Examen II - Distribuciones de Pérdidas

Shanthal Chavarría

2021-12-01

Contents

1. Considere los siguientes datos:

27	82	115	126	155	161	243	294	340	384
457	680	855	877	974	1.193	1.340	1.884	2.558	3.476

1. Calcule las funciones de densidad y probabilidad empíricas, y grafique en comparación con la respectiva distribución exponencial.

```
library(fitdistrplus)

datos <- c(27, 82 , 115,126, 155, 161, 243, 294, 340, 384, 457, 680, 855, 877, 974,  1193, 1340, 1884, 
```

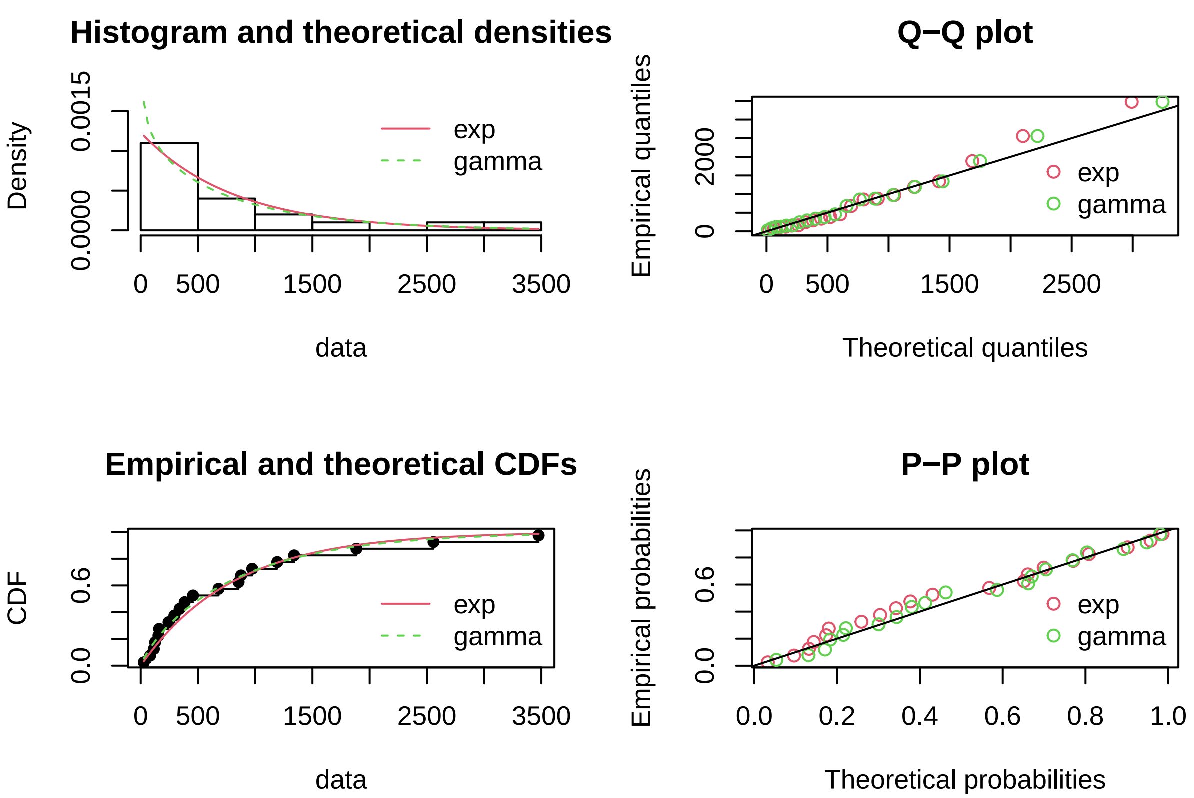
```
f.exp<-fitdist(datos, "exp",method = "mme")
f.exp
```

```
Fitting of the distribution ' exp ' by matching moments
Parameters:
       estimate
rate 0.00123297
```

```
f.gamma<-fitdist(datos, "gamma", method = "mme")
f.gamma
```

```
Fitting of the distribution ' gamma ' by matching moments
Parameters:
         estimate
shape 0.829860219
rate  0.001023192
```

```
par(mfrow=c(2,2))
plot.legend<-c("exp","gamma")
denscomp(list(f.exp,f.gamma), legendtext=plot.legend)
qqcomp(list(f.exp, f.gamma), legendtext=plot.legend)
cdfcomp(list(f.exp, f.gamma), legendtext=plot.legend)
ppcomp(list(f.exp, f.gamma), legendtext=plot.legend)
```



2. Realice los test Kolmogorov-Smirnov y Anderson-Darling y analice los resultados.

```r
comparacion <- gofstat(f = list(f.exp, f.gamma))
comparacion
```

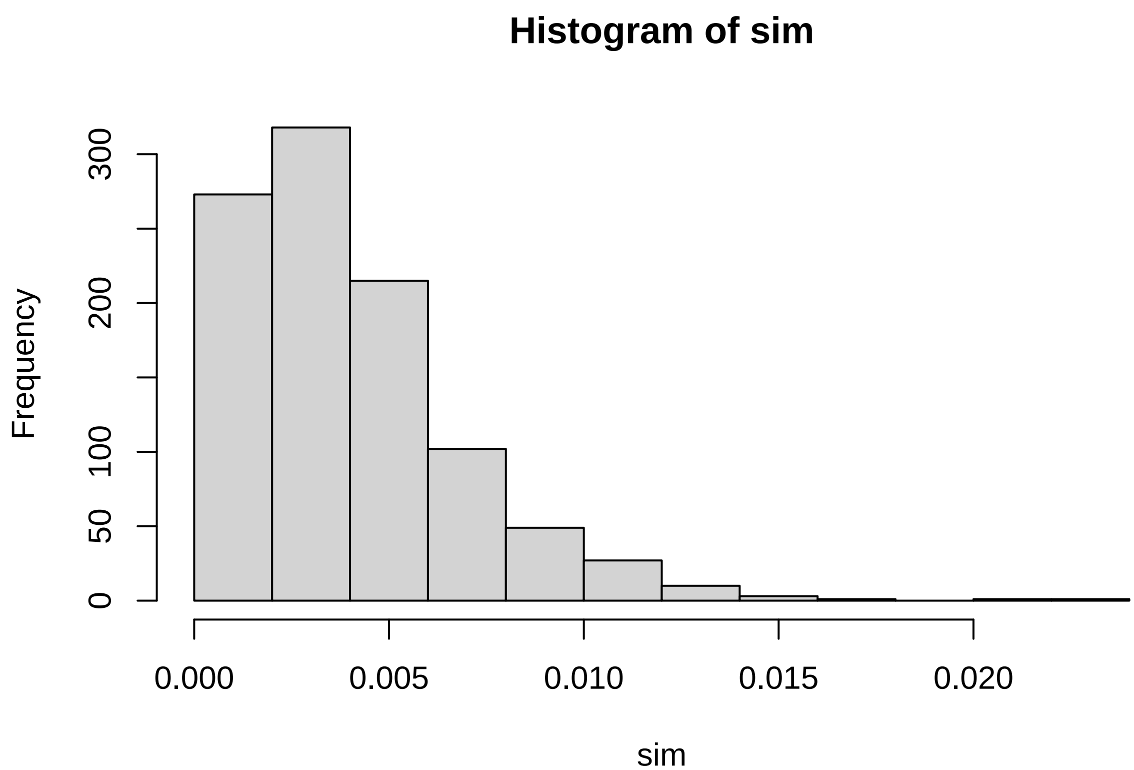
```
Goodness-of-fit statistics
                             1-mme-exp 2-mme-gamma
Kolmogorov-Smirnov statistic 0.12284329  0.08751566
Cramer-von Mises statistic   0.05581739  0.02785829
Anderson-Darling statistic   0.32587814  0.20105473

Goodness-of-fit criteria
                               1-mme-exp 2-mme-gamma
Akaike's Information Criterion  309.9332    311.9601
Bayesian Information Criterion  310.9289    313.9516
```

3. Determine si un modelo gamma es más apropiado que el modelo exponencial.

El más apropiado es gamma debido al resultado de las pruebas y el hecho de que los valores extremos se acercan más al ajuste de la gamma.

2. Simule 1,000 observaciones de una distribución gamma con $\alpha = 2$ y $\theta = 500$. Realice las pruebas de bondad de ajuste de chi-cuadrado y Kolmogorov-Smirnov para ver si los valores indicados eran en realidad de esa distribución

```r
library(EnvStats)

set.seed(3)

sim <- rgamma(1000,2,500)
hist(sim)
```




```
f<-fitdist(sim,"gamma",method = "mle")
f
```

```
Fitting of the distribution ' gamma ' by maximum likelihood
Parameters:
        estimate  Std. Error
shape   2.004136  0.08324291
rate  503.042024 23.72342133
```

```
test.chi <-  gofTest(sim, test = "chisq", distribution = "gamma",
    param.list = list(shape = 2, scale = 1/500))


test.chi$p.value
```

```
[1] 0.6425503
```

```
test.KS <- gofTest(sim, test = "ks", distribution = "gamma",
    param.list = list(shape = 2, scale = 1/500))


test.KS$p.value
```

```
[1] 0.6376003
```

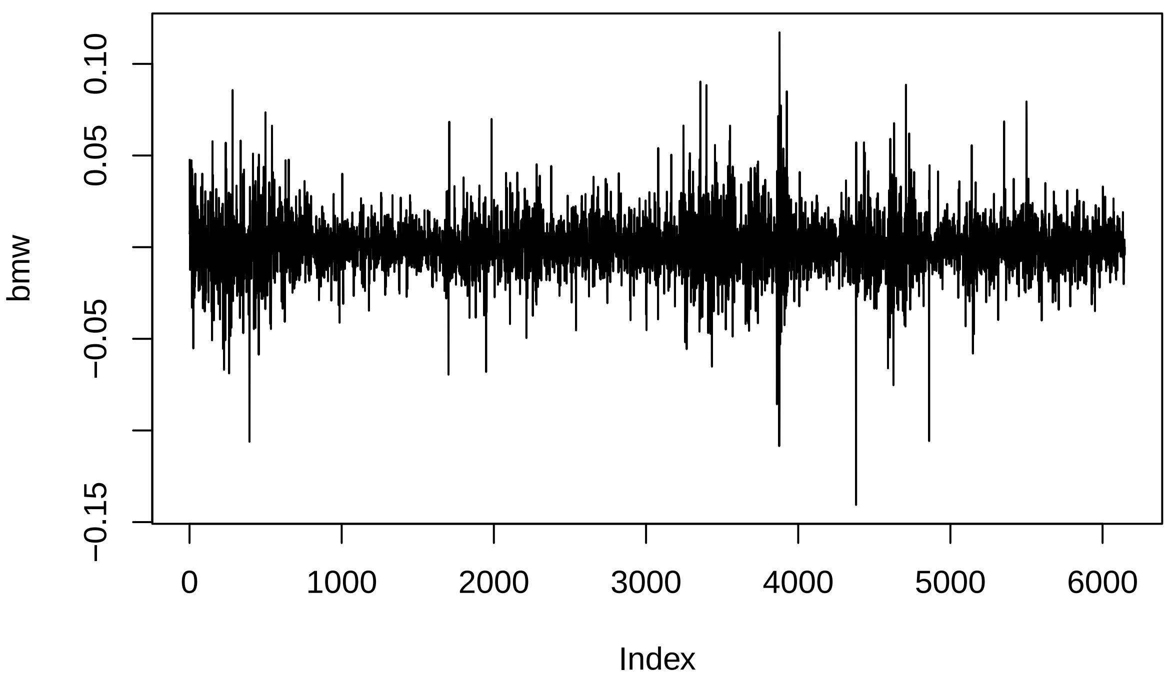
```r
ks.test(sim,"pgamma",2,500)
```

```
    One-sample Kolmogorov-Smirnov test

data:  sim
D = 0.023521, p-value = 0.6376
alternative hypothesis: two-sided
```

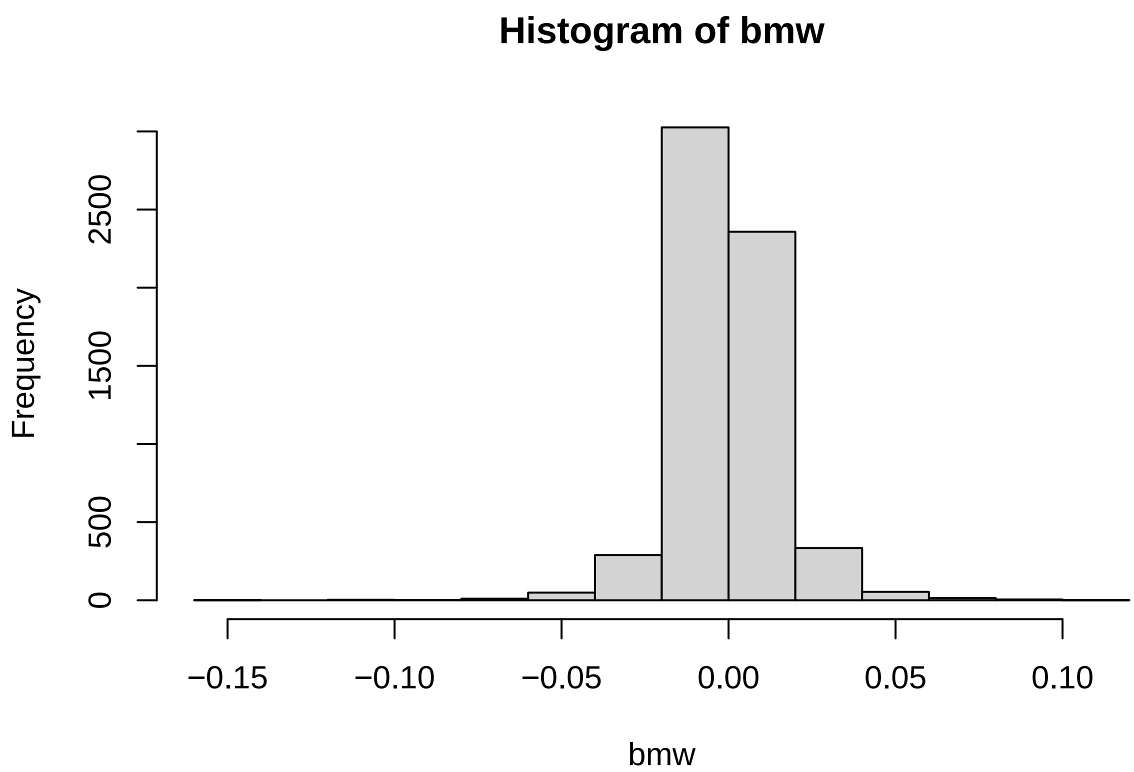
Tenemos un valor p de 0.63, por lo tanto no rechazamos la hipótesis nula y por lo tanto los datos simulados siguen una distribución Gamma.

3. Utilice los datos BMW, del paquete evir y ajuste un modelo para los valores extremos.

```r
library(evir)

data(bmw)

plot(bmw,type="l") # reclamos
```




```r
hist(bmw)
```

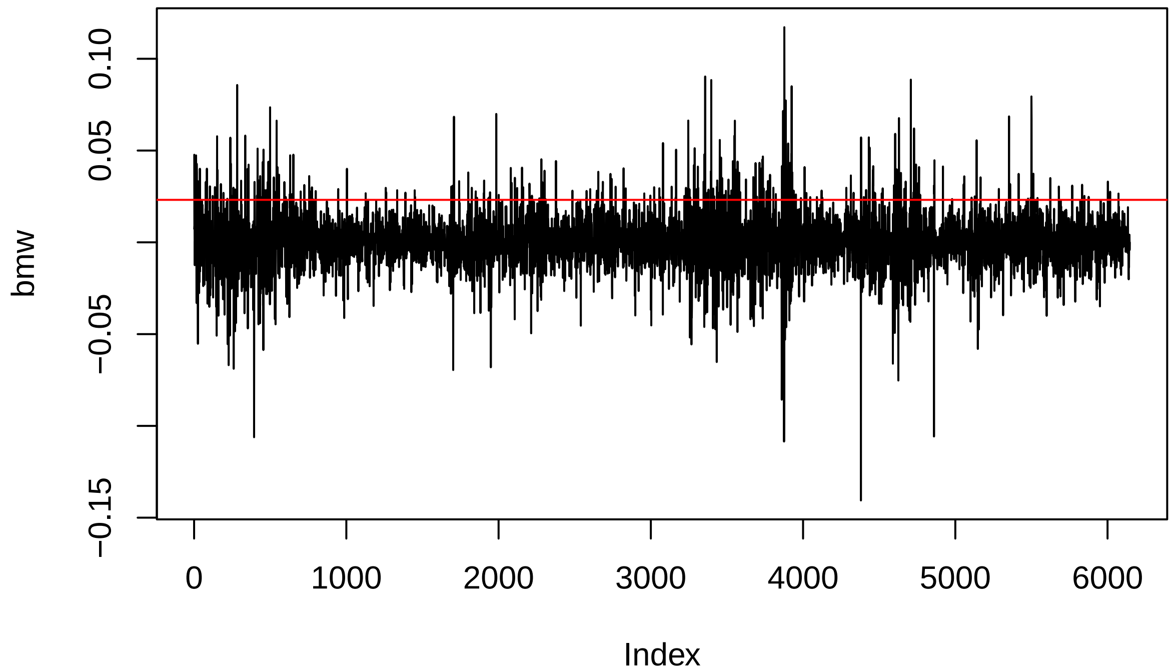



```r
#findthresh(danish, 100)
umbral=quantile(bmw,0.95)[[1]]
umbral
```

```
[1] 0.02313954
```

Aunque observamos que hay valores extremos muy grandes y muy pequeños, para efectos de examen solo ajustaremos para la cola derecha.

```r
plot(bmw,type="l")
abline(umbral,0,col="red")
```



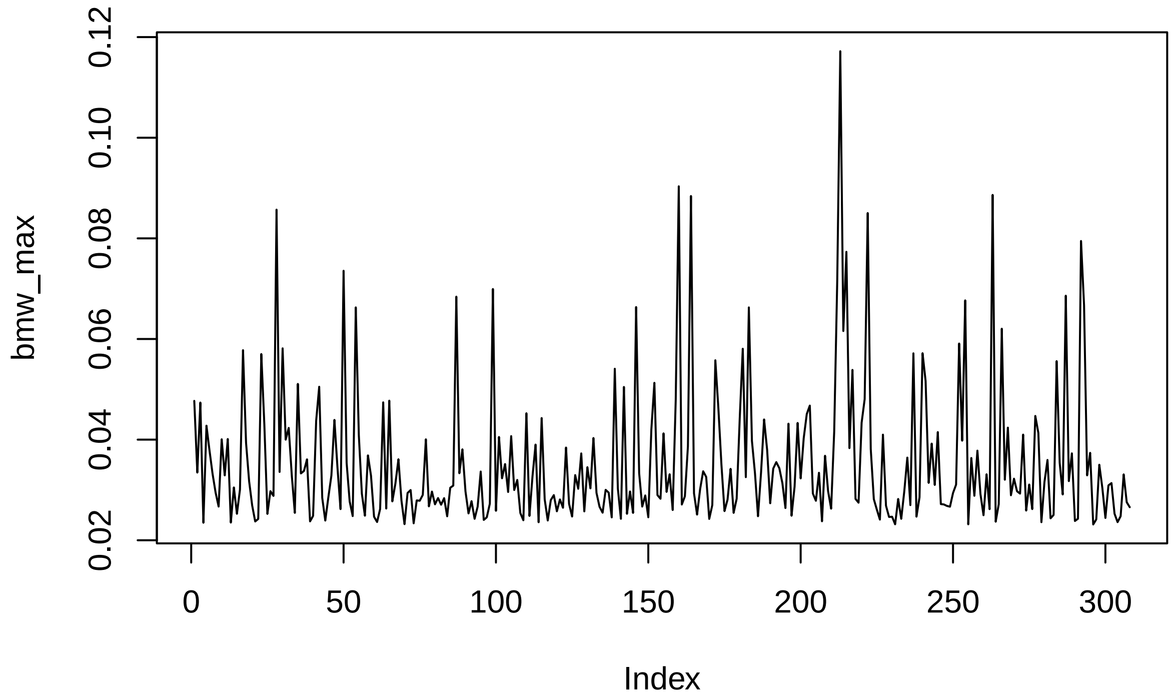
Seleccionamos los valores extremos muy grandes.

```
bmw_max = bmw [bmw > umbral]
bmw_max
```

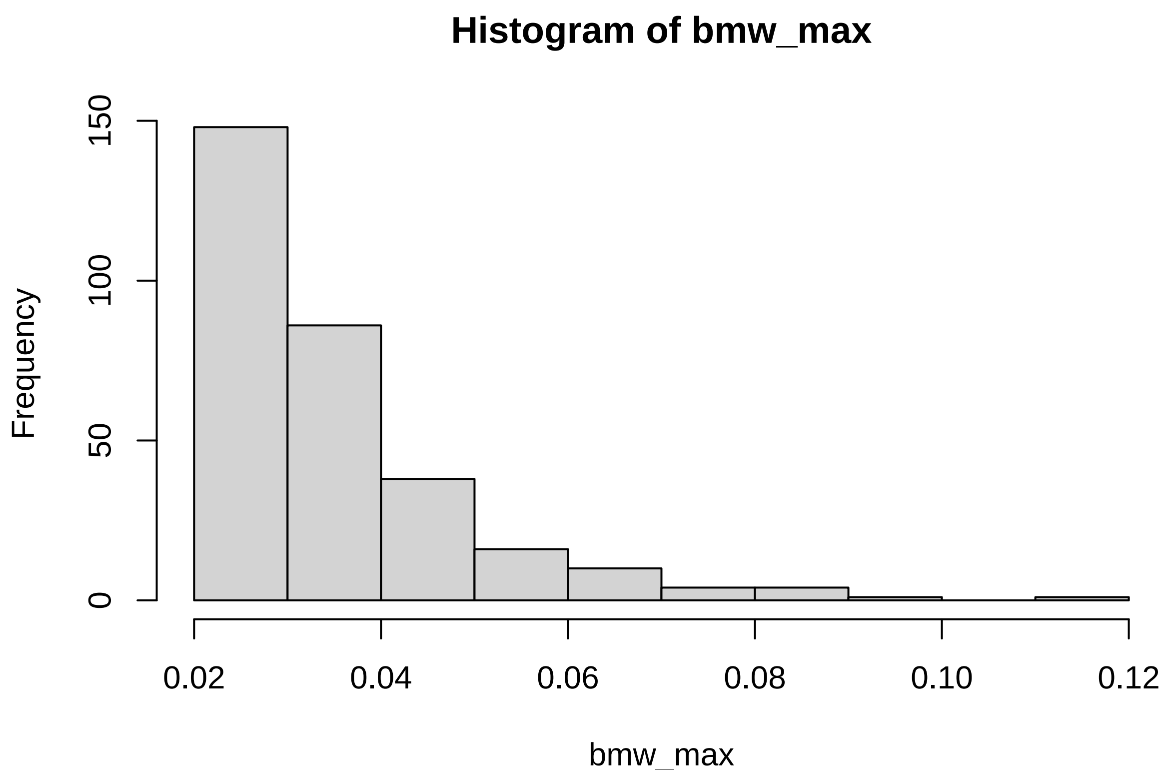
```
 [1] 0.04770410 0.03345517 0.04737021 0.02348919 0.04277424 0.03785610
 [7] 0.03323864 0.02949422 0.02669554 0.04007255 0.03289267 0.04012086
[13] 0.02351833 0.03049844 0.02526246 0.03007232 0.05777954 0.03938238
[19] 0.03174870 0.02694025 0.02373553 0.02424703 0.05700682 0.04296787
[25] 0.02524532 0.02975924 0.02882430 0.08570879 0.03358474 0.05814339
[31] 0.03999634 0.04232692 0.03290903 0.02545048 0.05106974 0.03328449
[37] 0.03378306 0.03608405 0.02375827 0.02484257 0.04373824 0.05050920
[43] 0.02842941 0.02391716 0.02854240 0.03285567 0.04391641 0.03443060
[49] 0.02619513 0.07356377 0.03538940 0.02771967 0.02480504 0.06627812
[55] 0.04093036 0.02926907 0.02488772 0.03686648 0.03282389 0.02468098
[61] 0.02363220 0.02621992 0.04740888 0.02630146 0.04774362 0.02773400
[67] 0.03121549 0.03609579 0.02786285 0.02320689 0.02939476 0.02998439
[73] 0.02337254 0.02792824 0.02785748
 [ reached getOption("max.print") -- omitted 233 entries ]
```

Valores fuera del umbral

```
plot(bmw_max,type="l")
```




```r
hist(bmw_max)
```

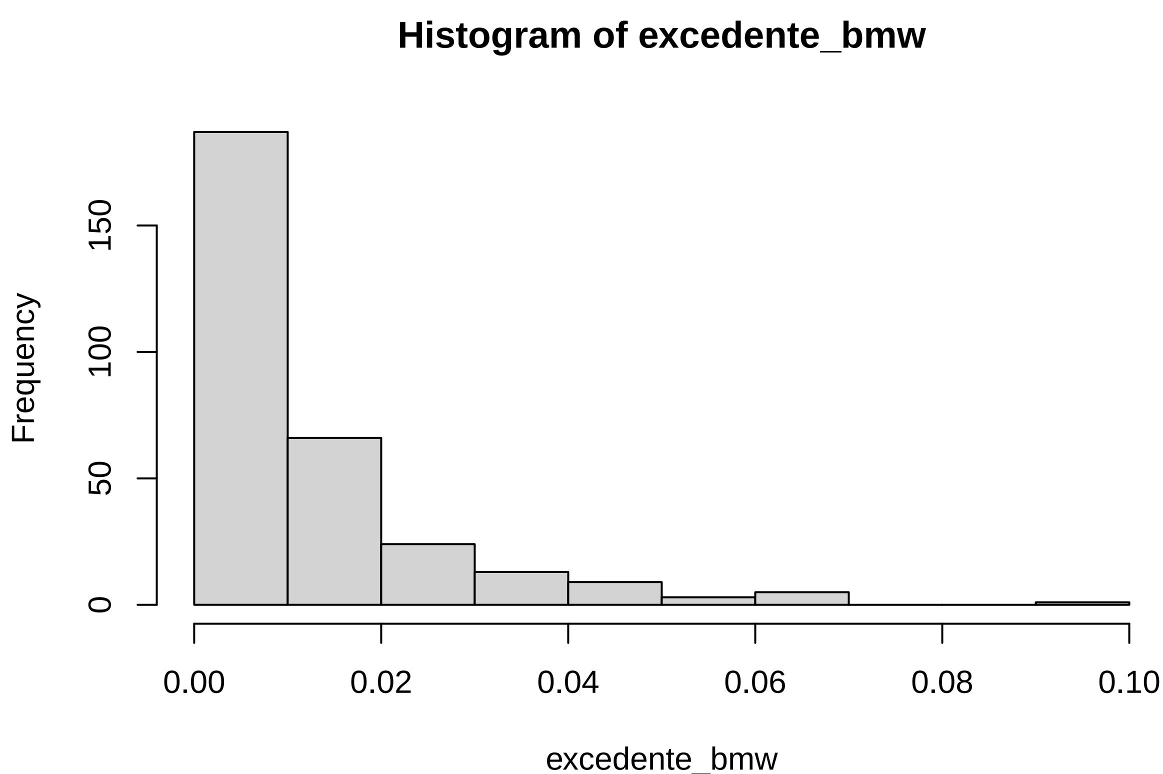



```r
## calculariamos los excedentes
excedente_bmw = bmw_max - umbral
excedente_bmw
```

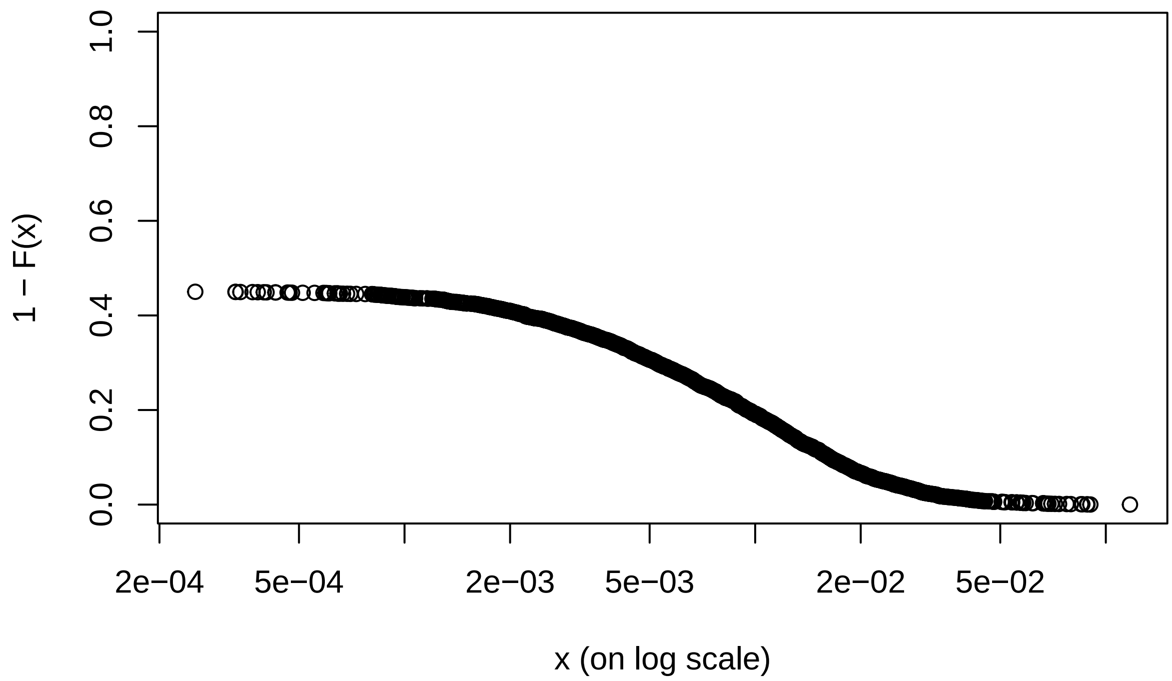
```
[1] 2.456456e-02 1.031564e-02 2.423068e-02 3.496522e-04 1.963470e-02
```

```
 [6] 1.471656e-02 1.009910e-02 6.354681e-03 3.556006e-03 1.693301e-02
[11] 9.753133e-03 1.698133e-02 3.787969e-04 7.358901e-03 2.122919e-03
[16] 6.932778e-03 3.464000e-02 1.624284e-02 8.609161e-03 3.800710e-03
[21] 5.959918e-04 1.107492e-03 3.386728e-02 1.982833e-02 2.105785e-03
[26] 6.619706e-03 5.684765e-03 6.256925e-02 1.044520e-02 3.500386e-02
[31] 1.685681e-02 1.918739e-02 9.769497e-03 2.310938e-03 2.793020e-02
[36] 1.014495e-02 1.064352e-02 1.294452e-02 6.187292e-04 1.703032e-03
[41] 2.059870e-02 2.736966e-02 5.289869e-03 7.776228e-04 5.402863e-03
[46] 9.716128e-03 2.077687e-02 1.129107e-02 3.055594e-03 5.042423e-02
[51] 1.224986e-02 4.580131e-03 1.665502e-03 4.313858e-02 1.779082e-02
[56] 6.129529e-03 1.748178e-03 1.372694e-02 9.684348e-03 1.541444e-03
[61] 4.926598e-04 3.080381e-03 2.426934e-02 3.161918e-03 2.460408e-02
[66] 4.594460e-03 8.075956e-03 1.295625e-02 4.723313e-03 6.734853e-05
[71] 6.255218e-03 6.844853e-03 2.330036e-04 4.788699e-03 4.717947e-03
 [ reached getOption("max.print") -- omitted 233 entries ]
```

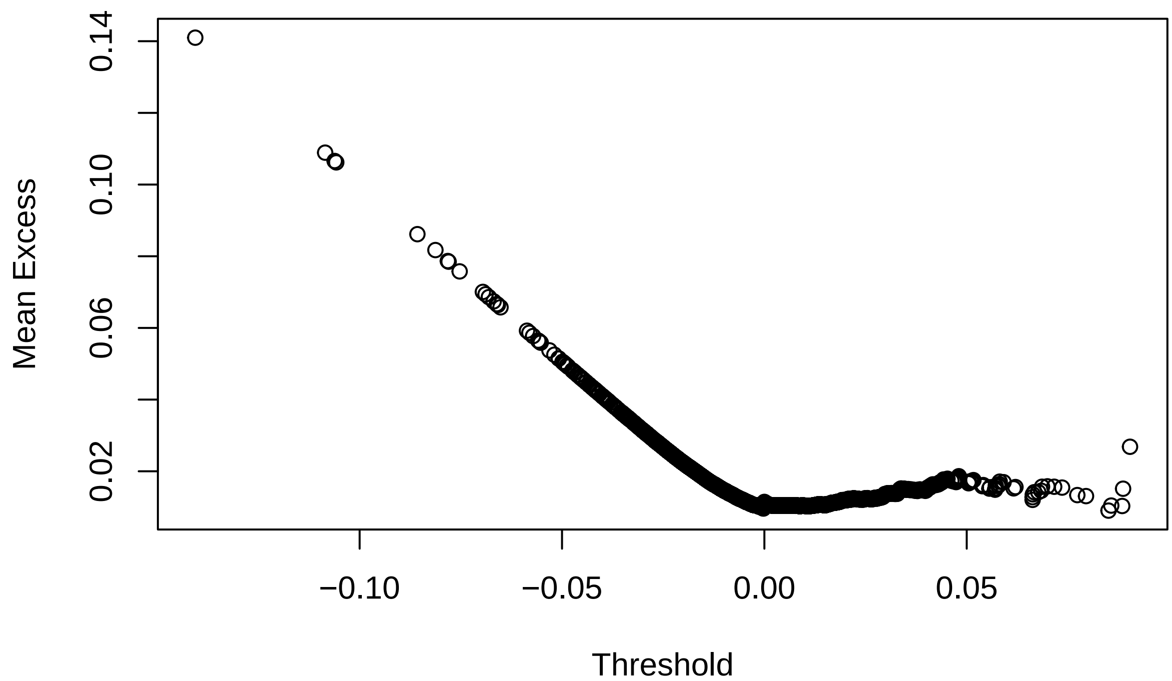
```r
hist(excedente_bmw)
```




```r
emplot(bmw) # distribución empírica
```




```
meplot(bmw, omit = 0) # gráfico los excesos medios de la muestra sobre los umbrales crecientes
```



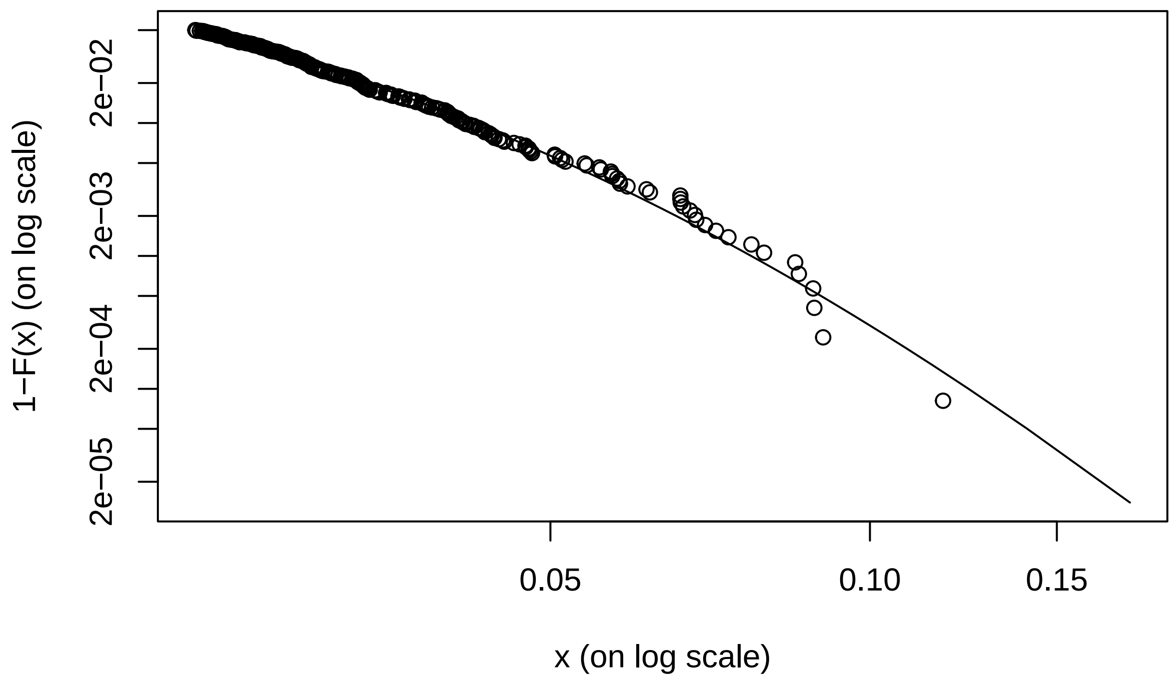

```
fit<- gpd(bmw,threshold=umbral, nextremes=NA) # umbral y cantidad
fit
```

```
$n
[1] 6146

$data
```

```
 [1] 0.04770410 0.03345517 0.04737021 0.02348919 0.04277424 0.03785610
 [7] 0.03323864 0.02949422 0.02669554 0.04007255 0.03289267 0.04012086
[13] 0.02351833 0.03049844 0.02526246 0.03007232 0.05777954 0.03938238
[19] 0.03174870 0.02694025 0.02373553 0.02424703 0.05700682 0.04296787
[25] 0.02524532 0.02975924 0.02882430 0.08570879 0.03358474 0.05814339
[31] 0.03999634 0.04232692 0.03290903 0.02545048 0.05106974 0.03328449
[37] 0.03378306 0.03608405 0.02375827 0.02484257 0.04373824 0.05050920
[43] 0.02842941 0.02391716 0.02854240 0.03285567 0.04391641 0.03443060
[49] 0.02619513 0.07356377 0.03538940 0.02771967 0.02480504 0.06627812
[55] 0.04093036 0.02926907 0.02488772 0.03686648 0.03282389 0.02468098
[61] 0.02363220 0.02621992 0.04740888 0.02630146 0.04774362 0.02773400
[67] 0.03121549 0.03609579 0.02786285 0.02320689 0.02939476 0.02998439
[73] 0.02337254 0.02792824 0.02785748
 [ reached getOption("max.print") -- omitted 233 entries ]

$threshold
[1] 0.02313954

$p.less.thresh
[1] 0.9498861

$n.exceed
[1] 308

$method
[1] "ml"

$par.ests
        xi       beta
0.12437449 0.01073653

$par.ses
          xi         beta
0.0654954914 0.0009074181

$varcov
              [,1]          [,2]
[1,]  4.289659e-03 -4.013864e-05
[2,] -4.013864e-05  8.234076e-07

$information
[1] "observed"

$converged
[1] 0

$nllh.final
[1] -1050.31

attr(,"class")
[1] "gpd"
```

Observamos que el modelo si ajusta bien.

```r
#gráfico de la cola de la distribución subyacente de los datos.
tp<-tailplot(fit)
```




```r
tp
```

```
$lastfit
$n
[1] 6146

$data
 [1] 0.04770410 0.03345517 0.04737021 0.02348919 0.04277424 0.03785610
 [7] 0.03323864 0.02949422 0.02669554 0.04007255 0.03289267 0.04012086
[13] 0.02351833 0.03049844 0.02526246 0.03007232 0.05777954 0.03938238
[19] 0.03174870 0.02694025 0.02373553 0.02424703 0.05700682 0.04296787
[25] 0.02524532 0.02975924 0.02882430 0.08570879 0.03358474 0.05814339
[31] 0.03999634 0.04232692 0.03290903 0.02545048 0.05106974 0.03328449
[37] 0.03378306 0.03608405 0.02375827 0.02484257 0.04373824 0.05050920
[43] 0.02842941 0.02391716 0.02854240 0.03285567 0.04391641 0.03443060
[49] 0.02619513 0.07356377 0.03538940 0.02771967 0.02480504 0.06627812
[55] 0.04093036 0.02926907 0.02488772 0.03686648 0.03282389 0.02468098
[61] 0.02363220 0.02621992 0.04740888 0.02630146 0.04774362 0.02773400
[67] 0.03121549 0.03609579 0.02786285 0.02320689 0.02939476 0.02998439
[73] 0.02337254 0.02792824 0.02785748
 [ reached getOption("max.print") -- omitted 233 entries ]

$threshold
[1] 0.02313954

$p.less.thresh
[1] 0.9498861
```

```
$n.exceed
[1] 308

$method
[1] "ml"

$par.ests
        xi       beta
0.12437449 0.01073653

$par.ses
          xi         beta
0.0654954914 0.0009074181

$varcov
              [,1]          [,2]
[1,]  4.289659e-03 -4.013864e-05
[2,] -4.013864e-05  8.234076e-07

$information
[1] "observed"

$converged
[1] 0

$nllh.final
[1] -1050.31

attr(,"class")
[1] "gpd"

$type
[1] "tail"

$dist
[1] "gpd"

$plotmin
[1] 0.02313954

$plotmax
[1] 0.1757877

$alog
[1] "xy"

$location
[1] -0.003695129

$shape
[1] 0.1243745

$scale
[1] 0.007398978
```

```r
# Los 3 parámetros - localización, escala, forma

loc<-tp$location
scal<-tp$scale
shape<-tp$shape

loc;scal;shape
```

```
[1] -0.003695129

[1] 0.007398978

[1] 0.1243745
```

```r
riskmeasures(fit, 0.99)
```

```
        p   quantile      sfall
[1,] 0.99 0.04230004 0.05728316
```

4. Considere la spline de suavizado cúbico natural que suaviza los puntos (0, 0), (1, 2), (2, 1), (3, 3) usando $p = 0.9$ y desviaciones estándar de 0.5.

(a) Obtenga los valores de las intersecciones de los nodos.

```r
p <- 0.9
H <- matrix(c(4,1,1,4),nrow = 2, byrow = T)
H
```

```
     [,1] [,2]
[1,]    4    1
[2,]    1    4
```

```r
R <- matrix(c(1,-2,1,0,0,1,-2,1),nrow = 2, byrow = T)
R
```

```
     [,1] [,2] [,3] [,4]
[1,]    1   -2    1    0
[2,]    0    1   -2    1
```

```r
R1 <- 6*R
R1
```

```
     [,1] [,2] [,3] [,4]
[1,]    6  -12    6    0
[2,]    0    6  -12    6
```

```r
S <- diag(x = 0.25, nrow = 4, ncol = 4)
S
```

```
     [,1] [,2] [,3] [,4]
[1,] 0.25 0.00 0.00 0.00
[2,] 0.00 0.25 0.00 0.00
[3,] 0.00 0.00 0.25 0.00
[4,] 0.00 0.00 0.00 0.25
```

```r
res <- R1%*%S%*%t(R1)
res
```

```
     [,1] [,2]
[1,]   54  -36
[2,]  -36   54
```

```r
res2 <- H + ((1-p)/(6*p))*res
res2
```

```
          [,1]      [,2]
[1,] 5.0000000 0.3333333
[2,] 0.3333333 5.0000000
```

```r
inversa <- solve(res2)
inversa
```

```
            [,1]        [,2]
[1,]  0.20089286 -0.01339286
[2,] -0.01339286  0.20089286
```

```r
y <-c(0,2,1,3)
y
```

```
[1] 0 2 1 3
```

```r
a <- y - (1-p)/(6*p) * S%*%t(R1) %*% inversa %*% R1%*%y

cat("Respuesta: a = ", a)
```

```
Respuesta: a =  0.1071429 1.678571 1.321429 2.892857
```

(b) Obtenga el spline de suavizado cúbico natural como el spline de interpolación natural.

```r
H
```

```
     [,1] [,2]
[1,]    4    1
[2,]    1    4
```

```r
u <- matrix(c(-11.571,11.571))

m <- solve(H)%*%u
m
```

```
       [,1]
[1,] -3.857
[2,]  3.857
```

```r
m1 <-c(0,m,0)
m1
```

```
[1]  0.000 -3.857  3.857  0.000
```

```r
y<- c(0,2,1,3)
x<- c(0,1,2,3)

h <- c(1,1,1)
g <- c(4,4)
u<- c(-18,18)

xj<-c()
aj<- c()
bj <- c()
cj <- c()
dj <- c()
for (i in 1:3){

  xj[i] <- x[i]

  aj[i] <- a[i]

  bj[i] <- ((a[i+1]-a[i])/h[i]) - ((h[i]*(2*m1[i] + m1[i+1]))/6)

  cj[i] <- m1[i]/2

  dj[i] <- (m1[i+1]-m1[i])/(6*h[i])

  print(i)
}
```

```
[1] 1
[1] 2
[1] 3
```

```r
data.frame(xj,aj,bj,cj,dj)
```

```
  xj        aj        bj      cj         dj
1  0 0.1071429 2.2142619  0.0000 -0.6428333
2  1 1.6785714 0.2856905 -1.9285  1.2856667
3  2 1.3214286 0.2857619  1.9285 -0.6428333
```

(c) Grafique el spline resultante de $x = 0.5$ a $x = 2.5$.

```
plot(x, y, main = "spline(.)")
lines(spline(x, y, n = 201), col = 2)
lines(spline(x, y, n = 201, method = "natural"), col = 3)
lines(spline(x, y, n = 201, method = "periodic"), col = 4)
legend(6,25, c("fmm","natural","periodic"), col=2:4, lty=1)
```