

Ajuste de probabilidad de reclamos de responsabilidad civil

Equipo 2

9/17/2021

Cargamos Paqueterías

Paqueterías

```
library(MASS) #distr de proba  
library(actuar) #distr adicionales de proba  
library(fitdistrplus) #ajuste de curvas de proba  
library(goftest) #pruebas de bondad de ajuste  
library(readxl) #importar de excel  
library(ggplot2) #gráficos  
library(dplyr) #manipulación de df
```

Cargamos los Datos

```
### Importamos Montos ya ajustados a Inflación
library(readxl)
datos <- read_excel("datosR.xlsx")
#Ajustamos las unidades de los datos (mdp)
datos_uni <- datos %>%
  mutate(Montos_mdp = Montos2020/1000000)
#Escalamos los Datos
datos_soft <- datos_uni%>%
  mutate(Monto_escala = log10((Montos_mdp) * 10^7))
```

Obtenemos los Datos

```
datos_soft
```

```
## # A tibble: 43,074 x 4
```

```
##       Año Montos2020 Montos_mdp Monto_escal
```

```
##    <dbl>      <dbl>      <dbl>      <dbl>
```

```
##  1  2015    129822.    0.130      6.11
```

```
##  2  2015    498133.    0.498      6.70
```

```
##  3  2015     4257.    0.00426    4.63
```

```
##  4  2015     4271.    0.00427    4.63
```

```
##  5  2015    54124.    0.0541    5.73
```

```
##  6  2015   928667.    0.929     6.97
```

```
##  7  2015    20775.    0.0208    5.32
```

```
##  8  2015    67780.    0.0678    5.83
```

```
##  9  2015   253901.    0.254     6.40
```

```
## 10  2015     9369.    0.00937    4.97
```

```
## # ... with 43,064 more rows
```

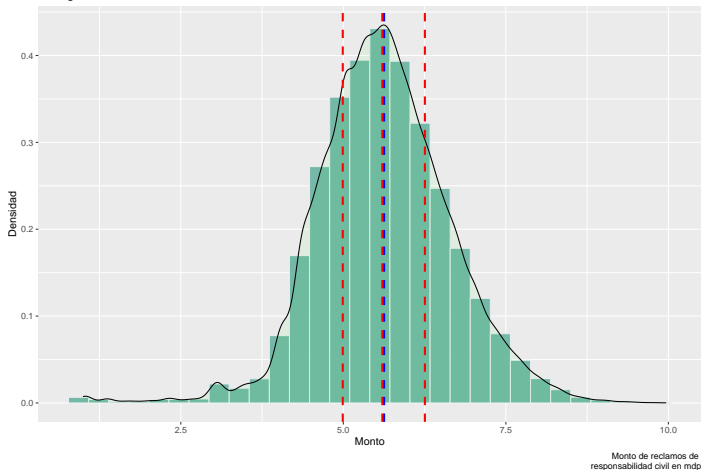
Histograma

```
# Histograma
ggplot(datos_soft, aes(x = Monto_escala ,
                        y = ..density..)) +
  geom_histogram(fill="#69b3a2",
                 color="#e9ecef", alpha = 0.9) +
  labs(title = 'Histograma de Montos actualizados al 2020',
        x = 'Monto',
        y = 'Densidad',
        caption = 'Monto de reclamos de
responsabilidad civil en mdp')
```

```
est_des <- summary(datos_soft$Monto_escala)
```

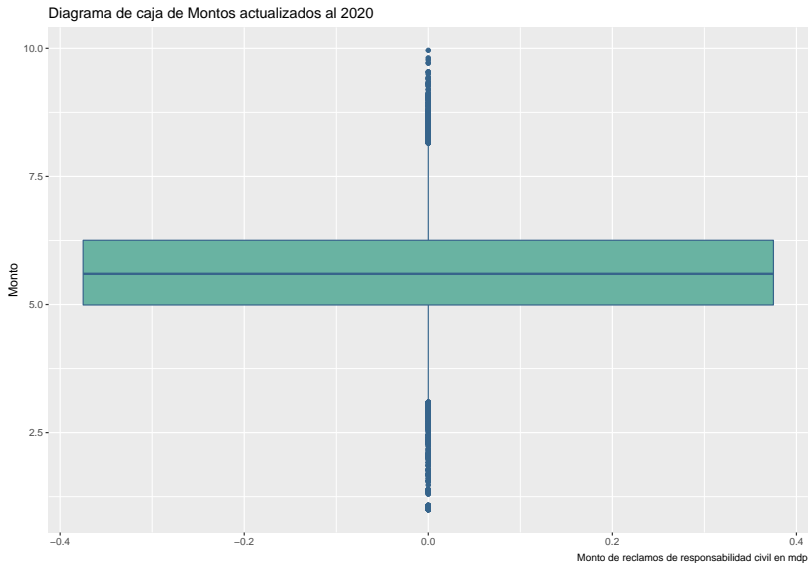
Histograma

Histograma de Montos actualizados al 2020

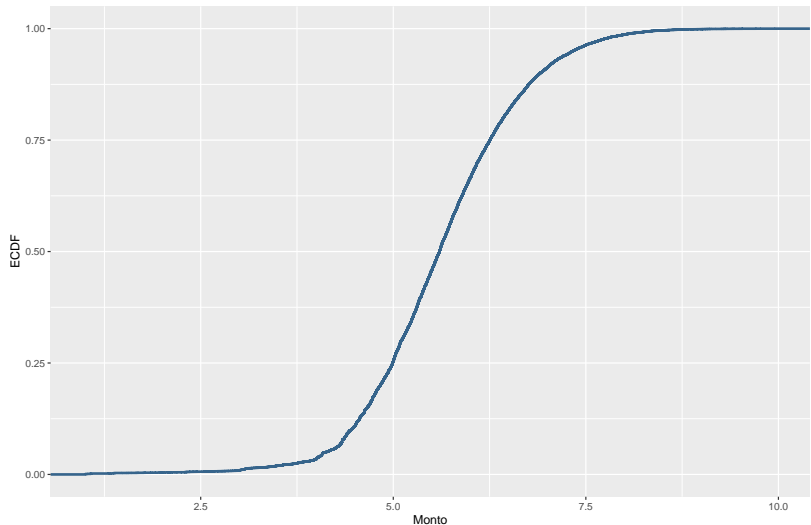


##	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
##	1.000	4.991	5.599	5.627	6.254	9.961

Boxplot



Func. de distribución acumulativa de Montos actualizados al 2015



Monto de reclamos de responsabilidad civil en mdp

Generamos Modelos

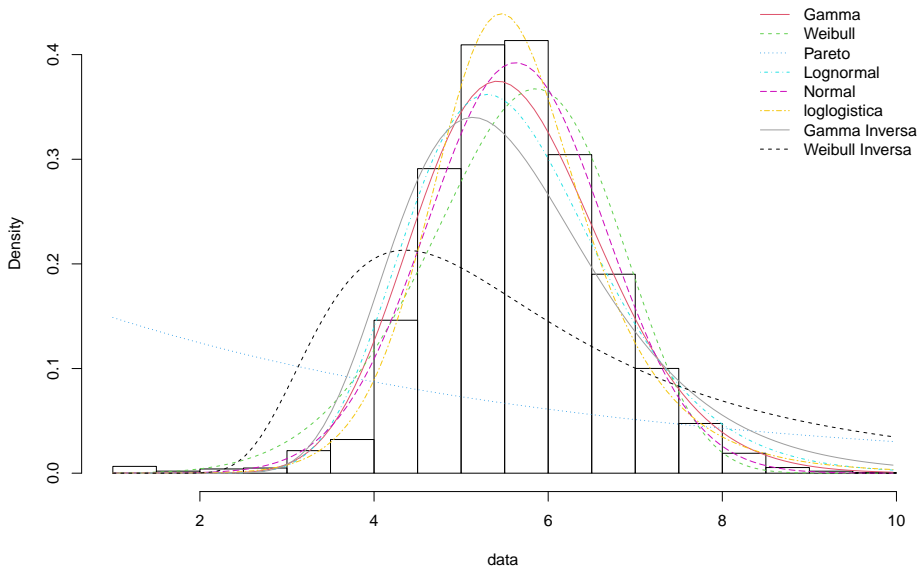
```
mod1 <- fitdist(datos_soft$Monto_escalas, "gamma",  
               method = c("mle"))  
mod2 <- fitdist(datos_soft$Monto_escalas, "weibull",  
               method = "mle")  
mod3 <- fitdist(datos_soft$Monto_escalas, "pareto",  
               method = "mle",  
               start = list(shape = 1, scale = 300))  
mod4 <- fitdist(datos_soft$Monto_escalas, "lnorm",  
               method = "mle")  
mod5 <- fitdist(datos_soft$Monto_escalas, "norm",  
               method = "mle")  
mod6 <- fitdist(datos_soft$Monto_escalas, "llogis",  
               method = "mle")  
mod7 <- fitdist(datos_soft$Monto_escalas, "invgamma",  
               method = "mle")  
mod8 <- fitdist(datos_soft$Monto_escalas, "invweibull",  
               method = "mle")
```

Section 1

Comparamos los Modelos Gráficamente

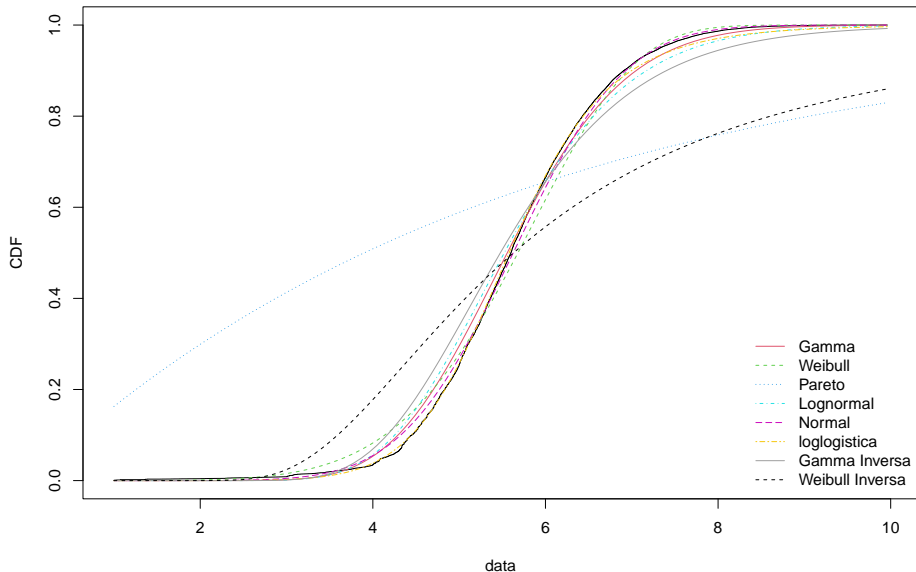
Función de Densidad

Histogram and theoretical densities

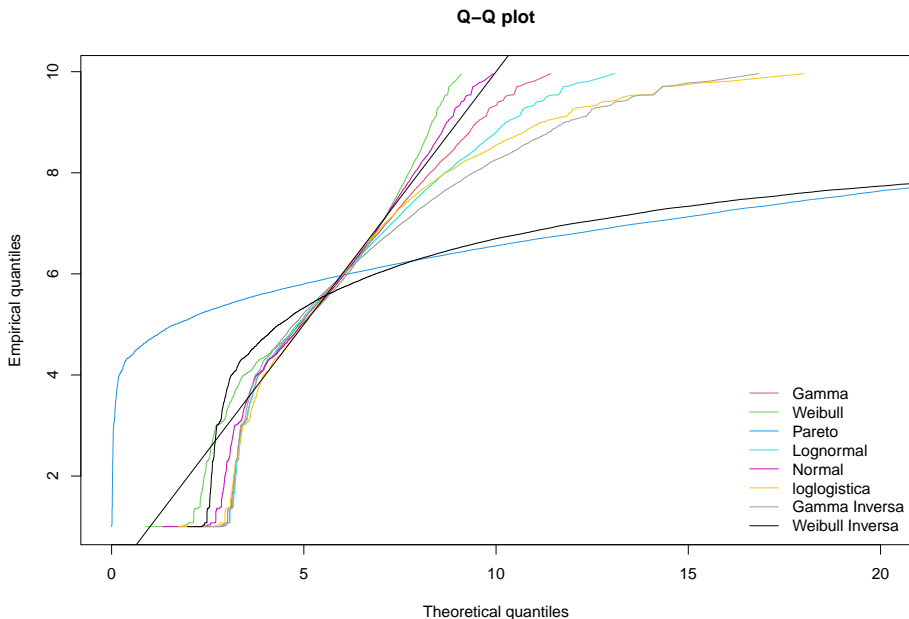


Función de Distribución

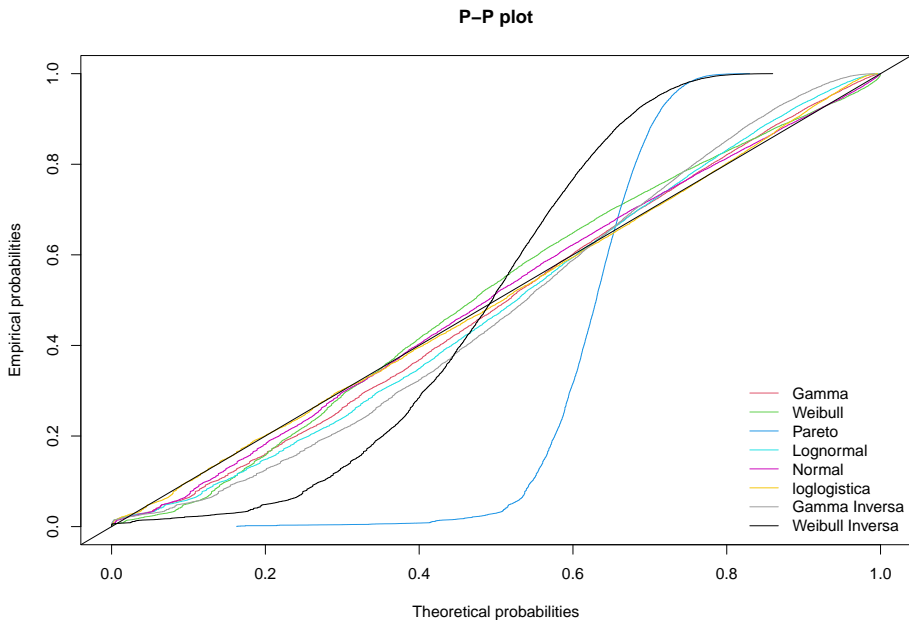
Empirical and theoretical CDFs



Cuantiles Teóricos vs Empíricos



Probabilidades Teóricas vs Empíricas



Pruebas de hipótesis

#K-S test

```
(hip1_ks <- ks.test(datos$Montos2020,  
  'pweibull', mod1$estimate[1], mod1$estimate[2]))  
(hip2_ks <- ks.test(datos$Montos2020,  
  'plnorm', mod2$estimate[1], mod2$estimate[2]))  
(hip3_ks <- ks.test(datos$Montos2020,  
  'pgamma', mod3$estimate[1], mod3$estimate[2]))  
(hip4_ks <- ks.test(datos$Montos2020,  
  'pcauchy', mod4$estimate[1], mod4$estimate[2]))  
(hip5_ks <- ks.test(datos$Montos2020,  
  'ppareto', mod5$estimate[1], mod5$estimate[2]))  
(hip6_ks <- ks.test(datos$Montos2020,  
  'pllogis', mod6$estimate[1], mod6$estimate[2]))
```

AD test

```
(hip1_ad <- ad.test(datos$Montos2020,  
  'pweibull', mod1$estimate[1],mod1$estimate[2]))  
(hip2_ad <- ad.test(datos$Montos2020,  
  'plnorm', mod2$estimate[1],mod2$estimate[2]))  
(hip3_ad <- ad.test(datos$Montos2020,  
  'pgamma', mod3$estimate[1],mod3$estimate[2]))  
(hip4_ad <- ad.test(datos$Montos2020,  
  'pcauchy', mod4$estimate[1],mod4$estimate[2]))  
(hip5_ad <- ad.test(datos$Montos2020,  
  'ppareto', mod5$estimate[1],mod5$estimate[2]))  
(hip6_ad <- ad.test(datos$Montos2020,  
  'pllogis', mod6$estimate[1],mod6$estimate[2]))
```


Wilcoxon test

```
(hip1_w <- wilcox.test(x = datos$Montos2020,  
  y = rweibull(464,mod1$estimate[1],mod1$estimate[2]),  
  alternative = "two.sided", mu = 0,paired = FALSE))  
  
(hip2_w <- wilcox.test(x = datos$Montos2020,  
  y = rlnorm(464,mod2$estimate[1],mod2$estimate[2]),  
  alternative = "two.sided", mu = 0,paired = FALSE))  
  
(hip3_w <- wilcox.test(x = datos$Montos2020,  
  y = rgamma(464,mod3$estimate[1],mod3$estimate[2]),  
  alternative = "two.sided", mu = 0,paired = FALSE))  
  
(hip4_w <- wilcox.test(x = datos$Montos2020,  
  y = rcauchy(464,mod4$estimate[1],mod4$estimate[2]),  
  alternative = "two.sided", mu = 0,paired = FALSE))  
  
(hip5_w <- wilcox.test(x = datos$Montos2020,  
  y = rpareto(464,mod5$estimate[1],mod5$estimate[2]),  
  alternative = "two.sided", mu = 0,paired = FALSE))  
  
(hip6_w <- wilcox.test(x = datos$Montos2020,  
  y = rlloris(464,mod6$estimate[1],mod6$estimate[2])).
```

Cramer Von Mises

```
(hip1_cv <- cvm.test(datos$Montos2020,  
  'pweibull', mod1$estimate[1],mod1$estimate[2]))  
(hip2_cv <- cvm.test(datos$Montos2020,  
  'plnorm', mod2$estimate[1],mod2$estimate[2]))  
(hip3_cv <- cvm.test(datos$Montos2020,  
  'pgamma', mod3$estimate[1],mod3$estimate[2]))  
(hip4_cv <- cvm.test(datos$Montos2020,  
  'pcauchy', mod4$estimate[1],mod4$estimate[2]))  
(hip5_cv <- cvm.test(datos$Montos2020,  
  'ppareto', mod5$estimate[1],mod5$estimate[2]))  
(hip6_cv <- cvm.test(datos$Montos2020,  
  'pllogis', mod6$estimate[1],mod6$estimate[2]))
```

Pruebas de Bondad de Ajuste

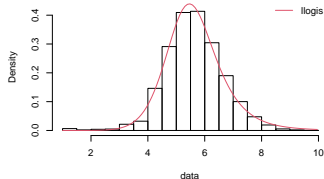
```
gofstat(list(mod1, mod2, mod3, mod4, mod5, mod6, mod7, mod8))
```

```
## Goodness-of-fit statistics
##
## 1-mle-gamma 2-mle-weibull 3-mle-pareto
## Kolmogorov-Smirnov statistic 0.06531506 0.05668035 4.482823e-01
## Cramer-von Mises statistic 65.95761599 39.05111286 2.628817e+03
## Anderson-Darling statistic 463.51318523 276.62129133 1.241871e+04
##
## 4-mle-lnorm 5-mle-norm 6-mle-llogis
## Kolmogorov-Smirnov statistic 0.1027788 0.03105447 0.02524038
## Cramer-von Mises statistic 192.8341189 10.38383066 5.37632028
## Anderson-Darling statistic 1279.3278801 79.69653583 89.66741604
##
## 7-mle-invgamma 8-mle-invweibull
## Kolmogorov-Smirnov statistic 0.2210578 0.3467066
## Cramer-von Mises statistic 863.3745541 1790.5464741
## Anderson-Darling statistic 4868.2150636 9053.0418843
##
## Goodness-of-fit criteria
##
## 1-mle-gamma 2-mle-weibull 3-mle-pareto
## Akaike's Information Criterion 205704.4 198414.2 291446.4
## Bayesian Information Criterion 205721.7 198431.5 291463.7
##
## 4-mle-lnorm 5-mle-norm 6-mle-llogis
## Akaike's Information Criterion 219408.7 195516.6 199677.0
## Bayesian Information Criterion 219426.0 195533.9 199694.3
##
## 7-mle-invgamma 8-mle-invweibull
## Akaike's Information Criterion 261677.1 299148.6
## Bayesian Information Criterion 261694.4 299165.9
```

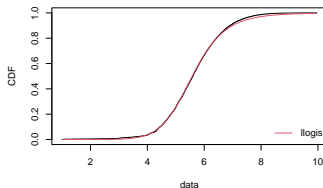
Conclusión

El menor AIC lo tiene la loglogística seguida de la pareto y el menor BIC lo tiene la loglogística seguida de la pareto Por lo tanto, concluimos que la distribución que mejor se ajusta es la Loglogística con parámetro de forma 7.833596 y de escala 10.709578

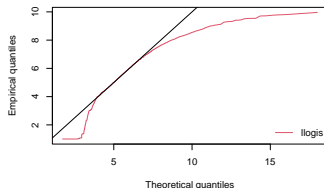
Histogram and theoretical densities



Empirical and theoretical CDFs



Q-Q plot



P-P plot

