Tarea 1

NO opta a Titulación

Juan Pablo Zamora Alarcón

2025-10-20

Contenido

0.1. Datos

A continuación, trabajarémos una base de datos que contiene montos de pagos de siniestros, agrupados según el ramo, y fecha de ocurrencia del siniestro. Los montos están expresando en Unidad de Fomento (UF)[1], y se han excluido los siniestros atípicos definidos por la compañía.

Antes de realizar un análisis exploratorio de la información, vamos a depurar nuestra base eliminando los montos cero y nulos, como también valores repetidos que puedan existir.

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

```
# Semilla para mantener la misma información al ejecutar el proceso.
set.seed(123)
# Paquete para lectura archivos excel xlsx
library(readxl)
data <- read_excel("D:/Diplomado Solvencia II/Mod 1 Estd Aplcda Seg/Tarea 1/Input.xlsx")
```

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

[1] Unidad de Fomento: La unidad de fomento (UF) es un índice de reajustabilidad, calculado y autorizado por el Banco Central de Chile, para las operaciones de crédito en moneda nacional que efectúen las empresas bancarias y las cooperativas de ahorro y crédito.

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

```
## # A tibble: 6 x 1
##   `Total Pago (UF)`
##               <dbl>
## 1                 0
## 2               279.
## 3               261.
## 4              1086.
## 5              4526.
## 6              6073.
```

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

```
## # A tibble: 6 x 1
##   `Total Pago (UF)`
##               <dbl>
## 1              9.99
## 2           1060.
## 3                 5
## 4                 0
## 5                 0
## 6                 0
```

Nuestra base contiene un total de registros igual a:

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

```
## [1] 785
```

Realizando la eliminación de registros con montos ceros, nulos y valores duplicados:

```r
# Limpiamos los valores ceros
datos <- data$`Total Pago (UF)`[data$`Total Pago (UF)` > 0]
# Eliminamos valores NaN
datos <- na.omit(datos)
# Eliminamos datos repetidos
datos <- unique(datos)
```

Obtenemos que nuestra base se reduce a un total de:

```
n <- length(datos) # Cantidad de datos totales
n
```

```
## [1] 703
```

0.1.1. Descripción de mis datos

Dado que ya tenemos nuestra información cargada y depurada, ahora realizarémos un análisis descriptivo:

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

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##     0.52    62.73   465.83  1377.05  1670.92 35336.11
```

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

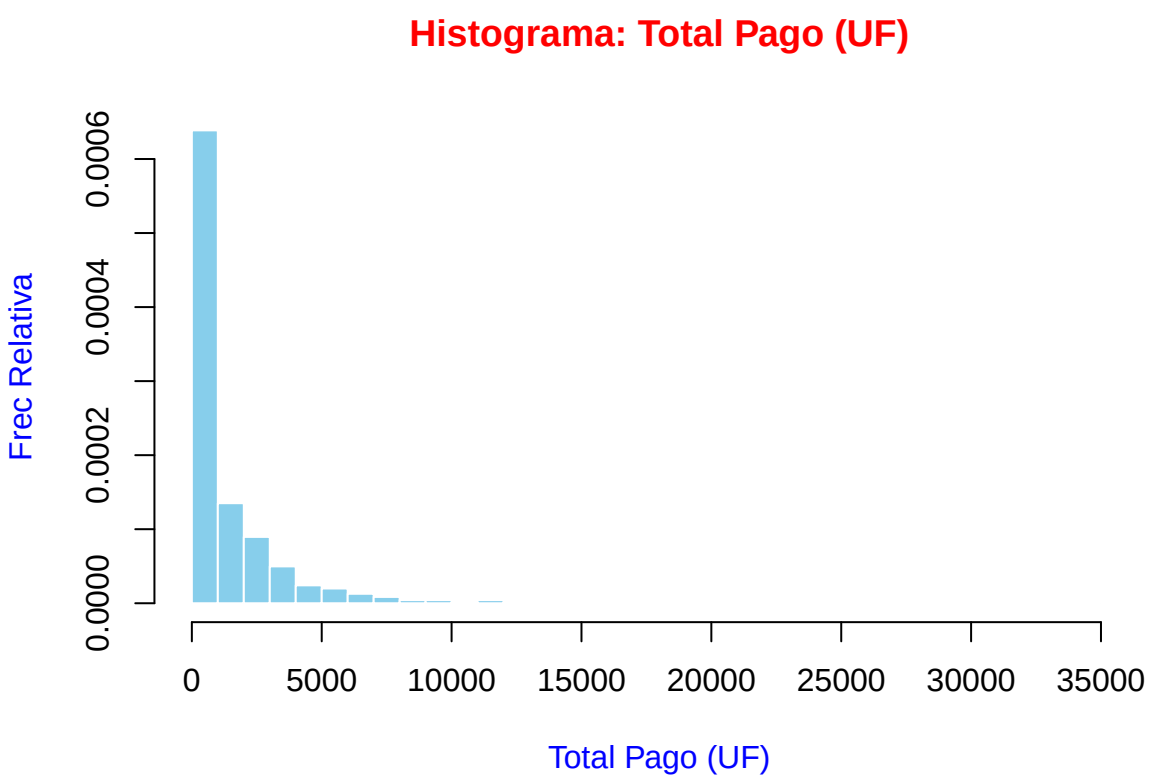
```
## [1] 6668473
```

Observamos de la salida anterior, que la media es mayor a la mediana, y la varianza es significativamente mayor a la media. También, se identifican montos muy pequeños y otros muy grandes respecto a la media. Esto implica que estámos ante la presencia de una distribución sesgada a la derecha o distribución con "cola derecha".

0.1.2. Gráficos de mis datos

En el siguiente gráfico (histograma) se muestra el comportamiento de los datos anteriormente descritos:

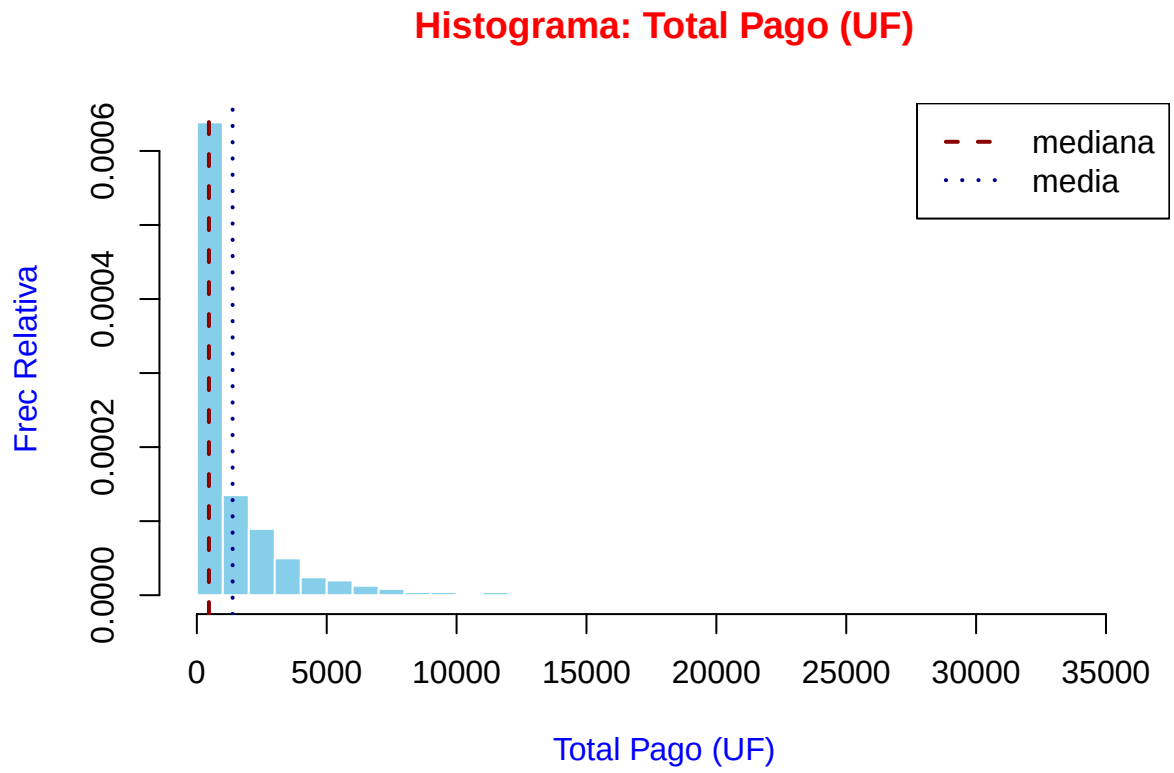
```
options(scipen=999) # Se agrega este código para no mostrar notación científica
# Se grafica el histograma en 50 intervalos de igual dimensión.
# Se agrega la condicion probability = TRUE para mostrar frecuencias relativa y no absolutas
hist(datos,
     col = "skyblue",
     border = "white",
     main="Histograma: Total Pago (UF)",
     ylab = "Frec Relativa",
     xlab = "Total Pago (UF)",
     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".

```
options(scipen=999) # Se agrega este código para no mostrar notación científica
hist(datos,
     col = "skyblue",
     border = "white",
     main="Histograma: Total Pago (UF)",
     ylab = "Frec Relativa",
     xlab = "Total Pago (UF)",
     col.main="red",
     col.lab="blue",
     breaks=50,
     probability = TRUE)
abline(v=median(datos),
       col = "darkred",
       lwd = 2,
       lty = 2)
# Marcamos la mediana en el histograma.
abline(v=mean(datos),
       col="darkblue",
```

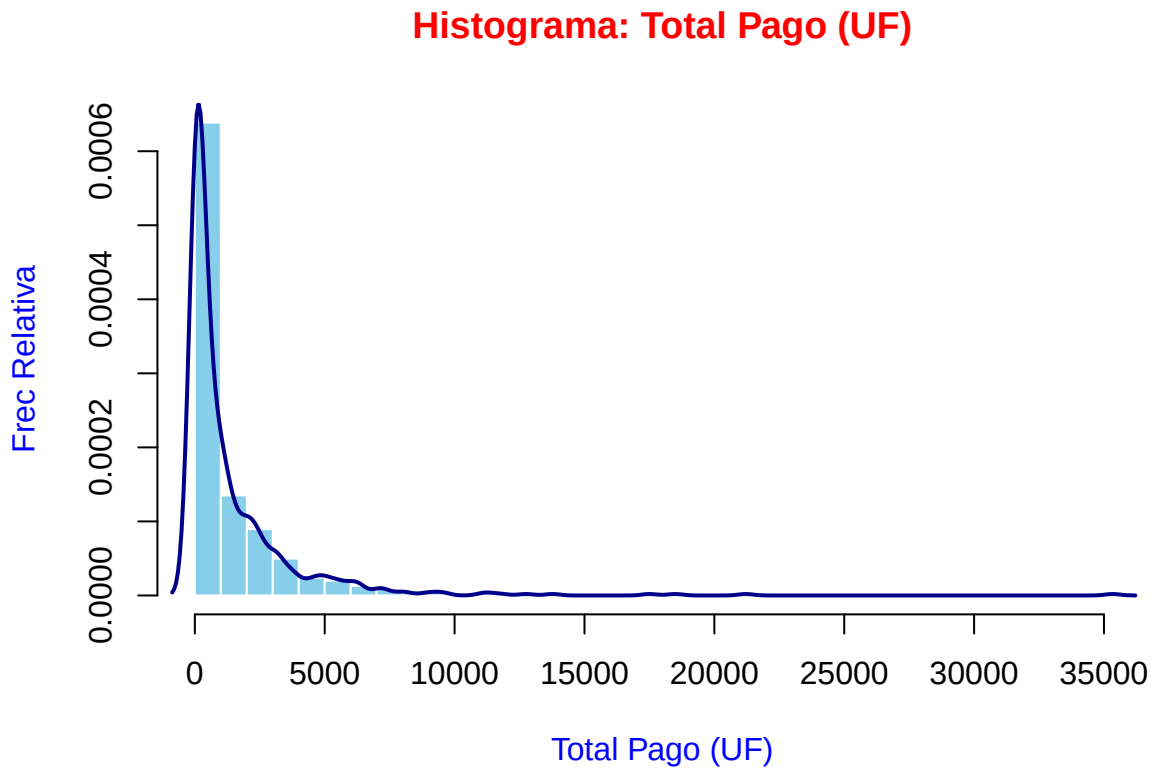
```
       lwd = 2,
       lty=3)
# Marcamos la media en el histograma.
legend("topright",
       legend=c("mediana","media")
       ,col=c("darkred","darkblue"),
       lwd=2,
       lty=c(2,3))
```



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

```
options(scipen=999) # Se agrega este código para no mostrar notación científica
# Se grafica el histograma en 50 intervalos de igual dimensión.
# Graficamos la función de densidad y el histograma superpuestos.
hist(datos,
     col = "skyblue",
     border = "white",
     main="Histograma: Total Pago (UF)",
     ylab = "Frec Relativa",
     xlab = "Total Pago (UF)",
```

```
     col.main="red",
     col.lab="blue",
     breaks=50,
     probability = TRUE)
lines(density(datos),
      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 10.000 UF.

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

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

```
## [1] 5.66218
```

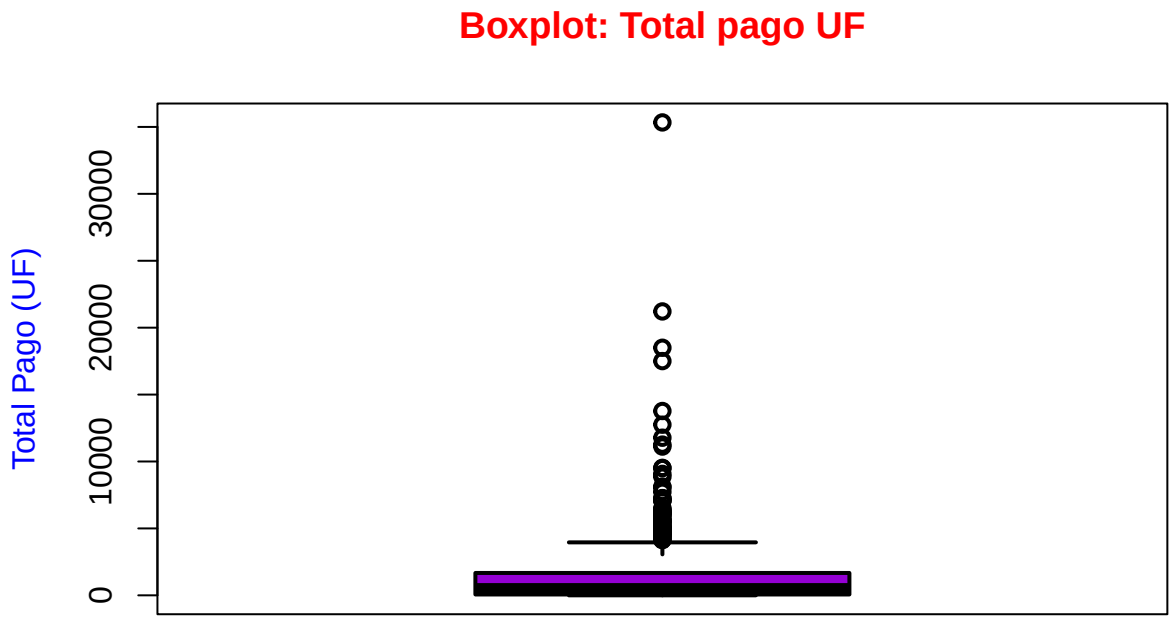
```r
kurt <- sum((datos - mu)^4)/(n*s^4) - 3
kurt # Curtosis.
```

```
## [1] 52.9216
```

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 positiva, es decir, presenta una curva puntiaguda con "cola" más pronunciada.

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

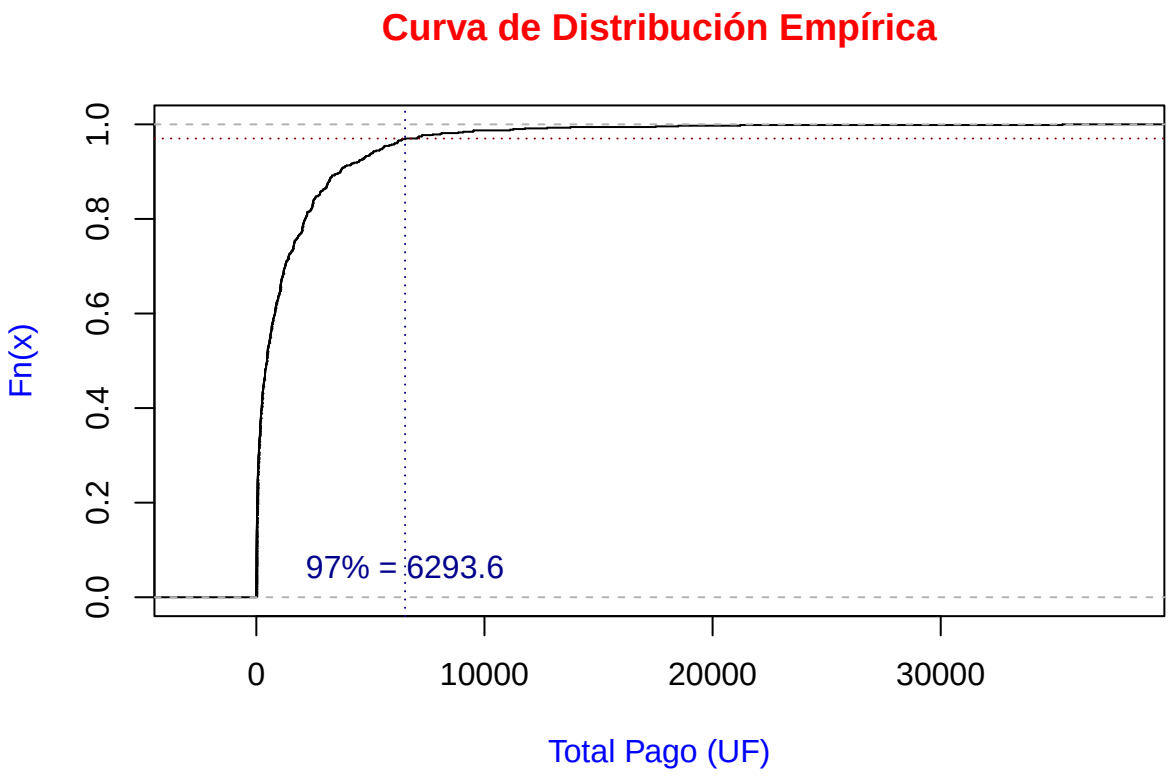
```r
# Diagrama de caja
boxplot(datos,
        col="darkviolet",
        main="Boxplot: Total pago UF",
        col.main="red",
        col.lab="blue",
        lwd=2,
        ylab = "Total Pago (UF)")
```



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:

```
#Graficamos la curva de distribución empírica, en conjunto con el percentil 97%.
plot(ecdf(datos),
     main ="Curva de Distribución Empírica",
     xlab = "Total Pago (UF)",
     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(datos,0.97),
       col ="darkblue",
       lty = 3)
text(quantile(datos,0.97),
     0,
     labels = paste("97% =", round(quantile(data$`Total Pago (UF)`,0.97),2)),
     pos=3,
     col="darkblue")
```



Se observa que el 97 % de los montos están distribuidos por debajo de 6.293 UF, 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.

0.2. Ajuste de densidades

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(datos, dist="lnorm",
                         estimate.params = TRUE,
                         add.line = TRUE,
                         xlab = "Cuantiles Teóricos",
                         ylab = "Cuantiles observados",
                         main = "Distribución Log Normal",
```

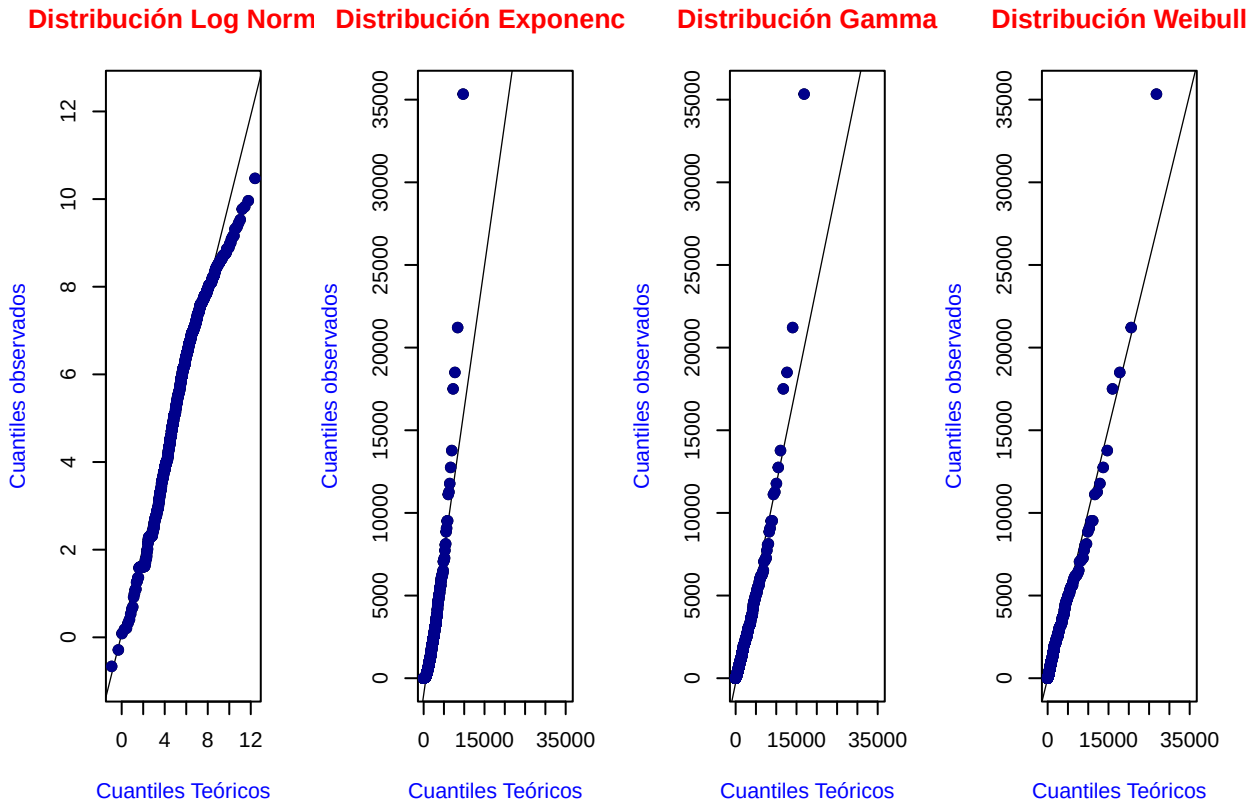
```
                        col.main="red",
                        col.lab="blue"),
       col = "darkblue",
       pch = 16,
       cex = 1.2)

# Gráfica de los cuantiles teóricos y observados de la distribución Exponencial.
points(EnvStats:: qqPlot(datos,
                        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(datos,
                        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(datos,
                        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)
```



En la distribución Exponencial, los datos se alinean sobre una recta con pendiente superior a 45°, lo que sugiere que sería necesario ajustar los parámetros para lograr que los cuantiles se ajusten a la diagonal teórica.

Para las distribuciones de Gamma y Log Normal, la recta está más próxima a la diagonal, pero en menor medida que la Weibull.

La distribución Log Normal presenta una curva en una de sus colas, indicando un desajuste en los extremos. Por último, la distribución Weibull muestra el mejor ajuste entre los cuantiles observados y los teóricos.

0.2.1. Estimación de parámetros

A partir de la comparación entre los cuantíles teóricos y empíricos, se observó que 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
library(fitdistrplus)# Cargamos la libreria para estimar los parámetros.
# Ajusta la distribución Weibull a mis datos.
fit.weibull <- fitdist(datos,
                       "weibull",
                       method="mle")
```

0.2.2. Test Bondad de Ajuste

Luego de estimar los parámetros de nuestra mejor distribución (Weibull), 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.

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

```
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  datos
## D = 0.048117, p-value = 0.07714
## alternative hypothesis: two-sided
```

```r
# Test de bondad de ajuste Anderson-Darling
ad.test(datos,
        "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 = 0.565458067637719, scale = 844.983617994804
##  Parameters assumed to be fixed
##
## data:  datos
## An = 2.179, p-value = 0.07341
```

A partir de los resultados obtenidos, se observa un valor p-value superior a un 5 %, por lo tanto no rechazamos nuestra distribución propuesta.

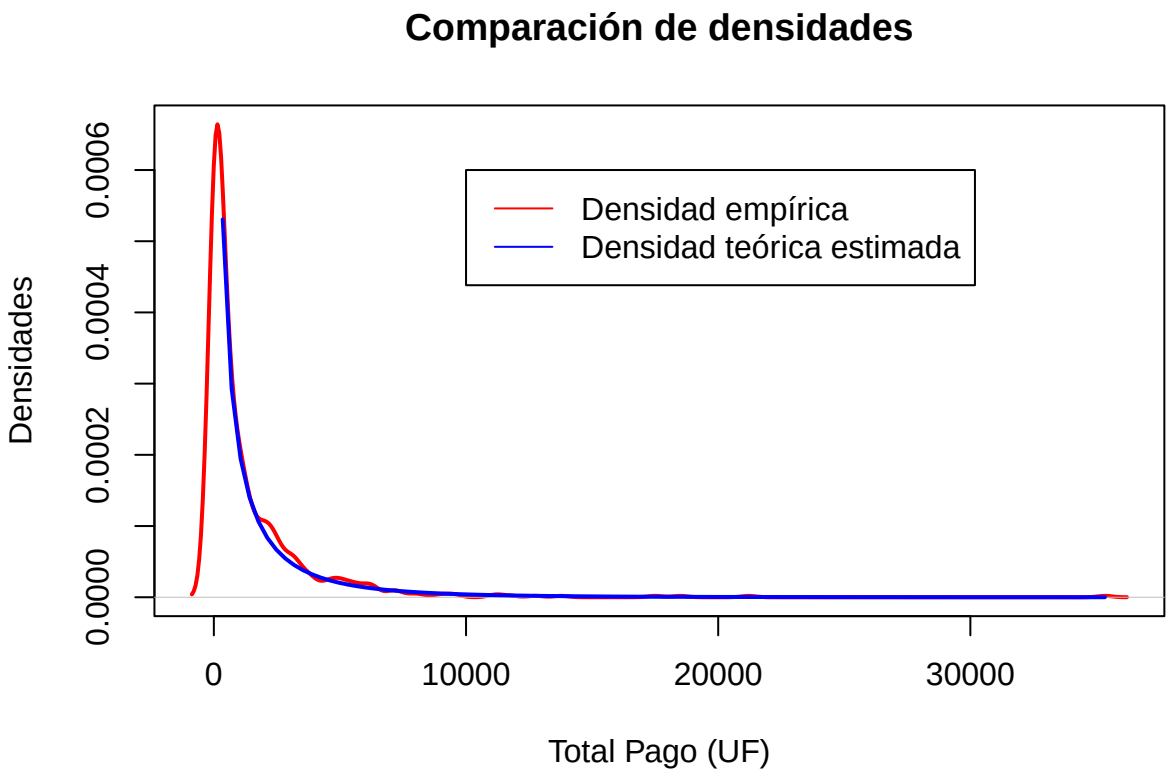
Así podemos concluir que la mejor distribución para nuestros datos es la distribución Weibull.

0.3. Comparación empírica versus teórica

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:

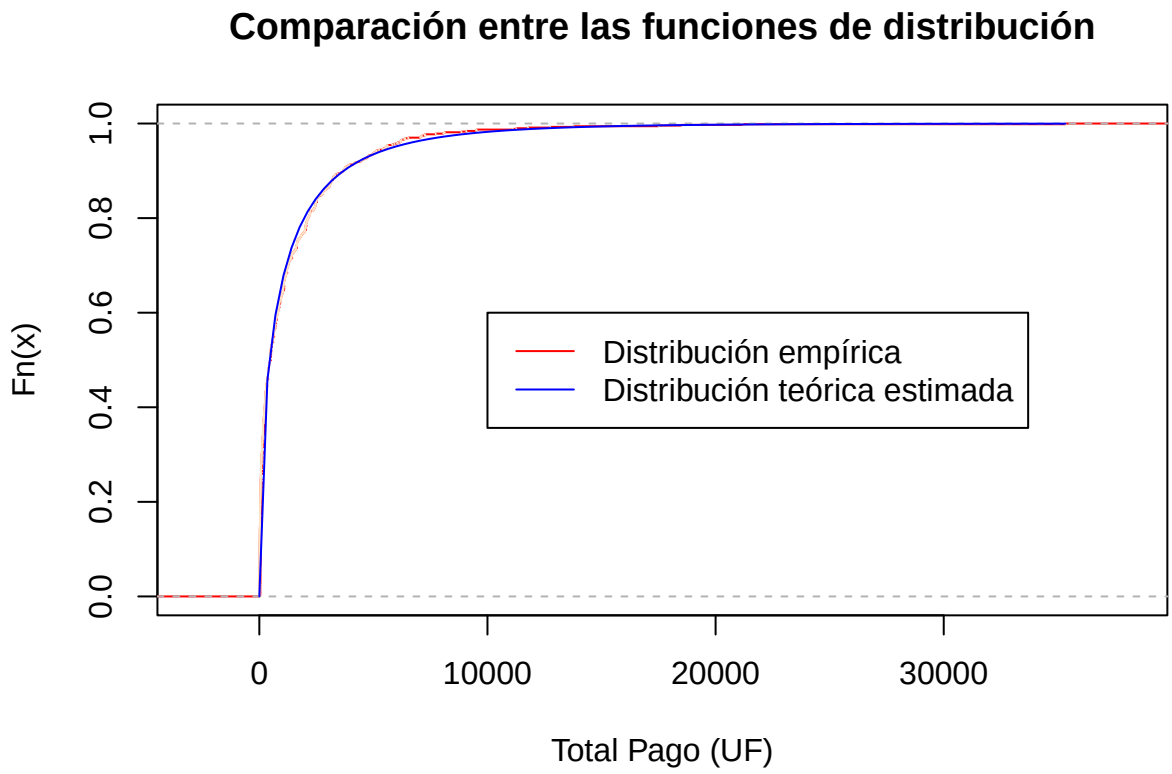
```
options(scipen=999) # Se agrega este código para no mostrar notación científica
# Comparación densidades empírica vs teórica
plot(density(datos),
     col="red",
     main="Comparación de densidades",
     xlab="Total Pago (UF)",
     ylab="Densidades",
     lwd=2)
curve(dweibull(x,
               shape = fit.weibull$estimate["shape"],
               scale = fit.weibull$estimate["scale"]),
      from=0,
      to=max(datos),
      add=TRUE,
      col="blue",
      lwd=2)
k<-c ("Densidad empírica",
      "Densidad teórica estimada")
legend (10000,
        0.0006,
        paste(k),
        lty=1,
        col=c("red","blue"))
```



Luego la comparación de la función de distribución:

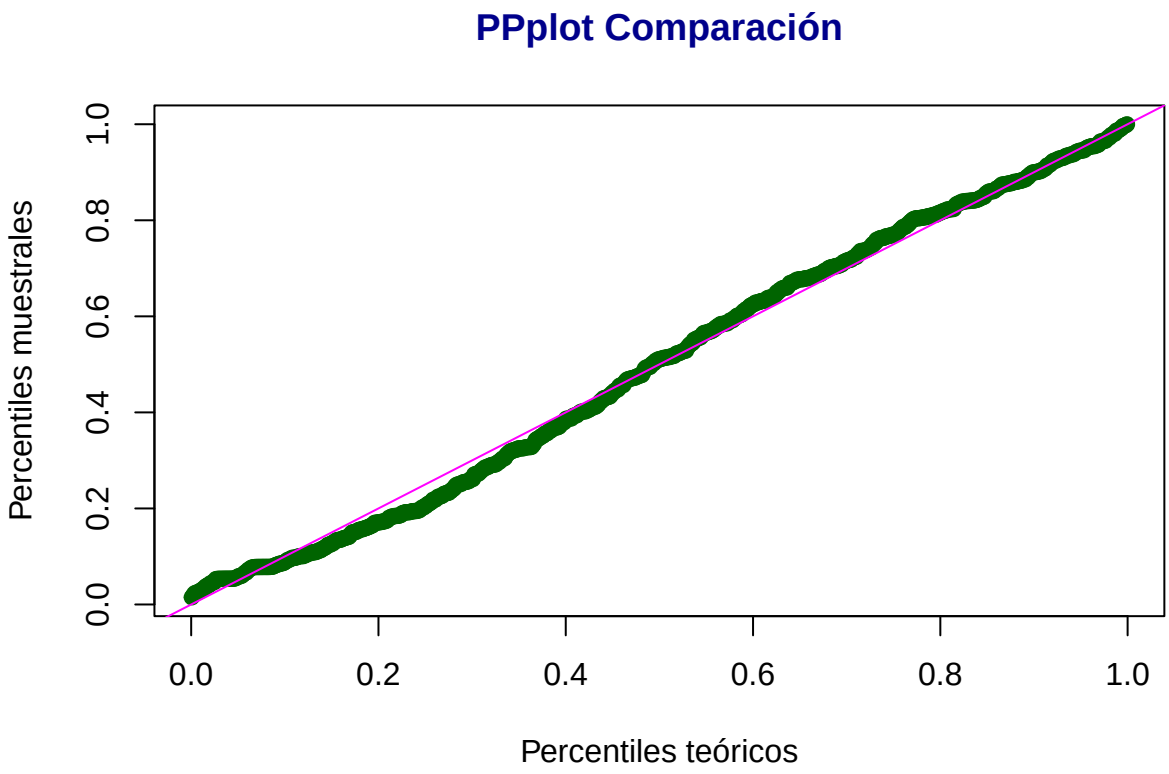
```
# Comparación entre las funciones de distribucion
plot(ecdf(datos),
     col.hor="red",
     col.vert="bisque",
     main="Comparación entre las funciones de distribución",
     xlab = "Total Pago (UF)",
     ylab = "Fn(x)",
     verticals = TRUE,
     do.points = FALSE)
curve(pweibull(x,
               shape = fit.weibull$estimate["shape"],
               scale = fit.weibull$estimate["scale"]),
      from=0,
      to=max(datos),
      add=TRUE,
      col="blue")
k<-c ("Distribución empírica",
      "Distribución teórica estimada")
legend (10000,
```

```
        0.6,
        paste(k),
        lty=1,
        col=c("red","blue"))
```



Y por último, la comparación de percentiles:

```
# Comparación entre percentiles
plot(ppoints(length(datos)),
     sort(pweibull(datos,
                   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 Weibull 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.

0.4. Simulación

A continuación, realizarémos una simulación de la distribución Weibull ajustada a los parámetros previamente estimados.

```
n<-5000  # Número de veces que se realiza el proceso.
m<-5000  # Tamaño de la muestra.
# Matrix cero's de 5000x6 (filas x columnas)
y<-matrix(0,
          n,
          6)
# Matrix cero's de 5000x5000 (filas x columnas)
x<-matrix(0,
          n,
          m)
# Se guardan valores aleatorios en la matriz X,
# de una distribución Weibull con nuestros parámetros estimados.
```

```
# En la matriz Y, guardamos los estadísticos calculados a las columnas de X.
for(i in 1:n){
  x[i,]<-rweibull(m,
                  shape = fit.weibull$estimate["shape"],
                  scale = fit.weibull$estimate["scale"])
  y[i,]<-c(mean(x[i,]),
           median(x[i,]),
           var(x[i,]),
           quantile(x[i,],
                    probs=c(0.9,0.95,0.99)
                    )
           )
}
```

En el código anterior se calcularon la media, mediana, varianza y los percentiles al 90 %, 95 % y 99 %, de valores aleatorios generados a partir de la distribución Weibull ajustada a los parámetros antes estimados.

A continuación, se calcularán los estadísticos descriptivos:

```
# Calculamos el promedio a las columnas de la Matriz Y
sim1<-apply(y,
            2,
            mean
            )
names(sim1) <- c("Media",
                 "Mediana",
                 "Varianza",
                 "q90",
                 "q95",
                 "q99"
                 )
sim1
```

```
##        Media      Mediana     Varianza          q90          q95          q99
##    1379.7300     442.3188 6825147.4307    3692.3977    5879.4440   12525.5693
```

Además, se calculará el intervalo de confianza al 95 %:

```
# Creamos la función q,
# Calcula el intervalo de confianza al 95% a cada columna de la matrix Y.
q<-function(x){quantile(x,
                        probs=c(0.025,
                                0.975)
                        )
```

```
  }
# Calcula el intervalo de confianza a las columnas de la matrix Y
sim2<-apply(y,
            2,
            q)
colnames(sim2)<-c("Media",
                  "Mediana",
                  "Varianza",
                  "q90",
                  "q95",
                  "q99")
sim2
```

```
##          Media  Mediana Varianza      q90      q95      q99
## 2.5%  1308.977 412.0161  5628104 3464.675 5480.948 11243.79
## 97.5% 1455.798 473.9329  8366442 3933.700 6309.163 13958.55
```

Dado que tenemos la simulación, vamos a comparar los estadísticos descriptivos de los datos empíricos, y de los valores teóricos de la distribución ajustada y los resultados generados mediante simulación.

```
# Media teórica
mean.mod<-fit.weibull$estimate["scale"]*
  gamma(1+1/fit.weibull$estimate["shape"])

# Mediana teórica
median.mod<-qweibull(p=0.5,
                     shape = fit.weibull$estimate["shape"],
                     scale = fit.weibull$estimate["scale"],
                     lower.tail = TRUE,
                     log.p = FALSE)

# Varianza teórica
var.mod<-fit.weibull$estimate["scale"]^2*
  (gamma(1+2/fit.weibull$estimate["shape"])
   -(gamma(1+1/fit.weibull$estimate["shape"]))^2)

# Percentil 90%, 95% y 99% teóricos
q.mod<-qweibull(p=c(0.9,
                    0.95,
                    0.99),
                shape = fit.weibull$estimate["shape"],
                scale = fit.weibull$estimate["scale"],
                lower.tail = TRUE,
```

```
                log.p = FALSE)

# Media empírica
mean.dat<-mean(datos)

# Mediana empírica
median.dat<-median(datos)

# Varianza empírica
var.dat<-var(datos)

# Percentil 90%, 95% y 99% empíricos
q.dat<-quantile(datos,
                prob=c(0.9,
                       0.95,
                       0.99)
                )
# Guardamos los valores anteriores en la siguiente matriz
# Cada fila es el estadístico calculado teóricamente, empíricamente, y simulado.
options(scipen=999)
Resultados<-matrix(c(mean.dat,
                     median.dat,
                     var.dat,
                     q.dat,
                     mean.mod,
                     median.mod,
                     var.mod,
                     q.mod,
                     sim1,
                     sim2[1,],
                     sim2[2,]
                     ),
                   6,
                   5
                   )
colnames(Resultados)<-c("Datos",
                        "Modelo",
                        "Simulación",
                        "simLi",
                        "simLs"
                        )
rownames(Resultados)<-c("Media",
                        "Mediana",
```

```
                        "Varianza",
                        "q90",
                        "q95",
                        "q99"
                        )
Resultados
```

```
##                Datos       Modelo   Simulación        simLi        simLs
## Media       1377.048    1379.8607    1379.7300    1308.9769    1455.7979
## Mediana      465.831     441.9281     442.3188     412.0161     473.9329
## Varianza 6668473.352 6839883.2805 6825147.4307 5628103.7199 8366442.1875
## q90         3685.483    3693.3347    3692.3977    3464.6752    3933.6996
## q95         5537.751    5882.1112    5879.4440    5480.9483    6309.1628
## q99        11259.708   12582.9802   12525.5693   11243.7890   13958.5530
```

La comparación muestra que la distribución Weibull ajustada estima de buena forma la media y mediana del conjunto de datos.

De manera análoga, la varianza presenta un ajuste sin diferencias significativas, lo que valida la consistencia del modelo para la dispersión de los montos.

Para los cuantiles, se observan que hay ciertas diferencias entre el modelo teórico y la simulación de los datos empíricos. Por tanto, el modelo como la simulación tienden a sobreestimar los montos superiores.

0.5. Conclusión

Como conclusión, el modelo propuesto es aceptable en términos generales, sin embargo se debe tener precaución en la predicción de montos altos en la cola derecha de la distribución, ya que pueden ser sobreestimados.