

# Metodología\_1

Seguros

2026-01-04

## Abstract

En este reporte se presentan los códigos empleados para el cálculo y generación de elementos cuantitativos para LMR 2026

## Contents

<b>Introducción</b>	<b>1</b>
<b>Diversos Misceláneos</b>	<b>1</b>
Preparación del modelado . . . . .	1
Modelaje de la suma asegurada . . . . .	3
Orden estocástico . . . . .	3
Recuperación de la transformación original . . . . .	7
Prueba de escenarios . . . . .	8
Comparación grafica del modelo . . . . .	9
Prueba de estrés . . . . .	14
Análisis de sensibilidad . . . . .	14
<b>Diversos Técnicos</b>	<b>16</b>
Preparación del modelado . . . . .	16
Ajuste de los datos . . . . .	17
Ajuste estimador kernel . . . . .	17
Ajuste para encontrar función acotadora . . . . .	18
Aplicación método de la función inversa . . . . .	19
Prueba de escenarios . . . . .	24
Modelando frecuencia siniestral . . . . .	24
Enfoque Zinar(p) . . . . .	26
Enfoque Proyectivo . . . . .	27
Análisis de sensibilidad . . . . .	29
Análisis usando Modelo individual de riesgo . . . . .	29

# Introducción

El Límite Máximo de Retención (LMR) es un parámetro crítico en la gestión de riesgos de aseguradoras, ya que determina la exposición máxima que la compañía puede asumir sin comprometer su solvencia. Su cálculo debe considerar la normativa vigente, la estructura de reaseguro y la suficiencia de capital, garantizando la estabilidad financiera ante eventos adversos.

```
#Validación
length(Vigor$POL[which(!is.na(Vigor$POL))])
```

```
## [1] 611200
```

```
length(B$POL)
```

```
## [1] 611200
```

## Diversos Miscelaneos

### Preparación del modelado

En el siguiente código se aplica el tipo de cambio, y se construye la tabla P1 con los cuantiles de las sumas aseguradas.

```
for(i in 1:nrow(Mi)){
  if(Mi$MON == 20){
    Mi$SA[i] = Mi$SA[i]*18.3147
  }
}
Mi$SA[which(Mi$MON == 20)[4]]
```

```
## [1] NA
```

```
#####
#Creación de tablas
decil = seq(from = 0, to = 1, by = 1/10)
P1 = as.matrix(quantile(probs = decil, x = Mi$SA), nrow = 11, ncol = 1)
Q = P1
Q = data.frame(Q)
Q$categoría = data.frame(categoría = c("1","2","3","4","5","6","7",
                                         "8","9","10","11"))
Q$porcentaje = rep(.1, times = 11)
df = Q %>%
  group_by(categoría) %>%
  summarise(max(Q),Frecuencia = sum(porcentaje))
Q = cbind(as.numeric(df[1,2]),paste(as.character(as.numeric(df[1,2])), "-", as.character(as.numeric(df[2,2]))))
for(j in 2:11){
  P = cbind(as.numeric(df[j,2]),paste(as.character(as.numeric(df[j,2])), "-", as.character(as.numeric(df[j,2]))))
  Q = rbind(Q,P)
}
s = cbind(Q[-11,],df$Frecuencia[-1])
```

```

x = Mi$SA
x = length(x)
s1 = as.numeric(s[,3])
s1 = s1*x
sum(s1) == x

```

```
## [1] TRUE
```

```

s = cbind(s,s1)
s

```

```

##
##          [,1]           s1
## [1,] "25.21"    "25.21 - 1069000"    "0.1" "11019"
## [2,] "1069000"   "1069000 - 9.5e+07"   "0.1" "11019"
## [3,] "9.5e+07"   "9.5e+07 - 269.25"    "0.1" "11019"
## [4,] "269.25"    "269.25 - 3184.43"    "0.1" "11019"
## [5,] "3184.43"   "3184.43 - 5791.99"   "0.1" "11019"
## [6,] "5791.99"   "5791.99 - 13120.7"   "0.1" "11019"
## [7,] "13120.7"   "13120.7 - 24259.02"  "0.1" "11019"
## [8,] "24259.02"   "24259.02 - 108830.8" "0.1" "11019"
## [9,] "108830.8"  "108830.8 - 570000"   "0.1" "11019"
## [10,] "570000"    "570000 - 780000"    "0.1" "11019"

```

P1

```

##          [,1]
## 0%      25.21
## 10%     269.25
## 20%     3184.43
## 30%     5791.99
## 40%     13120.70
## 50%     24259.02
## 60%     108830.80
## 70%     570000.00
## 80%     780000.00
## 90%     1069000.00
## 100%    95000000.00

```

*#Variable más importante con la que se trabajará*  
C=Mi

## Modelaje de la suma asegurada

Sea  $Y$  una variable aleatoria la cual se desea modelar. Supongamos que podemos encontrar otra variable aleatoria  $X$  tal que casi seguramente  $Y \leq X$ . Esta propiedad se define como orden estocástico y se dice que  $Y$  es estocásticamente menor a  $X$  (en símbolos)  $Y \leq_{st} X$ .

El fundamento primordial de la prueba de Kolmogorov Smirnov es el lema de Glivenko-Cantelli que establece que la función de distribución empírica

$$F_n(x) = \frac{1}{n} \sum_{i=1}^n \chi_{(-\infty, x]}(x_i),$$

converge uniformemente en probabilidad a la distribución teórica propuesta, donde  $\{x_i\}_{i=1}^n$  es una muestra aleatoria de la variable aleatoria  $X$

$$F(x) = \mathbb{P}(X \leq x)$$

La hipótesis nula es que la distriución teórica de la muestra es algun modelo propuesto por el estadístico y se rechaza siempre que la distancia Kolmogorov Smirnov entre la distribución propuesta y la empírica bajo norma uniforme sea pequeña; es decir:

$$\lim_{n \rightarrow \infty} D_n = \lim_{n \rightarrow \infty} \sup_x |F_n(x) - F(x)| = 0.$$

Esta norma denotada por  $D_n$  es el estadístico de Kolmogorov Smirnov y fueron computados métodos de determinación de p-values por Kolmogorov y Smirnov, aunque puede relajarse la hipótesis alternativa a dos colas ( $|D_n| > p_\alpha$ ), cola izquierda ( $D_n < p_\alpha$ ) y cola derecha ( $D_n > p_\alpha$ ). En este caso como no se pudo ajustar en la prueba de dos colas, se hará uso de la prueba de cola izquierda.

Esta noción viene interrelacionada al concepto de orden estocástico:

### Orden estocástico

Para dos variables aleatorias  $X$  y  $Y$ ,  $X$  es estocásticamente menor a  $Y$  ( $X \leq_{st} Y$ ) si y sólo si

$$\bar{F}_X(x) = 1 - F_X(x) \leq 1 - F_Y(x) = \bar{F}_Y(x)$$

Para cada  $x \in \mathbb{R}$ .

En este caso la definición de prueba Kolmogorov Smirnov de cola derecha quiere decir que la distribución teórica propuesta, en este caso  $t - Student(n - 1)$  es estocásticamente menor a la distribución de nuestra suma asegurada transformada.

Una de las principales ventajas del orden estocástico es que se preserva tambien a modelos de pérdida agregada; esto quiere decir que si pudieramos simular una  $t - Student(n - 1)$  que es estocásticamente menor a nuestra suma asegurada y sumaramos varias reclamaciones identicamente distribuidas  $t - Student(n - 1)$ , la suma, aleatoria o no sería por ende también estocásticamente menor a la distribución desconocida de nuestra suma asegurada.

Se empieza con el modelado aplicando orden estocástico, barrido de atípicos y pruebas de cola derecha. Se aplica una transformación tipo Box Cox para realizar el ajuste. Para  $M$  la suma asegurada se define:

$$X = \log(M) - 6.64,$$

y se ajustará un modelo  $t - Student(m)$  para  $X$ .

```
#Diversos miscelaneos
set.seed(1234)
#####inicio
library(MASS)

#Solo tomamos sumas aseguradas mayores que 0
C = C%>%filter(SA >0)
#En la variale M capturamos la suma asegurada
M = C$SA
#Se aplica una transformación de Box Cox para suavizar los datos
#Posteriormente esto puede deshacerse
M = .5*log(M)-6.2
#Se calculan los whisklers del box plot y se quitan outliers
l = boxplot.stats(M)$stats
M= M[M >= l[1] & M <= l[5]]
```

```

#Se calcula por máxima verosimilitud los degree freedom de una t-student
ajuste = fitdistr(M, densfun = "t")
ajuste

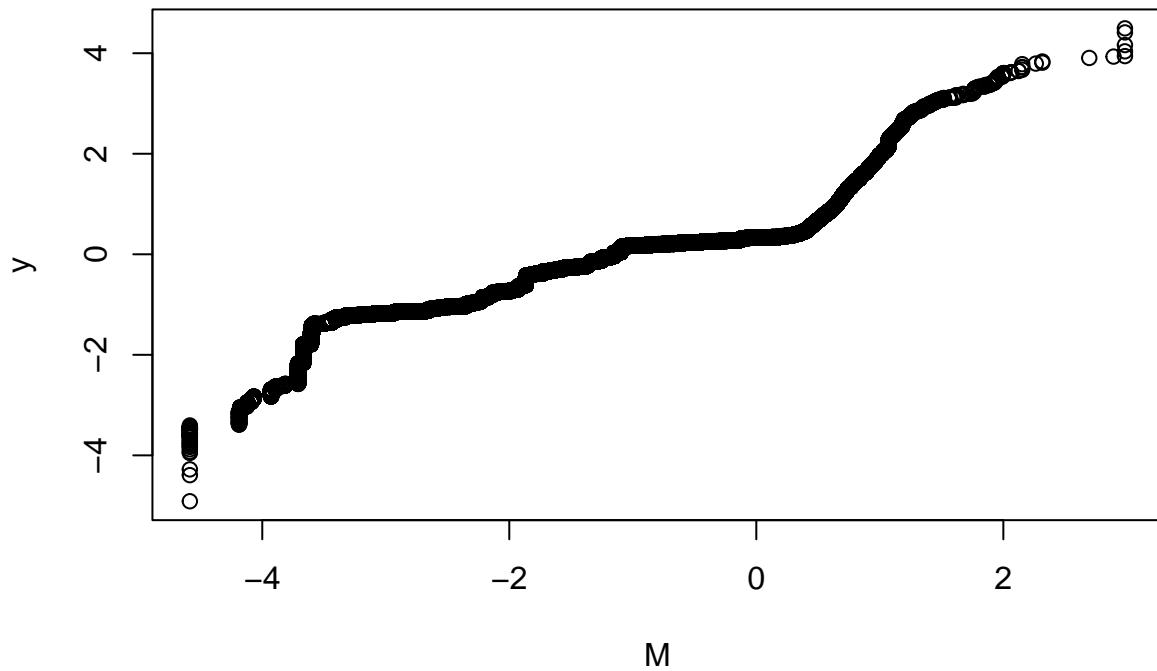
##           m             s            df
## -0.990414297   1.464497735 177.623698347
## ( 0.004437966) ( 0.003143511) ( 9.686489449)

#Se guardan los degree freedom en la variable m
m = ajuste$estimate["df"]
#Se simulan 135502 corridas de t-student con esos degree freedom's
set.seed(1234)
y <- rt(length(M),m)
#Se comparan nuestros datos con la simulación
#La hipótesis alternativa es que la función de distribución o de probabilidad
#acumulada empírica queda por encima de
#la distribución t-student con esos grados de libertad y la hipótesis nula
#es que nuestros datos se distribuyan igual que la simulación.
ks.test(M,y, alternative = "less")

## 
##  Asymptotic two-sample Kolmogorov-Smirnov test
##
##  data: M and y
##  D^- = 9.0752e-06, p-value = 1
##  alternative hypothesis: the CDF of x lies below that of y

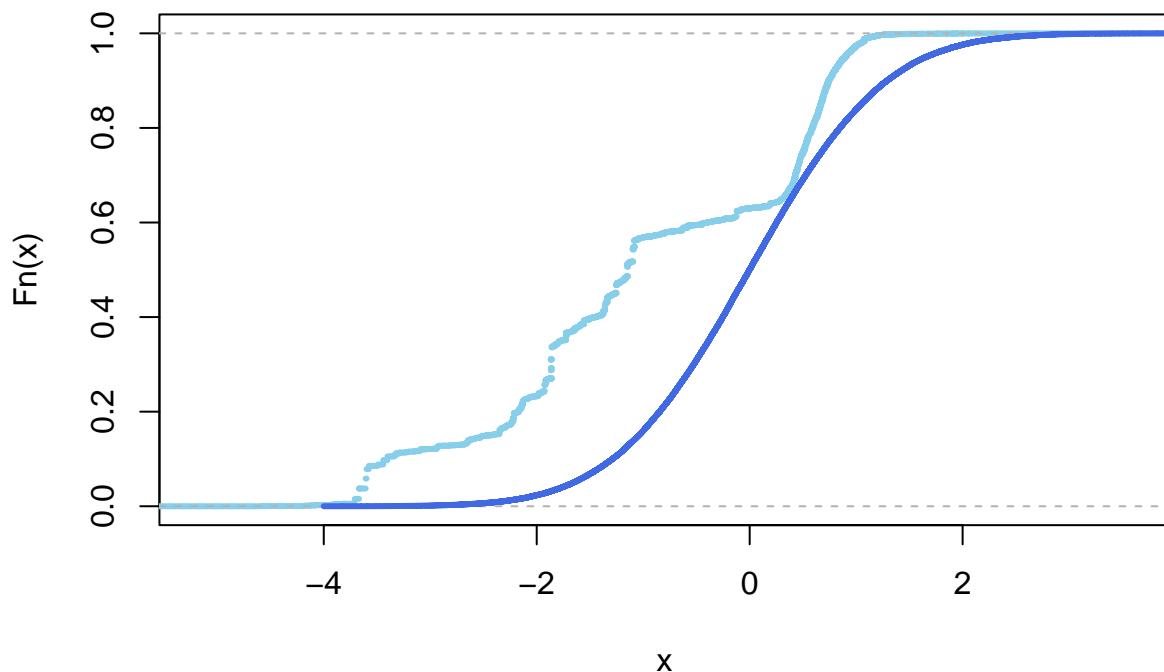
#El qqplot confirma nuestro análisis
qqplot(M,y)

```



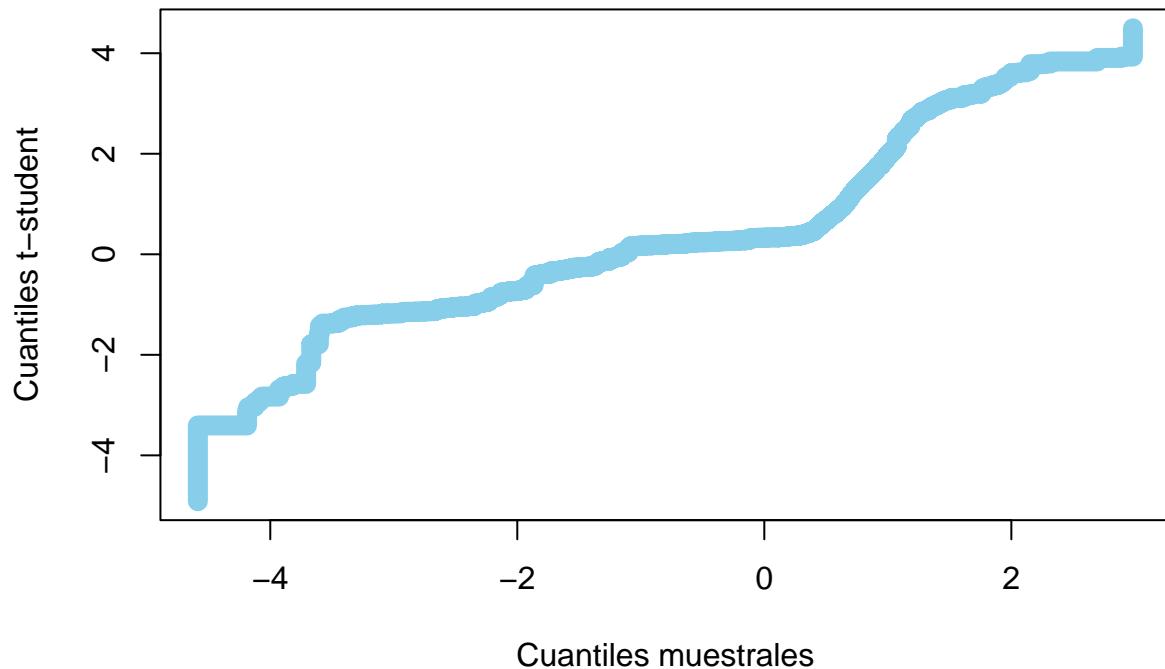
```
plot(ecdf(M), col = "skyblue", lwd = 3)
points(seq(from = -4, to=15,by=1/1000),ecdf(y)(seq(from = -4, to=15,by=1/1000))
      ,lwd = 3,type = "s", col = "royalblue")
```

### ecdf(M)



```
qqplot(M,y,col = "skyblue", ylab = "Cuantiles t-student",
       xlab = "Cuantiles muestrales", main = "Gráfico Q-Q",type = "s",lwd = 10,)
```

## Gráfico Q-Q



Aquí  $m$  toma el valor

$m$

```
##      df
## 177.6237
```

Claramente en este caso se presenta el hecho de que

$$SA \leq_{st} X$$

Claramente se desconoce la distribución de la suma asegurada; únicamente sabemos que  $t\text{-Student} \leq F_n(x) \leftarrow F_{SA}(x)$ . Esto es equivalente a que casi seguramente en probabilidad  $SA \leq X$  con  $X \sim t\text{-Student}$ . Por ende al restar el límite máximo de retención  $t\text{-Student} \geq F_{SA}(x)$ . Ahora basta mostrar que  $LMR \geq t\text{-Student} \geq F_{SA}(x)$  en menos del 5% de los escenarios únicamente.

### Recuperación de la transformación original

Al haberse aplicado la transformación

$$X = .5 * \log(M) - 6.2$$

Destransformando, queda:

$$\exp\left(\frac{M + 6.2}{.5}\right) = M.$$

## Prueba de escenarios

```
#####
##Prueba de escenarios
set.seed(1234)
longitud = 100000
z = rt(longitud,df=m)
z = (z+6.2)/.5
z = exp(z)
LMR = 12000000
longitud1 = length(z)
z1 = z-LMR
length(which(z1>0))/longitud1

## [1] 0.02692

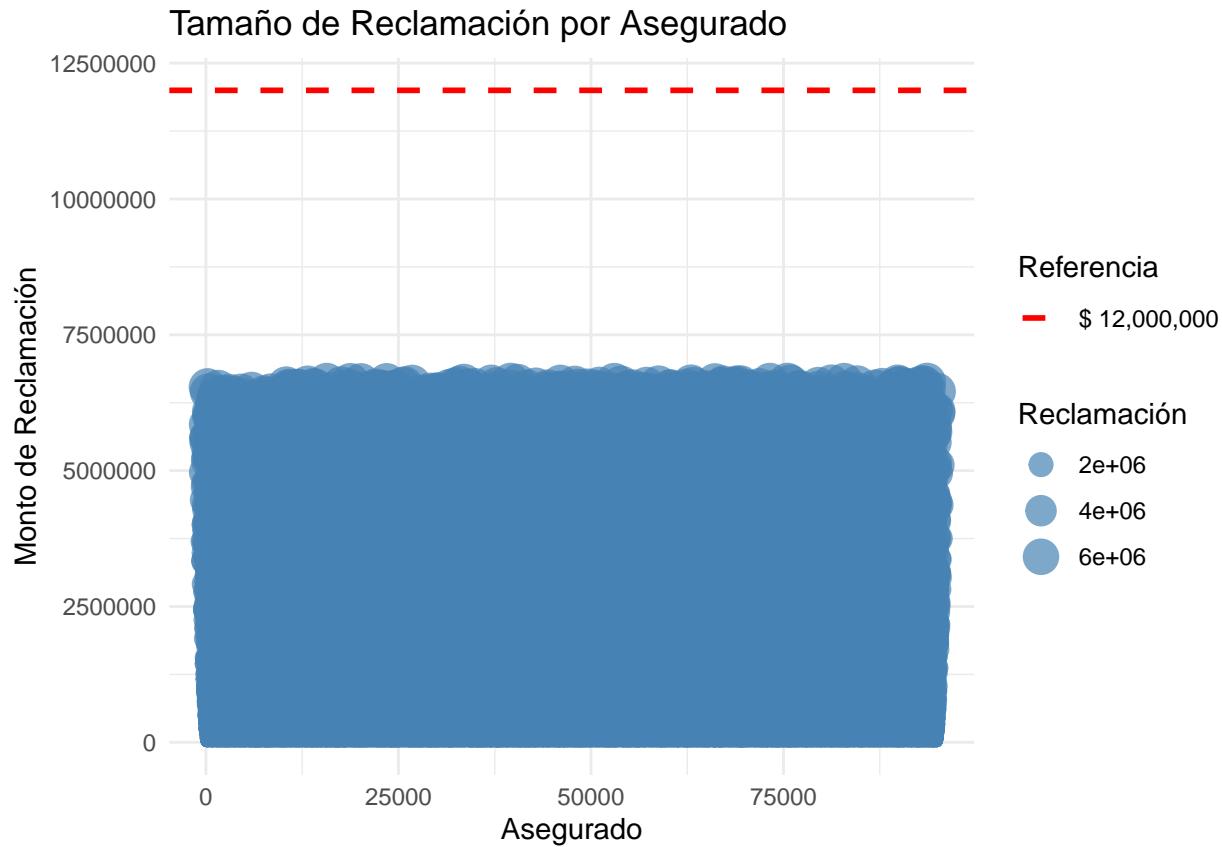
summary(z1)

##      Min.    1st Qu.     Median      Mean    3rd Qu.      Max.
## -11999987 -11937801 -11759492 -10130326 -11059160 1934679143
```

Nuevamente y observando que la distribución desconocida de las sumas aseguradas están acotadas casi seguramente por la distribución t destransformada de la forma siguiente, claramente las sumas aseguradas serán menores al límite máximo propuesto.

```
l = boxplot.stats(z)$stats
z2= z[ z <= quantile(prob = .95,z)]
H = data.frame(asignado=c(1:length(z2)), reclamacion = z2)
cutoff =12000000
ggplot(H, aes(x = asignado, y = reclamacion, size = reclamacion)) +
  geom_point(color = "steelblue", alpha = 0.7) +
  geom_hline(aes(yintercept = cutoff, color = "$ 12,000,000"), linetype = "dashed", size = 1) +
  scale_color_manual(values = c("$ 12,000,000" = "red")) +
  labs(color = "Referencia") +
  labs(title = "Tamaño de Reclamación por Asegurado",
       x = "Asegurado",
       y = "Monto de Reclamación",
       size = "Reclamación") +
  theme_minimal()

## Warning: Using 'size' aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use 'linewidth' instead.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.
```



### Comparación grafica del modelo

En lo que sigue se realiza un análisis comparativo con tres modelos representativos y se descartan los mismos pues sus p-values son inaceptables ya que de los mismos se rechaza la hipótesis de las distribuciones.

Lognormal, normal y gamma son los modelos probados, dos distribuciones sesgadas tipo colas pesadas, mesocurtica y leptocurtica. Se analiza la bondad de ajuste

```
e = fitdistr(Mi$SA,"lognormal")
x1 = e$estimate[1]
w1= e$estimate[2]
y1 = rlnorm(length(Mi$SA),x1,w1)
k = ks.test(y1,Mi$SA)

## Warning in ks.test.default(y1, Mi$SA): p-value will be approximate in the
## presence of ties
```

```
# Diversos Misceláneos
set.seed(1234)
e = fitdistr(Mi$SA,"normal")
x2 = e$estimate[1]
w2= e$estimate[2]
y2 = rnorm(length(Mi$SA),x2,w2)
k1 = ks.test(y2,Mi$SA)
```

```

## Warning in ks.test.default(y2, Mi$SA): p-value will be approximate in the
## presence of ties

set.seed(1234)
y3 = rgamma(length(Mi$SA), rate = 1/(var(Mi$SA)/mean(Mi$SA)), shape = (mean(Mi$SA))^2/var(Mi$SA))
k2 = ks.test(y3,Mi$SA)

## Warning in ks.test.default(y3, Mi$SA): p-value will be approximate in the
## presence of ties

c(k$statistic,k$p.value)

##          D
## 0.163454 0.000000

c(k1$statistic,k1$p.value)

##          D
## 0.3423813 0.0000000

c(k2$statistic,k2$p.value)

##          D
## 0.2241038 0.0000000

#Cramer Von Mises
library(goftest)
# Diversos Misceláneos
cv = cvm.test(Mi$SA, null = "plnorm", meanlog = x1, sdlog = w1)

## 
## Cramer-von Mises test of goodness-of-fit
## Null hypothesis: log-normal distribution
## with parameters meanlog = 10.414904027357, sdlog = 2.93638473721471
## Parameters assumed to be fixed
##
## data: Mi$SA
## omega2 = 436.75, p-value < 2.2e-16

# Diversos Misceláneos
cv1 = cvm.test(Mi$SA, null = "norm", mean = x2, sd = w2)
cv1

## 
## Cramer-von Mises test of goodness-of-fit
## Null hypothesis: Normal distribution
## with parameters mean = 377199.581094217, sd = 943541.084871166
## Parameters assumed to be fixed
##
## data: Mi$SA
## omega2 = 2741.9, p-value < 2.2e-16

```

```

# Diversos Misceláneos
cv2 = cvm.test(Mi$SA, null = "gamma", rate = 1/(var(Mi$SA)/mean(Mi$SA)), shape = (mean(Mi$SA))^2/var(Mi$SA))
cv2

##
## Cramer-von Mises test of goodness-of-fit
## Null hypothesis: Gamma distribution
## with parameters rate = 4.23687478655526e-07, shape = 0.15981473946373
## Parameters assumed to be fixed
##
## data: Mi$SA
## omega2 = 1391.1, p-value < 2.2e-16

c(cv$statistic, cv$p.value)

##     omega2
## 436.7515  0.0000

c(cv1$statistic, cv1$p.value)

##     omega2
## 2741.868   0.000

c(cv2$statistic, cv2$p.value)

##     omega2
## 1391.114   0.000

#####Anderson Darling
#Accidentes personales
ad = ad.test(Mi$SA, "plnorm", meanlog = mean(Mi$SA), sdlog = sd(Mi$SA))

#Accidentes personales
ad1 = ad.test(Mi$SA, "norm", mean = mean(Mi$SA), sd = sd(Mi$SA))

#Accidentes personales
ad2 = ad.test(Mi$SA, "gamma", rate = 1/(var(Mi$SA)/mean(Mi$SA)), shape = (mean(Mi$SA))^2/var(Mi$SA))

c(ad$statistic, ad$p.value)

##          An
## 5.374898e+04 5.445140e-09

c(ad1$statistic, ad1$p.value)

##          An
## Inf 5.44514e-09

```

```

c(ad2$statistic,ad2$p.value)

##          An
##      Inf 5.44514e-09

t(t(summary(Mi$SA)))

##      [,1]
## Min.       25
## 1st Qu.    5148
## Median     24259
## Mean      377200
## 3rd Qu.   661000
## Max.    95000000

Mi[which(Mi$SA == max(Mi$SA)),]$POL

## [1] 41197 297728 415753 457253 523362

#### Graficos
##1
B_1 = ecdf(Mi$SA)(seq(from=0, to=4500000, 4000))
B_2 = ecdf(y1)(seq(from=0, to=4500000, 4000))
C_2 = ecdf(y2)(seq(from=0, to=4500000, 4000))
D_2 = ecdf(y3)(seq(from=0, to=4500000, 4000))
# Crear data frames para cada comparación
pp_AB <- qq_data(B_1, B_2, "SA vs lognormal")
pp_AC <- qq_data(B_1, C_2, "SA vs normal")
pp_AD <- qq_data(B_1, D_2, "SA vs gamma")

# Unir todos los datos
pp_all <- bind_rows(pp_AB, pp_AC, pp_AD)

# Graficar con ggplot2
pp <- ggplot(pp_all, aes(x = x, y = y, color = grupo)) +
  geom_point() + geom_step() + coord_cartesian(ylim = c(0, 1)) +
  geom_abline(slope = 1, intercept = 0, linetype = "solid") +
  labs(title = "PP-Plot", x = "Probabilidades de SA", y = "Probabilidades teóricas") +
  theme_minimal()

```

Ahora se comparan las frecuencias y distribuciones empíricas para observar como a pesar de existir un cierto parecido las pruebas indican rechazar las hipótesis de igualdad de distribución.

```

#####2
par(mfrow = c(1,2))
plot(ecdf(Mi$SA), main = "CDF empírica y teórica")
lines(ecdf(y1), col = "turquoise")
lines(ecdf(y2), col = "orange")
lines(ecdf(y3), col = "blue")
z = Mi$SA
l1 = boxplot.stats(z)$stats

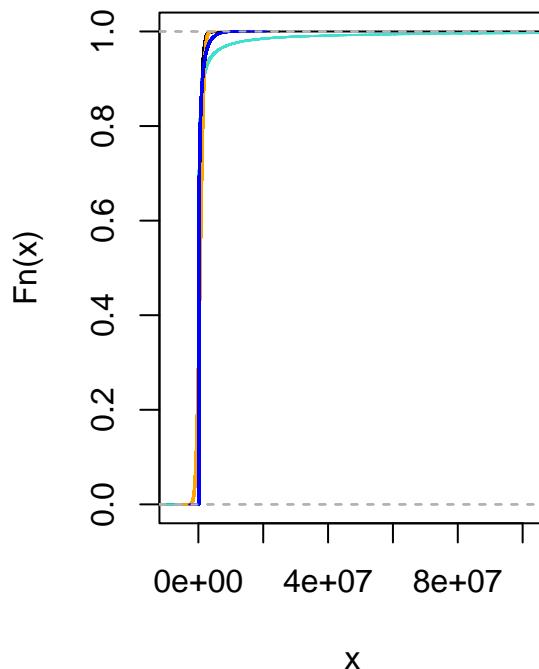
```

```

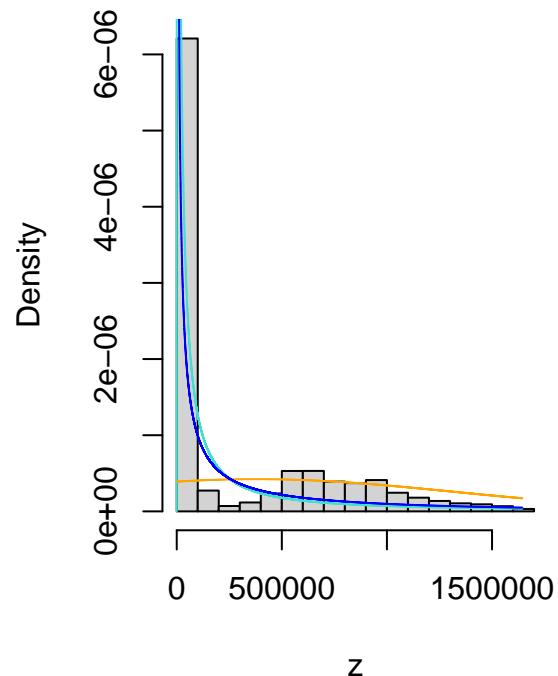
z= z[z >= 11[1] & z <= 11[5]]
hist(z,freq = FALSE,main = "Histograma y densidades teóricas")
d = seq(from = 0, to = max(z), by =1)
lines(d, dlnorm(d,x1,w1), col = "turquoise")
lines(d, dnorm(d,x2,w2), col = "orange")
lines(d, dgamma(d,rate = 1/(var(Mi$SA)/mean(Mi$SA)),shape = (mean(Mi$SA))^2/var(Mi$SA)), col = "blue")

```

**CDF empírica y teórica**



**Histograma y densidades teóricas**



```

A_1 = Mi$SA
B = y1
C = y2
D = y3
# Crear data frames para cada comparación
qq_AB <- qq_data(A_1, B, "SA vs lognormal")
qq_AC <- qq_data(A_1, C, "SA vs normal")
qq_AD <- qq_data(A_1, D, "SA vs gamma")

# Unir todos los datos
qq_all <- bind_rows(qq_AB, qq_AC, qq_AD)

# Graficar con ggplot2
p <- ggplot(qq_all, aes(x = x, y = y, color = grupo)) +
  geom_point() + geom_step() + coord_cartesian(ylim = c(0, 4500000)) +
  geom_abline(slope = 1, intercept = 0, linetype = "solid") +
  labs(title = "QQ-Plot", x = "Cuantiles de SA", y = "Cuantiles teóricos") +
  theme_minimal()

```

```

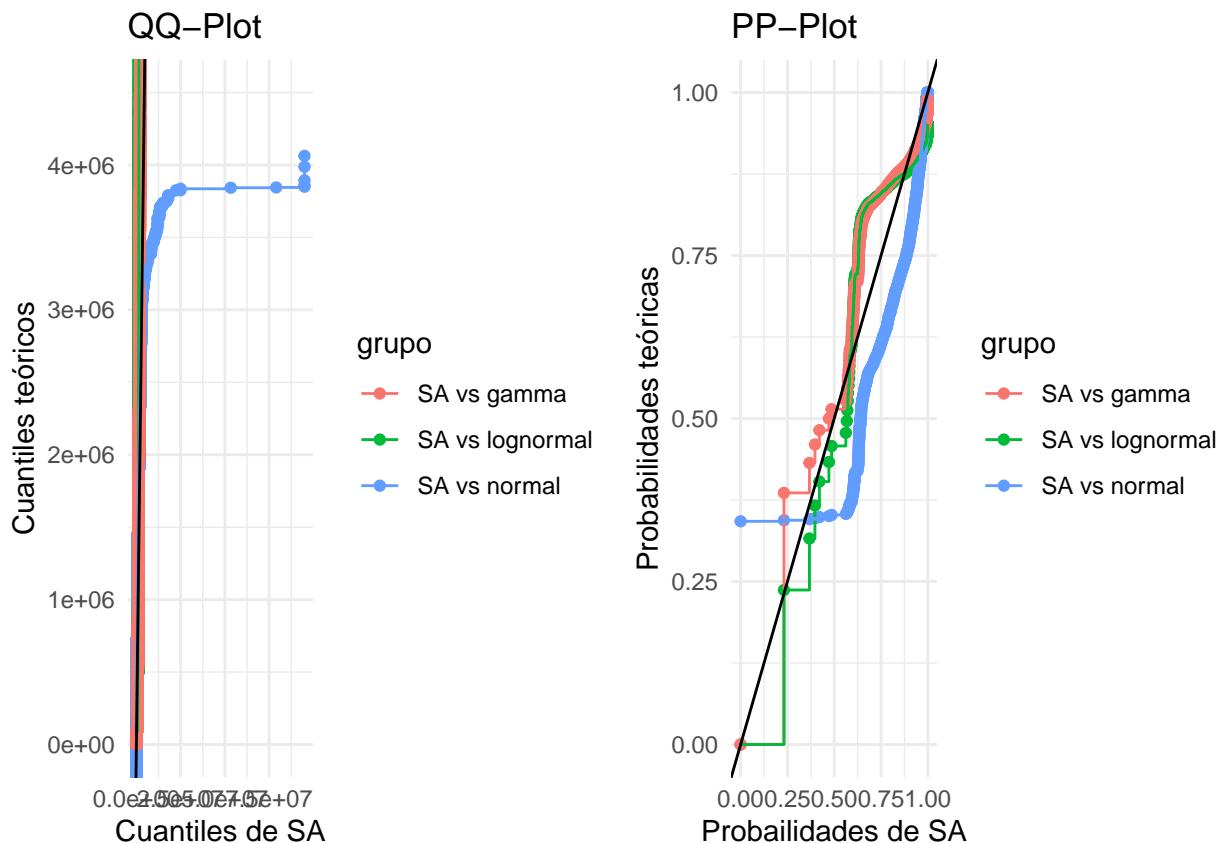
par(mfrow = c(1,1))
library(gridExtra)

##
## Adjuntando el paquete: 'gridExtra'

## The following object is masked from 'package:dplyr':
##     combine

grid.arrange(p, pp, ncol = 2)

```



## Prueba de estrés

La simulación adquiere las siguientes características

## Análisis de sensibilidad

```

#Sensibilidad
prueba = function(i,LMR,k){
  k = (1+i)*k
  z1 = k-LMR

```

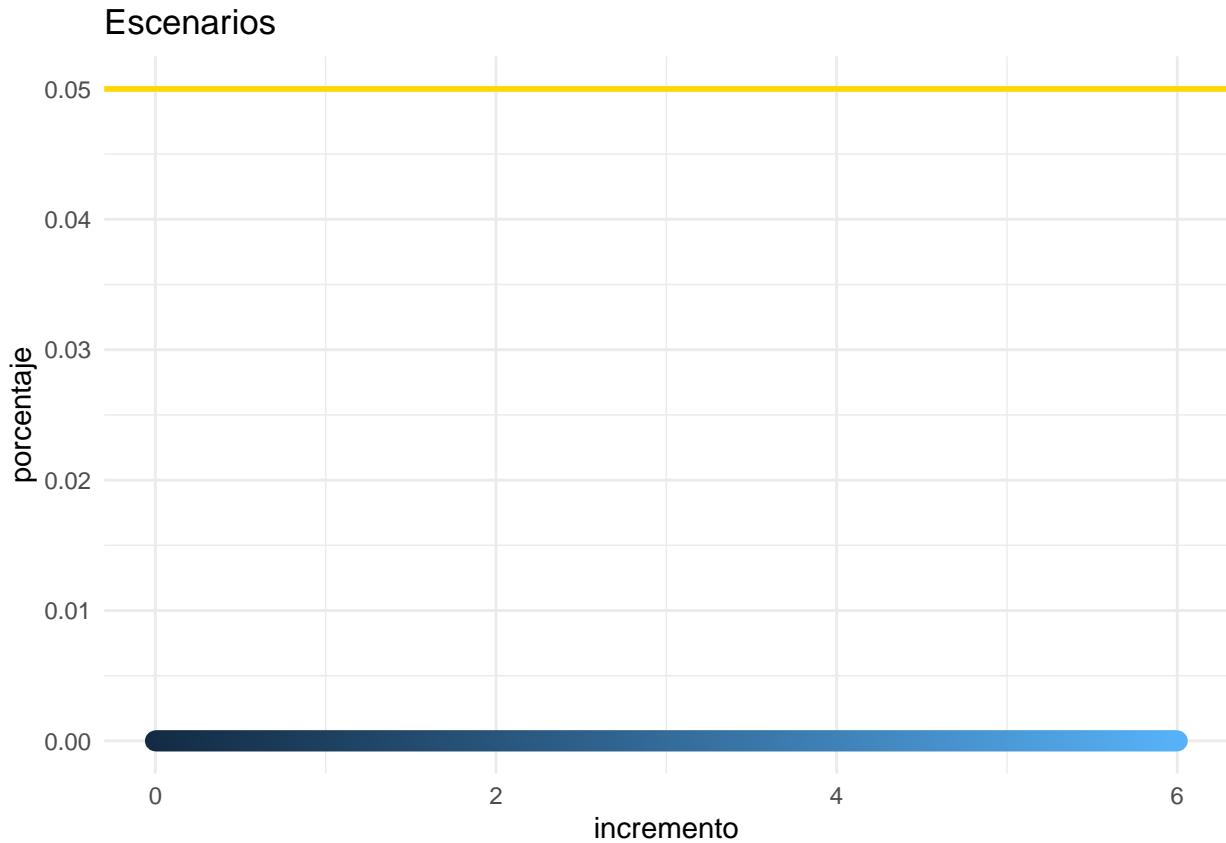
```

f = length(which(z1>0))/length(z1)
return(f)
}
library(ggplot2)
max(z)

## [1] 1644000

LMR = 12000000
d2 = seq ( from = 0, to = 6, by = 1/ 100)
escenarios = sapply (d2 , prueba , LMR = LMR, k=z )
# Datos de ejemplo
sensibilidad <- data.frame(
  incremento = d2,
  porcentaje = escenarios,
  categoria = c(1:length(escenarios))
)
# Scatter plot
ggplot(sensibilidad, aes(x = incremento, y = porcentaje, color = categoria)) +
  geom_point(size = 3) +
  labs(title = "Escenarios") +
  theme_minimal() + theme(legend.position = "none") +
  geom_hline(yintercept = .05, color = "gold", linetype = "solid", size = 1)

```



# Diversos Técnicos

## Preparación del modelado

En el siguiente código se aplica el tipo de cambio, y se construye la tabla P1 con los cuantiles de las sumas aseguradas.

Se ocupa el inciso 1 de la metodología basada en el algoritmo de simulación estocástica **Aceptación y Rechazo**

Las cuentas en dólares se convierten en dólares.

Se puede ver la estructura de los datos con los cuantiles. Los cuantiles permiten observar los datos a modelar ordenados y mostrando las frecuencias.

```
decil = seq(from = 0, to = 1, by = 1/10)
P1 = as.matrix(quantile(probs = decil, x = Te$SA), nrow = 11, ncol = 1)
Q = P1
Q = data.frame(Q)
Q$categoría = data.frame(categoría = c("1","2","3","4","5","6","7","8","9","10","11"))
Q$porcentaje = rep(.1, times = 11)
df = Q %>%
  group_by(categoría) %>%
  summarise(max(Q), Frecuencia = sum(porcentaje))
Q = cbind(as.numeric(df[1,2]), paste(as.character(as.numeric(df[1,2])), "-", as.character(as.numeric(df[2,2])), "-"))
for(j in 2:11){
  P = cbind(as.numeric(df[j,2]), paste(as.character(as.numeric(df[j,2])), "-", as.character(as.numeric(df[j,2])), "-"))
  Q = rbind(Q,P)
}
s = cbind(Q[-11,], df$Frecuencia[-1])
x = Te$SA
x = length(x)
s1 = as.numeric(s[,3])
s1 = s1*x
sum(s1) == x

## [1] TRUE

s = cbind(s,s1)
s

##                                     s1
## [1,] "25.21"      "25.21 - 1075000"    "0.1" "30050"
## [2,] "1075000"    "1075000 - 9.5e+07"   "0.1" "30050"
## [3,] "9.5e+07"    "9.5e+07 - 269.25"    "0.1" "30050"
## [4,] "269.25"     "269.25 - 3184.43"   "0.1" "30050"
## [5,] "3184.43"    "3184.43 - 5791.99"   "0.1" "30050"
## [6,] "5791.99"    "5791.99 - 13654.41"  "0.1" "30050"
## [7,] "13654.41"   "13654.41 - 24259.02" "0.1" "30050"
## [8,] "24259.02"   "24259.02 - 121498.93" "0.1" "30050"
## [9,] "121498.93"  "121498.93 - 573000"   "0.1" "30050"
## [10,] "573000"     "573000 - 794000"     "0.1" "30050"
```

```
## [,1]
## 0%    25.21
## 10%   269.25
## 20%   3184.43
## 30%   5791.99
## 40%   13654.41
## 50%   24259.02
## 60%   121498.93
## 70%   573000.00
## 80%   794000.00
## 90%   1075000.00
## 100%  95000000.00
```

Se actualiza la base para diversos técnicos.

### Ajuste de los datos

Dentro de la metodología se expondrá el inciso 2, **Aceptación y Rechazo**.

```
#Diversos tecnicos
#####inicio
library(MASS)
#Solo tomamos sumas aseguradas mayores que 0
C = C%>%filter(SA >0)
#En la variale M capturamos la suma asegurada
M = C$SA
#Se aplica una transformación Box Plot
M = log(M)-6.461
#Limpieza de outliers vía Box Plot
l = boxplot.stats(M)$stats
M= M[M >= l[1] & M <= l[5]]
```

### Ajuste estimador kernel

El kernel es el núcleo y consiste en una distribución con el mismo soporte que la distriución a modelar.

Primeramente se sabe por el teorema de lema de Glivenko-Cantelli que establece que la función de distribución empírica

$$F_n(x) = \frac{1}{n} \sum_{i=1}^n \chi_{(-\infty, x]}(x_i),$$

converge uniformemente en probabilidad a la distribución teórica propuesta, donde  $\{x_i\}_{i=1}^n$  es una muestra aleatoria de la variable aleatoria  $X$

$$F(x) = \mathbb{P}(X \leq x)$$

A pesar de ser una técnica usada ampliamente en las aplicaciones, tiene la limitante de que la distribución empírica no necesariamente es absolutamente continua, es decir, puede no tener función de densidad.

El método de aproximación Kernel pretende dada una muestra aleatoria  $\{x_i\}_{i=1}^n$ , definir una función  $\hat{f}_d(x)$  que satisface ser la densidad de

$$\hat{F}_n(x) = \int_{(-\infty, x)} \text{Kernel}\left(\frac{x - x_i}{h}\right) dF_n(x)$$

Al ser esta una convolución puede calcularse  $\int f * g = \int f \int g = \int f$  si  $g$  es función de densidad; en este caso si el kernel se escoge como una densidad digamos normal estandar, entonces esto se satisfacera y utilizando la media muestral entonces podrá estimarse la densidad buscada.

```
set.seed(1234)
#Kernel normal
cociente = function(x,h,M){
  return((x-M)/h)
}
kernel1= function(x,h,M){
  sumando = sum(dnorm(cociente(x,h,M),mean = 0, sd=1))
  return(sumando/(length(M)*h))
}
#La constante de escalamiento
h = bw.nrd0(M)
```

## Ajuste para encontrar función acotadora

Se define un cierto dominio para verificar que el Kernel usado es función de densidad. Se observa por inspección que la distribución Cauchy multiplicada por cierta constante acota por arriba a la densidad objetivo estimada con el método Kernel. Se usa la regla de Silverman ( $h$ ) para seleccionar el **Bandwidth**.

```
#Límites del eje x
a=min(M)
b=max(M)
dominio = seq(from = a, to = b, length.out= 1000)
#Densidad suavizada objetivo a modelar
#Aplicar el kernel al dominio
ya = sapply(dominio, FUN = kernel1,h=h,M=M)
#Densidad que acota por arriba a la densidad objetivo
#Aplicar densidad Cauchy al dominio
ye = sapply(dominio, FUN = dcauchy,location = 6, scale =5.9)
ye = 9*ye
```

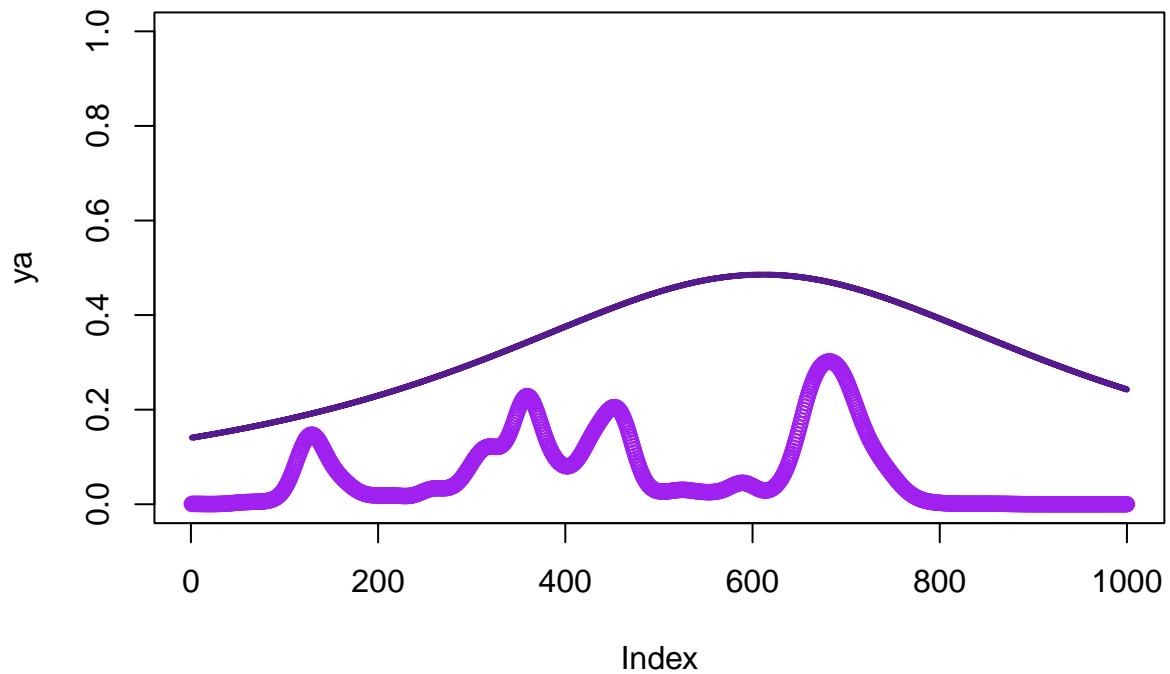
Se verifica que es una función de densidad

```
sum(ya)*diff(dominio)[1]
```

```
## [1] 0.9998075
```

Ahora se observa la aplicación del método de la función inversa conforme al inciso 2 de la metodología:

```
#curve(dlnorm(x,m,m1))
plot(ya, ylim=c(0,1), col="purple")
lines(ye,lwd = 3,type = "s", col = "purple4")
```



## Aplicación método de la función inversa

En aplicación del método de la función inversa pueden prepararse las variables insumo.

```
M = C$SA
M = log(M)-6.461
l = boxplot.stats(M)$stats
M= M[M >= l[1] & M <= l[5]]
h = bw.nrd0(M)
ter = h*length(M)
```

Definimos la función de densidad objetivo a modelar.

```
Objetivo = function(x,M1){
cociente =(x-M1)/h
  sumando = sum(dnorm(cociente,mean = 0, sd=1))
  return(sumando/(length(M1)*h))
}
```

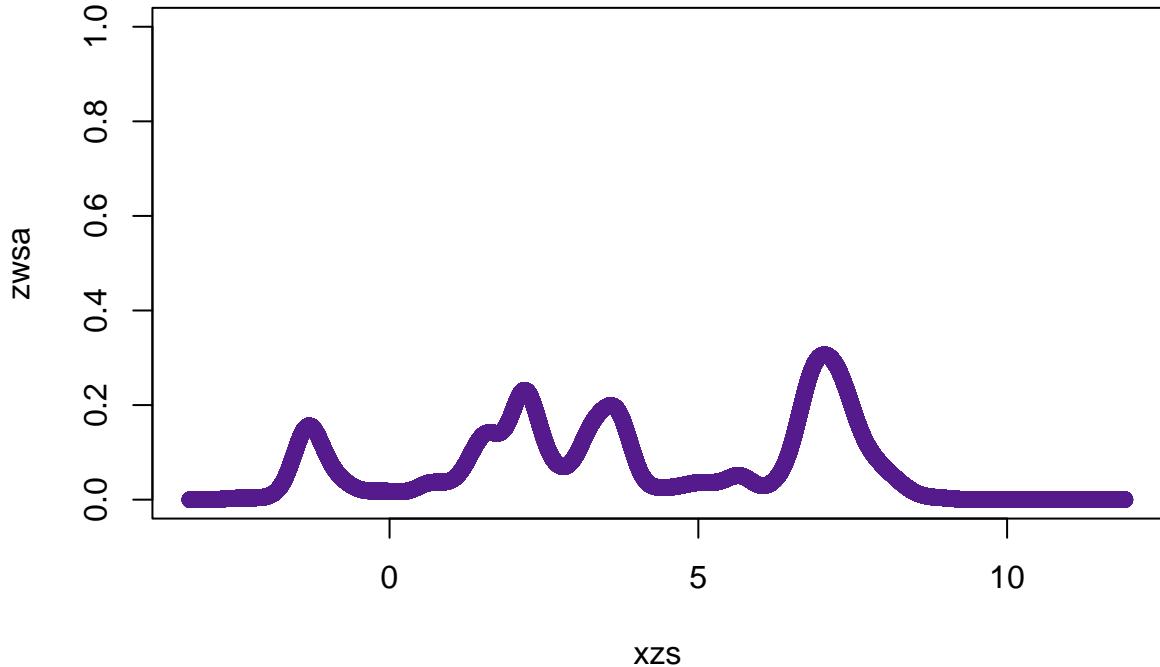
Se definen los parametros para observar gráficamente el comportamiento.

```
a = min(M)
b=max(M)
xzs = seq(from =a, to =b, length.out = 100000)
```

```

kwe =seq(from =1, to  =length(M), length.out = 2000)
M1 = M[kwe]
zwsa =sapply(xzs, Objetivo, M1)
plot(xzs,zwsa,ylim = c(0,1), col = "purple4")

```



Se generan números pseudoaleatorios Cauchy.

```

set.seed(1234)
longitud = 100000
u =c(1:longitud)
c = rcauchy(longitud,location = 6, scale =5.9)
c = c[c>0]
SA = list()

```

Una vez generados estos números pseudoaleatorios, se generan vectores con valuaciones  $U \sim \text{unif}(0, 9 \cdot f_{\text{cauchy}}(X))$ , aquí  $X \sim \text{Cauchy}$ .

```

for(i in 1:longitud){
  cota = dcauchy(c[i],location = 6, scale =5.9)
  cota = 9*cota
  u[i] = runif(1, 0, cota)
}

```

Ahora se generaran las corridas de la función objetivo  $\hat{f}_d(X)$

```

simulac = list()
M = C$SA
M = log(M)-6.461
l = boxplot.stats(M)$stats
M= M[M >= l[1] & M <= l[5]]
h = bw.nrd0(M)

```

Se generan las mallas de valores para valuar el Kernel.

```

kwe =seq(from =1, to =length(M), length.out = 2000)
M1 = M[kwe]
ter = h*length(M)

```

Se generan las valuaciones  $\hat{f}_d(X) = E[Kernel(\frac{X-M}{h})]$ .

```

for(i in 1:length(c)){
cociente = (c[i]-M1)/h
sumando = sum(dnorm(cociente,mean = 0, sd=1))
simulac[[i]] = sumando/ter
}
simulac = unlist(simulac)

```

Se aplica el algoritmo de la función inversa.

```

for(i in 1:length(c)){
  if(u[i] <= simulac[i]){
    SA[[i]] = c[i]
  }
}
SA = unlist(SA)

```

Se deshacen las transformaciones  $\log(M) - 6.461$  y se valida con bondad de ajuste.

```

Fer= C$SA
l = boxplot.stats(Fer)$stats
Fer= Fer[Fer >= l[1] & Fer <= l[5]]
z = exp(SA + 6.461)
ks.test(z,Fer)

```

```

## Warning in ks.test.default(z, Fer): p-value will be approximate in the presence
## of ties

##
##  Asymptotic two-sample Kolmogorov-Smirnov test
##
## data: z and Fer
## D = 0.14977, p-value = 0.2651
## alternative hypothesis: two-sided

```

Se valida ahora con datos transformados:

```

ks.test(M,SA)

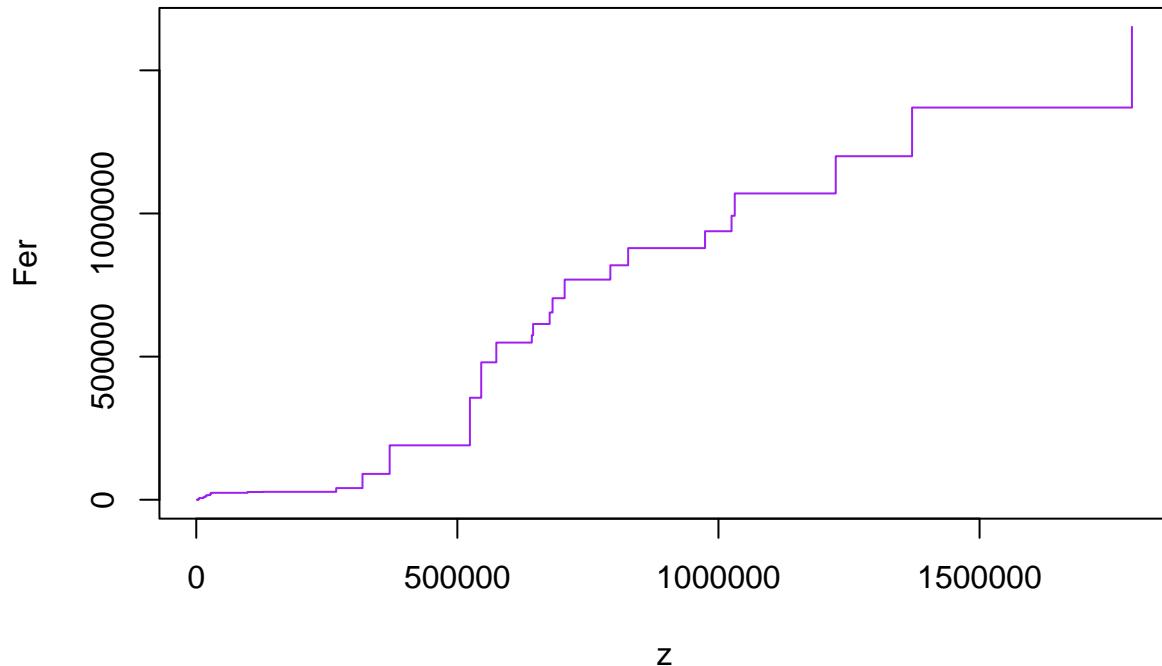
## Warning in ks.test.default(M, SA): p-value will be approximate in the presence
## of ties

##
## Asymptotic two-sample Kolmogorov-Smirnov test
##
## data: M and SA
## D = 0.14219, p-value = 0.3229
## alternative hypothesis: two-sided

```

Se observa el gráfico cuantil-cuantil

```
qqplot(z,Fer, col = "purple", type = "s")
```



Observemos la estructura de la simulación

```

decil = seq(from = 0, to = 1, by = 1/10)
P11 = as.matrix(quantile(probs = decil, x = z), nrow = 11, ncol = 1)
Q11 = P11
Q11 = data.frame(Q11)

```

P1

```
## [,1]
## 0%    25.21
## 10%   269.25
## 20%  3184.43
## 30% 5791.99
## 40% 13654.41
## 50% 24259.02
## 60% 121498.93
## 70% 573000.00
## 80% 794000.00
## 90% 1075000.00
## 100% 95000000.00
```

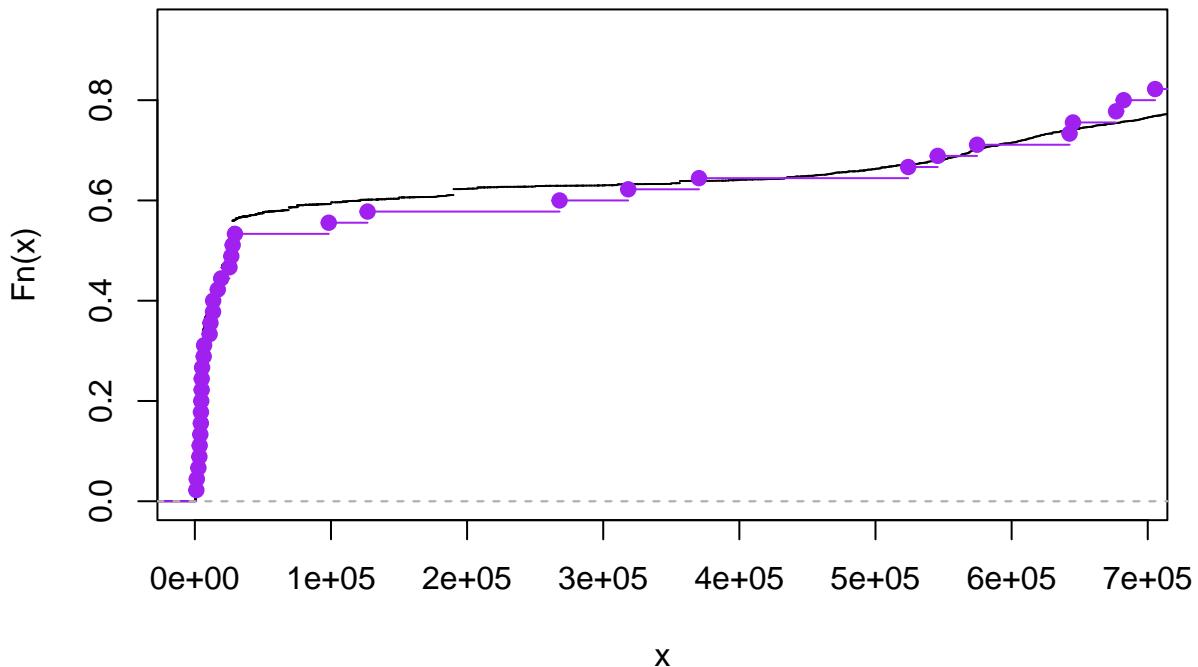
P11

```
## [,1]
## 0%    1085.038
## 10%   3739.092
## 20%  4954.027
## 30% 7612.417
## 40% 15490.599
## 50% 27789.318
## 60% 288086.359
## 70% 568824.107
## 80% 686922.536
## 90% 1004754.893
## 100% 1791838.667
```

En gráficos

```
plot(ecdf(Te$SA), main = "CDF empírica y teórica", xlim = c(0,686922.536))
lines(ecdf(z), col = "purple")
```

## CDF empírica y teórica



### Prueba de escenarios

```
#####
##Prueba de escenarios
LMR = 12000000
longitud1 = length(z)
z1 = z-LMR
length(which(z1>0))/longitud1

## [1] 0

summary(z1)

##      Min.    1st Qu.     Median      Mean    3rd Qu.      Max.
## -11998915 -11994675 -11972211 -11656241 -11354945 -10208161
```

### Modelando frecuencia siniestral

En el archivo anexo al presente trabajo puede obtenerse la frecuencia siniestral en el ejercicio 2025: Como insumo

```

library("readr")
library("stats")
library("tseries")

## Registered S3 method overwritten by 'quantmod':
##   method           from
##   as.zoo.data.frame zoo

library("nortest")

##
## Adjuntando el paquete: 'nortest'

## The following objects are masked from 'package:goftest':
##   ad.test, cvm.test

library("FinTS")

## Cargando paquete requerido: zoo

##
## Adjuntando el paquete: 'zoo'

## The following objects are masked from 'package:base':
##   as.Date, as.Date.numeric

library("ZINARp")

serie_tiempo <- read_csv("C:/Users/agarciadeleon/R_Studio/Proyectos_R/TS/serie_tiempo.csv")

## Rows: 48 Columns: 1

## -- Column specification -----
## Delimiter: ","
## dbl (1): S
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.

serie = ts(serie_tiempo$S, start=c(2022, 1), end = c(2025, 9), frequency=12)

```

Como insumo

```

library("readr")
library("stats")
library("tseries")
library("nortest")
library("FinTS")
library("ZINARp")

serie_tiempo <- read_csv("C:/Users/agarciadeleon/R_Studio/Proyectos_R/TS/serie_tiempo.csv")

```

```

## Rows: 48 Columns: 1
## -- Column specification -----
## Delimiter: ","
## dbl (1): S
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

serie = ts(serie_tiempo$S, start=c(2022, 1), end = c(2025, 9), frequency=12)

```

## Enfoque Zinar(p)

Se usará la técnica series de tiempo generalizadas usando distribución de parte media móvil

$$X_t - \sum_{i=1}^p \alpha_i X_{t-i} \sim Poisson(\lambda)$$

Usando optimización Bayesiana pueden estimarse los parámetros del modelo asumiendo verosimilitud poisson; En nuestro caso probamos con el orden  $p$  desde 1 hasta 4 donde se consideró se tuvo un excelente ajuste.

```

set.seed(1234)
serie

##      Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
## 2022    0   0   0   0   0   0   0   0   0   2   0   0
## 2023    3   0   0   0   0   0   0   0   0   0   0   2
## 2024    7   2   1   0   3   8   5   11  0   5   6  15
## 2025   16  11   1  12   9  10   7   2   3

x = serie
p=4
m = estimate_zinarp(
x,
p,
iter = 5000,
thin = 2,
burn = 0.1,
innovation = "Poisson"
)

```

Se guardaron los parámetros del modelo

```

a1 = mean(m$alpha[,1])
a2 = mean(m$alpha[,2])
a3 = mean(m$alpha[,3])
a4 = mean(m$alpha[,4])
lambda = mean(m$lambda)

```

Se contrastó con la serie original

```

set.seed(1234)
serie1 = serie[-length(serie)]
serie2 = serie[-c(length(serie),(length(serie)-1))]
serie3 = serie[-c(length(serie),(length(serie)-1),(length(serie)-2))]
serie4 = serie[-c(length(serie),(length(serie)-1),(length(serie)-2),length(serie)-3)]
poi = rpois(length(serie1),lambda = lambda)
Proceso = a1*serie1 +a2*serie2 + a3*serie3+a4*serie4 +poi

## Warning in a1 * serie1 + a2 * serie2: longitud de objeto mayor no es múltiplo
## de la longitud de uno menor

## Warning in a1 * serie1 + a2 * serie2 + a3 * serie3: longitud de objeto mayor no
## es múltiplo de la longitud de uno menor

## Warning in a1 * serie1 + a2 * serie2 + a3 * serie3 + a4 * serie4: longitud de
## objeto mayor no es múltiplo de la longitud de uno menor

ks.test(Proceso,serie)

##
##  Exact two-sample Kolmogorov-Smirnov test
##
## data: Proceso and serie
## D = 0.22222, p-value = 0.1056
## alternative hypothesis: two-sided

```

Dando un valor de excelente ajuste.

## Enfoque Proyectivo

Este enfoque no depende de supuestos probabilísticos; en álgebra lineal una proyección ortogonal es una transformación idempotente, y tal que es normal, es decir en el caso real, commuta con su adjunto  $T \circ T^* = T^* \circ T$ . Adicionalmente, en este caso, el adjunto es la proyección sobre el complemento ortogonal y el vector proyección es el que satisface que la distancia de este a el complemento ortogonal respectivo es mínima. Esto se traduce en que si la distancia es el error cuadrático medio, entonces este se minimizará para el caso del estimador pronóstico. Se usa este enfoque para no depender de un ajuste paramétrico particular sino de la proyección de la historia entera hacia el futuro. Se puede observar que el pronóstico es con aproximadamente 45 periodos de historia, es decir el pronóstico depende de 45 parámetros. Afortunadamente el modelo expresado arriba solo depende de 5 lo cual como se verá más adelante, aunado a que ambos satisfacen bondad de ajuste, hace que sea más facil de manejar en términos predictivos.

```

#####
#Ajuste por mínimos cuadrados
autocov <- function(x, k){
  n <- length(x)
  if(k == n){
    k = 0
  }
  x_bar <- mean(x)
  sum((x[(k+1):n] - x_bar) * (x[1:(n-k)] - x_bar)) / n
}

```

```

autocor <- function(x, k){
  autocov(x,k)/autocov(x,0)
}

##SIGMA
tamano = length(serie)
seriek = serie[1:tamano]
A = matrix(nrow = tamano, ncol = tamano)
for(i in 1:nrow(A)){
  for(j in 1:ncol(A)){
    if(j>i){
      A[i,j] = autocor(seriek,j-i)}else{
      A[i,j] = autocor(seriek,i-j)
    }
  }
}
tam = 12
b = matrix(rep(0, times = tamano*tam), ncol = tam)
#tendencia = rep(0, times = h)
for(h in 1:tam){
#b = gamma(n-s-k) k:1:n
#s horizonte prospectivo
for(b1 in 1:tamano){
b[b1,h] = autocor(seriek,tamano + h - b1)
}
tendencia = solve(A)%*%b[,h]
tendencia = sum(tendencia)
seriek = c(seriek,tendencia)
}

#Validación
length(seriek)

```

```
## [1] 57
```

```
length(serie)
```

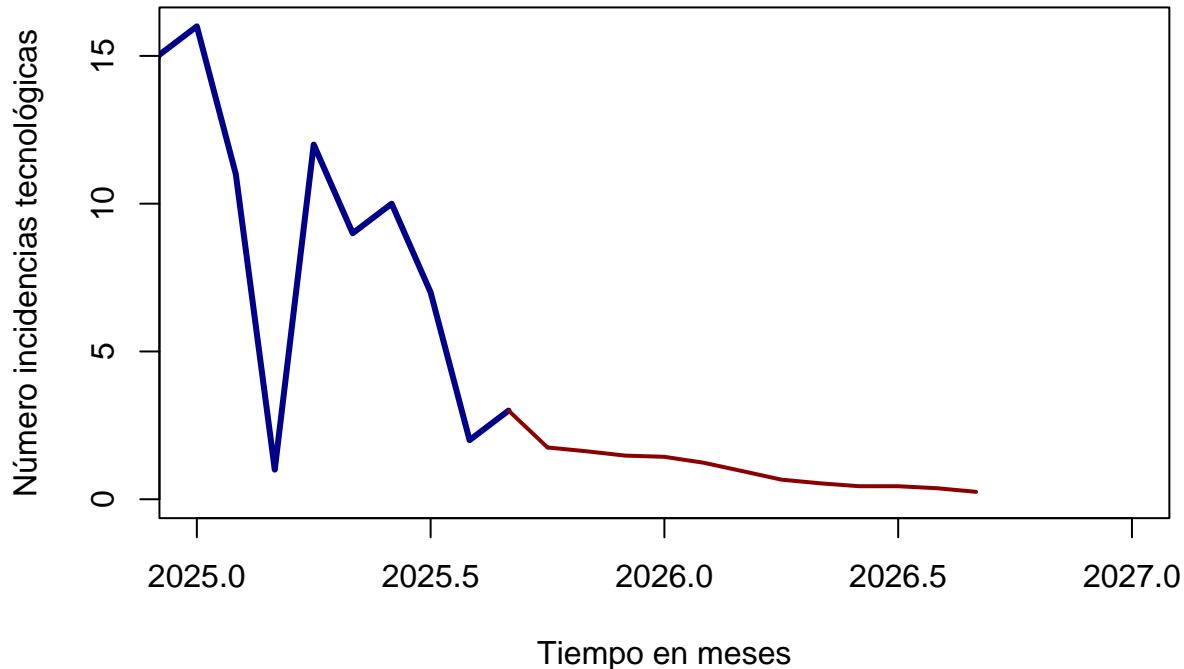
```
## [1] 45
```

```

plot(serie, col = "darkblue", lty = 1, lwd = 3, ylab = "Número incidencias tecnológicas", xlab = "Tiempo"
lines(ts(seriek[46:57],start = c(2025,10),frequency = 12), col = "darkred", lty = 1, lwd = 2)
segments(2026-4/12, seriek[length(serie)], 2026-3/12, ts(seriek[46],start = c(2025,10),frequency = 12),c

```

## Serie Original y Predicciones



```
ks.test(Proceso,seriek)
```

```
##  
##  Exact two-sample Kolmogorov-Smirnov test  
##  
## data: Proceso and seriek  
## D = 0.17544, p-value = 0.2667  
## alternative hypothesis: two-sided
```

Nuevamente ambos presentan excelente ajuste, enfoque proyectivo y enfoque paramétrico.

```
#####  
z1 = z -LMR  
length(which(z1>0))/longitud1  
  
## [1] 0  
  
summary(z1)  
  
##      Min.    1st Qu.     Median      Mean    3rd Qu.      Max.  
## -11998915 -11994675 -11972211 -11656241 -11354945 -10208161
```

## Análisis de sensibilidad

Análisis usando Modelo individual de riesgo