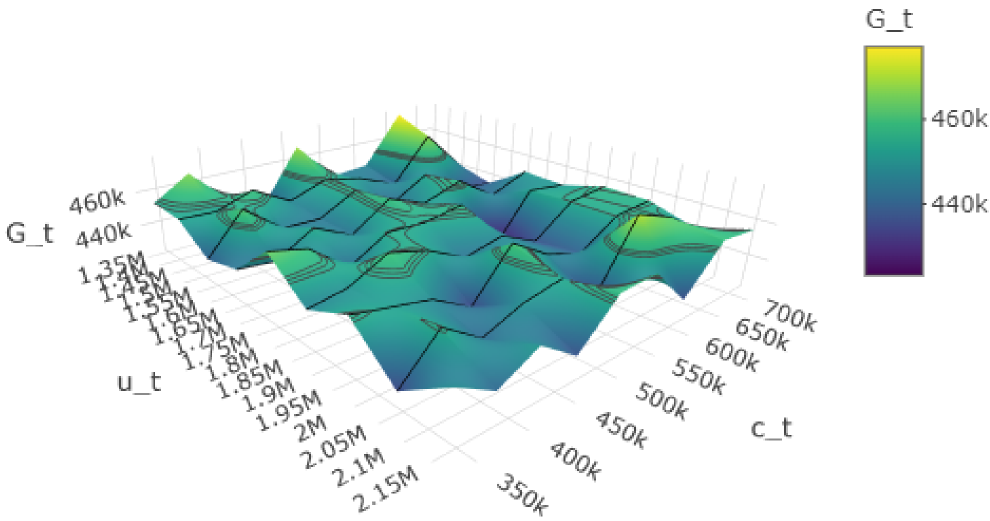
```
                        c*tiempo_llegada,
                      Cramer_trayectoria[length(Cramer_trayectoria)]-
                       c*tiempo_llegada +  Y_i )
    if(Cramer_trayectoria[length(Cramer_trayectoria)] < 0)
    {
      ruina = 1
    }
    else
    {
      ruina = 0
    }
  }
  df_post_pandemia<- data.frame(tiempo, Cramer_trayectoria)
  df_trayectoria_post_pandemia <- data.frame(df_post_pandemia%>%
                                  rename(Tiempo = tiempo,
                                         Ct = Cramer_trayectoria))
  return(df_trayectoria_post_pandemia$Ct[length
                                (df_trayectoria_post_pandemia$Ct)])
}

# La función generador_mediana nos calcula la mediana de
# las ganancias finales de 100 trayectorias,
# fijando el u= surplus y el c
generador_mediana_post_pandemia<- function(ui,cj)
  {
  ganancia_final_replicas_post_pandemia <- replicate(100,
           trayectoria_CLt_post_pandemia(u, c,lambda_Nt, t_final))
    return( median(ganancia_final_replicas_post_pandemia))
  }
# Se crea la rejilla donde se hace el analisis de sensibilidad
# para diferentes valores de u y c
grid_u_post_pandemia <- seq(from = (u-100000*4),
                            to = (u+100000*4),
                            by = 100000)
grid_c_post_pandemia <- seq(from = (decimal_c-0.01*4),
                            to = (decimal_c+0.01*4),
                            by = 0.01)
u_t <- grid_u_post_pandemia
matriz_mediana_post_pandemia <- matrix(rep(0,
      length(grid_u_post_pandemia)*length(grid_c_post_pandemia)),
      nrow= length(grid_u_post_pandemia),
      ncol= length(grid_c_post_pandemia))
```

```r
G_t <- matriz_mediana_post_pandemia

for (i in 1:length(u_t))
{
  for (j in 1:length(grid_c_post_pandemia))
  {
    G_t[i,j] <- generador_mediana_post_pandemia(u_t[i],
                              grid_c_post_pandemia[j]*u_t[i])
  }
}
# matriz_mediana_post_pandemia
# Grafica del ánalisis de sensibilidad
library(plotly)
library(ggplot2)
c_t <- u_t*grid_c_post_pandemia
fig2 <- plot_ly(
  type = 'surface',
  contours = list(x = list(show = TRUE, start = u_t[1],
                  end = grid_u_post_pandemia[length(u_t)],
                  size =100000 , color = 'black'),
                  z = list(show = TRUE, start = G_t[1],
                  end = G_t[length(G_t)],
                  size = 0.01*100000)),
  x = ~u_t,
  y = ~c_t,
  z = ~G_t)
fig2 <- fig2 %>% layout(
    scene = list(
    xaxis = list(nticks = 20),
    zaxis = list(nticks = 4),
    camera = list(eye = list(x = 0,
                             y = -1,
                             z = 1)),
    aspectratio = list(x = .9, y = .8, z = 0.2)))

fig2
```



Se presenta también el análisis de sensibilidad para $t = 19$, debido a que en 19 semanas se presenta el caso donde se pueden registrar ventas por debajo del cero, es decir, casos donde $G(t) \leq 0$.

```
library(dplyr) # libreria para poder
# renombrar las cabeceras de los dataframes

# Parametros
set.seed(13) #semilla fija
u = 1759629 #surplus(capital inicial de salvamento)
# Es un estimado a partir de la media de las ganancias por semana,
# multiplicado por 10/3,
# siendo una proporción para evitar la ruina
# u (sum(medias))*(10/3)
decimal_c = 0.29
c =decimal_c*u #prima de pago cada timepo t. c=0.5*u
lambda_Nt = 0.5
#lambda_Xi = 3
t_final = 19
# S(t) = \sum_{i=1}^{N(t)}X_i
# donde N(t)~ Poisson (lambda*t)
# X_i ~ exponencial (lambda_Xi)
```

```r
# CL = REPRESENTA EL MODELO DE CRAMER LUNDBERG
# Simulación de trayectoria de CL_t, cuando t < t_final.
trayectoria_CLt_post_pandemia <- function(u, c, lambda_Nt, t_final)
{
  tiempo <- c(0)
  Cramer_trayectoria <- c(u)
  while(tiempo[length(tiempo)] < t_final)
  {
    #tiempo_llegada <- rexp(1, rate = lambda_Nt)
    tiempo_llegada <- (1)
    Y_i <-  (rnorm(1, mean = 101919.050 , sd = 10877.075) +
              rnorm(1, mean =  39841.721 , sd= 7873.446)  +
              rnorm(1, mean =   41751.747, sd = 5687.030) +
              rnorm(1, mean =   43243.143, sd = 10517.841)+
              rnorm(1, mean = 43010.307  , sd = 7741.889) +
              rnorm(1, mean = 61191.300  , sd = 7202.989) +
              rnorm(1, mean =  85684.058 , sd = 8371.359))
    tiempo <- c(tiempo, tiempo[length(tiempo)]
                + tiempo_llegada,tiempo[length(tiempo)]
                + tiempo_llegada )
    Cramer_trayectoria <- c(Cramer_trayectoria,
                  Cramer_trayectoria[length(Cramer_trayectoria)]-
                    c*tiempo_llegada,
                  Cramer_trayectoria[length(Cramer_trayectoria)]-
                   c*tiempo_llegada +  Y_i )
    if(Cramer_trayectoria[length(Cramer_trayectoria)] < 0)
    {
      ruina = 1
    }
    else
    {
      ruina = 0
    }
  }
  df_post_pandemia<- data.frame(tiempo, Cramer_trayectoria)
  df_trayectoria_post_pandemia <- data.frame(df_post_pandemia%>%
                                 rename(Tiempo = tiempo,
                                        Ct = Cramer_trayectoria))
  return(df_trayectoria_post_pandemia$Ct[length
                                (df_trayectoria_post_pandemia$Ct)])
}

# La función generador_mediana nos calcula la mediana de
```

```
# las ganancias finales de 100 trayectorias,
# fijando el u= surplus y el c
generador_mediana_post_pandemia<- function(ui,cj)
  {
  ganancia_final_replicas_post_pandemia <- replicate(100,
           trayectoria_CLt_post_pandemia(u, c,lambda_Nt, t_final))
    return( median(ganancia_final_replicas_post_pandemia))
  }
# Se crea la rejilla donde se hace el analisis de sensibilidad
# para diferentes valores de u y c
grid_u_post_pandemia <- seq(from = (u-100000*4),
                            to = (u+100000*4),
                            by = 100000)
grid_c_post_pandemia <- seq(from = (decimal_c-0.01*4),
                            to = (decimal_c+0.01*4),
                            by = 0.01)
u_t <- grid_u_post_pandemia
matriz_mediana_post_pandemia <- matrix(rep(0,
      length(grid_u_post_pandemia)*length(grid_c_post_pandemia)),
      nrow= length(grid_u_post_pandemia),
      ncol= length(grid_c_post_pandemia))

G_t <- matriz_mediana_post_pandemia

for (i in 1:length(u_t))
{
  for (j in 1:length(grid_c_post_pandemia))
  {
    G_t[i,j] <- generador_mediana_post_pandemia(u_t[i],
                                  grid_c_post_pandemia[j]*u_t[i])
  }
}

#Grafica del ánalisis de sensibilidad
#Grafica del ánalisis de sensibilidad
library(plotly)
library(ggplot2)
c_t <- grid_u_post_pandemia*grid_c_post_pandemia

z1<- matrix(rep(0,
length(grid_u_post_pandemia)*length(grid_c_post_pandemia)), nrow=
length(grid_u_post_pandemia), ncol= length(grid_c_post_pandemia))
```

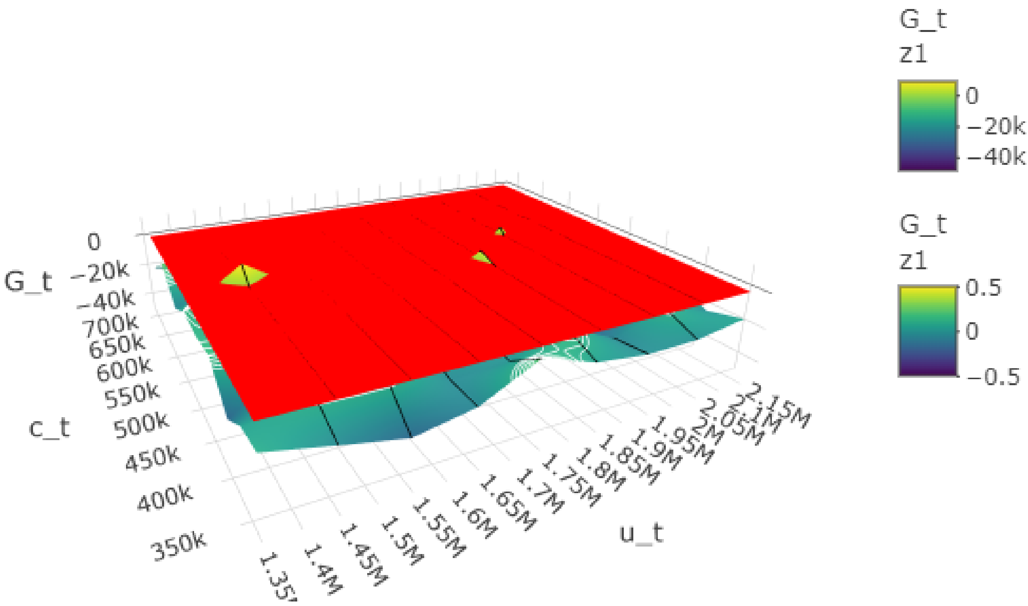
```
z2<- matrix(rep(u,
length(grid_u_post_pandemia)*length(grid_c_post_pandemia)), nrow=
length(grid_u_post_pandemia), ncol= length(grid_c_post_pandemia))

fig3 <- plot_ly(
  type = 'surface',
  contours = list(
    x = list(show = TRUE, start = u_t[1], end =
u_t[length(u_t)], size =100000 ,
color = 'black'),
    z = list(show = TRUE, start = G_t[1],
             end = G_t[length(G_t)], size =0.01*100000 ,
             color = 'white')),
  x = ~u_t,
  y = ~c_t,
  z = ~G_t)

fig3 <- fig3%>% add_surface(
  type = 'surface',
  contours = list(
    x = list(show = TRUE, start = u_t[1],
             end = u_t[length(u_t)], size =100000 ,
             color = 'black'),
    y = list(show = TRUE, start = c_t[1],
             end = c_t[length(c_t)],
             size = 0.01*100000 , color = 'red')),
  x = ~u_t,
  y = ~c_t,
  z = ~z1)

fig3 <- fig3 %>% layout(
  scene = list(
    xaxis = list(nticks = 20),
    zaxis = list(nticks = 4),
    camera = list(eye = list(x = 0, y = -1, z = 1)),
    aspectratio = list(x = .9, y = .8, z = 0.2)))


fig3
```



6.5. Análisis de resultados y recomendaciones.

Con los datos registrados de 90 semanas antes de pandemia en el modelo modificado (Ecuación 6.1) se obtiene un crecimiento positivo de las ganancias simuladas, en particular mayor que el capital inicial, es decir la probabilidad del evento donde $G(t) \geq u$ es positiva, con $t = 90$, como se puede observar en la imagen (Figura 6.5), por lo que en el intervalo de tiempo en el que se estudia la empresa generaría siempre ganancias positivas y se lograría suplir la deuda e incluso pensar en implementar más equipos inmobiliarios.

Mientras que después de la pandemia, se puede argumentar con base en las observaciones de las 14 semanas de las ventas registradas, usando los mismos parámetros obtenidos para los datos antes de la pandemia implementados, las ganancias son bajas como se muestra en la imagen (Figura 6.7). Aquí se observa que la simulación de las ganancias son menores que el capital inicial y al dar seguimiento en el intervalo de tiempo con los datos registrados después de pandemia hasta la semana 17 se observa que las ganancias son menores que cero, por lo que se puede concluir que se llega a una ruina total, como se muestra el caso en la imagen (Figura 6.8).

Analizando el caso en el que se modifica los parámetros de los costos del modelo (Ecuación 6.1) , se concluye que al calibrar el parámetro c, este se puede reducir a un 17% para cuando $t = 14$, es decir, al final de la semana 14 puede evitarse la perdida total de las ganancias que son mayores que el capital inicial, como se observa en la figura (**?@fig-fig-trayectoria4pdf**).

Como c, es el parámetro de los pagos que incluye el pago de la adquisición de la franquicia, compras de equipos inmobiliarios, pagos de empleados y prestamos, el anterior análisis quiere decir que se necesita un recorte de personal del 17% como mínimo y una reestructuración financiera del capital pendiente por pagar, haciendo los pagos más pequeños a un tiempo extendido, con el fin de la funcionalidad del negocio durante este proceso pandemico y evitar el cierre de quiebra.

Referencias

Castañeda, Liliana Blanco, Viswanathan Arunachalam, y Selvamuthu Dharmaraja. 2012. *Introduction to probability and stochastic processes with applications*. John Wiley & Sons.

Wickham, Hadley, Mine Çetinkaya-Rundel, y Garrett Grolemund. 2023. *R for data science*. " O'Reilly Media, Inc.".