Laboratorio 1- Análisis estadístico de datos extremales

laura montaldo

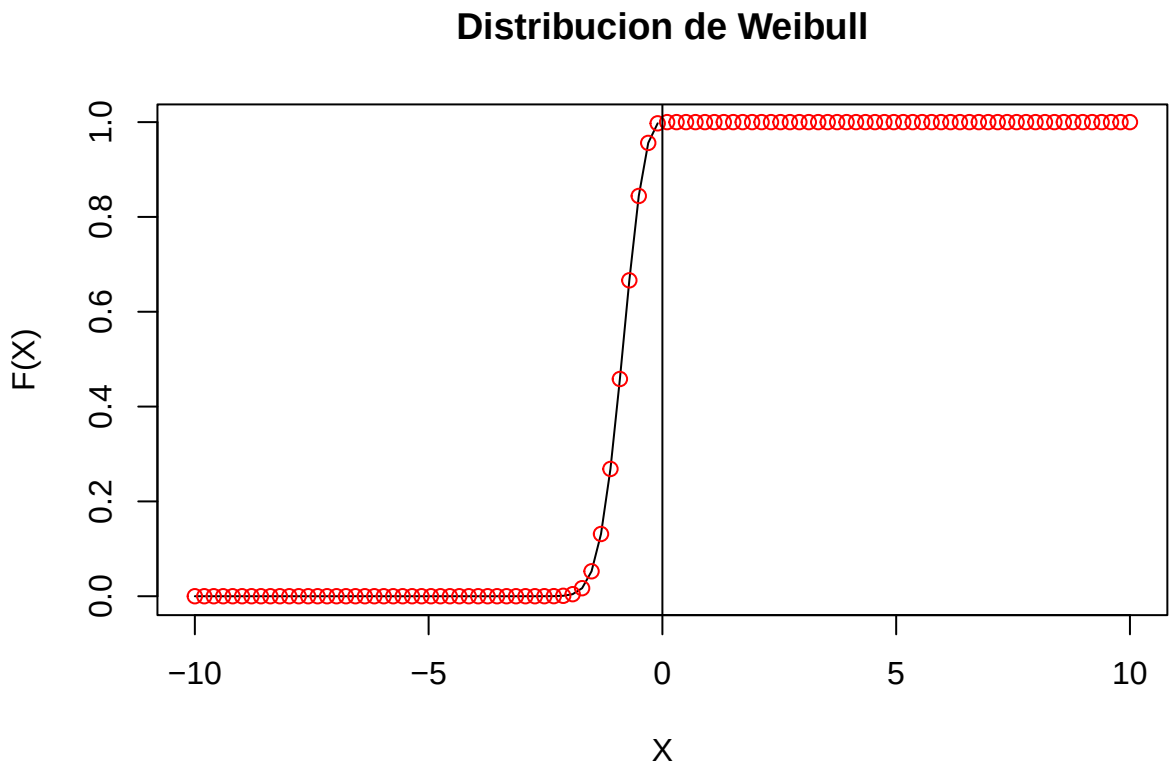
2023-09-08

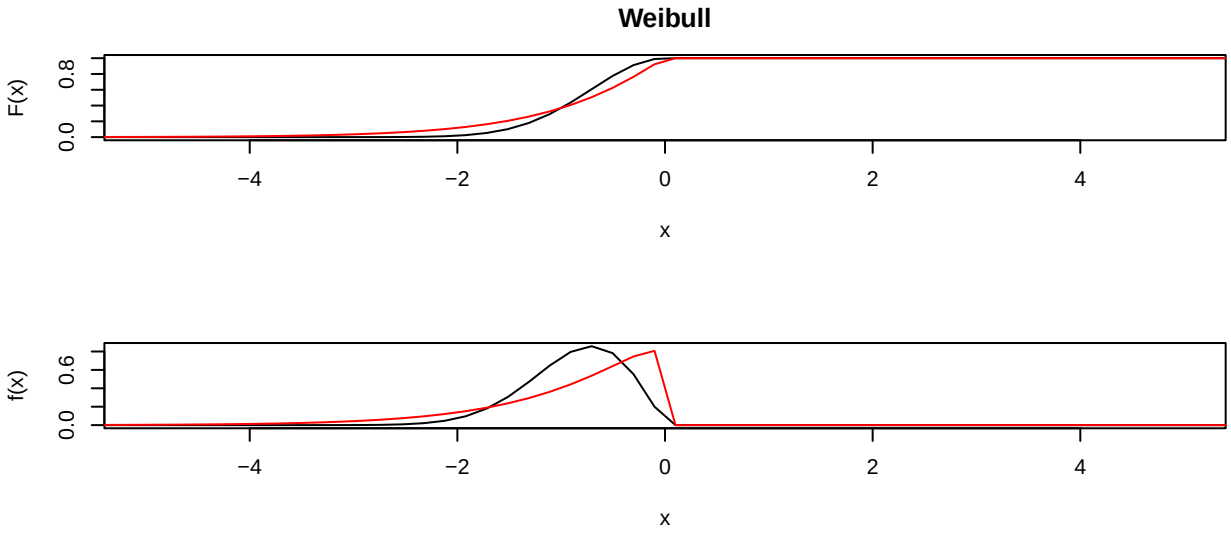
Distrubiciones extremales

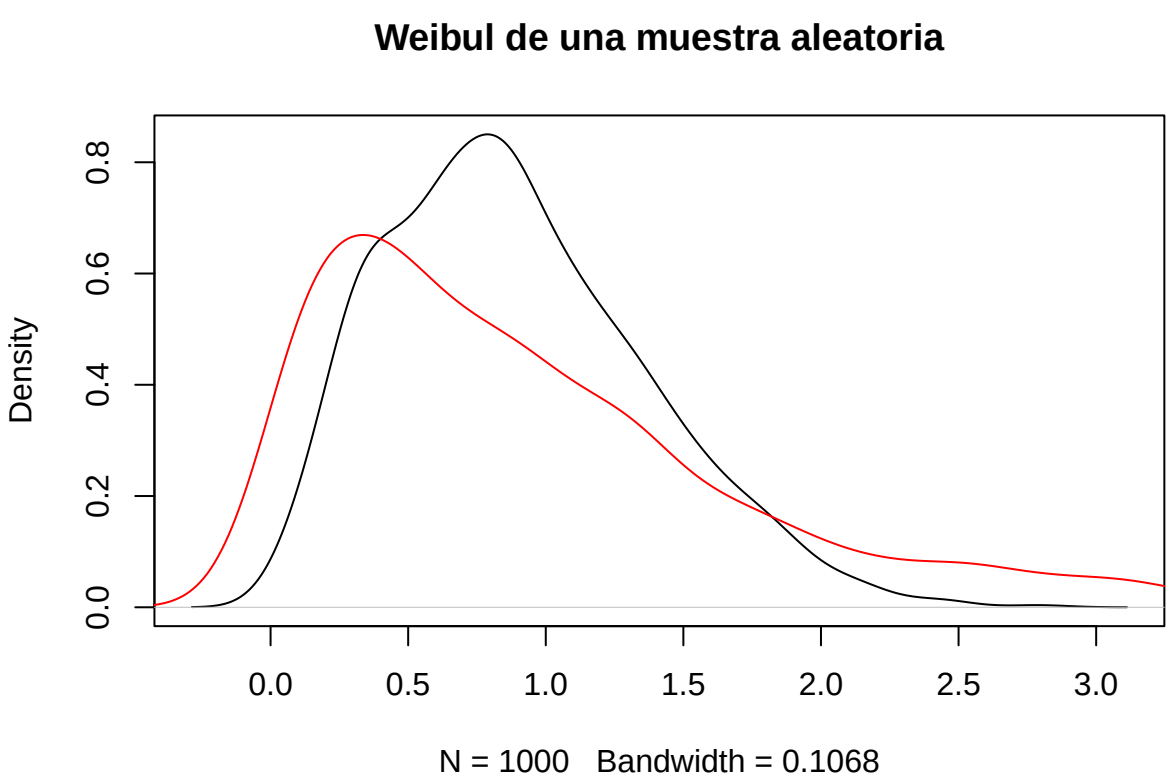
Weibull

```
 x_aux<-seq(-10, 10, length=100)
head( x_aux)
```

```
## [1] -10.000000  -9.797980  -9.595960  -9.393939  -9.191919  -8.989899
```




Weibull
F(x)
0.0
0.8
−4
−2
0
2
4
x
f(x)
0.0
0.6
−4
−2
0
2
4
x



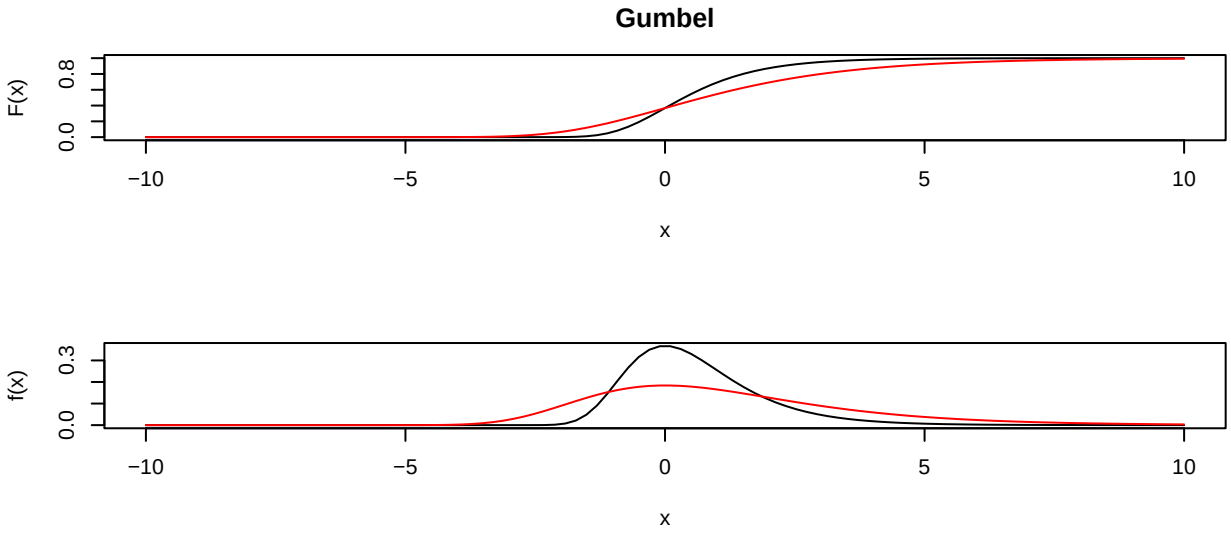
Gumbel

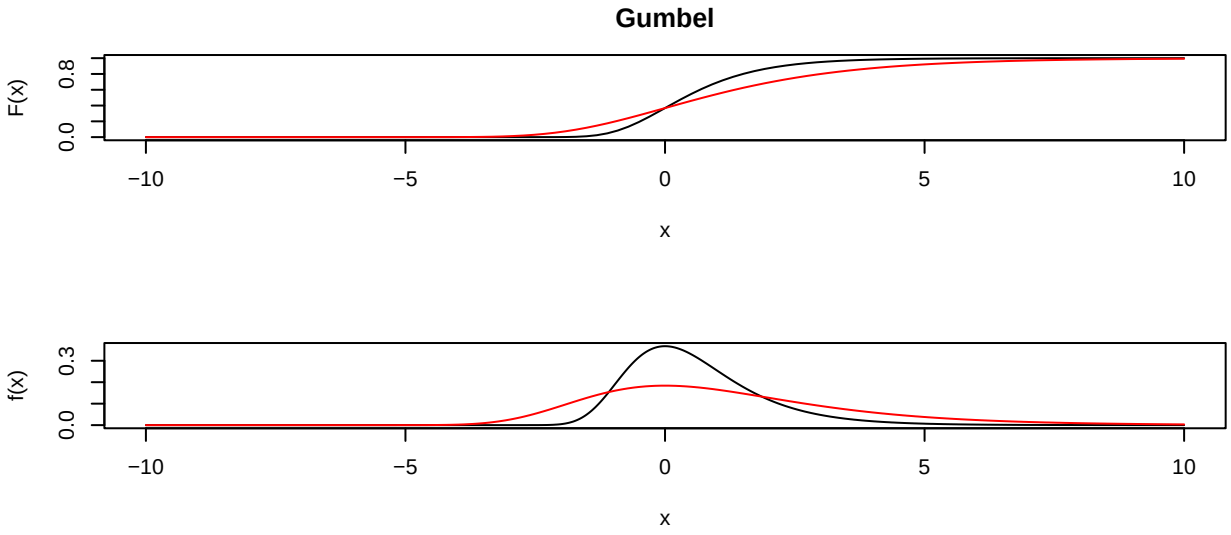
```
require(evd)
```

```
## Loading required package: evd
```

```
par(mfrow=c(3,1), mar=c(5,4,3,1))
plot(seq(-10,10,length=100), pgumbel(q=seq(-10,10,length=100), loc=0, scale=1) ,xlim=c(-10,10), type="l
lines(seq(-10,10,length=100), pgumbel(q=seq(-10,10,length=100), loc=0, scale=2), col="red")

plot(x_aux, dgumbel(x=x_aux, loc=0, scale=1, log = FALSE) ,xlim=c(-10,10), type="l", ylab="f(x)", xlab=
lines(x_aux, dgumbel(x=x_aux, loc=0, scale=2, log = FALSE), col="red") # Cambio la escala de la Gumbel
```


Gumbel
F(x)
0.0
0.8
−10
−5
0
5
10
x
f(x)
0.0
0.3
−10
−5
0
5
10
x




```
GumbelAleatorio<-rgumbel(100)
-digamma(1)
```

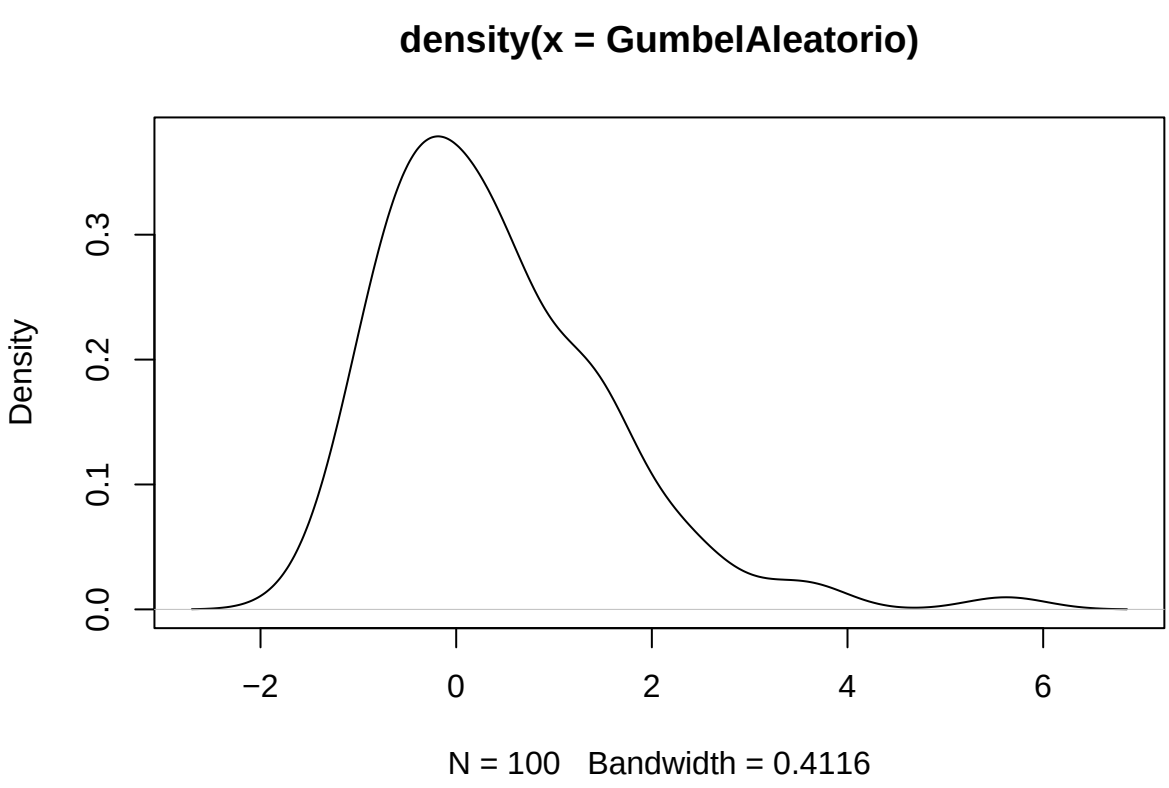
```
## [1] 0.5772157
```

```
mean(rgumbel(1000))
```

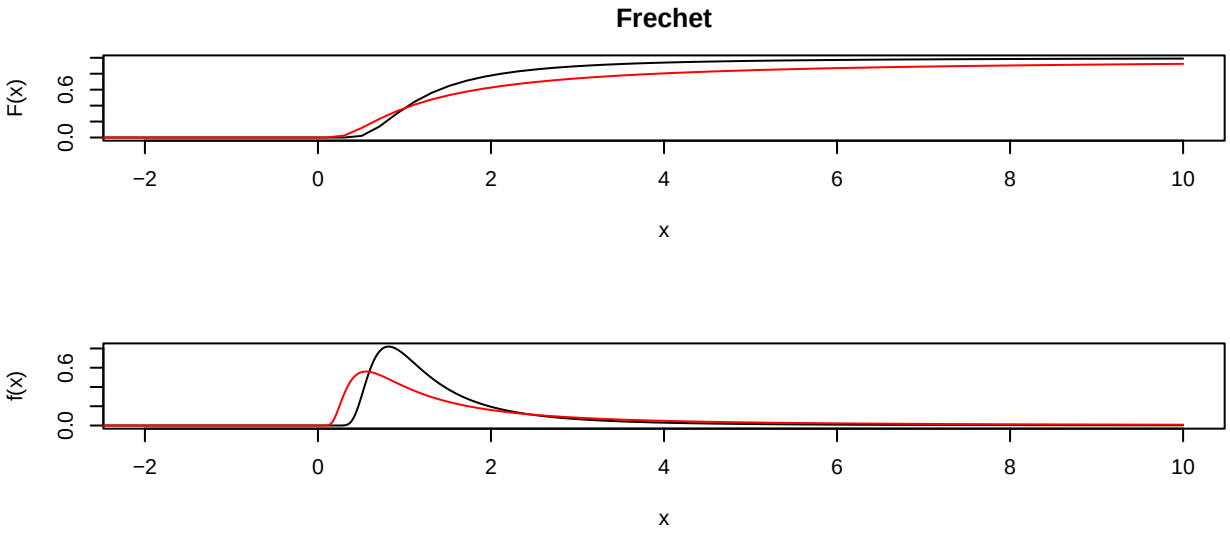
```
## [1] 0.5628083
```

```
sd(rgumbel(1000))
```

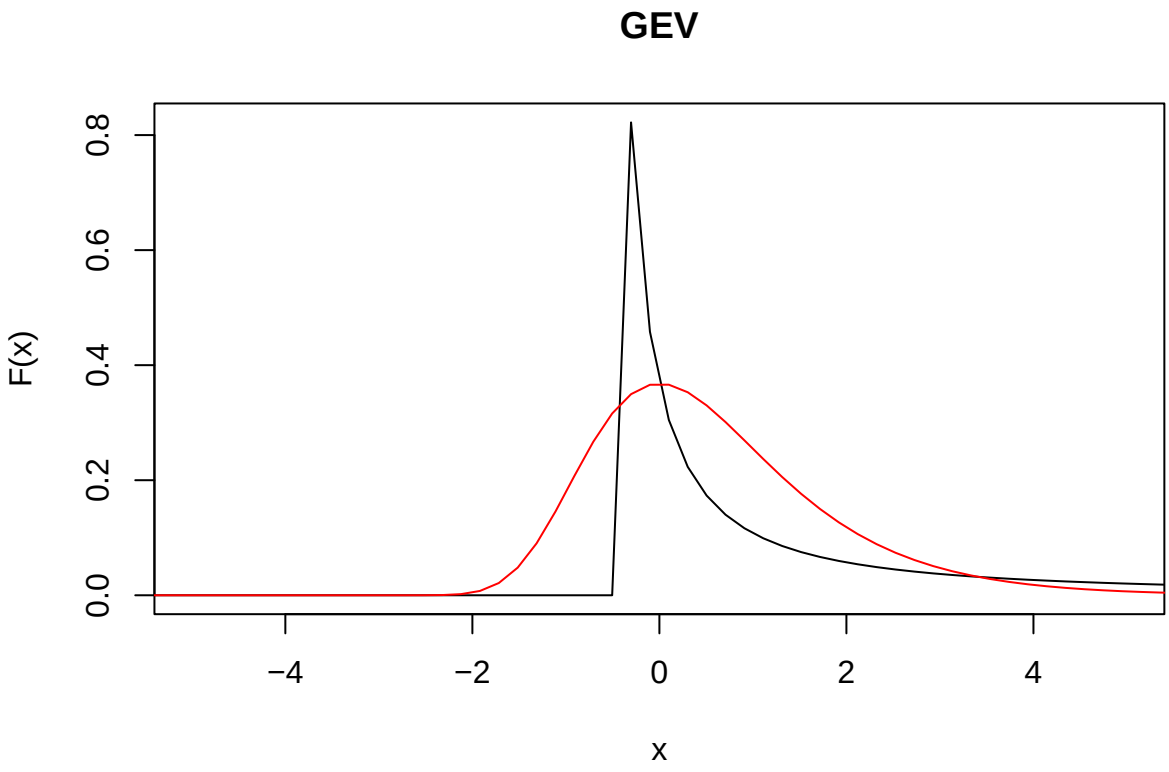
```
## [1] 1.308391
```


density(x = GumbelAleatorio)
Density
0.0
0.1
0.2
0.3
−2
0
2
4
6
N = 100 Bandwidth = 0.4116

Frechet



GEV Distribución Extremal Generalizada (DEG)




```
require(ismev)
```

```
## Loading required package: ismev
```

```
## Loading required package: mgcv
```

```
## Loading required package: nlme
```

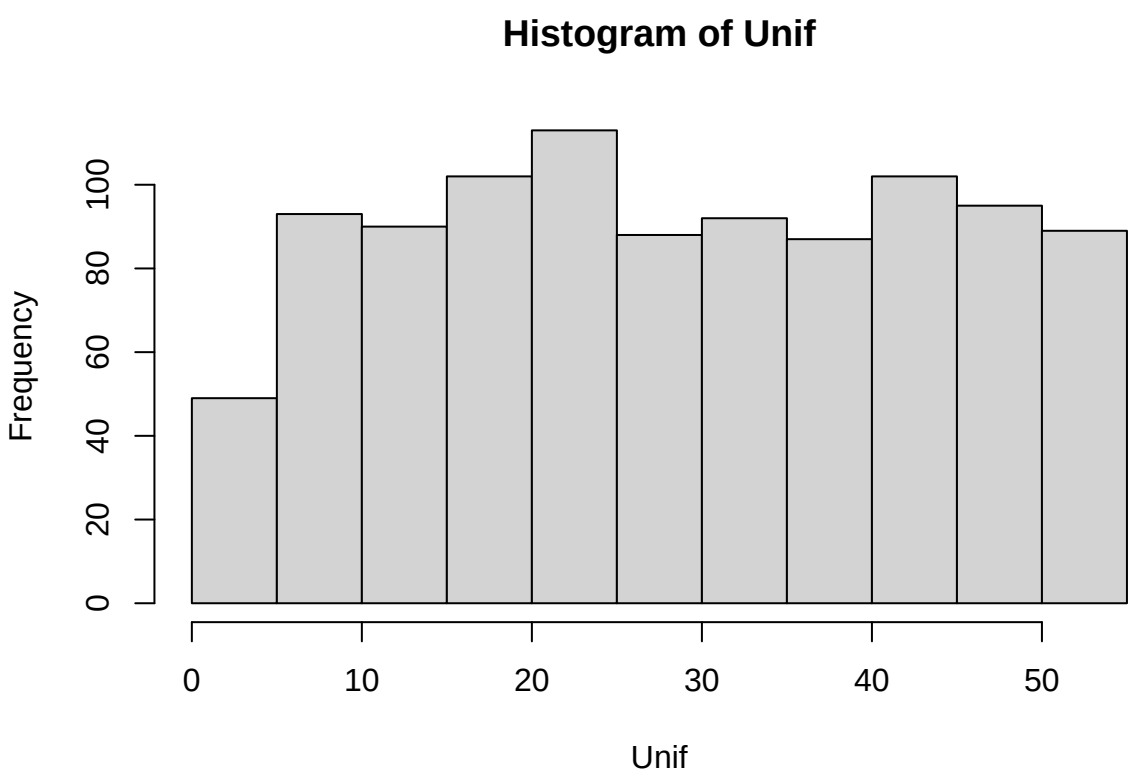
```
## This is mgcv 1.8-42. For overview type 'help("mgcv-package")'.
```

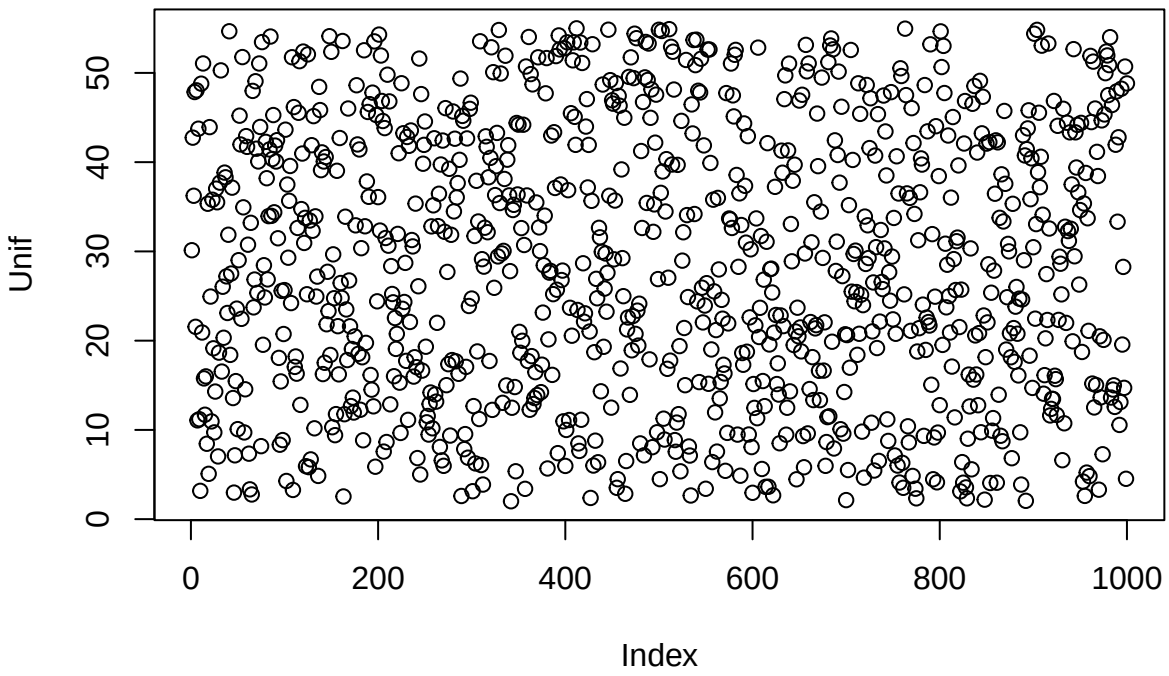
```
set.seed(69)
Unif<-runif(1000, 2, 55)
max(Unif)
```

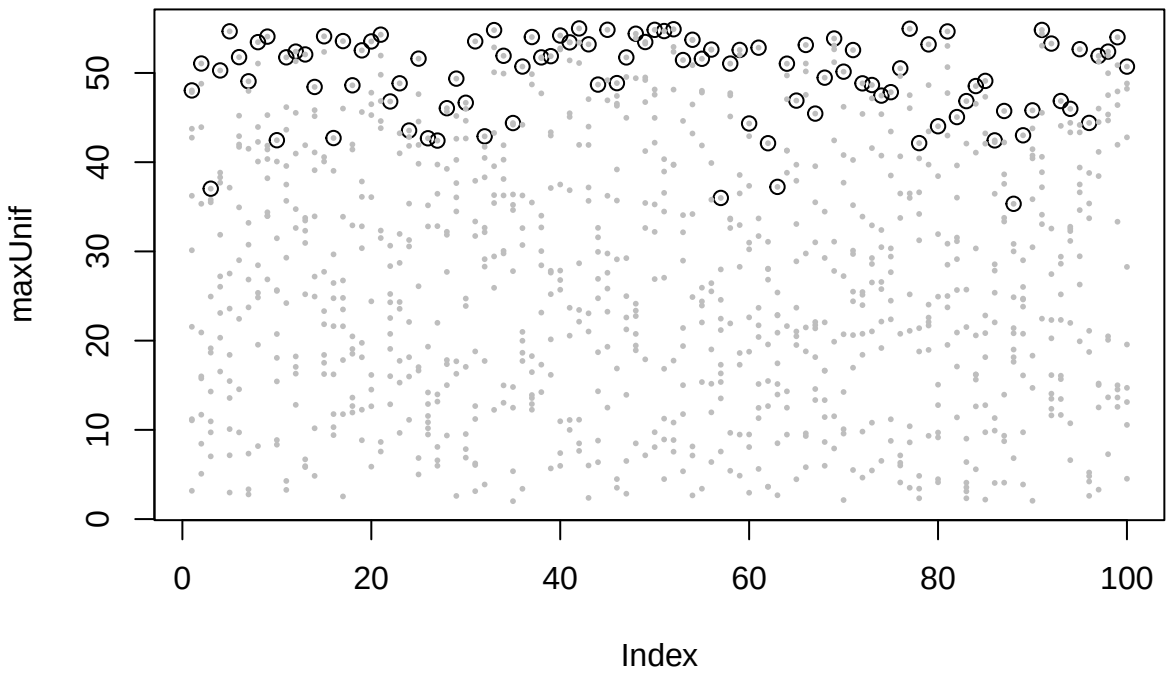
```
## [1] 54.99179
```

```
min(Unif)
```

```
## [1] 2.007008
```


Histogram of Unif
Frequency
0
20
40
60
80
100
0
10
20
30
40
50
Unif


Unif
0
10
20
30
40
50
0
200
400
600
800
1000
Index




```
# Segun el Teorema de Fischer-Tippet-Gnedenko (FTG), Teorema 3, pg 12 y por la Observación 9 pg 13, deb
# Que en el caso de la distribucion de valores extremos generalizada (GEV, pg 34), corresponde a un val
# Esta funcion ajusta por maxima verosimilitud los parametros de posicion, escala e indice
modUnif<-gev.fit(maxUnif) # la funcion gev.fit ajusta una GEV por mve y lo salva en el objeto "modUnif"
```

```
## $conv
## [1] 0
##
## $nllh
## [1] 270.0331
##
## $mle
## [1] 49.4491181  5.3686186 -0.9682121
##
## $se
## [1] 8.439391e-02 7.672301e-02 2.000270e-06
```

```
str(modUnif)
```

```
## List of 10
##  $ trans: logi FALSE
##  $ model:List of 3
##   ..$ : NULL
##   ..$ : NULL
##   ..$ : NULL
##  $ link : chr "c(identity, identity, identity)"
##  $ conv : int 0
```

```
##  $ nllh : num 270
##  $ data : Named num [1:100] 48 51.1 37 50.3 54.7 ...
##   ..- attr(*, "names")= chr [1:100] "1" "2" "3" "4" ...
##  $ mle  : num [1:3] 49.449 5.369 -0.968
##  $ cov  : num [1:3, 1:3] 7.12e-03 -6.44e-03 -1.00e-11 -6.44e-03 5.89e-03 ...
##  $ se   : num [1:3] 0.084394 0.076723 0.000002
##  $ vals : num [1:100, 1:3] 49.4 49.4 49.4 49.4 49.4 ...
##  - attr(*, "class")= chr "gev.fit"
```

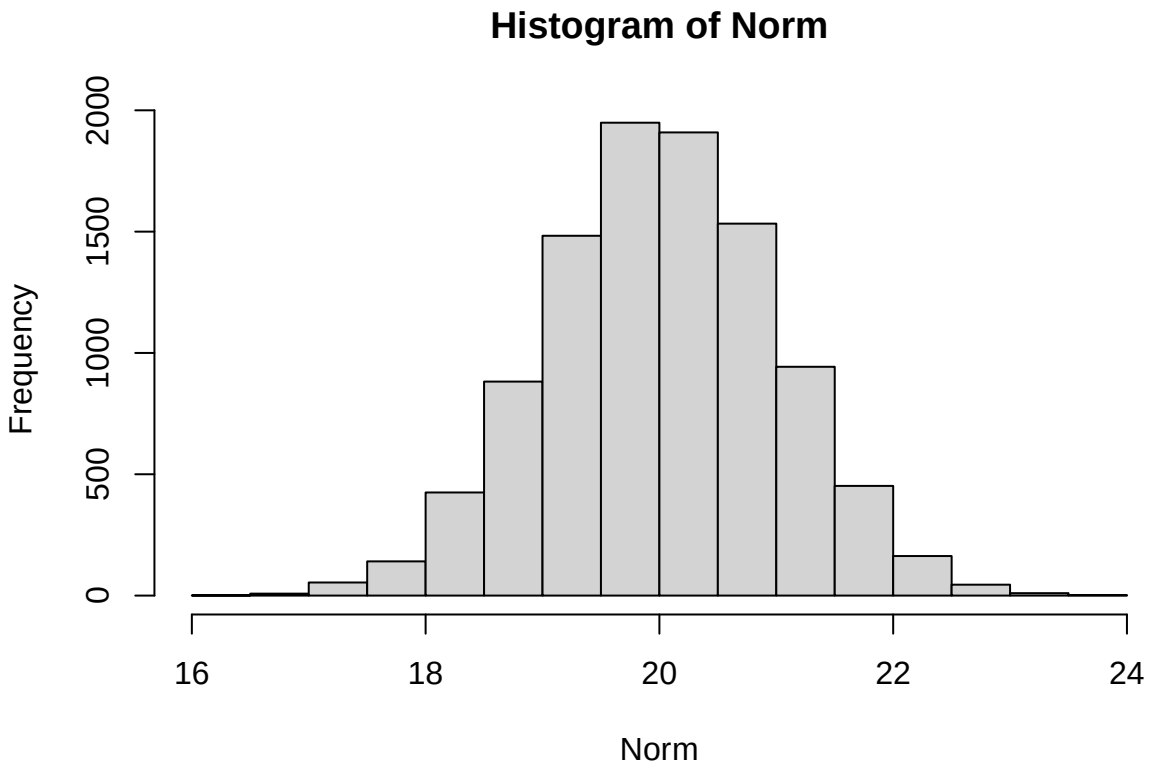
```
modUnif$mle
```

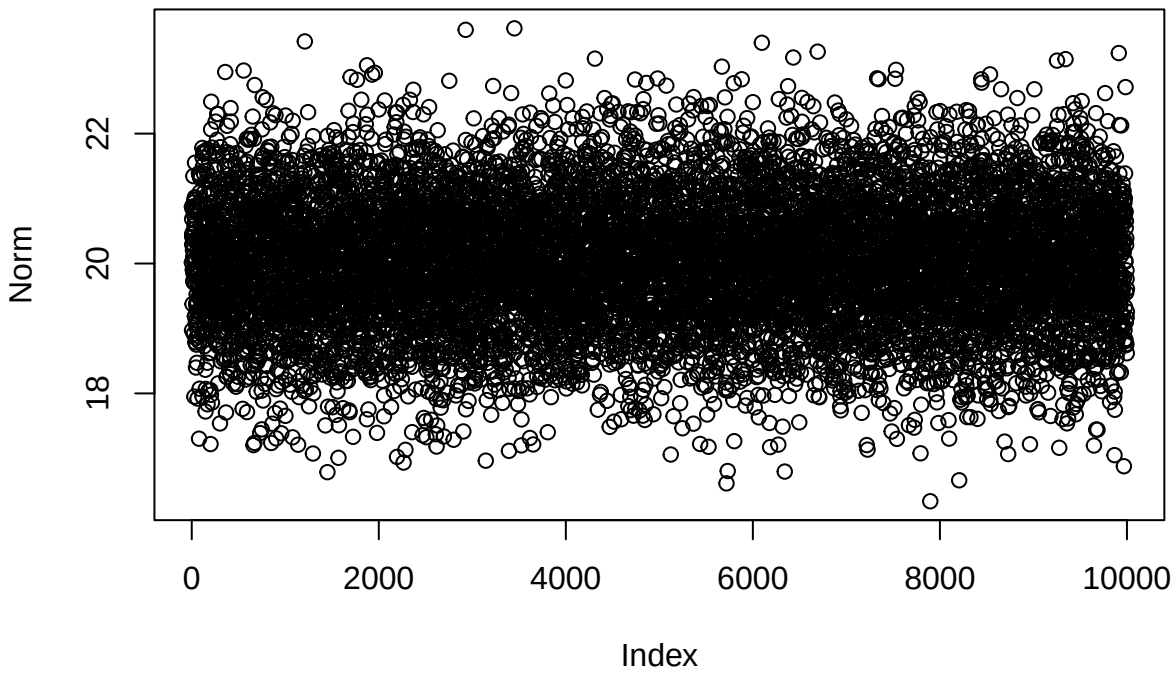
```
## [1] 49.4491181  5.3686186 -0.9682121
```

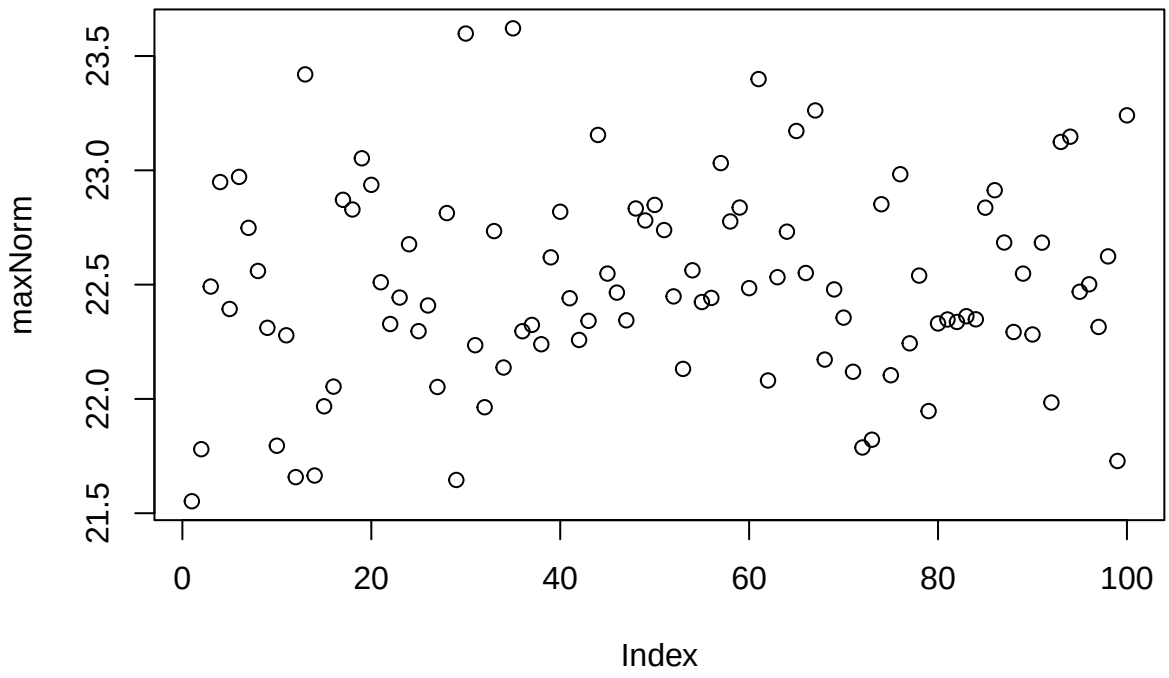
```
data.frame(parameter=c("posicion", "escala","indice"), Estimado=round(modUnif$mle,3), SE=modUnif$se)
```

```
##   parameter Estimado           SE
## 1  posicion   49.449 8.439391e-02
## 2    escala    5.369 7.672301e-02
## 3    indice   -0.968 2.000270e-06
```

NORMAL




Norm
18
20
22
0
2000
4000
6000
8000
10000
Index




```
# ¿Cual es el valor esperado del indice?
modNorm<-gev.fit(maxNorm)
```

```
## $conv
## [1] 0
##
## $nllh
## [1] 58.30997
##
## $mle
## [1] 22.3374486  0.4217606 -0.2266639
##
## $se
## [1] 0.04645616 0.03226844 0.06136402
```

```
data.frame(parameter=c("posicion", "escala","indice"), Estimate=round(modNorm$mle,3), SE=modNorm$se)
```

```
##   parameter Estimate         SE
## 1  posicion   22.337 0.04645616
## 2    escala    0.422 0.03226844
## 3    indice   -0.227 0.06136402
```

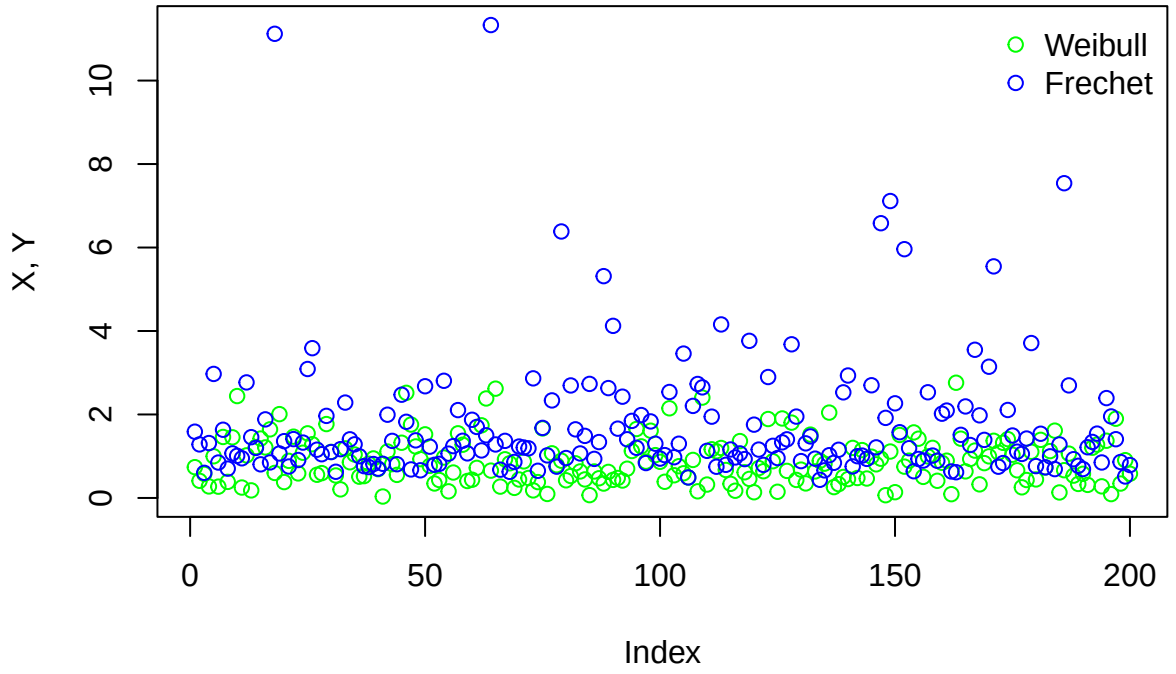
```
modNorm$mle[3]+ 1.96* modNorm$se[3]
```

```
## [1] -0.1063905
```

```
modNorm$mle[3]- 1.96* modNorm$se[3]
```

```
## [1] -0.3469374
```

```r
require(evd)
n=200
set.seed(69)
randomWeibull<-rweibull(n=n, shape=1.5, scale=1)
randomFrechet<-rfrechet(n=n, loc=0, shape= 2, scale=1) # loc es posicion, shape= es el parametro de for
MixtureDist<-cbind(randomWeibull, randomFrechet)
```




```r
#max(X,Y)
maxDist<-apply(MixtureDist, 1 ,max)
modMixture<-gev.fit(maxDist)
```

```
## $conv
## [1] 0
##
## $nllh
## [1] 239.2991
##
## $mle
## [1] 1.1841214 0.5308165 0.4388088
##
## $se
## [1] 0.04278010 0.03952826 0.06713168
```

```r
modFrech<-gev.fit(randomFrechet)
```

```
## $conv
## [1] 0
```

```
##
## $nllh
## [1] 232.9541
##
## $mle
## [1] 1.0732692 0.5035662 0.4727500
##
## $se
## [1] 0.04027668 0.03814067 0.06611307
```

```
data.frame(round(modFrech$mle,3), c("posicion", "escala","indice"))
```

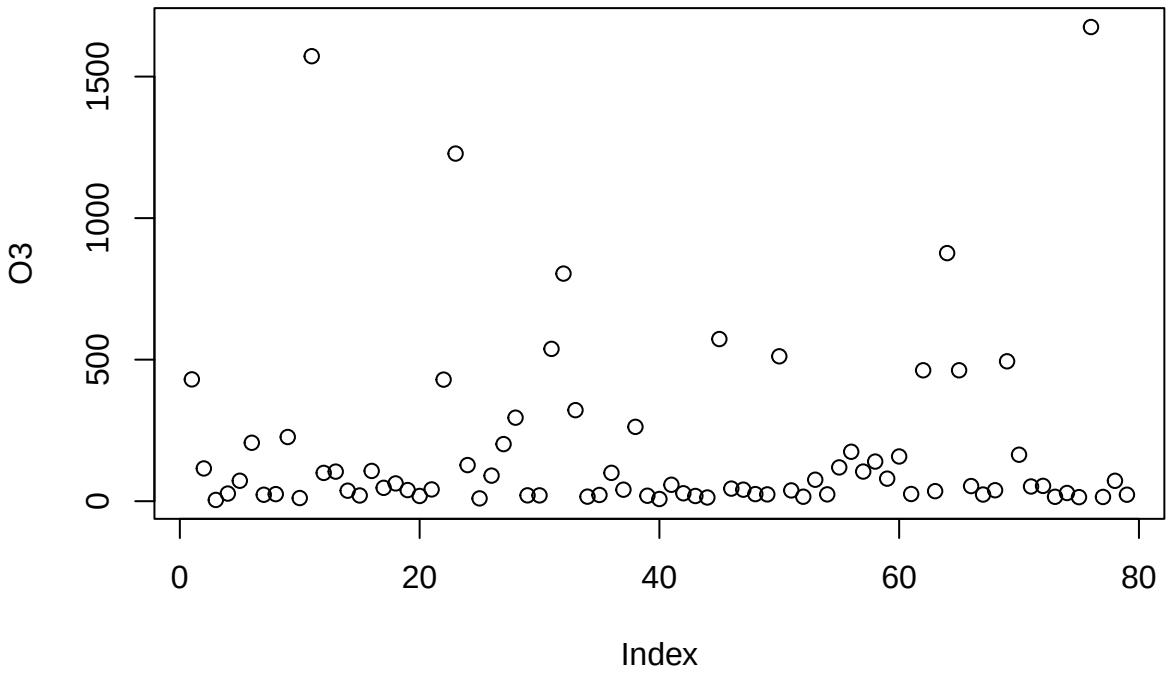
```
##   round.modFrech.mle..3. c..posicion....escala....indice..
## 1                  1.073                          posicion
## 2                  0.504                            escala
## 3                  0.473                            indice
```

```
1/0.44 # cercano a 2 que asignamos en la creación de la variable aleatoria (shape=2).
```

```
## [1] 2.272727
```

```
##   parameter Estimate         SE
## 1  posicion    1.184 0.04278010
## 2    escala    0.531 0.03952826
## 3    indice    0.439 0.06713168
```

OZONO



```r
# Ajustamos a estos datos por maxima verosimiliud una distribución extremal generalizada
mod03<-gev.fit(03)
```

```
## $conv
## [1] 0
##
## $nllh
## [1] 468.9038
##
## $mle
## [1] 37.367980 41.931248  1.096984
##
## $se
## [1] 5.2378582 7.5503443 0.1421071
```

```r
str(mod03)
```

```
## List of 10
##  $ trans: logi FALSE
##  $ model:List of 3
##   ..$ : NULL
##   ..$ : NULL
##   ..$ : NULL
##  $ link : chr "c(identity, identity, identity)"
##  $ conv : int 0
##  $ nllh : num 469
##  $ data : num [1:79] 430.3 115.7 4.48 26.95 72.27 ...
##  $ mle  : num [1:3] 37.4 41.9 1.1
##  $ cov  : num [1:3, 1:3] 27.435 34.442 -0.018 34.442 57.008 ...
##  $ se   : num [1:3] 5.238 7.55 0.142
##  $ vals : num [1:79, 1:3] 37.4 37.4 37.4 37.4 37.4 ...
##  - attr(*, "class")= chr "gev.fit"
```

```r
mod03$mle
```

```
## [1] 37.367980 41.931248  1.096984
```

```r
mod03shift<-gev.fit(03-15)
```

```
## $conv
## [1] 0
##
## $nllh
## [1] 468.9038
##
## $mle
## [1] 22.367980 41.931248  1.096984
##
## $se
## [1] 5.2378566 7.5503411 0.1421071
```

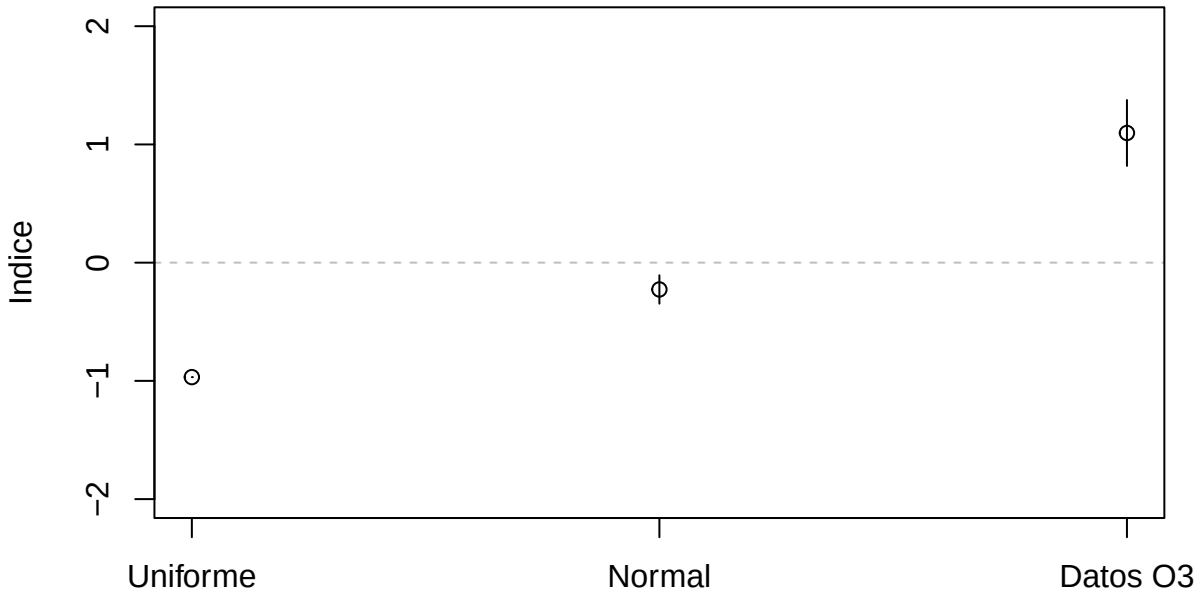
```r
# Resumen del ajuste (debo hacerlo manualmente pues no hay una funcion equivalente al summary)
data.frame(parameter=c("posicion", "escala","indice"), cbind(Estimate=mod03$mle, se=mod03$se))
```

```
##   parameter  Estimate       se
## 1  posicion 37.367980 5.237858
## 2    escala 41.931248 7.550344
```

```
## 3    indice  1.096984 0.1421071
```

```
data.frame(parameter=c("posicion", "escala","indice"), cbind(Estimate=mod03shift$mle, se=mod03shift$se))
```

```
##   parameter  Estimate        se
## 1  posicion 22.367980 5.2378566
## 2    escala 41.931248 7.5503411
## 3    indice  1.096984 0.1421071
```



Niveles y tiempo de retorno

```
# Ajustemos la distribución extremal con el paquete extRemes y la función "fevd". Ajusto por Maxima ver
library(extRemes)
```

```
## Loading required package: Lmoments
```

```
## Loading required package: distillery
```

```
##
## Attaching package: 'extRemes'
```

```
## The following objects are masked from 'package:evd':
##
##     fbvpot, mrlplot
```

```
## The following objects are masked from 'package:stats':
##
##     qqnorm, qqplot
```

```r
modNormext<-fevd(maxNorm)
modNormext$results$par
```

```
##   location      scale      shape
## 22.3373972  0.4216981 -0.2265683
```

```r
# nombro un vector con los parámetros ajustados
a<-modNormext$results$par
# comparamos con los resultados del ajuste que hicimos previamente
modNorm$mle
```

```
## [1] 22.3374486  0.4217606 -0.2266639
```

```r
# Estimemos el valor de retorno para los 10 años
return.level(modNormext, return.period = c(10), do.ci=TRUE)
```

```
## fevd(x = maxNorm)
##
## [1] "Normal Approx."
##
## [1] "10-year return level: 23.081"
##
## [1] "95% Confidence Interval: (22.9616, 23.2)"
```