Metodologia_1

Seguros

2025-12-25

Abstract

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

Contents

Introducción **1**

Diversos Miscelaneos **1**

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

Diversos Técnicos **16**

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 . . . 31

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$MON[i] == 20){
    Mi$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$SA), nrow  =11, ncol = 1)
Q = P1
Q = data.frame(Q)
Q$categoria = data.frame(categoria = c("1","2","3","4","5","6","7"
                                       ,"8","9","10","11"))
Q$porcentaje = rep(.1, times = 11)
df = Q %>%
  group_by(categoria) %>%
  summarise(max(Q),Frecuencia = sum(porcentaje))
Q = cbind(as.numeric(df[1,2]),paste(as.character(as.numeric(df[1,2])),"-", as.character(as.numeric(df[2
for(j in 2:11){
  P = cbind(as.numeric(df[j,2]),paste(as.character(as.numeric(df[j,2])),"-", as.character(as.numeric(df
  Q  =rbind(Q,P)
}
s = cbind(Q[-11,],df$Frecuencia[-1])
```

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

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

```r
s = cbind(s,s1)
s
```

```
##                                                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"
```

```r
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
```

```r
#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:

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

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.

```r
#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$SA
#Se aplica una transformación de Box Cox para suavizar los datos
#Posteriormente esto puede deshacerse
M = .5*log(M)-6.2
#Se calculan los whisclers del box plot y se quitan outliers
l =  boxplot.stats(M)$stats
M= M[M >= l[1] & M <= l[5]]
```

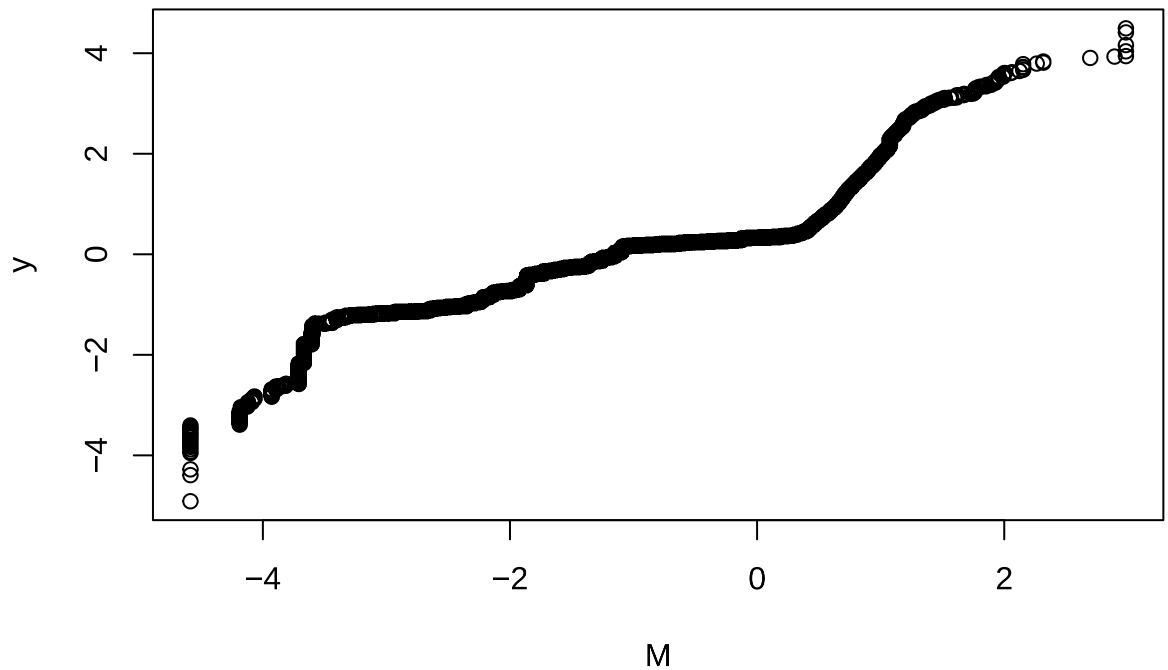
```r
#Se calcula por máxima verosimilitud los degree freedom de una t-student
ajuste = fitdistr(M, densfun = "t")
ajuste
```

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

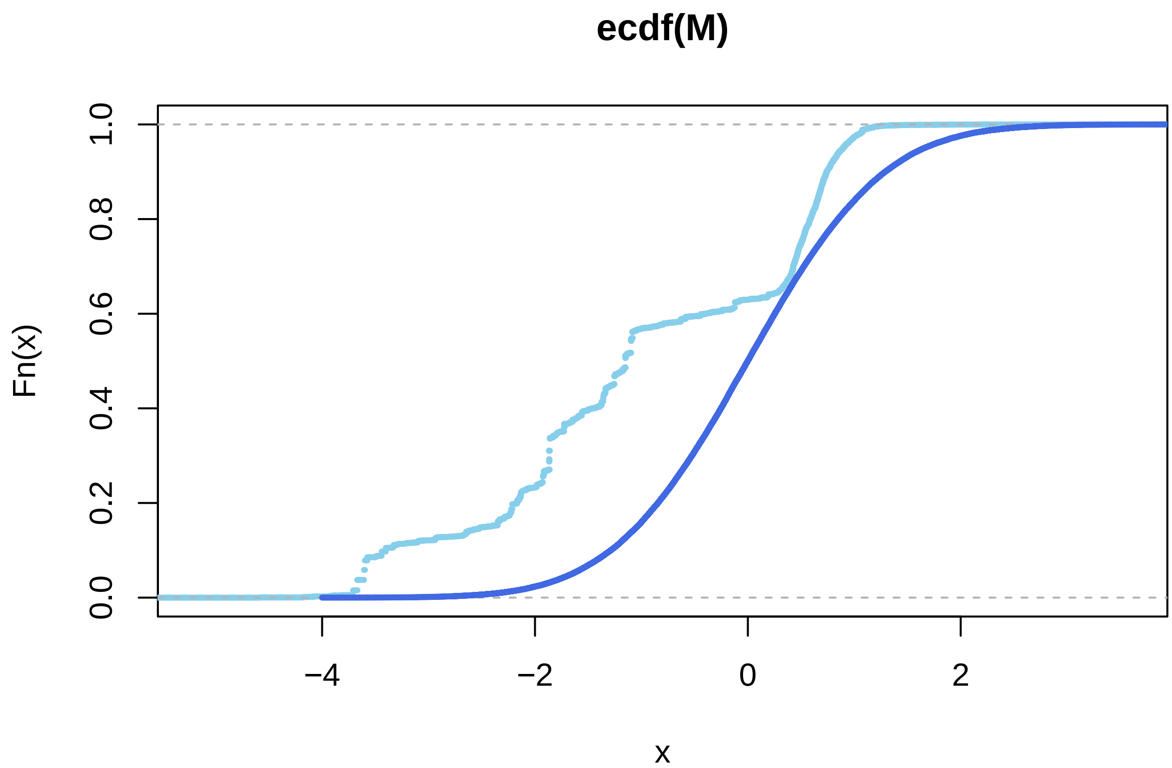
```r
#Se guardan los degree freedom en la variable m
m = ajuste$estimate["df"]
#Se simulan 135502 corridas de t-student con esos degree freedom's
set.seed(1234)
y <- rt(length(M),m)
#Se comparan nuestros datos con la simulación
#La hipótesis alternativa es que la función de distribución o de proabilidad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

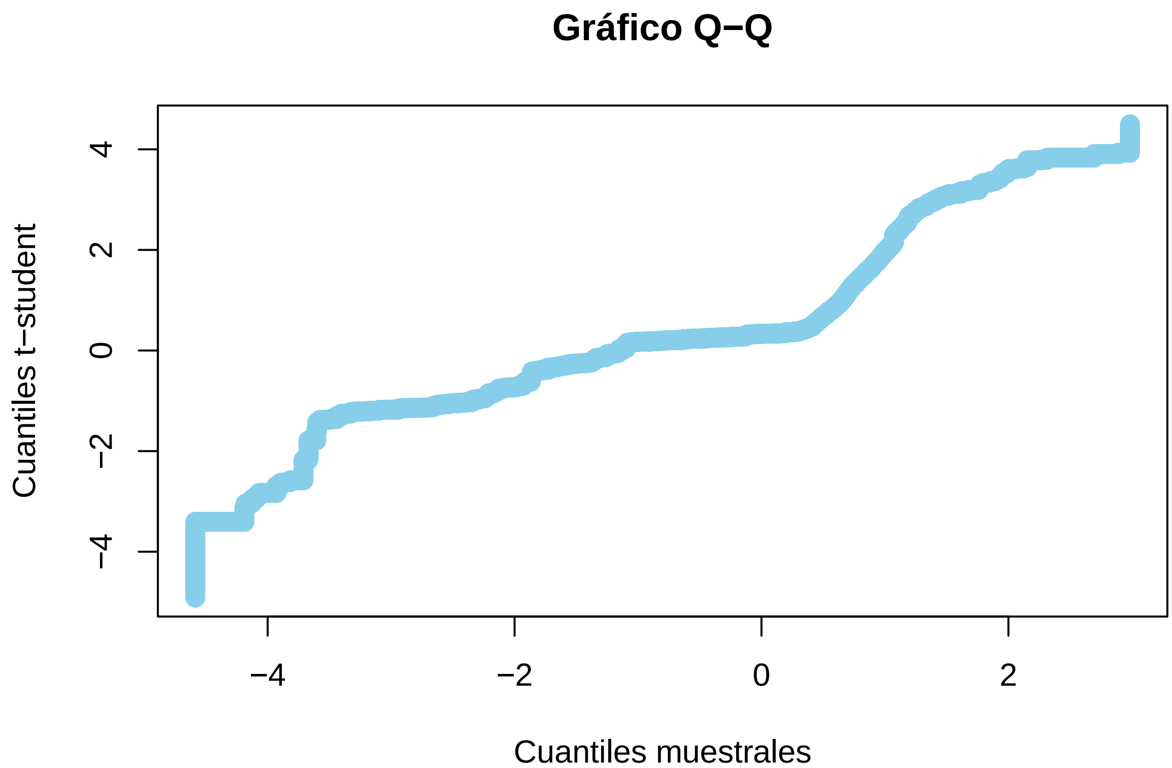
```r
#El qqplot confirma nuestro análisis
qqplot(M,y)
```




```r
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")
```




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



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-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-Student` . Por ende al restar el límite máximo de retención `t-Student` $\geq F_{SA}(x)$. Ahora basta mostrar que $LMR \geq$ `t-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:

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

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
```

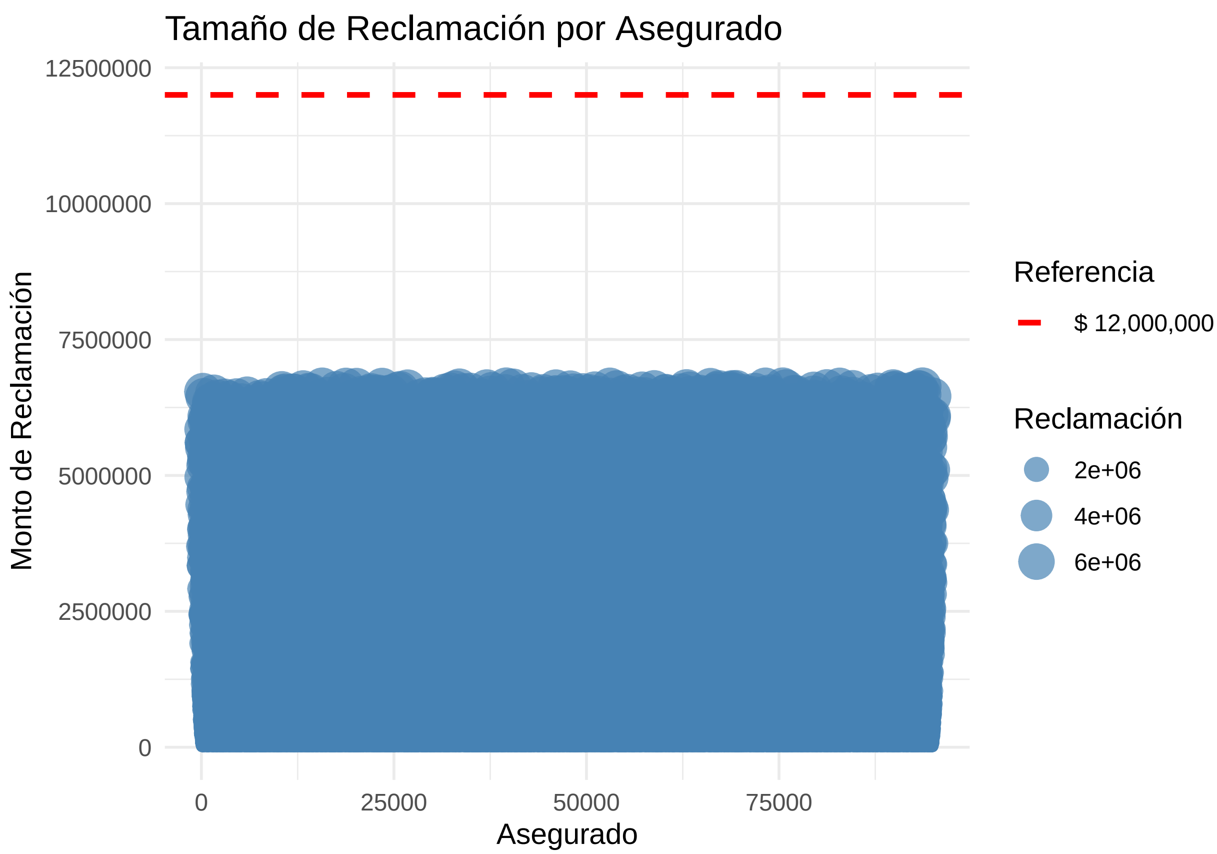
```
summary(z1)
```

```
##       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)$stats
z2= z[ z <= quantile(prob = .95,z)]
H = data.frame(asegurado=c(1:length(z2)), reclamacion = z2)
cutoff =12000000
ggplot(H, aes(x = asegurado, y = reclamacion, size = reclamacion)) +
  geom_point(color = "steelblue", alpha = 0.7) +
  geom_hline(aes(yintercept = cutoff, color  = "$ 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 represantivos 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)
cv
```

```
##
##  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
```

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

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

```
##   omega2
## 436.7515   0.0000
```

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

```
##   omega2
## 2741.868    0.000
```

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

```
##   omega2
## 1391.114    0.000
```

```r
############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
```

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

```
##          An
##         Inf 5.44514e-09
```

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

```
##           An
##          Inf 5.44514e-09
```

```r
t(t(summary(Mi$SA)))
```

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

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

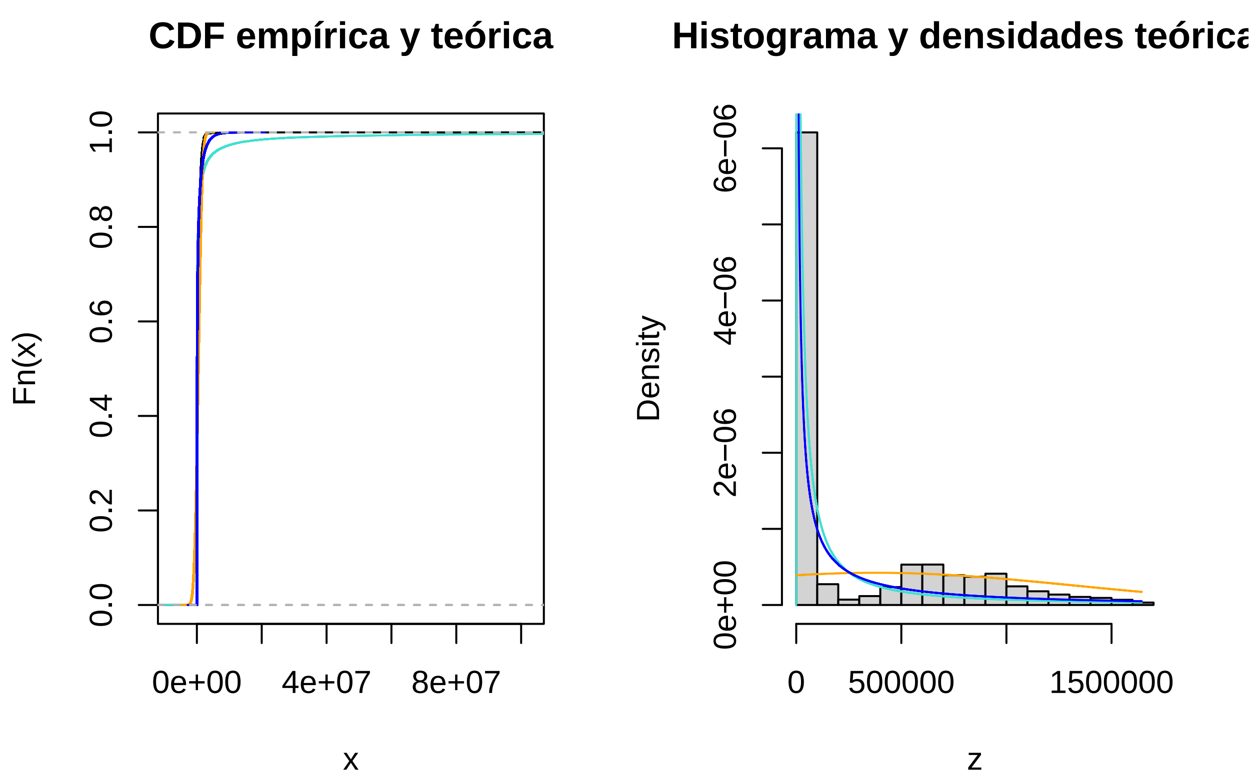
```
## [1]  41197 297728 415753 457253 523362
```

```r
###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 = "Probailidades 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.

```r
####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
```

```r
z= z[z >= l1[1] & z <= l1[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")
```

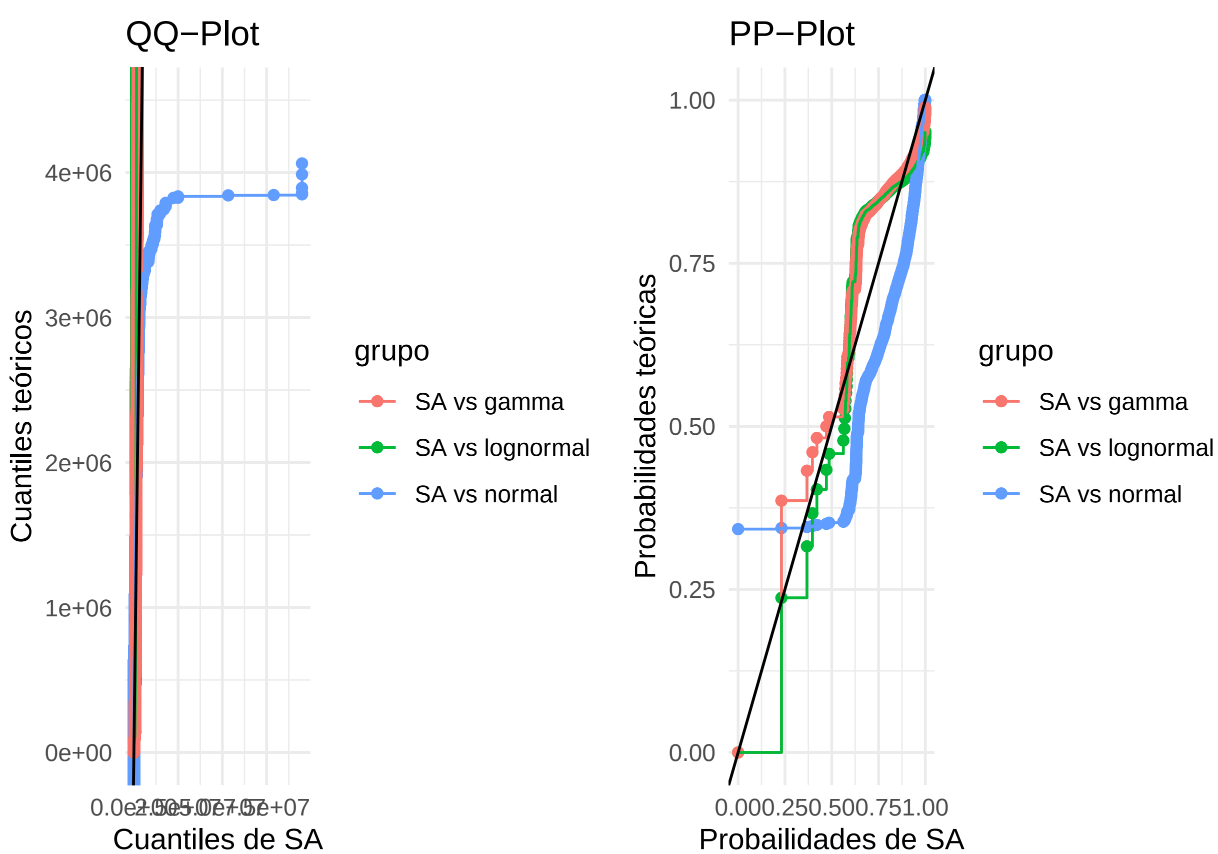



```r
A_1 = Mi$SA
B = y1
C = y2
D = y3
# Crear data frames para cada comparación
qq_AB <- qq_data(A_1, B, "SA vs lognormal")
qq_AC <- qq_data(A_1, C, "SA vs normal")
qq_AD <- qq_data(A_1, D, "SA vs gamma")

# 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()
```

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

```
##
## Adjuntando el paquete: 'gridExtra'

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

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



Prueba de estrés

La simulación adquiere las siguientes características

Análisis de sensibilidad

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