Ejemplo máximos: Altura diques *The sealevel data frame has 81 rows and 2 columns. The columns contain annual sea level maxima from 1912 to 1992 at Dover and Harwich respectively, two sites on the coast of Britain. The row names give the years of observation. There are 39 missing values.*

```
if(!require(evd)){install.packages("evd")} # Instala la librería EVD si no la tienes

library(evd)
datos = evd::sealevel
dover = datos$dover[!is.na(datos$dover)]
```

Se busca modelar el máximo anual del nivel del mar en Dover, para proponer la altura que debe tener el dique que le contenga, con el fin de evitar inundaciones, considerando una probabilidad de 0.1% de que el nivel máximo del agua rebase al muro.

- Encuentra la mejor distribución paramétrica para modelar la variable.
- Estima el cuantil requerido.
- ¿Cuál es la probabilidad de que alguno de los niveles máximos de los siguientes 10 años sea superior a 5 metros? Compara la respuesta analítica con la simulada.

```
# Se pudo haber pensado en modelar como una lognormal o exponencial:

library(fitdistrplus)
lognormal = fitdist(dover, distr = "lnorm")
exponencial = fitdist(dover, distr = "exp")

# Se pudo probar con la prueba KS si había evidencia para rechazar estas distribuciones:

ks.test(dover, "plnorm", lognormal$estimate[1], lognormal$estimate[2])
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  dover
## D = 0.13384, p-value = 0.1516
## alternative hypothesis: two-sided
```

```
ks.test(dover, "pexp", exponencial$estimate[1])
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  dover
## D = 0.58294, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

```
# Estimando el cuantil con la lognormal:

qlnorm(.999, lognormal$estimate[1], lognormal$estimate[2])
```

```
## [1] 4.541995
```

La lognormal no fue rechazada, pero como sabemos que las observaciones son máximos, entonces podemos usar otros resultados para modelarlos.

Breve introducción a Teoría de Valores Extremos

Esta rama de la estadística se encarga del estudio de observaciones extremas (ya sea máximas o mínimas, o a veces simplemente muy raras de observar), en una variable aleatoria. Es necesaria porque muchos resultados, como el TLC, no se cumplen o tardan mucho en cumplirse (requieren una muestra muy grande).

Sean $X_1, X_2, ..., X_n$ una secuencia de variables aleatorias independientes e idénticamente distribuidas. Si quisiéramos encontrar la distribución asintótica del máximo/mínimo de las mismas, podemos aprovechar el Teorema de **Fisher-Tippett-Gnedenko** (FTG) que nos dice que, si existen secuencias de constantes a_n, y b_n, para realizar una especie de estandarización, entonces:

$lim_{n\to\infty} F^n(a_n x + b_n) = G(X)$

Donde $G(X)$ es la distribución generalizada de valores extremos.

Esta distribución puede tomar las siguientes formas, dependiendo del parámetro de forma o de la cola:

$G_\gamma(x) = e^{-(1+\gamma x)^{-1/\gamma}}$, cuando $1 + \gamma x > 0$

$G_\gamma(x) = e^{-e^{-x}}$, en otro caso.

La distribución generalizada de valores extremos (GEV) puede tomar la forma de 3 distintas distribuciones paramétricas:

- Fréchet (cuando el parámetro de cola es mayor que 0).
- Gumbel (cuando es igual a 0).
- Negative or reversed Weibull (cuando es menor que 0).

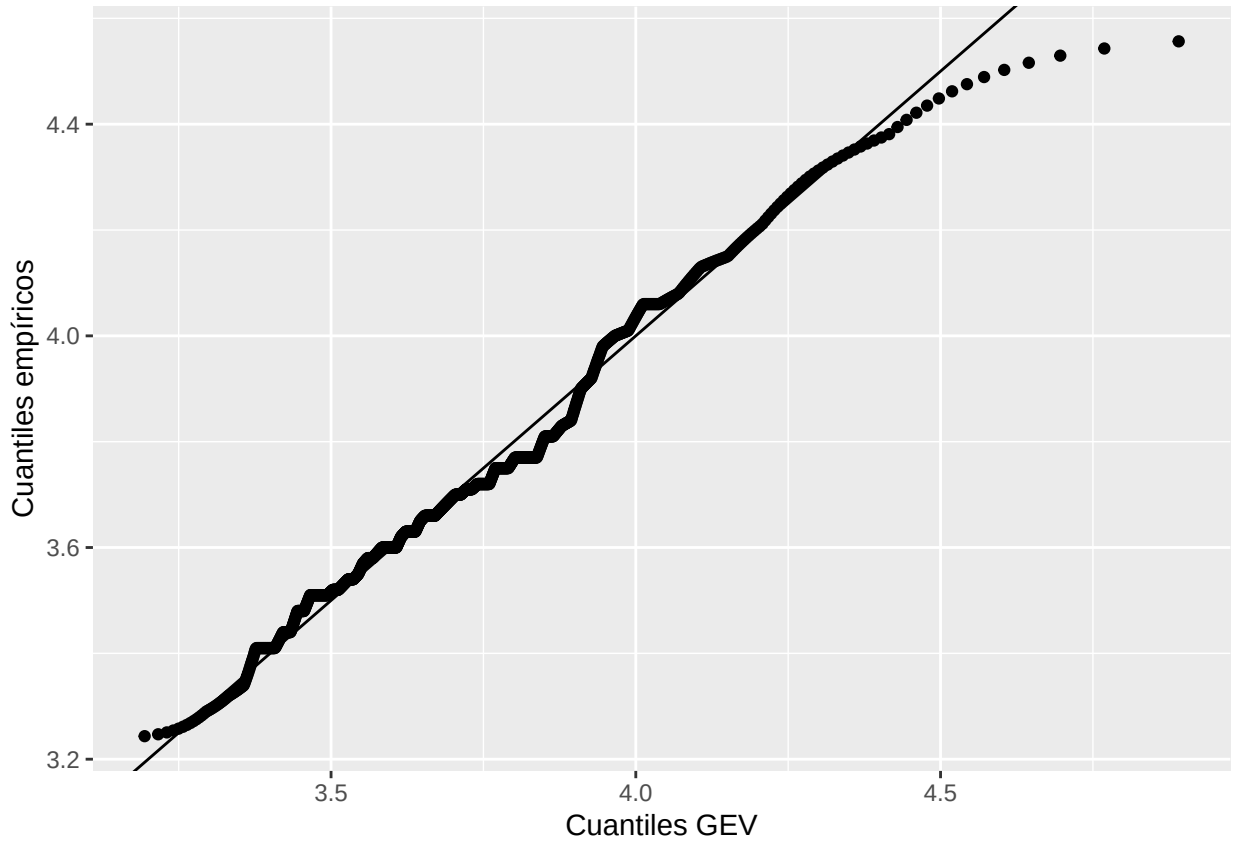
En la práctica, lo que se hace es estimar los parámetros para "estandarizar" los máximos y el parámetro de la cola (γ).

Así, sabiendo que en nuestro ejemplo tenemos solamente máximos en la muestra, podemos ajustar una distribución GEV:

```
dgve = fgev(x = dover)

# Podemos ver que el parámetro de cola estimado es de -0.0211, lo que probablemente nos hable de una di

a = dgve$estimate[1]
b = dgve$estimate[2]
gamma = dgve$estimate[3]
```

Revisamos qué tan bien se ajustan los datos a esta distribución, por ejemplo con un QQ-plot y una prueba KS:

```
cuantiles = 1:999/1000
Q_empiricos = quantile(dover, cuantiles)
Q_GEV = qgev(cuantiles, a, b, gamma)

library(ggplot2)
ggplot()+
  geom_point(aes(Q_GEV, Q_empiricos))+
  labs(x = "Cuantiles GEV",
       y = "Cuantiles empíricos")+
  geom_abline(intercept = 0, slope = 1)
```




```
# Prueba KS:

ks.test(dover, "pgev", a, b, gamma)
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  dover
## D = 0.087582, p-value = 0.6387
## alternative hypothesis: two-sided
```

Se ve un buen ajuste y además no se rechaza la distribución con la prueba Kolmogorov-Smirnov, por lo que la usaremos para resolver el problema.

Podemos calcular la altura del dique para que la probabilidad de que sea rebasado por el siguiente máximo sea de 0.1% o 0.001:

```
altura = qgev(.999, a, b, gamma)
altura
```

```
##      loc
## 4.890708
```

Obtenemos una altura mayor que con la distribución lognormal. Si hubiéramos utilizado el resultado con esta distribución, la probabilidad de que el muro fuera rebasado por el nivel máximo del agua, sería de:

```r
pgev(4.541995, a, b, gamma, lower.tail = F)
```

```
##       shape
## 0.007051714
```

Sería de 0.7%, algo mayor al 0.1%, por lo que estábamos subestimando el nivel máximo con la lognormal.

¿Cuál es la probabilidad de que alguno de los niveles máximos de los siguientes 10 años sea superior a 5 metros? Compara la respuesta analítica con la simulada.

Analíticamente:

```r
# Puedo calcular la probabilidad de que en un año el máximo sea superior a 5:
pgev(5, a, b, gamma, lower.tail = F)
```

```
##        shape
## 0.0005326552
```

```r
# Para obtener la probabilidad de que por lo menos en uno de los 10 años se rebasen los 5, tendríamos q
pgev(5, a, b, gamma, lower.tail = F)*10
```

```
##       shape
## 0.005326552
```

Obtenemos una probabilidad de 0.5%.

Con simulación:

```r
# Simulando un escenario:

rgev(n = 10, loc = a, scale = b, shape = gamma)
```

```
##  [1] 3.773182 4.258812 3.903763 3.616822 3.974221 3.686784 3.877788 3.667762
##  [9] 3.444577 4.292614
```

```r
# Simulando un millón de escenarios:

simulaciones = 1e6

escenarios = replicate(simulaciones,
                       rgev(n = 10, loc = a, scale = b, shape = gamma))

cumplen = apply(escenarios, MARGIN = 2, function(x){any(x>5)})

mean(cumplen)
```

```
## [1] 0.005297
```

Con un millón de escenarios, obtenemos también una probabilidad de 0.5%, así como en la solución analítica.

Ejemplo Pérdidas agregadas (EVT):

Tenemos una base de datos de siniestros por incendios, con la que podemos conocer la fecha y la cantidad que se pagó por parte de la aseguradora encargada del siniestro.

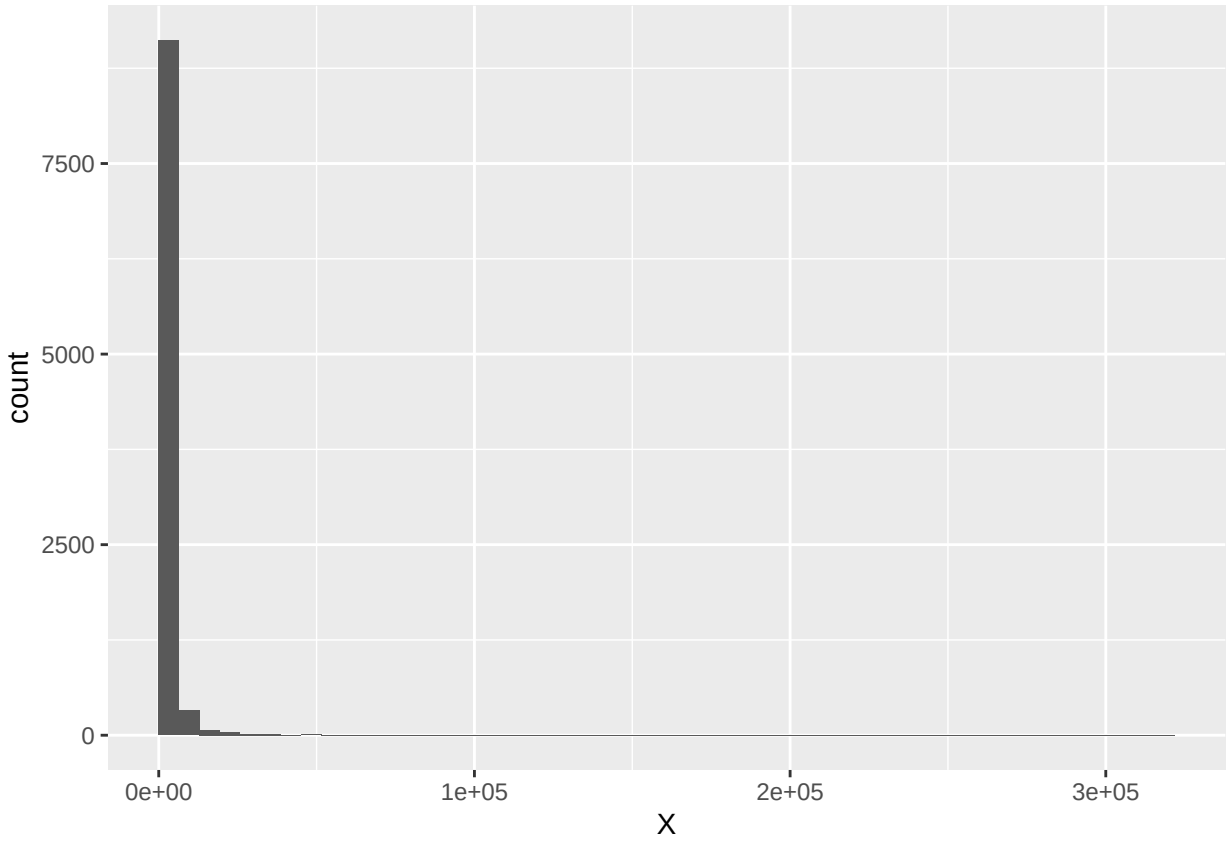
```
base = read.csv("D:/UAG/Modelación actuarial/2022/clases/28. EVD/frecomfire.csv")
```

Se buscará modelar la variable de pérdidas agregadas anuales por este tipo de siniestros.

Para esto, modelará primero la severidad y luego la frecuencia, para con estos dos modelos, realizar simulaciones y encontrar una distribución para las pérdidas agregadas.

Severidad:

```
X = base$ClaimCost2007

ggplot()+
  geom_histogram(aes(X), boundary = 0, bins = 50)
```




```
summary(X)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##    186.3    412.5    763.3   1981.1   1750.5 315543.8
```

Propondremos algunas distribuciones para modelar la variable:

- Gamma
- Lognormal
- Weibull

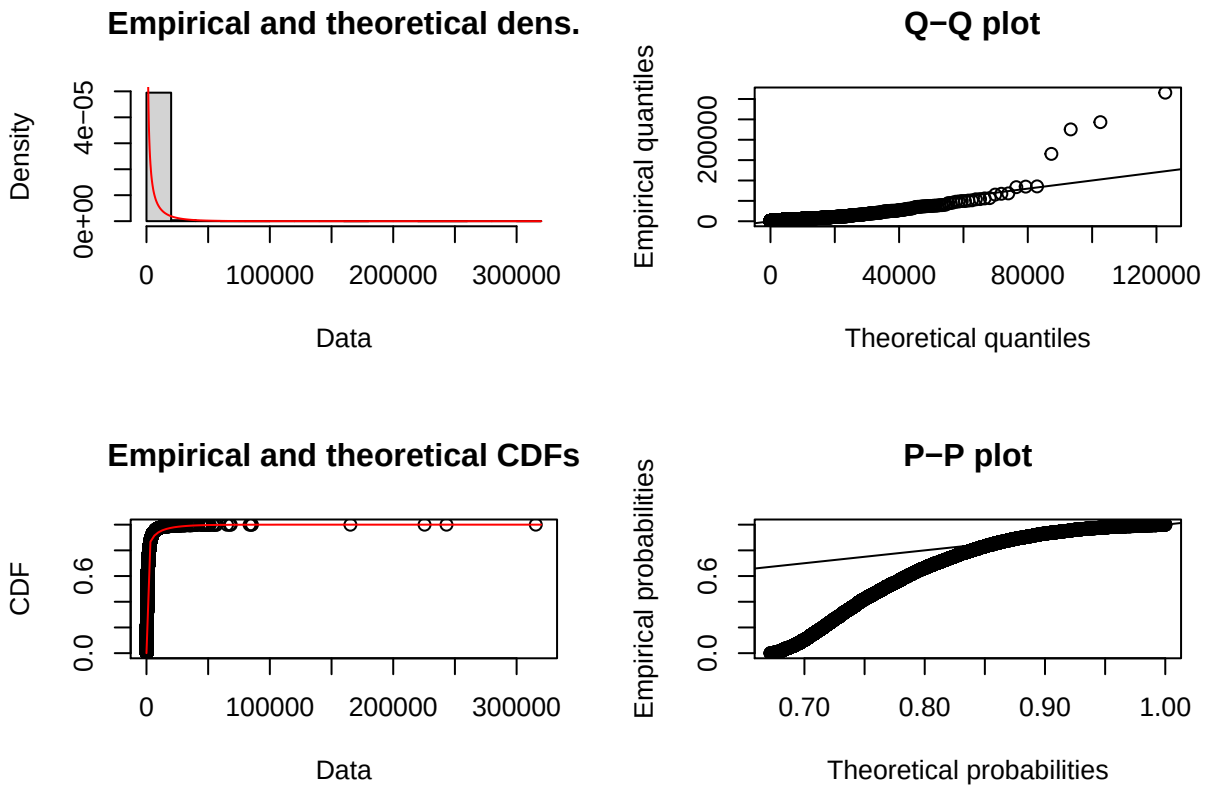
Ajustando modelos:

mod_lnorm = fitdist(X, distr = "lnorm")

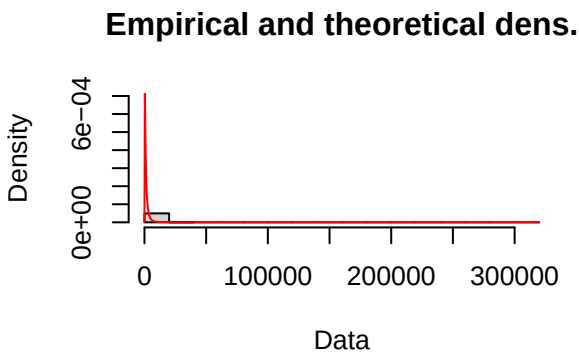
```
library(fitdistrplus)
mod_gamma = fitdist(X, distr = "gamma", method = "mme")
mod_lnorm = fitdist(X, distr = "lnorm")
mod_weibull = fitdist(X, distr = "weibull")
```

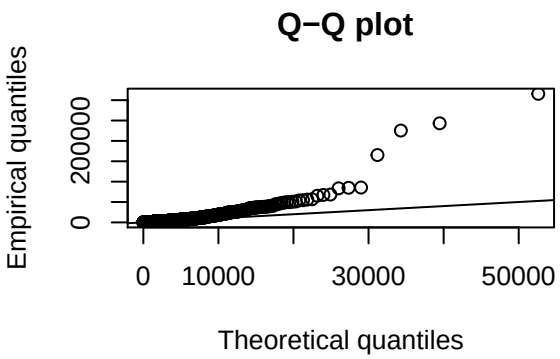
Podemos revisar gráficamente el ajuste de los 3 modelos:

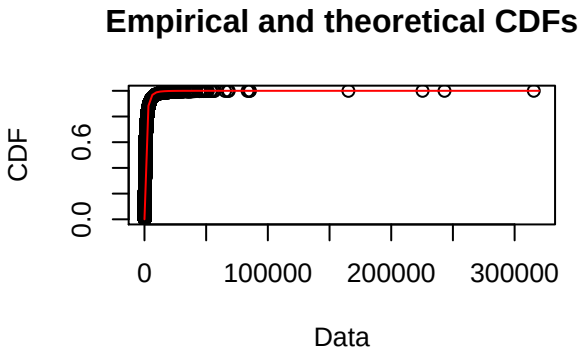
```
# Gamma:
plot(mod_gamma)
```

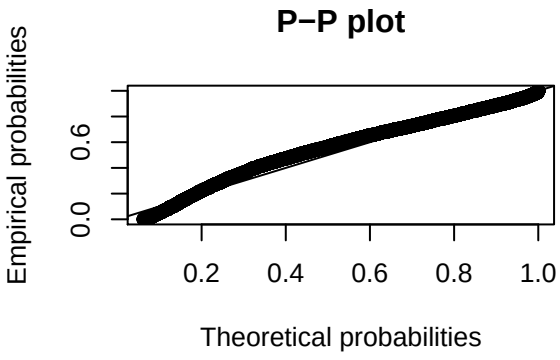



```
# Lognormal:
plot(mod_lnorm)
```

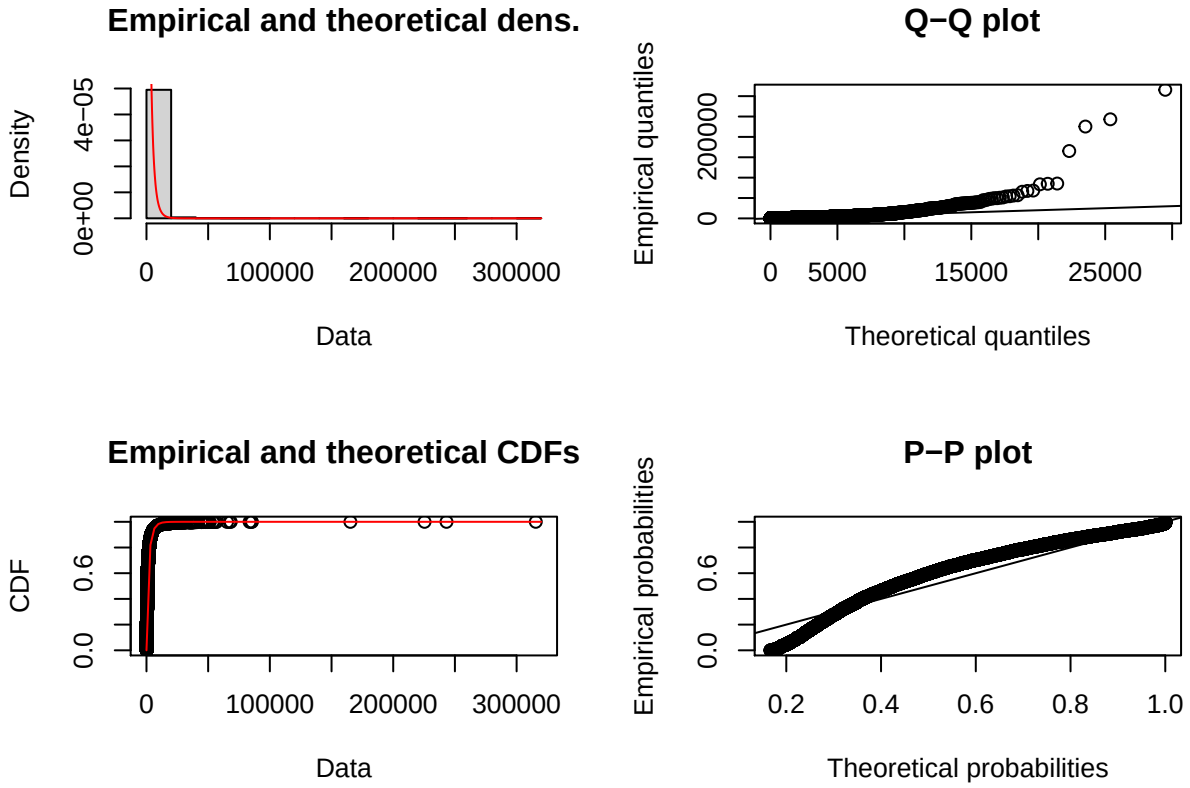









```
# Weibull:
plot(mod_weibull)
```



Podemos compararlas por sus criterios de información:

```
mod_gamma$aic
```

```
## [1] 183325.8
```

```
mod_lnorm$aic
```

```
## [1] 159370.8
```

```
mod_weibull$aic
```

```
## [1] 163566.2
```

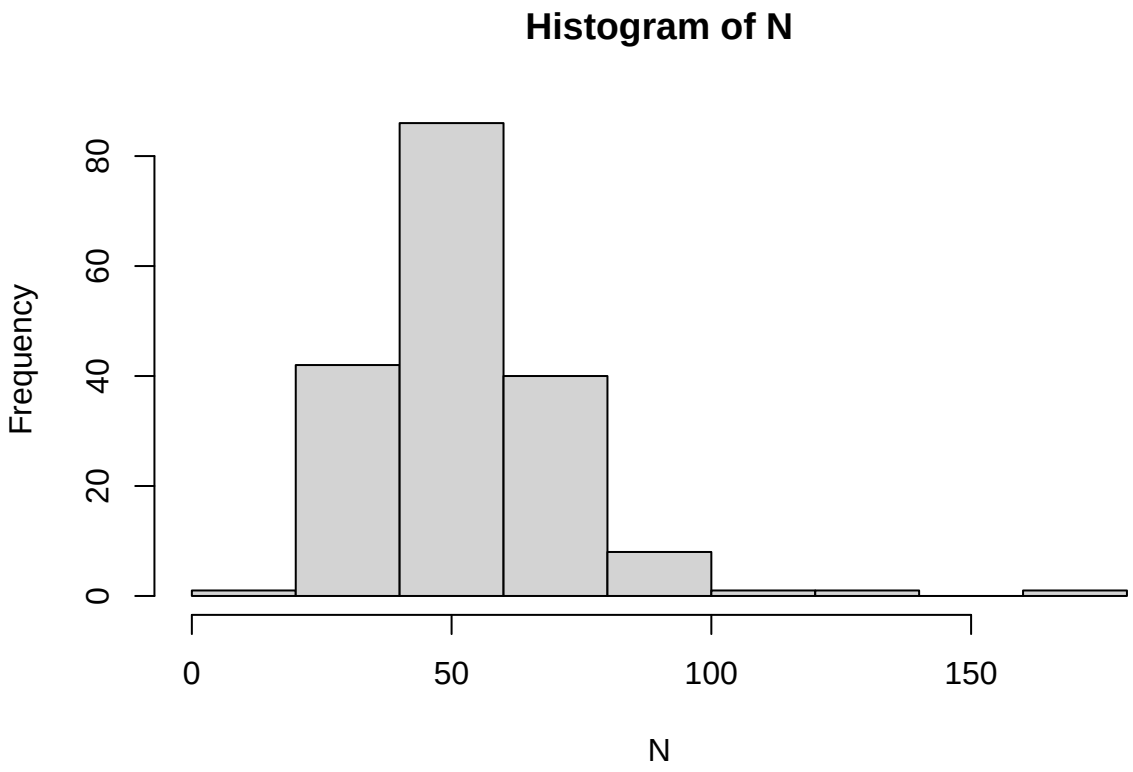
Omitiendo las pruebas estadísticas[1], nos quedamos con la lognormal por su AIC.

Frecuencia:

Si realizamos un modelo para la frecuencia mensual de siniestros, tendremos 180 observaciones.

[1] Esto está mal.

```
base$Mes = zoo::as.yearmon(base$OccurDate)
N = as.vector(table(base$Mes))

hist(N)
```




```
summary(N)
```

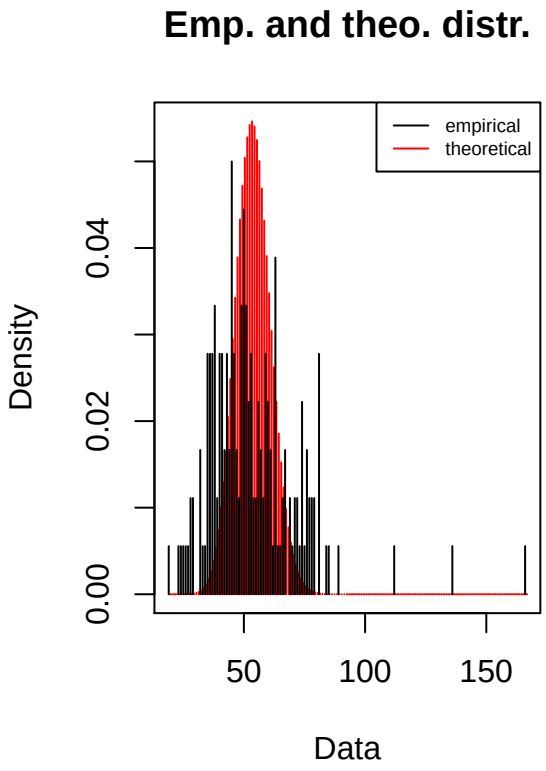
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   19.00   41.00   50.00   53.41   63.00  166.00
```

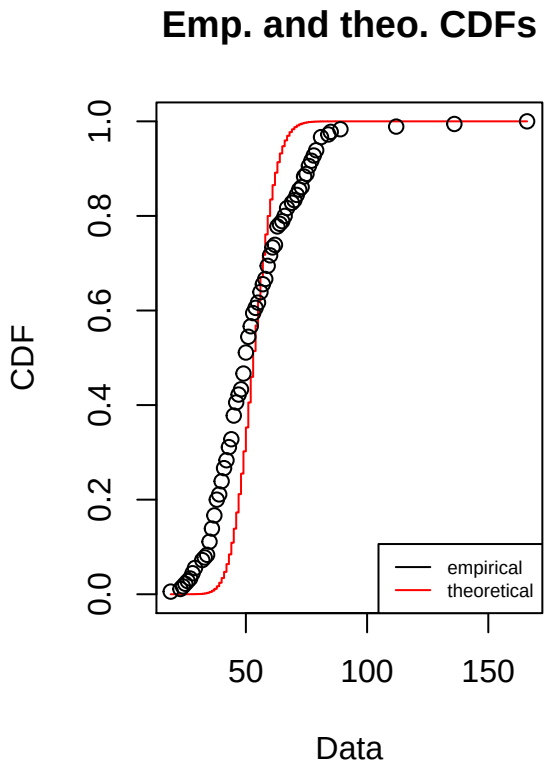
Se propone utilizar la distribución Poisson y la Binomial negativa para modelar la frecuencia.

```
mod_pois = fitdist(N, distr = "pois")
mod_nbinom = fitdist(N, distr = "nbinom")
```

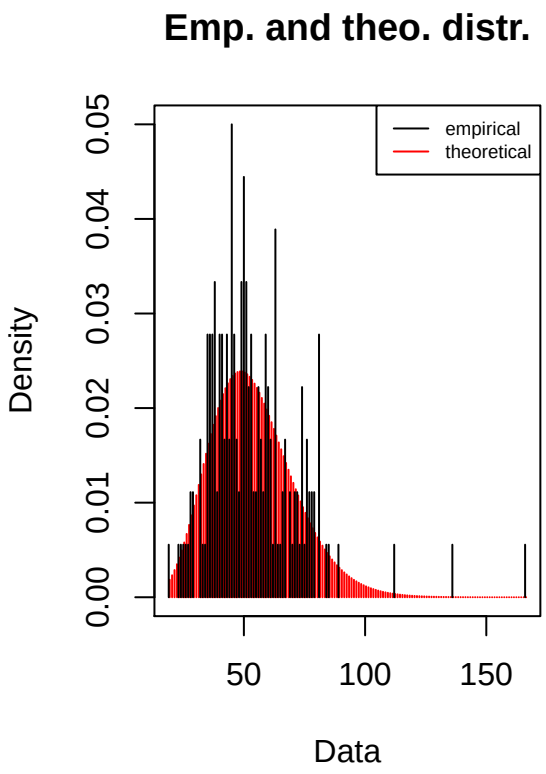
Vemos su ajuste gráfico:

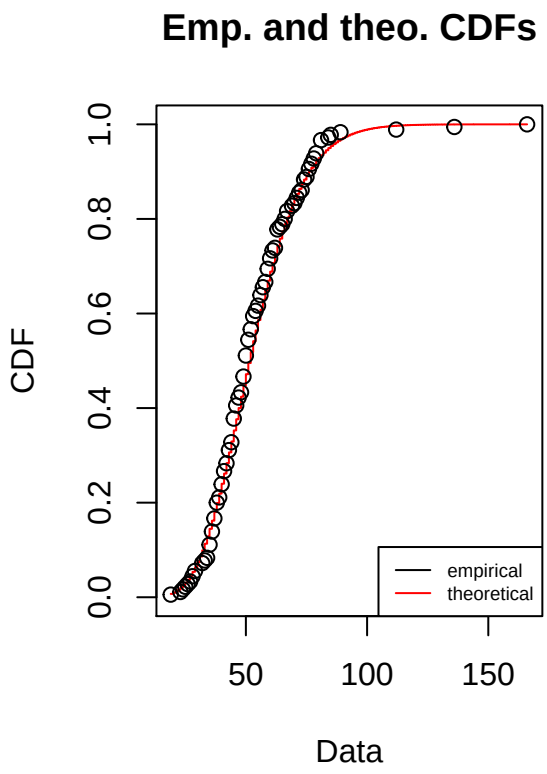
```
plot(mod_pois)
```






```
plot(mod_nbinom)
```






```
mod_pois$aic
```

```
## [1] 2093.498
```

```
mod_nbinom$aic
```

```
## [1] 1529.369
```

Se observa un mejor comportamiento con la negativa binomial y un menor AIC, por lo que esta será la elección para modelar la frecuencia mensual.

Pérdidas agregadas:

Vamos a simular así ambas variables (frecuencia y severidad) para obtener una distribución para las pérdidas agregadas $L = \sum_{i=1}^{N} X_i$.

Podemos empezar simulando un año de pérdidas:

```
# Primero simulamos el número de siniestros en un año:
size = mod_nbinom$estimate[1]
mu = mod_nbinom$estimate[2]
Nsim = sum(rnbinom(12, size, mu = mu))

# Ahora, simulamos la severidad para el número de siniestros obtenido:
```

```r
meanlog = mod_lnorm$estimate[1]
sdlog = mod_lnorm$estimate[2]
Xsim = rlnorm(Nsim, meanlog, sdlog)

# Con las severidades obtenidas, podemos calcular las pérdidas agregadas simplemente sumando la totalid
Lsim = sum(Xsim)
Lsim
```

```
## [1] 989164
```

Ahora, podemos repetir esto una gran cantidad de escenarios para obtener una distribución para las pérdidas agregadas anuales:

Para definir el número de simulaciones, usaremos la expresión derivada para estimar una media:

$n \geq (\frac{q_{(1-\alpha/2)}\sigma}{\epsilon\mu})^2$

```r
escenarios = 1
nmin = 2
alpha = .05
epsilon = .001
cuantil = qnorm(1-alpha/2)
while(nmin>escenarios){

  escenarios = escenarios*2

  Nsim = replicate(escenarios,
                   sum(rnbinom(12, size, mu = mu)))

  Xsim = sapply(1:escenarios,
                function(i){
                  rlnorm(Nsim[i], meanlog, sdlog)
                })

  Lsim = sapply(1:escenarios, function(i){sum(Xsim[[i]])})

  media = mean(Lsim)
  desv = sd(Lsim)
  nmin = (cuantil*desv/(epsilon*media))^2

  print(list(escenarios = escenarios,
             nmin = nmin,
             media = media))
}
```

```
## $escenarios
## [1] 2
##
## $nmin
## [1] 121898
##
## $media
## [1] 1072489
##
```

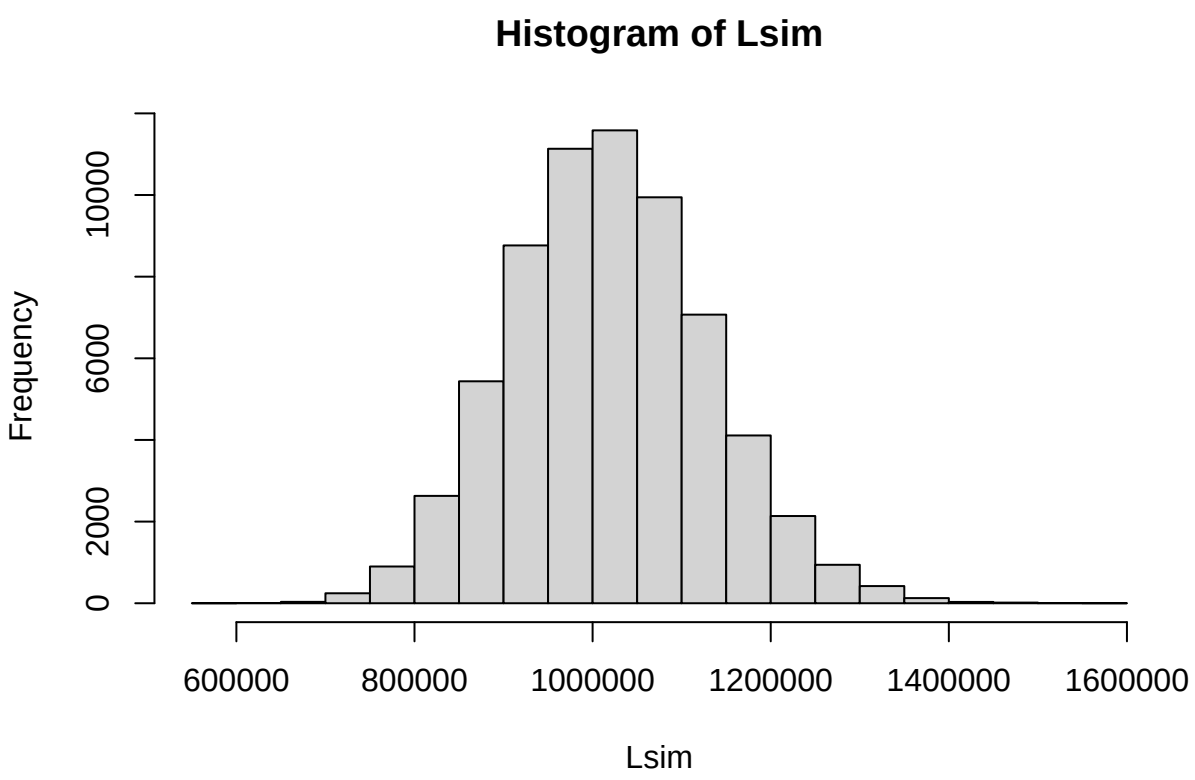
```
## $escenarios
## [1] 4
##
## $nmin
## [1] 38141.55
##
## $media
## [1] 1052161
##
## $escenarios
## [1] 8
##
## $nmin
## [1] 27065.12
##
## $media
## [1] 1048641
##
## $escenarios
## [1] 16
##
## $nmin
## [1] 50754.29
##
## $media
## [1] 994841.9
##
## $escenarios
## [1] 32
##
## $nmin
## [1] 33286.62
##
## $media
## [1] 993233.5
##
## $escenarios
## [1] 64
##
## $nmin
## [1] 63600.01
##
## $media
## [1] 1011695
##
## $escenarios
## [1] 128
##
## $nmin
## [1] 42963.56
##
## $media
## [1] 1003871
##
```

```
## $escenarios
## [1] 256
##
## $nmin
## [1] 47359.11
##
## $media
## [1] 1012454
##
## $escenarios
## [1] 512
##
## $nmin
## [1] 43575.29
##
## $media
## [1] 1019653
##
## $escenarios
## [1] 1024
##
## $nmin
## [1] 44636.61
##
## $media
## [1] 1021948
##
## $escenarios
## [1] 2048
##
## $nmin
## [1] 46488.12
##
## $media
## [1] 1023627
##
## $escenarios
## [1] 4096
##
## $nmin
## [1] 46759.71
##
## $media
## [1] 1020224
##
## $escenarios
## [1] 8192
##
## $nmin
## [1] 45651.3
##
## $media
## [1] 1018717
##
```

```
## $escenarios
## [1] 16384
##
## $nmin
## [1] 45558.27
##
## $media
## [1] 1018256
##
## $escenarios
## [1] 32768
##
## $nmin
## [1] 45236.55
##
## $media
## [1] 1018633
##
## $escenarios
## [1] 65536
##
## $nmin
## [1] 45443.83
##
## $media
## [1] 1018710
```

Así, ya tenemos una gran cantidad de escenarios para las pérdidas agregadas, que podemos usar como su distribución empírica.

```
hist(Lsim)
```




```
summary(Lsim)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  566193  941835 1015569 1018711 1091573 1557421
```

Si la aseguradora tuviera 120 clientes, y la prima a cobrar fuera la misma para todos, e igual al valor esperado de las pérdidas agregadas, entonces debería costar:

```
mean(Lsim)/120
```

```
## [1] 8489.254
```

Si la aseguradora quisiera que la probabilidad de no poder pagar sus responsabilidades con lo ingresado por primas fuera de 10%, la prima que debería de cobrar a cada cliente sería de:

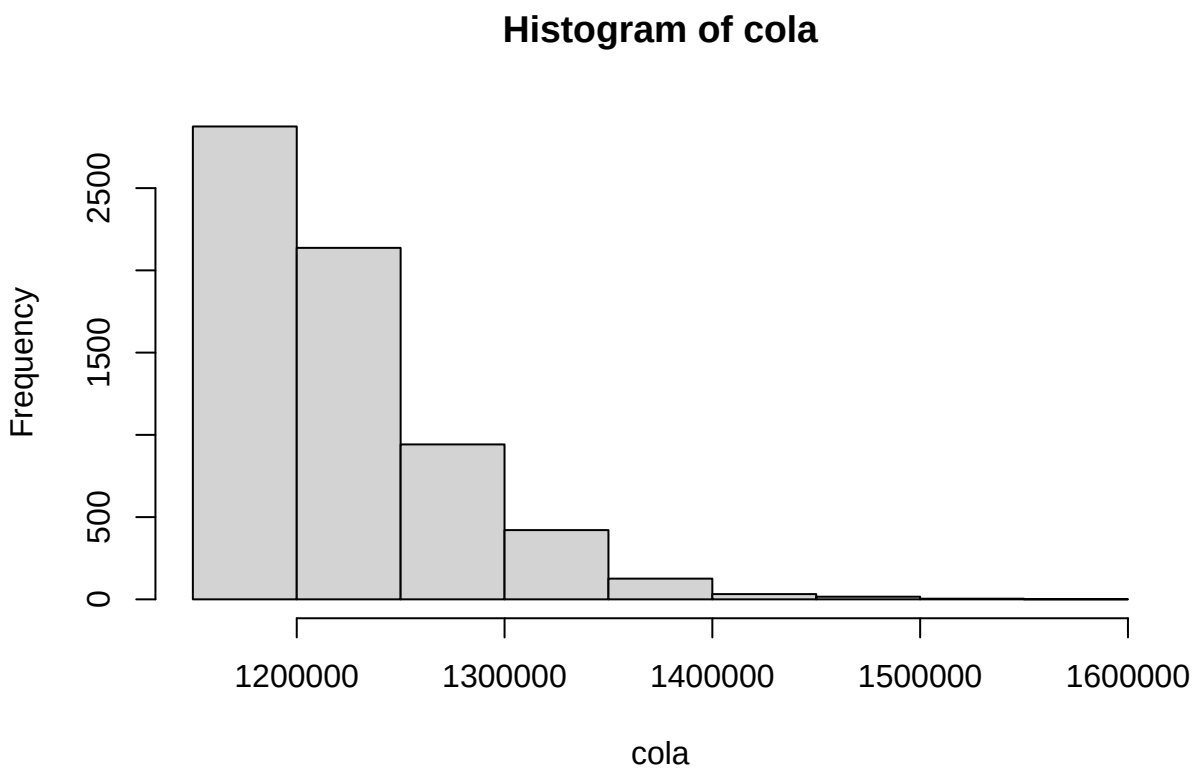
```
quantile(Lsim, .9)/120
```

```
##      90%
## 9679.917
```

Ahora, suponiendo que el reaseguro entra en caso de que las pérdidas superen el VaR al 90%. Calcula la prima anual justa (el valor esperado) que debe cobrar a la aseguradora.

Podemos resolver con la distribución empírica:

```r
VaR = quantile(Lsim, .9)
cola = Lsim[Lsim>VaR]
hist(cola)
```




```r
# Calculando el valor esperado:
mean(cola)
```

```
## [1] 1221461
```

También podríamos resolver teóricamente con algún modelo paramétrico:

Usaremos una distribución generalizada de valores extremos para modelar las pérdidas superiores al cuantil 0.9:

```r
library(evd)
mod_gev = fgev(cola, std.err = F)

# Podemos ver los parámetros estimados a, b y gamma:
mod_gev$estimate
```

```
##          loc        scale        shape
## 1.197587e+06 4.116218e+04 2.461326e-01
```

Vemos que el parámetro de forma o de la cola es mayor que 0, por lo que probablemente estemos en el caso de colas pesadas de la distribución Fréchet.

Podemos calcular la media teórica de la cola, si es que sus datos fueran generados por esta distribución GEV:

```
mu = mod_gev$estimate[1]
sigma = mod_gev$estimate[2]
xi = mod_gev$estimate[3]

g1 = gamma(1-1*xi)

# Media teórica:
mu+sigma*(g1-1)/xi
```

```
##     loc
## 1234430
```

```
# Media con simulaciones:

mean(rgev(1e6, mu, sigma, xi))
```

```
## [1] 1234501
```

Vemos que el modelo paramétrico estima una prima mayor que el modelo empírico, con lo que la reaseguradora podría cubrirse más en caso de que haya siniestros con un valor mucho más alto.