Tarea 2

NO opta a Titulación

Juan Pablo Zamora Alarcón

2025-11-03

Contenido

0.1. Datos 1

0.2. Metodos de Aproximación 27

0.1. Datos

La base a utilizar es "peragregas3.xlsx", contiene los montos individuales de reclamación (sev1,sev2,..,etc) por póliza de un seguro colectivo de vehículos, con la frecuencia de reclamación dada por la variable, *frecuencia*.

Entonces, antes del análisis, comenzamos con la lectura de nuestra base:

```r
# Semilla para mantener la misma información al ejecutar el proceso.
set.seed(123)
# Cargamos la lectura del archivo csv
data <- read.csv("D:/Diplomado Solvencia II/Mod 1 Estd Aplcda Seg/Tarea 2/Recibido/peragregas3.csv")
# Agregamos la función attach a nuestra base, para acceder a las variables.
attach(data)
```

También mostraremos los primeros y últimos 6 registros de nuestra base de datos:

```r
# Vista de las primeras 6 filas de la base.
head(data)
```

```
##   frecuencia       sev1      sev2      sev3      sev4      sev5       sev6
## 1          5  31.672877 154.57888  85.09500  44.18517 259.66508         NA
## 2          5  98.260324  18.13541 127.45057  21.31119  19.92799         NA
## 3          6  33.530148  34.00856 361.22576  91.15881 187.09583   9.946712
## 4          3 408.593003  85.61911  34.77149        NA        NA         NA
## 5          4   2.875061  35.11828 350.03364  40.07794        NA         NA
## 6          7  55.670380  55.45924  49.08410 203.09120   7.86949 132.897135
```

```
##        sev7 sev8 sev9 sev10 sev11 sev12
## 1        NA   NA   NA    NA    NA    NA
## 2        NA   NA   NA    NA    NA    NA
## 3        NA   NA   NA    NA    NA    NA
## 4        NA   NA   NA    NA    NA    NA
## 5        NA   NA   NA    NA    NA    NA
## 6 322.0671   NA   NA    NA    NA    NA
```

```
# Vista de las últimas 6 filas de la base.
tail(data)
```

```
##     frecuencia        sev1       sev2      sev3      sev4       sev5     sev6
## 195          2   0.3459765  77.521216        NA        NA         NA       NA
## 196          4  21.9438751  42.441581  87.82924  30.12584         NA       NA
## 197          8  16.4569999 203.939309 117.98043  52.32617 157.291352 283.5508
## 198          8 153.3175108   7.587017  23.91969  62.82738   2.354313 423.3370
## 199          7  31.8279457  91.729002 139.34800  47.98098 164.076066 149.6044
## 200          6  22.7985801 150.691710 202.23787 216.65636  85.332794 349.3575
##          sev7       sev8 sev9 sev10 sev11 sev12
## 195        NA         NA   NA    NA    NA    NA
## 196        NA         NA   NA    NA    NA    NA
## 197  30.31693   46.77929   NA    NA    NA    NA
## 198 106.14440 1153.81604   NA    NA    NA    NA
## 199  64.33534         NA   NA    NA    NA    NA
## 200        NA         NA   NA    NA    NA    NA
```

Nuestra base contiene un total de registros igual a:

```
# Cantidad de datos
n <- nrow(data)
n
```

```
## [1] 200
```

0.1.1. Descripción de mis datos de Frecuencia

La variable *frecuencia*, identificarémos si tiene valores NA:

```
# Identifica si hay valores NA
any(is.na(frecuencia))
```

```
## [1] FALSE
```

```r
# Cuenta la cantidad de NA
sum(is.na(frecuencia))
```

```
## [1] 0
```

Por tanto observamos que nuestra variable *frecuencia* no necesita modificarse.

Dado que ya tenemos nuestra información cargada, ahora realizarémos un análisis descriptivo de la variable *frecuencia*:

```r
# Resumen con datos descriptivos.
summary(frecuencia)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   3.000   5.000   4.925   6.000  12.000
```

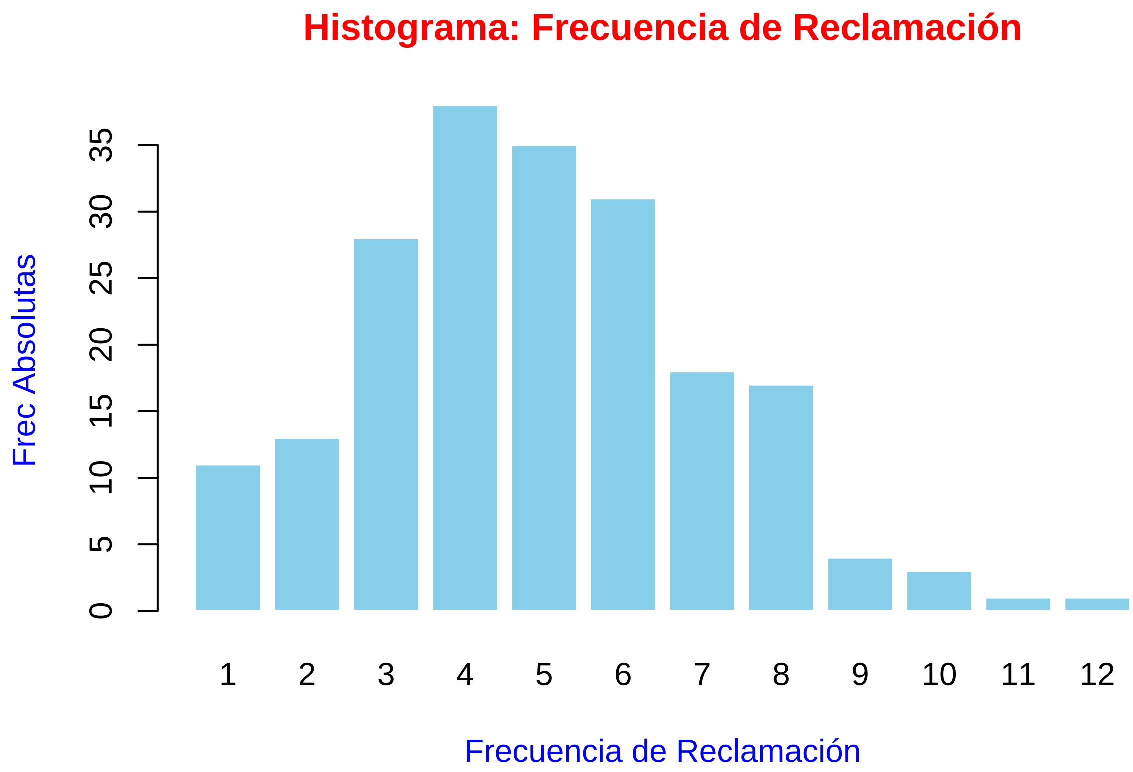
```r
# Varianza de mis datos.
var <- var(frecuencia)
var
```

```
## [1] 4.632538
```

Observamos de la salida anterior, que la media es muy cercana a la mediana, y la varianza es un poco menor a la media. También, se identifican montos mínimos de frecuencia *1* y montos máximos de frecuencia 12. Esto implica que estámos ante la presencia de una distribución sesgada a la derecha.

0.1.1.1. Gráficos de mis datos de Frecuencia En el siguiente gráfico (histograma) se muestra el comportamiento de los datos anteriormente descritos:

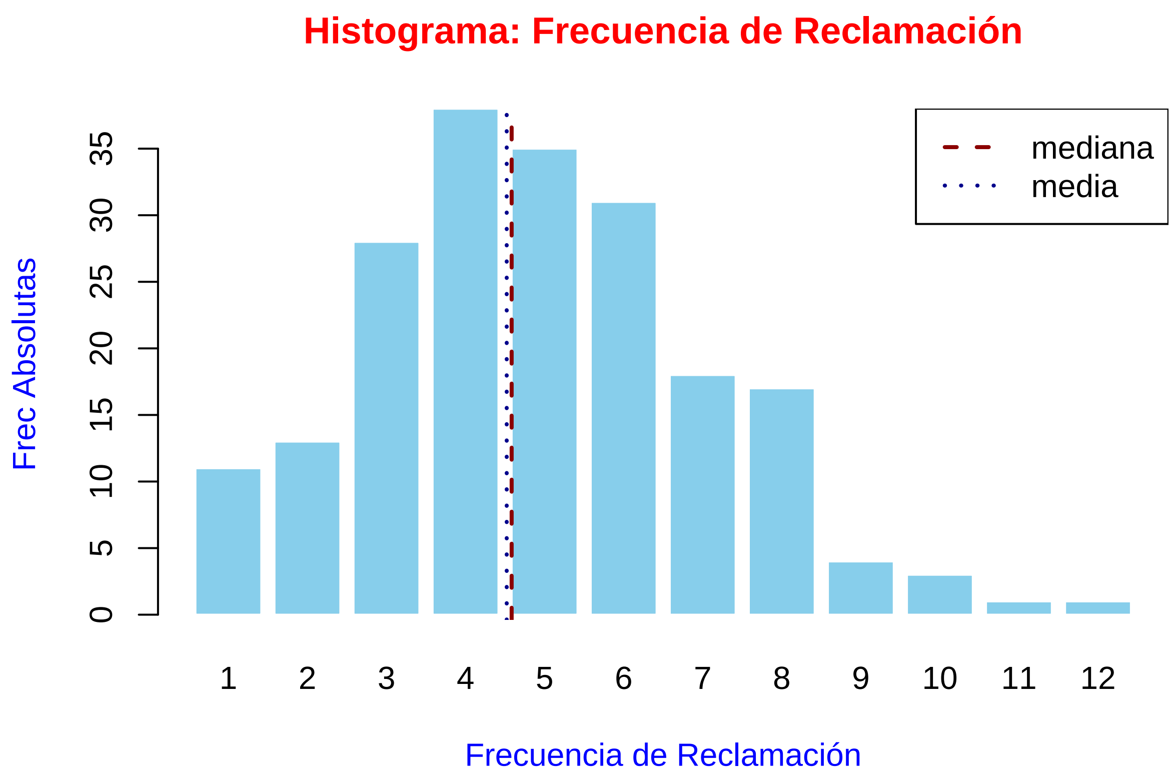
```r
barplot(table(frecuencia),
     col = "skyblue",
     border = "white",
     main="Histograma: Frecuencia de Reclamación",
     ylab = "Frec Absolutas",
     xlab = "Frecuencia de Reclamación",
     col.main="red",
     col.lab="blue"
     )
```



Se observa que el histograma valida el resumen descriptivo de los datos, mostrando que estos están concentrados en la media y mediana. Y tenemos frecuencias de reclamaciones altas con poca frecuencia, y frecuencias de reclamaciones bajas con alta frecuencia.

```
barplot(table(frecuencia),
     col = "skyblue",
     border = "white",
     main="Histograma: Frecuencia de Reclamación",
     ylab = "Frec Absolutas",
     xlab = "Frecuencia de Reclamación",
     col.main="red",
     col.lab="blue"
     )
abline(v=median(frecuencia),
       col = "darkred",
       lwd = 2,
       lty = 2)
# Marcamos la mediana en el histograma.
abline(v=mean(frecuencia),
       col="darkblue",
       lwd = 2,
```

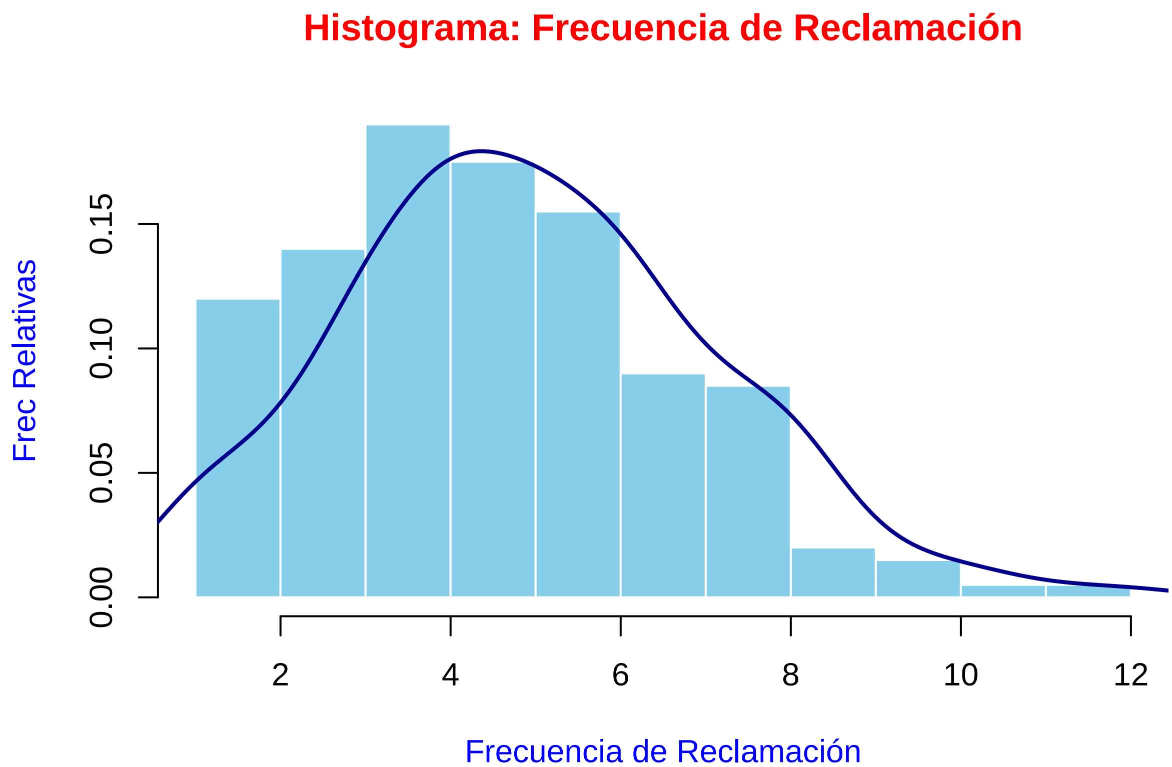
```
       lty=3)
# Marcamos la media en el histograma.
legend("topright",
       legend=c("mediana","media")
       ,col=c("darkred","darkblue"),
       lwd=2,
       lty=c(2,3))
```



Para comprender el comportamiento de los datos, graficaremos la función de densidad, lo que nos permitirá visualizar la distribución de manera continua:

```
# Graficamos el histograma con frecuencias relativas
# Graficamos la función de densidad y el histograma superpuestos.
hist(frecuencia,
     col = "skyblue",
     border = "white",
     main="Histograma: Frecuencia de Reclamación",
     ylab = "Frec Relativas",
     xlab = "Frecuencia de Reclamación",
     col.main="red",
     col.lab="blue",
```

```
       probability = TRUE)
lines(density(frecuencia),
        col="darkblue",
        lwd=2)
```



Se observa una distribución con "cola derecha" más alargada , y la densidad suavizada se ajusta adecuadamente al histograma.

Además, calcularemos la curtosis y el sesgo de los datos para validar nuestra interpretación de la densidad.

```
# Desviación estándar de mis datos.
s <- sd(frecuencia)
# Media de mis datos.
mu <- mean(frecuencia)
# Sesgo.
skew <- sum((frecuencia - mu)^3)/(n*s^3)
skew
```

```
## [1] 0.3583006
```

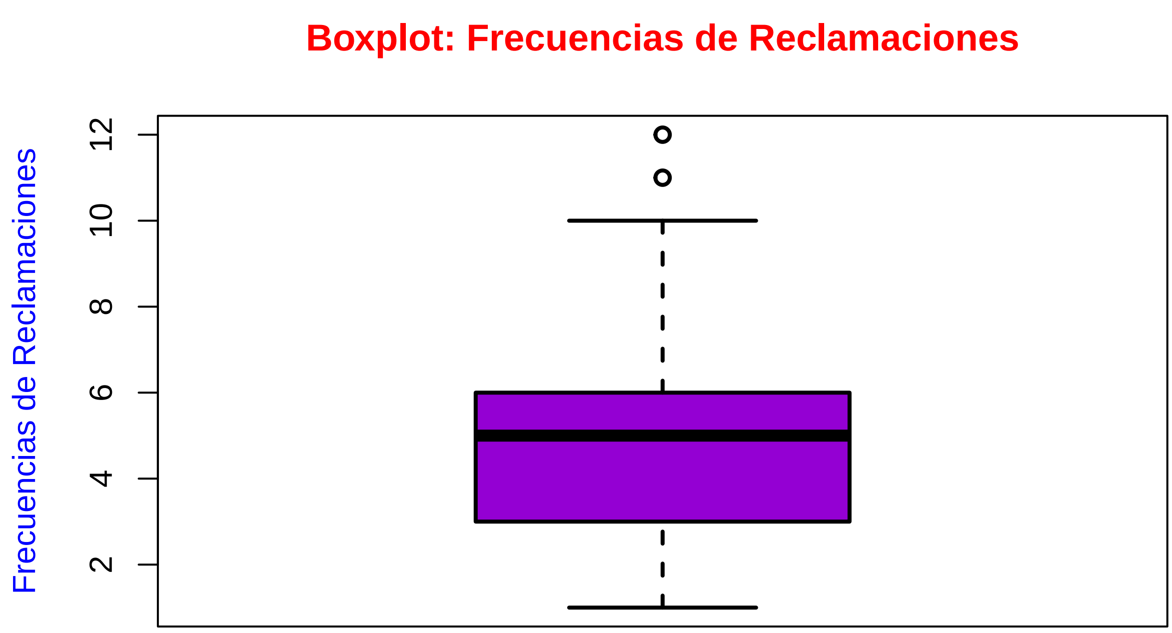
```r
# Curtosis.
kurt <- sum((frecuencia - mu)^4)/(n*s^4) - 3
kurt
```

```
## [1] -0.01071997
```

Dado que el sesgo es positivo, tenemos que la mayoría de los datos están concentrados en el extremo izquierdo de la distribución. Para la curtosis, tenemos que es negativa, es decir, presenta una curva baja con "cola" más corta que la normal.

La descripción anterior también la podemos visualizar en un diagrama de caja:

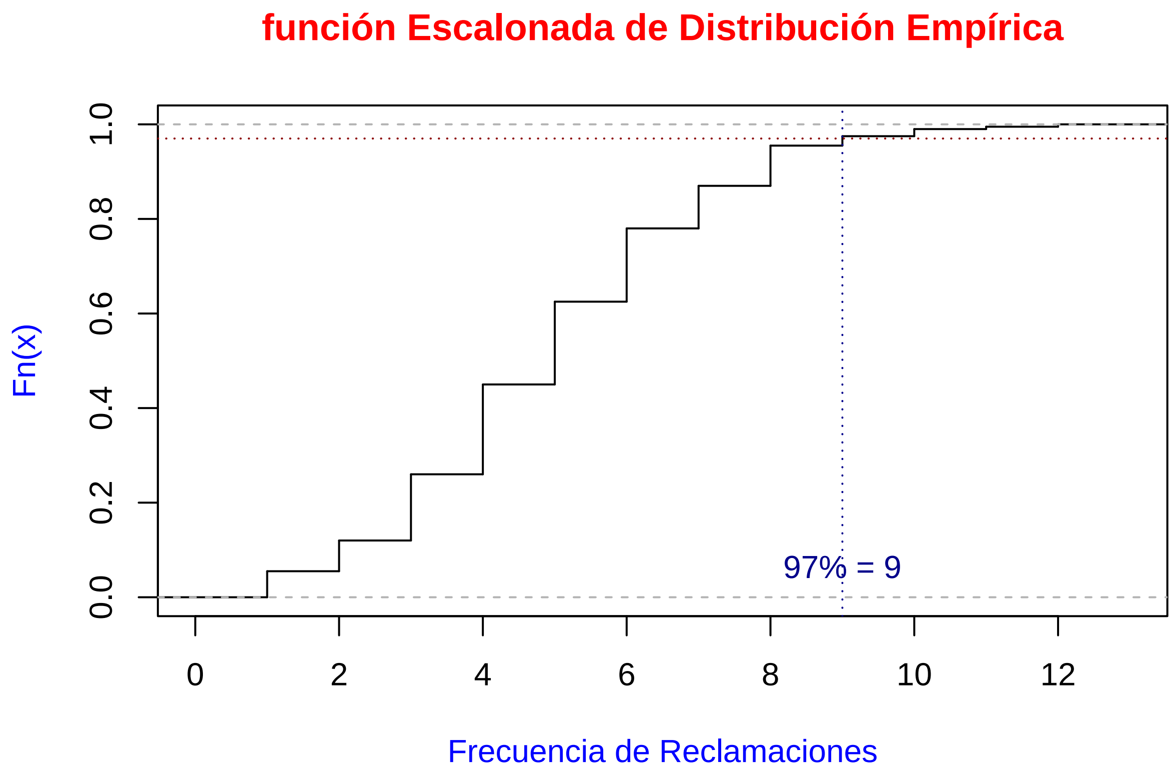
```r
# Diagrama de caja
boxplot(frecuencia,
        col="darkviolet",
        main="Boxplot: Frecuencias de Reclamaciones",
        col.main="red",
        col.lab="blue",
        lwd=2,
        ylab = "Frecuencias de Reclamaciones")
```



El diagrama muestra que tenemos una distribución sesgada hacia la derecha, con cola derecha muy corta, evidenciando también que los montos están más dispersos bajo la mediana. Y, se observan valores que podrían ser clasificados como atípicos.

Para comprender mejor el rango en el que se concentran los montos, graficaremos la función de distribución empírica:

```
#Graficamos la curva de distribución empírica, en conjunto con el percentil 97%.
plot(ecdf(frecuencia),
     main ="función Escalonada de Distribución Empírica",
     xlab = "Frecuencia de Reclamaciones",
     ylab = "Fn(x)",
     col.main="red",
     col.lab="blue",
     verticals = TRUE,
     do.points = FALSE)
abline(h=0.97,
       col ="darkred",
       lty = 3)
abline(v=quantile(frecuencia,0.97),
       col ="darkblue",
       lty = 3)
text(quantile(frecuencia,0.97),
     0,
     labels = paste("97% =", round(quantile(frecuencia,0.97),2)),
     pos=3,
     col="darkblue")
```



Se observa que el 97 % de los montos están distribuidos por debajo de 9 reclamaciones, mientras que solo un 3 % supera dicho valor. También, observamos que las reclamaciones de frecuencias de 3, 4 , 5 y 6 tienen un salto mayor al resto, eso indica que tenemos más reclamaciones con estás frecuencias. En cambio, las reclamaciones con frecuencias de 1, 2 y sobre 9 son muy pocas.

Esta evidencia ratifica lo observado en el diagrama de caja, en relación con la alta dispersión de los datos que son menores a la mediana y la presencia de reclamaciones altas con muy poca frecuencia.

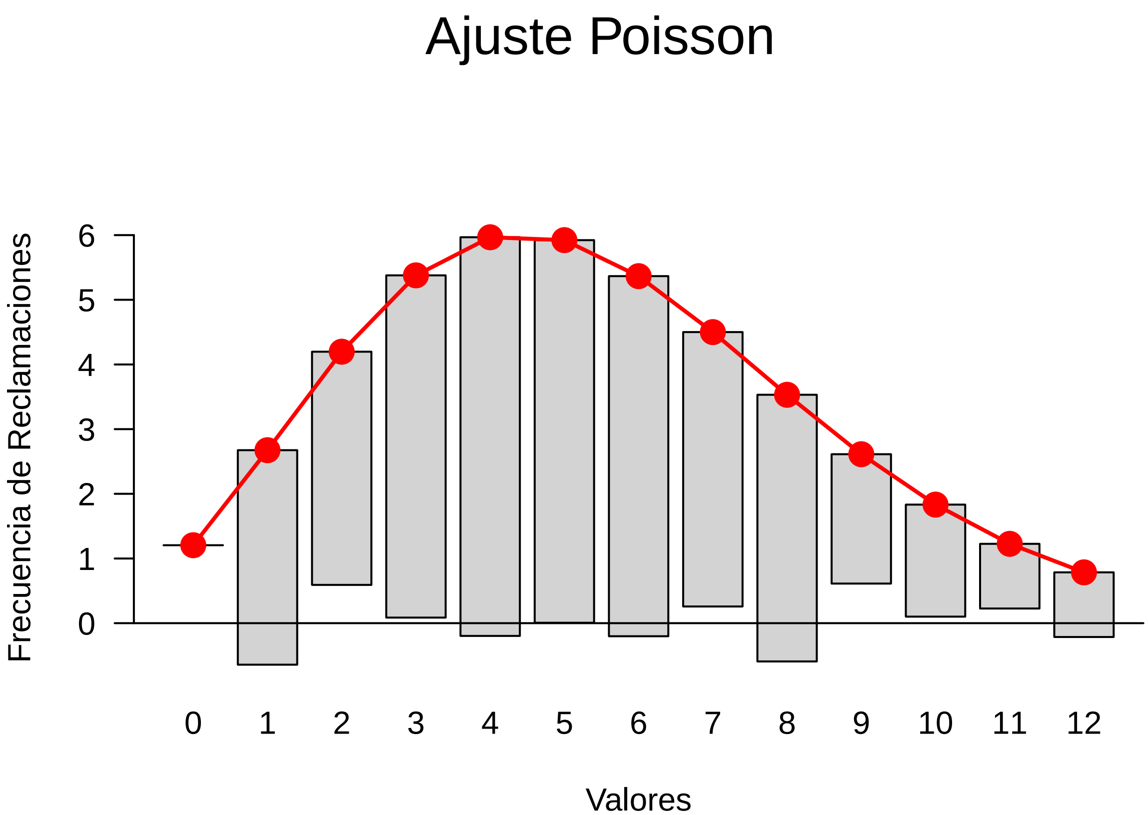
0.1.1.2. Ajuste de densidades a la Frecuencia de Reclamaciones

Primero, se ajustará el modelo teórico Poisson a nuestros datos empíricos, utilizando el método de máxima verosimilitud. Y luego el modelo teórico Binomial.

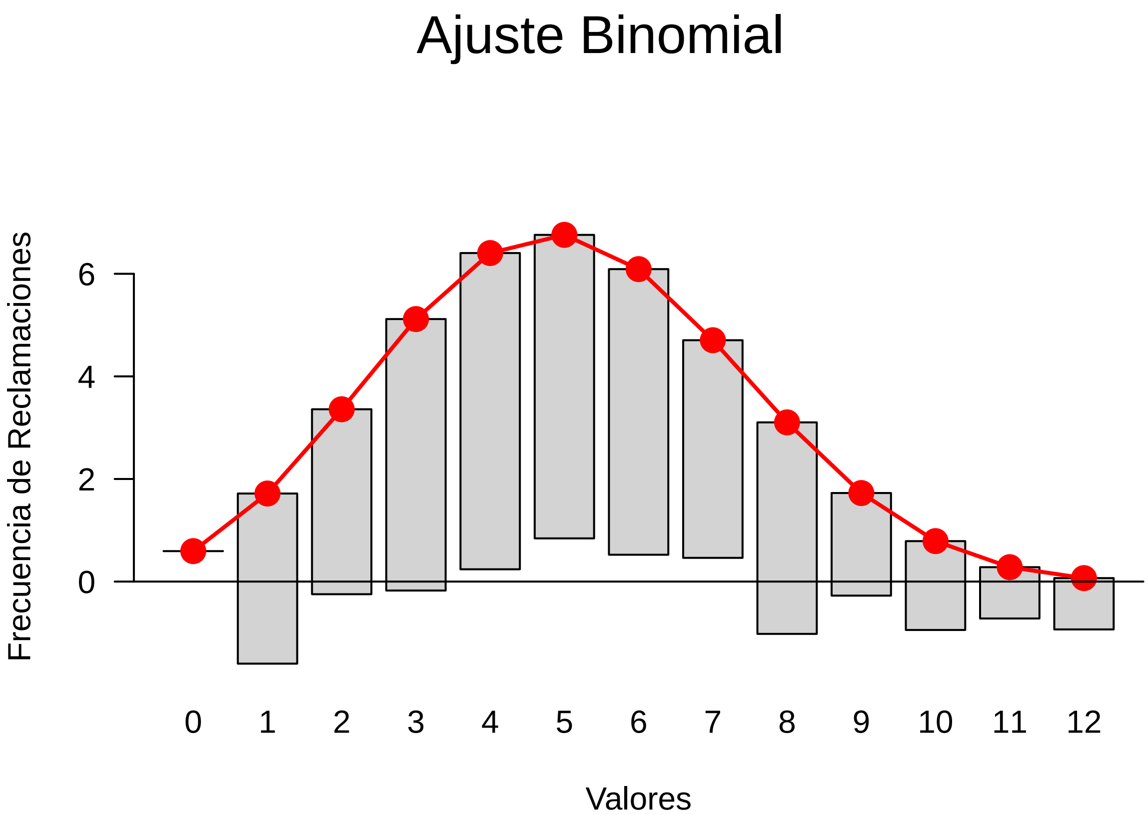
Dado que la varianza es menor a la media descartamos la distribución geométrica y binomial negativa.

```r
# Cargamos la libreria para gráficar
library(vcd)
par(mfrow=c(1,2))
disc.fit_p<-goodfit(frecuencia,type = "poisson", method = "ML")
plot(disc.fit_p,
     main = "Ajuste Poisson",
     xlab = "Valores",
     ylab="Frecuencia de Reclamaciones"
```

```
    )
```




```r
disc.fit_b<-goodfit(frecuencia,type = "binomial", method = "ML")
plot(disc.fit_b,
     main = "Ajuste Binomial",
     xlab = "Valores",
     ylab="Frecuencia de Reclamaciones")
```



Se obseva en los gráficos las barras con las frecuencias observadas, y los puntos rojos las frecuencias esperadas, según poisson o binomial.

Vemos que hay un buen ajuste entre lo observado y esperado. Aún teniendo ciertas discrepancias en algunas barras, que son tolerables.

0.1.1.3. Test Bondad de Ajuste Luego de ajustar el modelo teórico con los datos empíricos, realizarémos el test de bondad de ajuste para evaluar si dicho modelo se ajusta a nuestros datos empíricos.

Se calcularán los test de Chi-cuadrado:

```
# Test estadístico para la distribución Poisson.
summary(disc.fit_p)
```

```
##
##   Goodness-of-fit test for poisson distribution
##
##                         X^2 df  P(> X^2)
## Likelihood Ratio 10.61574 10 0.3882278
```

Y luego el test estadístico para la distribución binomial:

```r
summary(disc.fit_b)
```

```
##
##   Goodness-of-fit test for binomial distribution
##
##                       X^2 df     P(> X^2)
## Likelihood Ratio 40.38502 10 1.449324e-05
```

Dado que el valor-p de la distribución Poisson es mayor al 5 %, se acepta esta distribución para nuestros datos empíricos.

0.1.1.4. Comparación empírica versus teórica a la Frecuencia de Reclamaciones

Vamos a comparar nuestra distribución teórica y nuestros datos empíricos.

```r
# Datos observados
obs <- table(frecuencia)
# Valores que tiene la variable frecuencia
value <- as.numeric(names(obs))
# Frecuencia relativa
prop_obs <- obs/sum(obs)
```

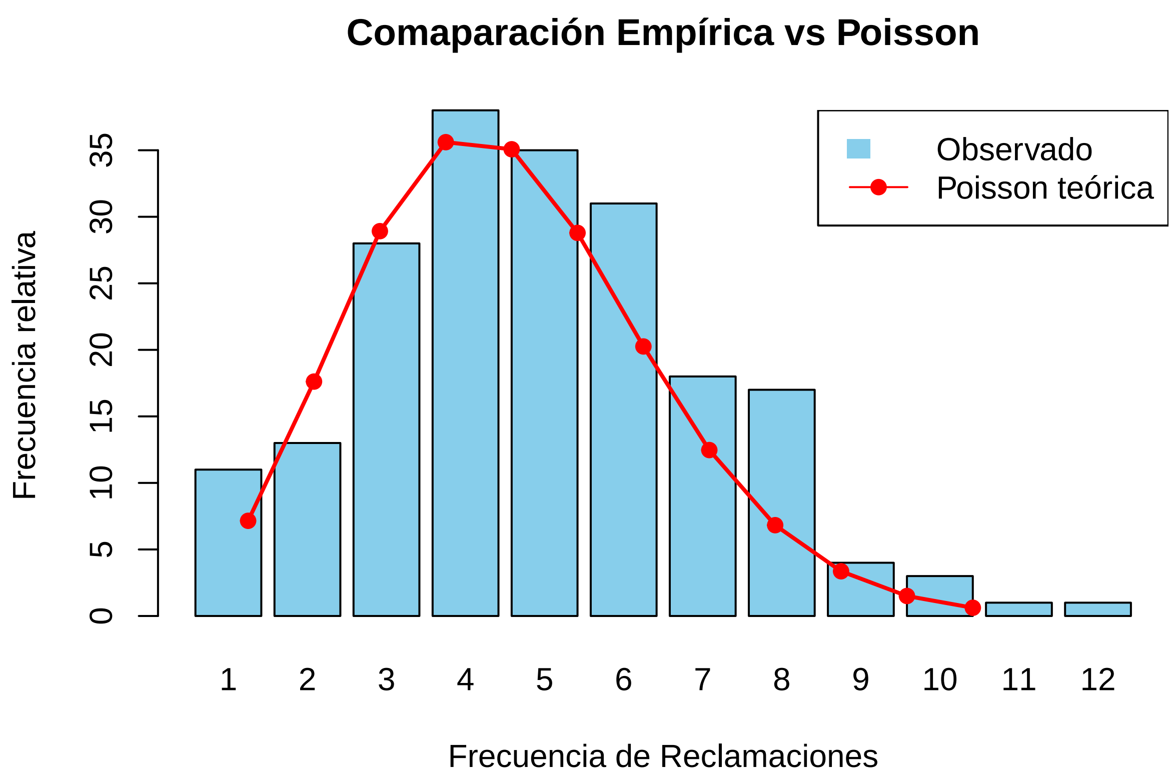
Ahora vamos a estimar el parámetro de la distribución poisson:

```r
# Parámetro lambda usamos la media por MLE
lambda <- mean(frecuencia)
# probabilidad teórica
prob_teo <- dpois(value, lambda = lambda)
# frecuencia esperada
esperadas <- prob_teo*length(frecuencia)
```

Ahora por último vamos a realizar la comparación gráfica:

```r
barplot(obs,
        col = "skyblue",
        main = "Comaparación Empírica vs Poisson",
        ylab = "Frecuencia relativa",
        xlab = "Frecuencia de Reclamaciones")
points(value,
      esperadas,
      col = "red",
      pch=19)
lines(value,
      esperadas,
```

```r
      col ="red",
      lwd=2)
legend("topright", legend=c("Observado", "Poisson teórica"),
       fill=c("skyblue", NA), border=NA, lty=c(NA,1), col=c("skyblue","red"), pch=c(NA,19))
```



Se observa que la distribución Poisson cumple en ser la distribución de nuestros datos, coincidiendo con el test de bondad de ajuste.

0.1.2. Descripción de mis datos de Severidad

La variable *severidad*, identificarémos si tiene valores NA:

```r
# En el siguiente vector concatenearémos las columnas de severidad
severidad<-c(data[,2],data[,3],data[,4],data[,5],data[,6],data[,7],data[,8],data[,9],data[,10],data[,11
```

Realizando un resumen de *severidad*:

```r
summary(severidad)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.     NA's
##    0.065   31.759   78.606  149.151  177.464 2369.608     1216
```

Se identifican 1216 casos NA's en el dataframe de *severidad.*

Vamos a filtrar los datos de los NA:

```
severidad1<-severidad[!is.na(severidad)]
length(severidad1)
```

```
## [1] 984
```

Tenemos 984 datos sin NA.

Dado que ya tenemos nuestra información cargada, ahora realizarémos un análisis descriptivo de la variable *severidad*:

```
# Resumen con datos descriptivos.
summary(severidad1)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##    0.065   31.759   78.606  149.151  177.464 2369.608
```

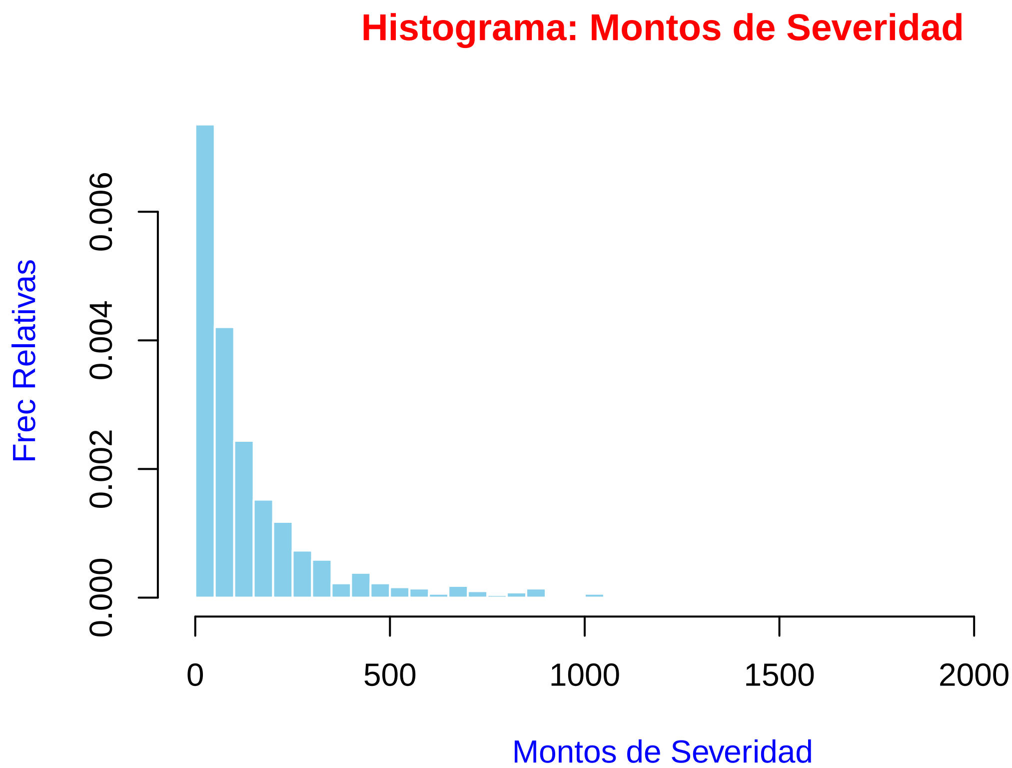
```
# Varianza de mis datos.
var <- var(severidad1)
var
```

```
## [1] 46281.25
```

Observamos de la salida anterior, que la media es aproximadamente dos veces a la mediana, y la varianza es mayor a la media. Esto implica que estámos ante la presencia de una distribución sesgada a la derecha.

0.1.2.1. Gráficos de mis datos de Severidad En el siguiente gráfico (histograma) se muestra el comportamiento de los datos anteriormente descritos:

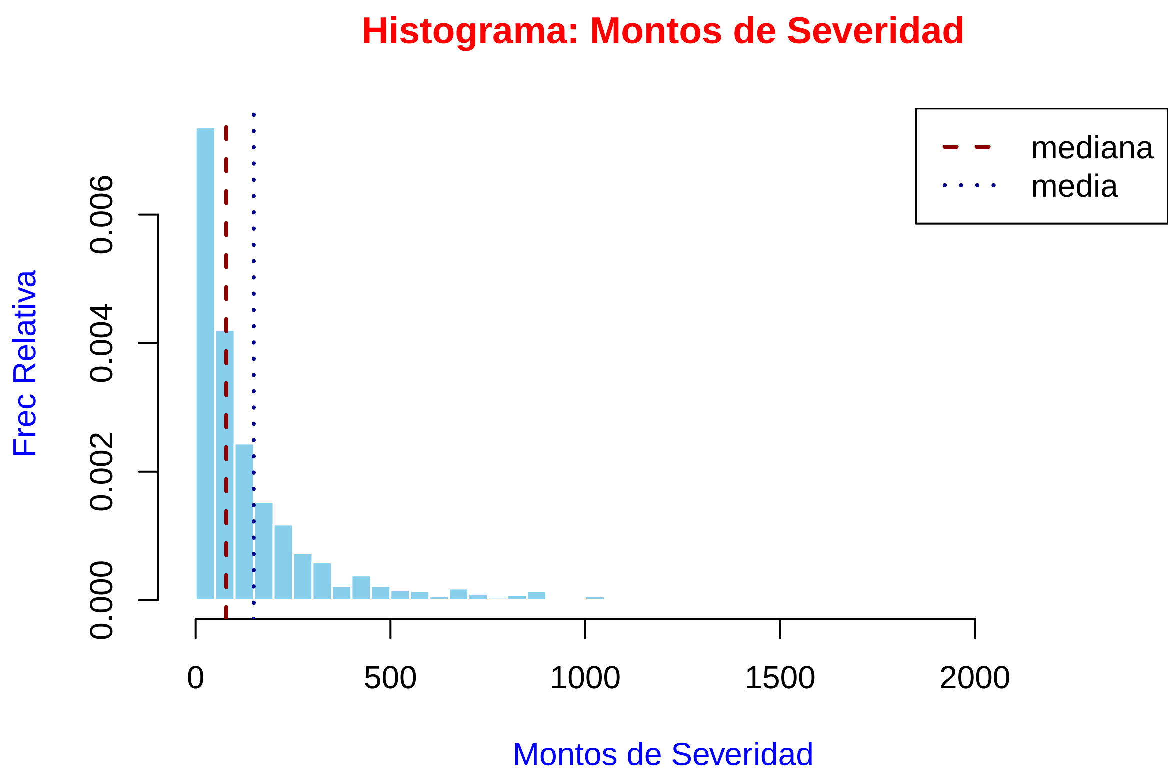
```
hist(severidad1,
     col = "skyblue",
     border = "white",
     main="Histograma: Montos de Severidad",
     ylab = "Frec Relativas",
     xlab = "Montos de Severidad",
     col.main="red",
     col.lab="blue",
     breaks=50,
     probability = TRUE
     )
```



Se observa que el histograma valida el resumen descriptivo de los datos, mostrando una alta dispersión y una distribución con "cola derecha".

```
hist(severidad1,
     col = "skyblue",
     border = "white",
     main="Histograma: Montos de Severidad",
     ylab = "Frec Relativa",
     xlab = "Montos de Severidad",
     col.main="red",
     col.lab="blue",
     breaks=50,
     probability = TRUE)
abline(v=median(severidad1),
       col = "darkred",
       lwd = 2,
       lty = 2)
# Marcamos la mediana en el histograma.
abline(v=mean(severidad1),
       col="darkblue",
       lwd = 2,
```

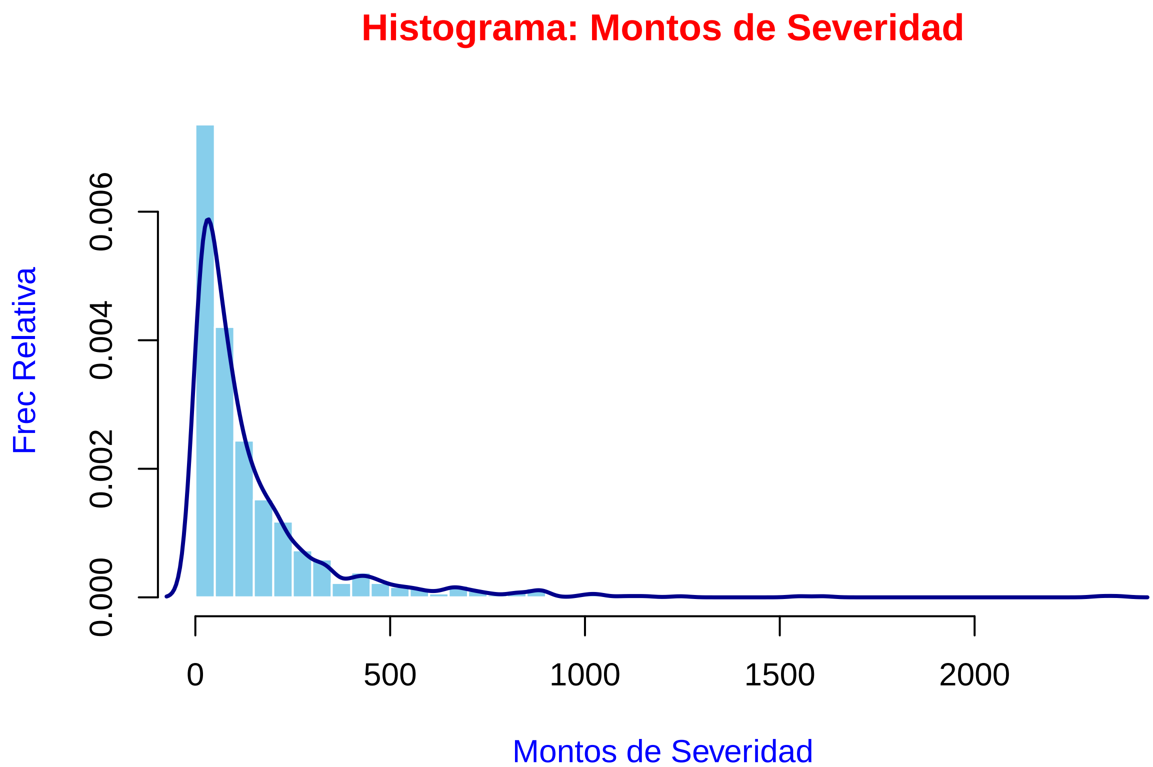
```
        lty=3)
# Marcamos la media en el histograma.
legend("topright",
        legend=c("mediana","media")
        ,col=c("darkred","darkblue"),
        lwd=2,
        lty=c(2,3))
```



Para comprender el comportamiento de los datos, graficaremos la función de densidad, lo que nos permitirá visualizar la distribución de manera continua:

```
# Graficamos la función de densidad y el histograma superpuestos.
hist(severidad1,
     col = "skyblue",
     border = "white",
     main="Histograma: Montos de Severidad",
     ylab = "Frec Relativa",
     xlab = "Montos de Severidad",
     col.main="red",
     col.lab="blue",
     breaks=50,
```

```
     probability = TRUE)
lines(density(severidad1),
      col="darkblue",
      lwd=2)
```



Se observa una distribución con "cola derecha", y la densidad suavizada se ajusta adecuadamente al histograma. Esta densidad nos permite identificar la presencia de montos grandes, superiores a 1000.

Además, calcularemos la curtosis y el sesgo de los datos para validar nuestra interpretación de la densidad.

```
s <- sd(severidad1) # Desviación estándar de mis datos.
mu <- mean(severidad1) # Media de mis datos.
skew <- sum((severidad1 - mu)^3)/(n*s^3)
skew # Sesgo.
```

```
## [1] 20.66907
```

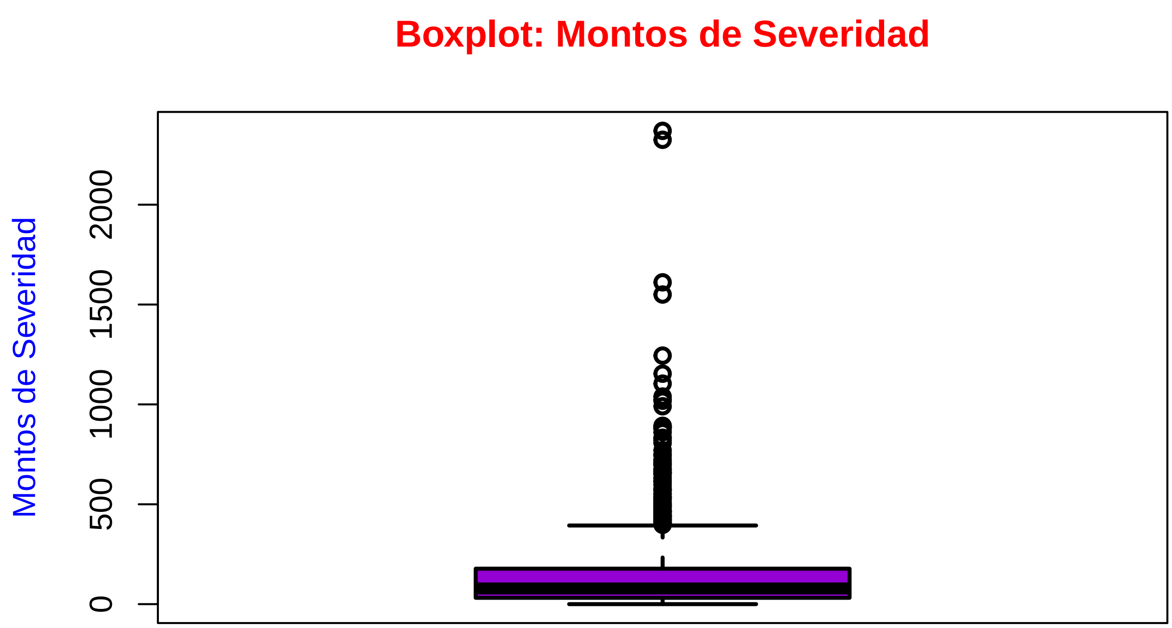
```
kurt <- sum((severidad1 - mu)^4)/(n*s^4) - 3
kurt # Curtosis.
```

```
## [1] 151.0067
```

Dado que el sesgo es positivo, se interpreta como una cola más larga a la derecha. En el caso de la curtosis, se interpreta como una cola derecha más pesada.

La descripción anterior también la podemos visualizar en un diagrama de caja:

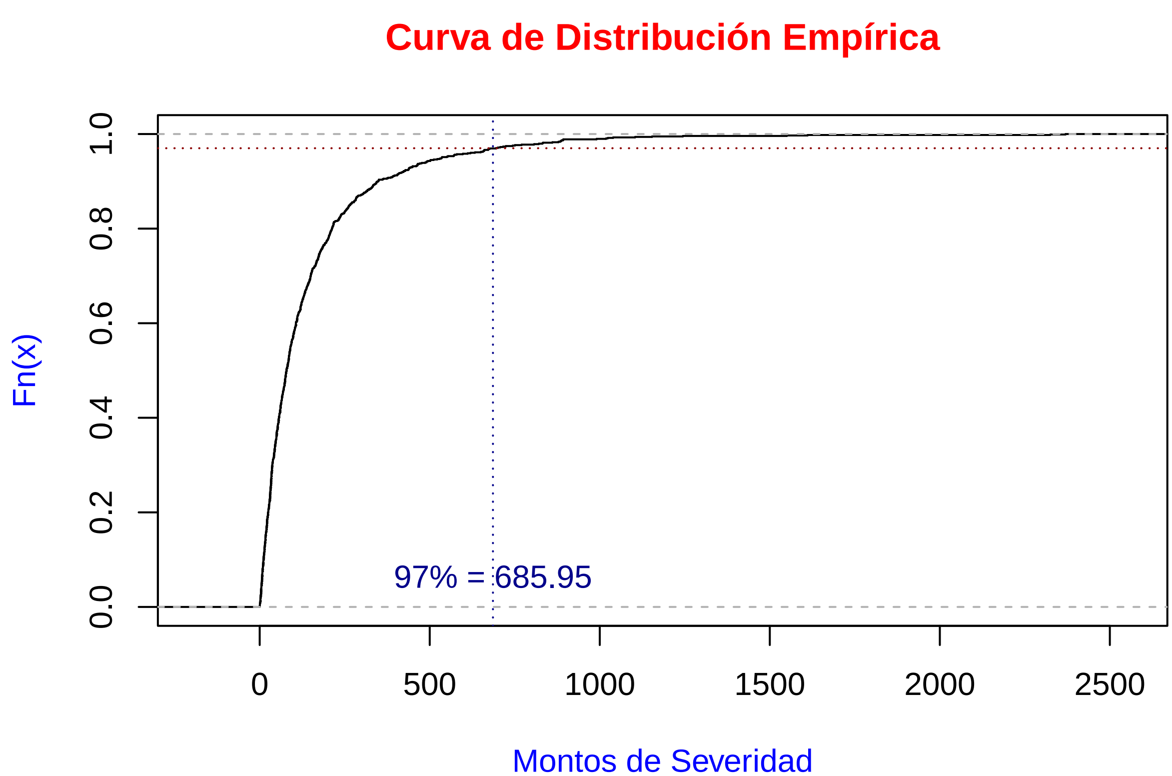
```
# Diagrama de caja
boxplot(severidad1,
        col="darkviolet",
        main="Boxplot: Montos de Severidad",
        col.main="red",
        col.lab="blue",
        lwd=2,
        ylab = "Montos de Severidad")
```



El diagrama muestra que tenemos una distribución fuertemente sesgada hacia la derecha, evidenciando una alta dispersión de los montos en relación con la media. Y, se observan valores que podrían ser clasificados como atípicos.

Para comprender mejor el rango en el que se concentran los montos, graficaremos la función de distribución empírica:

```r
#Graficamos la curva de distribución empírica, en conjunto con el percentil 97%.
plot(ecdf(severidad1),
     main ="Curva de Distribución Empírica",
     xlab = "Montos de Severidad",
     ylab = "Fn(x)",
     col.main="red",
     col.lab="blue",
     verticals = TRUE,
     do.points = FALSE)
abline(h=0.97,
       col ="darkred",
       lty = 3)
abline(v=quantile(severidad1,0.97),
       col ="darkblue",
       lty = 3)
text(quantile(severidad1,0.97),
     0,
     labels = paste("97% =", round(quantile(severidad1,0.97),2)),
     pos=3,
     col="darkblue")
```



Se observa que el 97 % de los montos están distribuidos por debajo de 685.95, mientras que solo un 3 % supera dicho valor. Esta evidencia ratifica lo observado en el diagrama de caja, en relación con la alta dispersión de los montos y su mayor concentración en el extremo izquierdo de la distribución.

0.1.2.2. Ajuste de densidades a la Severidad Una vez realizada la descripción de nuestros datos, vamos a comparar los cuantiles teóricos versus los cuantiles empíricos, con el objetivo para decidir qué distribución teórica se ajusta mejor a nuestros datos observados.

En las siguientes gráficas se explorarán las distribuciones Log Normal, Weibull, Gamma, y Exponencial, comparando los cuantiles de cada una con los cuantiles empíricos.

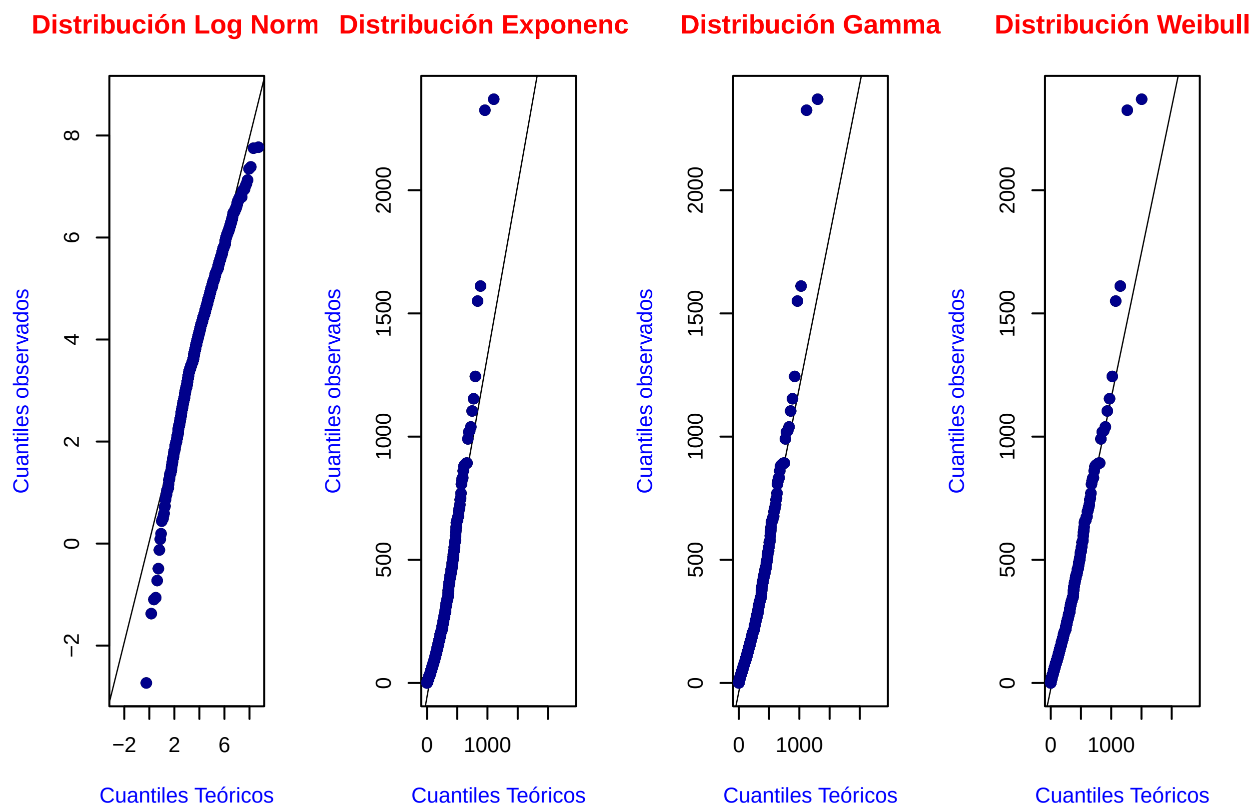
```
library(EnvStats) # Cargamos la libreria para gráficar qqplot de distribuciones.
par(mfrow=c(1,4))
# Gráfica de los cuantiles teóricos y observados de la distribución Log Normal
points(EnvStats:: qqPlot(severidad1, dist="lnorm",
                         estimate.params = TRUE,
                         add.line = TRUE,
                         xlab = "Cuantiles Teóricos",
                         ylab = "Cuantiles observados",
                         main = "Distribución Log Normal",
                         col.main="red",
                         col.lab="blue"),
       col = "darkblue",
       pch = 16,
       cex = 1.2)

# Gráfica de los cuantiles teóricos y observados de la distribución Exponencial.
points(EnvStats:: qqPlot(severidad1,
                         dist="exp",
                         estimate.params = TRUE,
                         add.line = TRUE,
                         xlab = "Cuantiles Teóricos",
                         ylab = "Cuantiles observados",
                         main = "Distribución Exponencial",
                         col.main="red",
                         col.lab="blue"),
       col = "darkblue",
       pch = 16,
       cex = 1.2)

# Gráfica de los cuantiles teóricos y observados de la distribución Gamma.
points(EnvStats:: qqPlot(severidad1,
                         dist="gamma",
                         estimate.params = TRUE,
```

```
                         add.line = TRUE,
                         xlab = "Cuantiles Teóricos",
                         ylab = "Cuantiles observados",
                         main = "Distribución Gamma" ,
                         col.main="red",
                         col.lab="blue"),
       col = "darkblue",
       pch = 16,
       cex = 1.2)


# Gráfica de los cuantiles teóricos y observados de la distribución Weibull.
points(EnvStats:: qqPlot(severidad1,
                         dist="weibull",
                         estimate.params = TRUE,
                         add.line = TRUE,
                         xlab = "Cuantiles Teóricos",
                         ylab = "Cuantiles observados",
                         main = "Distribución Weibull" ,
                         col.main="red",
                         col.lab="blue"),
       col = "darkblue",
       pch = 16,
       cex = 1.2)
```



Se observa que la distribución Log Normal presenta una curva en una de sus colas, indicando un desajuste en los extremos.

De las distribuciones Exponencial, Gamma, y Weibull, se observa que esta última presenta un mejor ajuste entre los cuantiles teórico y empíricos, aún existiendo un desajuste en la diagonal y en valores extremos.

0.1.2.3. Estimación de parámetros a la Severidad A partir de la comparación entre los cuantíles teóricos y empíricos, se observó que la distribución la distribución weibull presentan un mejor ajuste.

Primero, se ajustará el modelo teórico Weibull a nuestros datos empíricos, utilizando el método de máxima verosimilitud.

```r
# Cargamos la libreria para estimar los parámetros.
library(fitdistrplus)
# Ajusta la distribución weibull a mis datos.
fit.weibull <- fitdist(severidad1,
                       "weibull",
                       method="mle")
```

0.1.2.4. Test Bondad de Ajuste a la Severidad Luego de estimar los parámetros, realizarémos el test de bondad de ajuste para evaluar si dicho modelo se ajusta a nuestros datos empíricos.

Se calcularán los test de Kolmogorov-Smirnoff y de Anderson-Darling para la distribución Weibull.

```r
library(ADGofTest)
library(goftest)
# Test de bondad de ajuste Kolmogorov-Smirnoff
ks.test(severidad1,
        "pweibull",
        shape = fit.weibull$estimate["shape"],
        scale = fit.weibull$estimate["scale"])
```

```
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  severidad1
## D = 0.038172, p-value = 0.1136
## alternative hypothesis: two-sided
```

```r
# Test de bondad de ajuste Anderson-Darling
ad.test(severidad1,
        "pweibull",
        shape = fit.weibull$estimate["shape"],
        scale = fit.weibull$estimate["scale"])
```

```
##
##  Anderson-Darling test of goodness-of-fit
##  Null hypothesis: Weibull distribution
##  with parameters shape = 0.825380104968049, scale = 132.976905877508
##  Parameters assumed to be fixed
##
## data:  severidad1
## An = 3.0562, p-value = 0.02564
```

A partir de los resultados obtenidos, la estimación de parámetros de la distribución weibull se observa que en el test KS un valor p-value superior a un 5 %, sin embargo, en el test AD tenemos un valor p-value menor.

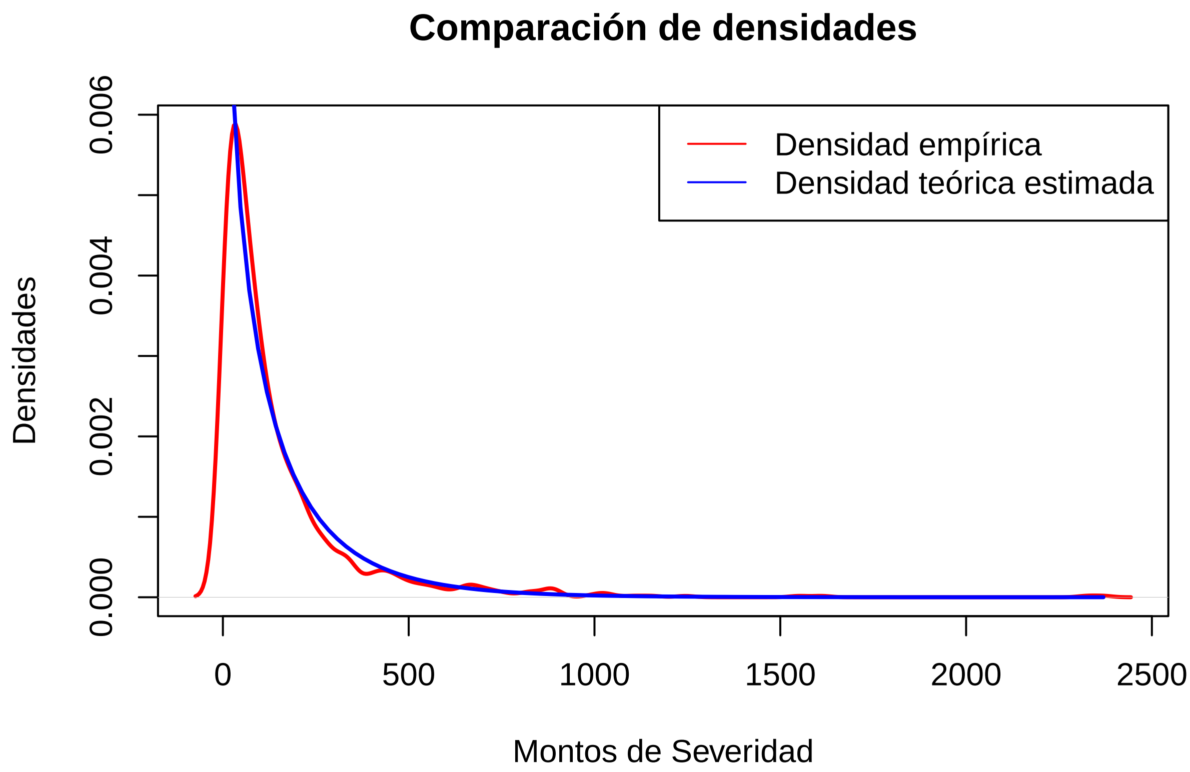
Del test de AD podemos decir que los valores extremos no se ajusta adecuadamente a la distribución.

Así podemos concluir que la mejor distribución para nuestros datos es la distribución Weibull.

0.1.2.5. Comparación empírica versus teórica a la Severidad Dado que ya tenemos los parámetros de nuestro modelo estimados, vamos a comparar el ajuste entre las funciones de densidad, la función de distribución y los percentiles.

Comenzamos con la funcion de densidad:

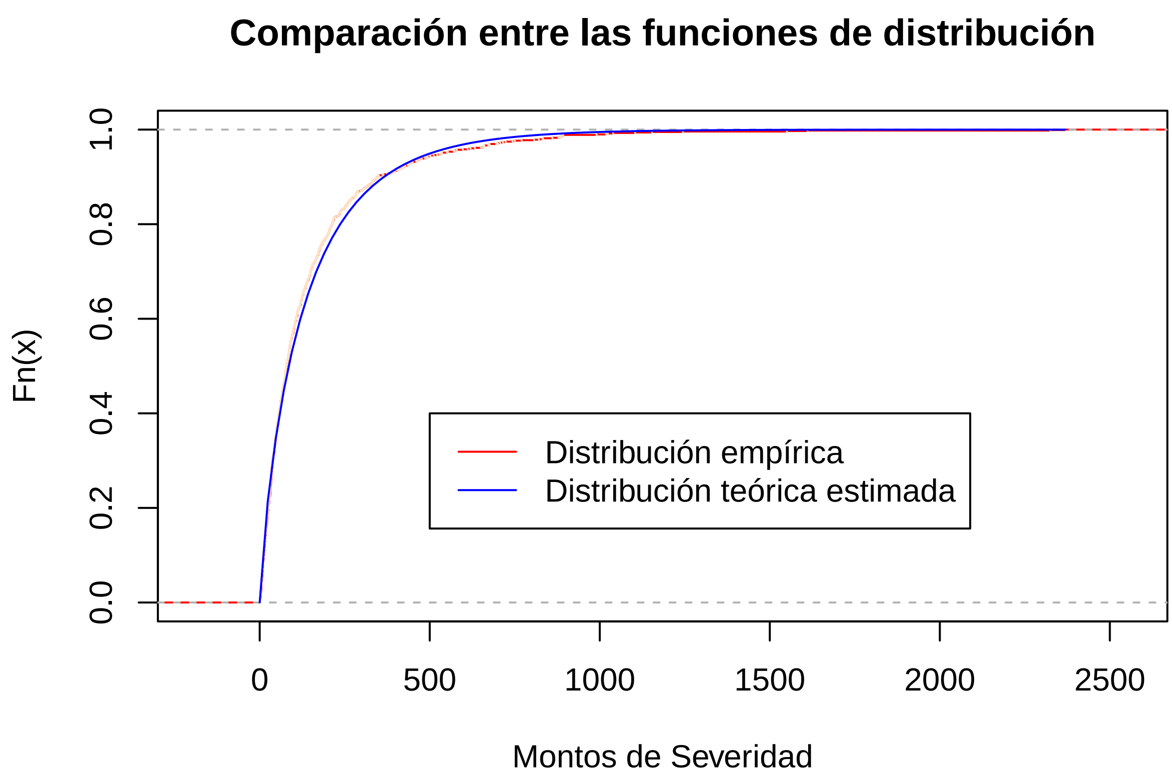
```r
# Comparación densidades empírica vs teórica
plot(density(severidad1),
     col="red",
     main="Comparación de densidades",
     xlab="Montos de Severidad",
     ylab="Densidades",
     lwd=2)
curve(dweibull(x,
               shape = fit.weibull$estimate["shape"],
        scale = fit.weibull$estimate["scale"]),
      from=0,
      to=max(severidad1),
      add=TRUE,
      col="blue",
      lwd=2)
k<-c ("Densidad empírica",
      "Densidad teórica estimada")
legend ("topright",
        paste(k),
        lty=1,
        col=c("red","blue"))
```



Luego la comparación de la función de distribución:

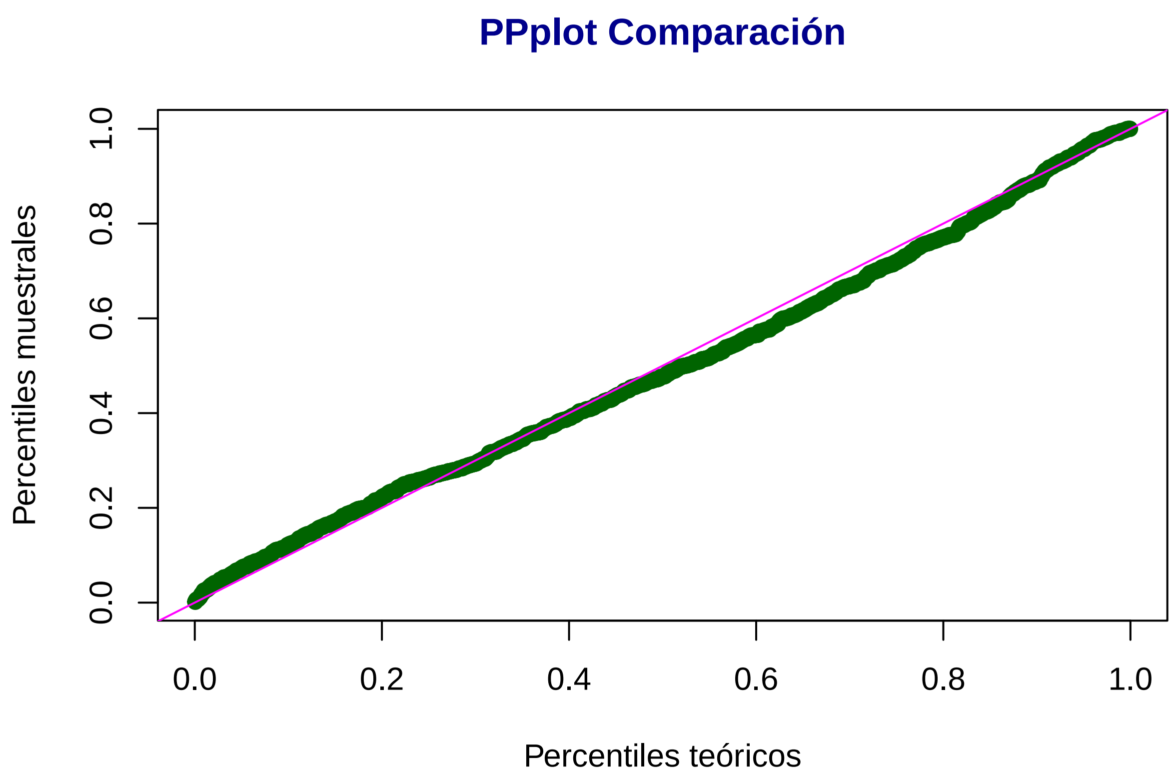
```
# Comparación entre las funciones de distribucion
plot(ecdf(severidad1),
     col.hor="red",
     col.vert="bisque",
     main="Comparación entre las funciones de distribución",
     xlab = "Montos de Severidad",
     ylab = "Fn(x)",
     verticals = TRUE,
     do.points = FALSE)
curve(pweibull(x,
               shape = fit.weibull$estimate["shape"],
        scale = fit.weibull$estimate["scale"]),
      from=0,
      to=max(severidad1),
      add=TRUE,
      col="blue")
k<-c ("Distribución empírica",
      "Distribución teórica estimada")
legend (500,0.4,
```

```
        paste(k),
        lty=1,
        col=c("red","blue"))
```



Y por último, la comparación de percentiles:

```
# Comparación entre percentiles
plot(ppoints(length(severidad1)),
     sort(pweibull(severidad1,
                   shape = fit.weibull$estimate["shape"],
        scale = fit.weibull$estimate["scale"])),
     pch=19,
     col="darkgreen",
     main="PPplot Comparación",
     xlab="Percentiles teóricos",
     ylab="Percentiles muestrales",
     col.main="darkblue")
abline(c(0,1),
       col="magenta")
```



Se observa que la distribución Log normal cumple en ser la distribución de nuestros datos, validando el test de bondad de ajuste, dado que la comparación entre la función de densidad, la función de distribución y la comparación de percentiles cumple un buen ajuste a los datos empíricos.

0.2. Metodos de Aproximación

En esta sección vamos a ajustar un modelo compuesto para la distribución de las pérdidas agregadas, utilizando los métodos de aproximación Normal, de Panjer y de Simulación.

0.2.1. Metodo de Aproximación Normal

El modelo de frecuencia, habíamos definido la distribución Poisson:

```
mod_frec <- dpois(severidad1,
                  lambda = mean(frecuencia))
```

El modelo de severidad, habíamos definido la distribución Weibull:

```
mod_sev <- dweibull(severidad1,
                    shape = fit.weibull$estimate["shape"],
                    scale = fit.weibull$estimate["scale"])
```

Vamos a calcular los momentos media y varianza de la variable severiad.

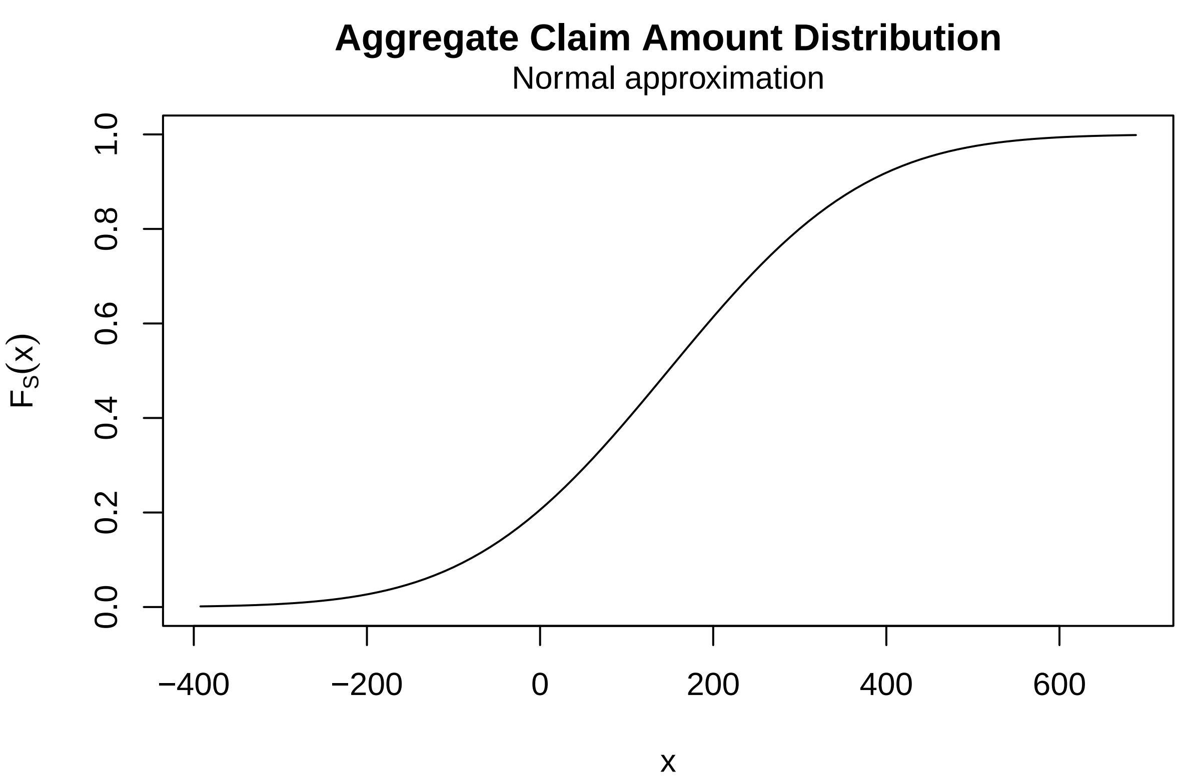
```
mean_sev <- mean(rweibull(10000,
                          shape = fit.weibull$estimate["shape"],
                          scale = fit.weibull$estimate["scale"]))
var_sev <- var(rweibull(10000,
                          shape = fit.weibull$estimate["shape"],
                          scale = fit.weibull$estimate["scale"]))
```

Por último, vamos a usar la función ***aggregateDist*** con el método **normal**.

```
library(actuar)
result <- aggregateDist(method = "normal",
                        model.freq = mod_frec,
                        modelo.sev = mod_sev,
                        moments = c(mean_sev,var_sev))
```

Ahora vamos a gráficar la función de dsitribución acumulada.

```
plot(result)
```



0.2.2. Metodo de Aproximación Panjer

Para la utilización de este método vamos a discretizar la distribución Weibull.

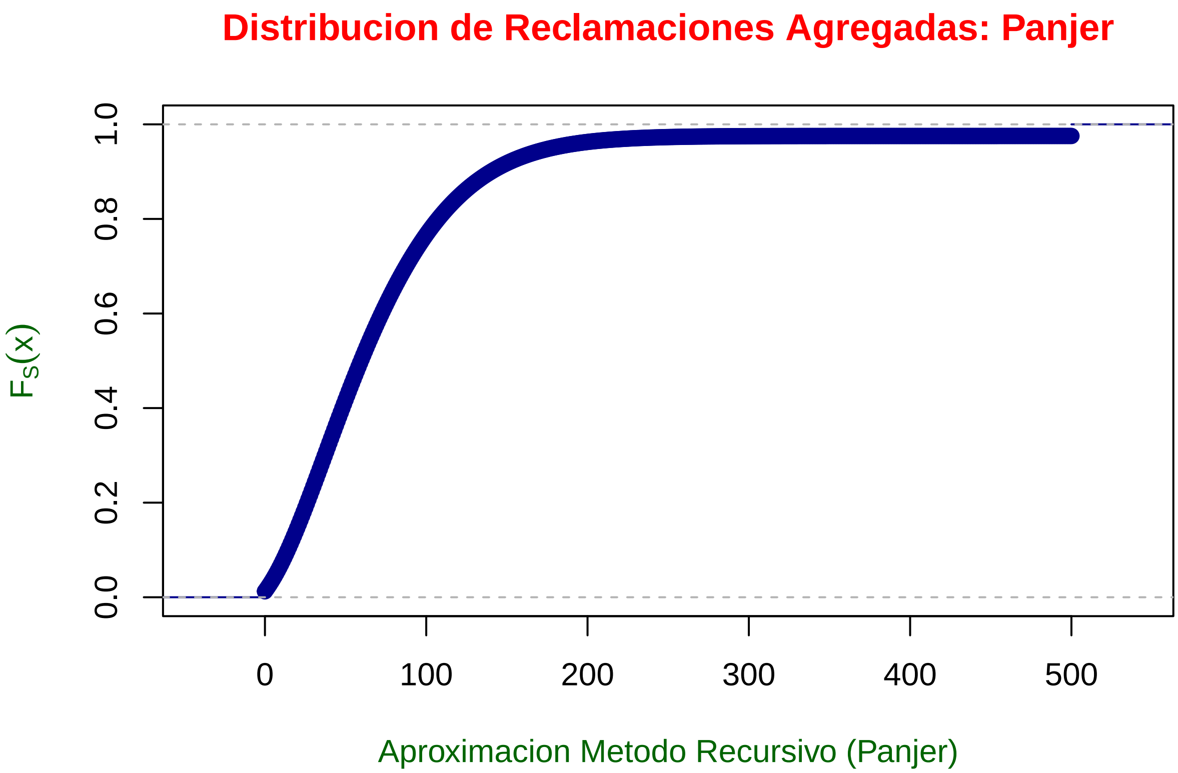
```
# Tamaño de discretización
x.scale <- 10
# Monto de valor máximo de pérdidas
max.loss <- 1000
# Creamos una secuencia
point <- seq(0, max.loss, by = x.scale)
# Calculamos las probabilidades
prob <- diff(pweibull(point,
                      shape = fit.weibull$estimate["shape"],
                      scale = fit.weibull$estimate["scale"]))
```

Método Panjer: Es recursivo, y el modelo de frecuencias es poisson, con la discretización de la distribución weibull.

```
library(actuar)
result.p <- aggregateDist(
  method = "recursive",
  model.freq = "poisson",
  model.sev = prob,
  lambda = mean(frecuencia))
```

Graficamos la distribución de reclamaciones agregadas:

```
plot(result.p,
     main="Distribucion de Reclamaciones Agregadas: Panjer",
     col="darkblue",
     col.main="red",
     sub="",
     xlab= "Aproximacion Metodo Recursivo (Panjer)",
     col.lab="darkgreen")
```



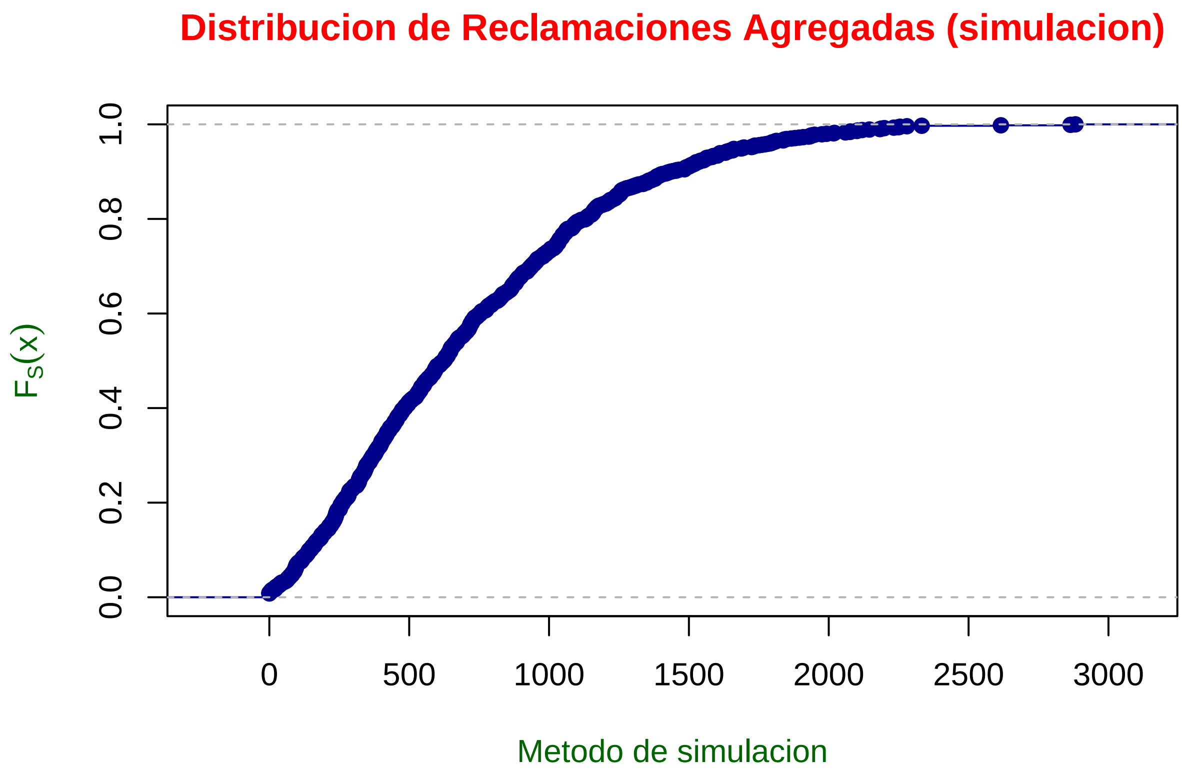
0.2.3. Metodo de Aproximación Simulación

A continuación vamos a realizar la aproximación con el método de simulación, manteniendo los modelos de frecuencia y de severidad del método de aproximación normal.

```
# realizarémos 10.000 simulaciones
model.freq<-expression(severidad1=rpois(mean(frecuencia)))
model.sev<-expression(severidad1=rweibull(shape= fit.weibull$estimate["shape"],scale=fit.weibull$estima
result.sim<-aggregateDist("simulation",nb.simul=1000,model.freq,model.sev)
```

Vamos a graficar la simulación:

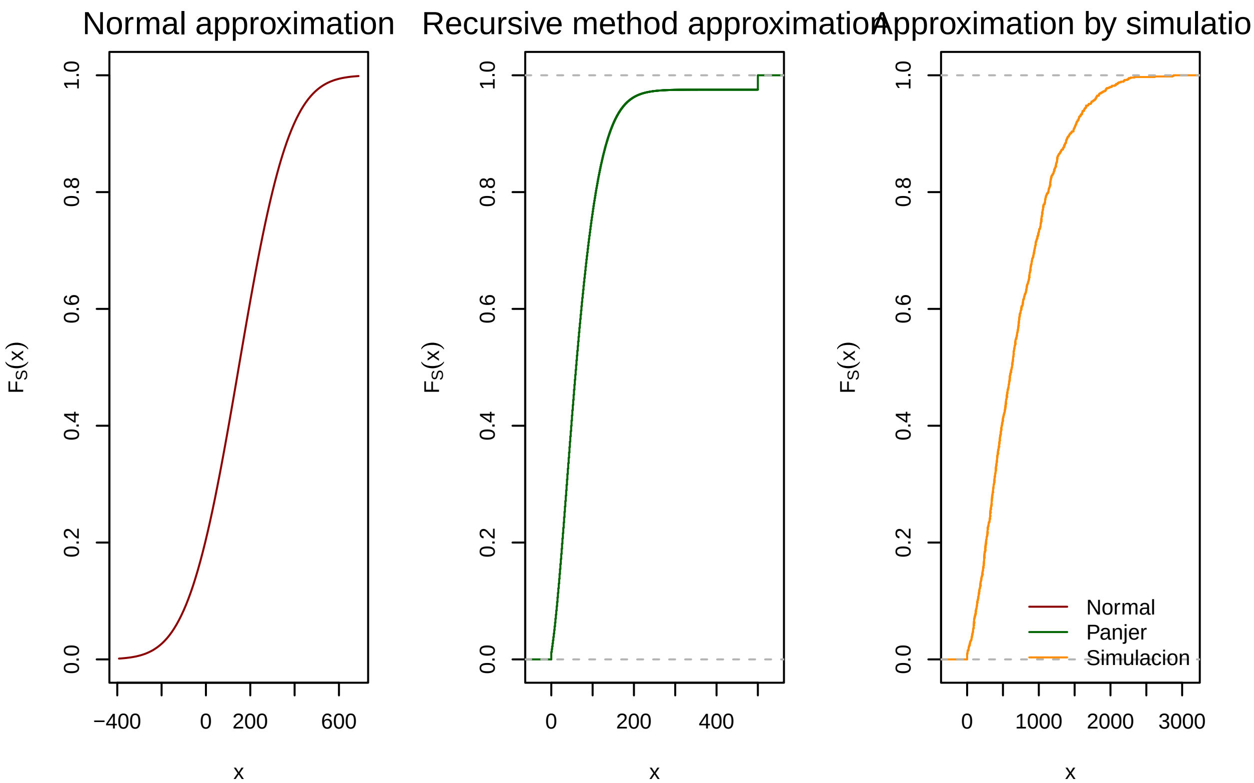
```
plot(result.sim, main="Distribucion de Reclamaciones Agregadas (simulacion)",col="darkblue",col.main="r
     xlab= "Metodo de simulacion",col.lab="darkgreen")
```



0.2.4. Comparación VaR y TVaR

Vamos a calcular el VaR y TvaR de los métodos anteriormente explicados.

```
par(mfrow=c(1,3))
plot(result,col="darkred",do.points=FALSE,verticals=TRUE,main="")
plot(result.p,col="darkgreen",do.points=FALSE,verticals=TRUE,main="")
plot(result.sim,,col="darkorange",do.points=FALSE,verticals=TRUE,main="")
legend("bottomright",legend=c("Normal","Panjer","Simulacion"),bty="n",col=c("darkred","darkgreen","dark
```



Por último calcularémos los percentiles 90 %, 95 % y 99 % de los tres métodos de aproximación.

```
resultados<-matrix(c(VaR(result),CTE(result),VaR(result.p),CTE(result.p),VaR(result.sim),CTE(result.sim
rownames(resultados)<-c("Normal","Panjer","Simulacion");
colnames(resultados)<-c("VaR90","VaR95","VaR99","TVaR90","TVaR95","TVaR99")
t(resultados)
```

```
##           Normal   Panjer Simulacion
## VaR90  378.8261 141.0000   1434.294
## VaR95  444.2624 178.0000   1692.668
## VaR99  567.0099 500.0000   2182.865
## TVaR90 464.0984 174.8822   1780.900
## TVaR95 519.5252 209.2031   2001.477
## TVaR99 628.0448      NaN   2409.770
```

Se observa que la aproximación por simulación se ajusta bien a los atos más grandes, para la aproximación normal hay una subestimación de los montos, y por última Panjer, el VaR y TVaR son muy bajos respecto al resto de métodos.