Actividad 3

Luis Daniel Dimas Ramirez

16/12/2022

1

a. Basandome en las distrubuciones de probabilidad discutidas en clase, considero que la distribución **Gamma** es la que mejor se ajusta al comportamiento de mi variable de Magnitud. En el caso de las estaciones considero que la distribución **Poisson** puede ajustarse a mis datos.

Para tener como referencia mis distrubuciónes, las pondre a continuación:

```
x <- read.csv("x1.csv")
est <- x$st

tabla_est <- table(est)
tablar_est <- prop.table(tabla_est)

range(na.omit(est))
```

```
## [1]  10 130
```

```
hist_est <- hist(est,breaks=seq(0,140,20),plot=F)
n <- length(hist_est$breaks)
tab_est <- cbind(hist_est$breaks[-n],hist_est$breaks[-1],
                hist_est$counts,
                 hist_est$counts/sum(hist_est$counts),
                 cumsum(hist_est$counts),
                   cumsum(hist_est$counts/sum(hist_est$counts)))
dimnames(tab_est)[[2]]<-c("Linf","Lsup","f","fr","F","Fr")
print(tab_est)
```

```
##      Linf Lsup   f    fr    F    Fr
## [1,]    0   20 473 0.473  473 0.473
## [2,]   20   40 313 0.313  786 0.786
## [3,]   40   60 113 0.113  899 0.899
## [4,]   60   80  61 0.061  960 0.960
## [5,]   80  100  27 0.027  987 0.987
## [6,]  100  120  11 0.011  998 0.998
## [7,]  120  140   2 0.002 1000 1.000
```

```
mag <- x$mag
(tabla_mag <- table(mag))
```

```
## mag
##   4 4.1 4.2 4.3 4.4 4.5 4.6 4.7 4.8 4.9   5 5.1 5.2 5.3 5.4 5.5 5.6 5.7 5.9   6
##  46  55  90  85 101 107 101  98  65  54  47  43  29  21  20  14   9   8   2   3
## 6.1 6.4
##   1   1
```

```
(tablar_mag <- prop.table(tabla_mag))
```

```
## mag
##     4   4.1   4.2   4.3   4.4   4.5   4.6   4.7   4.8   4.9     5   5.1   5.2
## 0.046 0.055 0.090 0.085 0.101 0.107 0.101 0.098 0.065 0.054 0.047 0.043 0.029
##   5.3   5.4   5.5   5.6   5.7   5.9     6   6.1   6.4
## 0.021 0.020 0.014 0.009 0.008 0.002 0.003 0.001 0.001
```

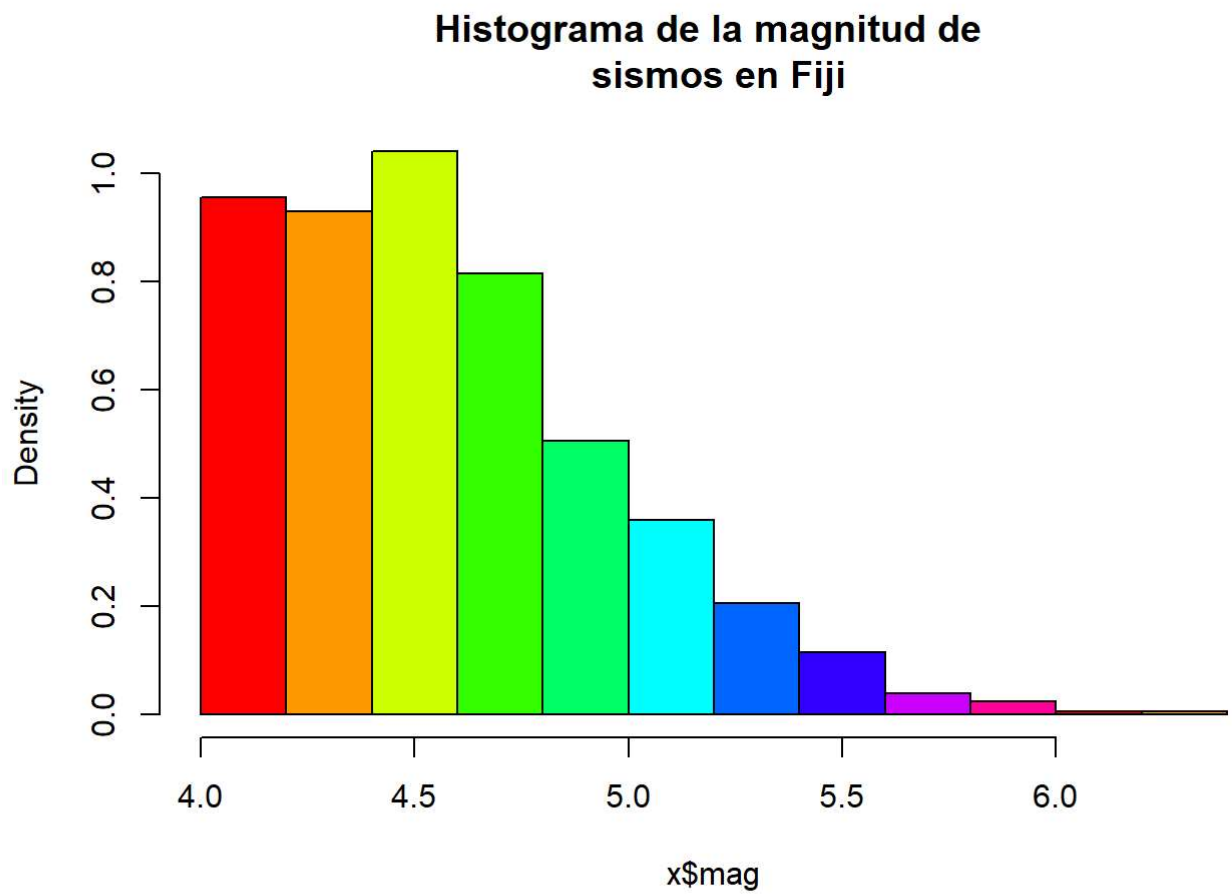
```
range(na.omit(mag))
```

```
## [1] 4.0 6.4
```

```
hist_mag <- hist(mag,breaks=seq(4,6.5,0.5),plot=F)
n <- length(hist_mag$breaks)
tab_mag <- cbind(hist_mag$breaks[-n],hist_mag$breaks[-1],
                hist_mag$counts,
                 hist_mag$counts/sum(hist_mag$counts),
                 cumsum(hist_mag$counts),
                 cumsum(hist_mag$counts/sum(hist_mag$counts)))
dimnames(tab_mag)[[2]]<-c("Linf","Lsup","f","fr","F","Fr")
print(tab_mag)
```

```
##      Linf Lsup   f    fr    F    Fr
## [1,]  4.0  4.5 484 0.484  484 0.484
## [2,]  4.5  5.0 365 0.365  849 0.849
## [3,]  5.0  5.5 127 0.127  976 0.976
## [4,]  5.5  6.0  22 0.022  998 0.998
## [5,]  6.0  6.5   2 0.002 1000 1.000
```

```
par(mfrow=c(1,1))
hist(x$mag, col=rainbow(10), main="Histograma de la magnitud de \n sismos en Fiji", prob=T)
```



b.

Variables	Distribución propuesta
Magnitud	Gamma
Estaciones	Poisson

2 Estimación puntual de los parametros

Discreta - Estaciones

Escribe una funcion que calcule la funcion de verosimilitud para la distribucion Poisson

Para este caso, debido a que trabajare con la distribución de Poisson, sabemos lo siguiente:

$$f(x;\lambda) = \frac{e^{-\lambda}\lambda^{x}}{x!}$$

Realizando el proceso para calcular la funcion $L(\theta)$ obtengo que

$$\lambda = \frac{\sum x_i}{n} = E(x) = \bar{x}$$

Por lo tanto:

```
lambda<- function(x,n){
  sum(x)/n
}

lambda(x$st, length(x$st))
```

```
## [1] 32.84
```

Estimando el parametro con el optmizador obtengo lo siguiente

```
library(fitdistrplus)
```

```
## Loading required package: MASS
```

```
## Loading required package: survival
```

```
(fitp <- fitdistr(x$st, densfun = "Poisson"))
```

```
##      lambda
##   32.8400000
##  ( 0.1812181)
```

Metodo	λ
$L(\theta)$	32.8
fitdistr()	32.84

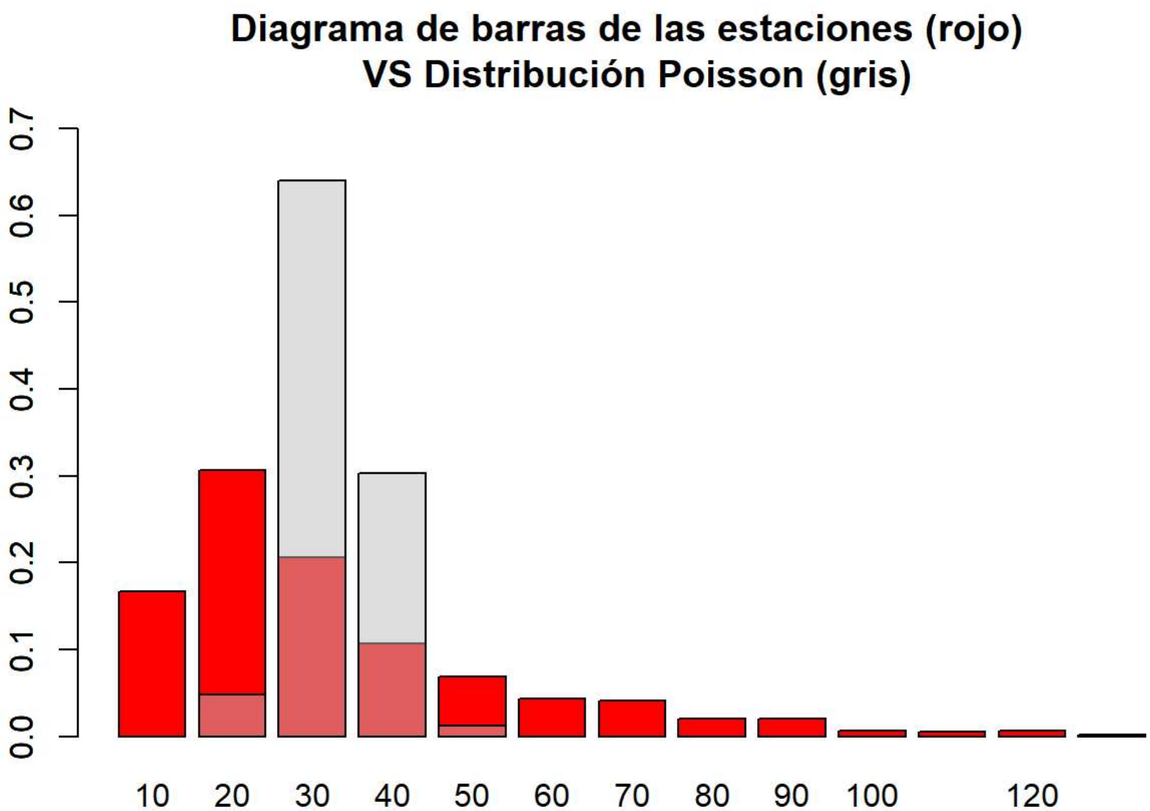
```
f <- 1:130
barplot(tablar_est, pch=20 ,main="Diagrama de barras de las estaciones (rojo) \n VS Distribución
Poisson (gris)", ylim = c(0,0.7), col = "red")

g<- seq(10,130, by=10)
modelo_estaciones<- dpois(g,lambda= fitp$estimate)
rbind(tablar_est,modelo_estaciones)
```

```
##                             10          20         30         40          50
## tablar_est        1.67000e-01 0.306000000 0.20600000 0.10700000 0.069000000
## modelo_estaciones 2.19805e-06 0.004783079 0.06400348 0.03035639 0.001188097
##                             60           70          80           90
## tablar_est        4.400000e-02 4.100000e-02 2.00000e-02 2.000000e-02
## modelo_estaciones 6.335535e-06 6.420732e-09 1.56781e-12 1.101836e-16
##                            100          110          120          130
## tablar_est        7.00000e-03 5.000000e-03 6.000000e-03 2.000000e-03
## modelo_estaciones 2.55906e-21 2.193804e-26 7.598943e-32 1.146794e-37
```

```
library(grDevices)  # paquete basico
micolor<- adjustcolor( "gray", alpha.f = 0.5)  # alpha.f ajusta lo transparente

barplot(modelo_estaciones*10,col=micolor,add=T)  # encimamos la graficas
```



Despues de observar esta grafica, y tratar de ajustarla con diferentes lambdas, la que más se acerca es cuando lambda es igual a 20 pero aun asi creo que no se ajusta bien. Me parece que tratare con otra distribución esperando que se ajuste mejor.

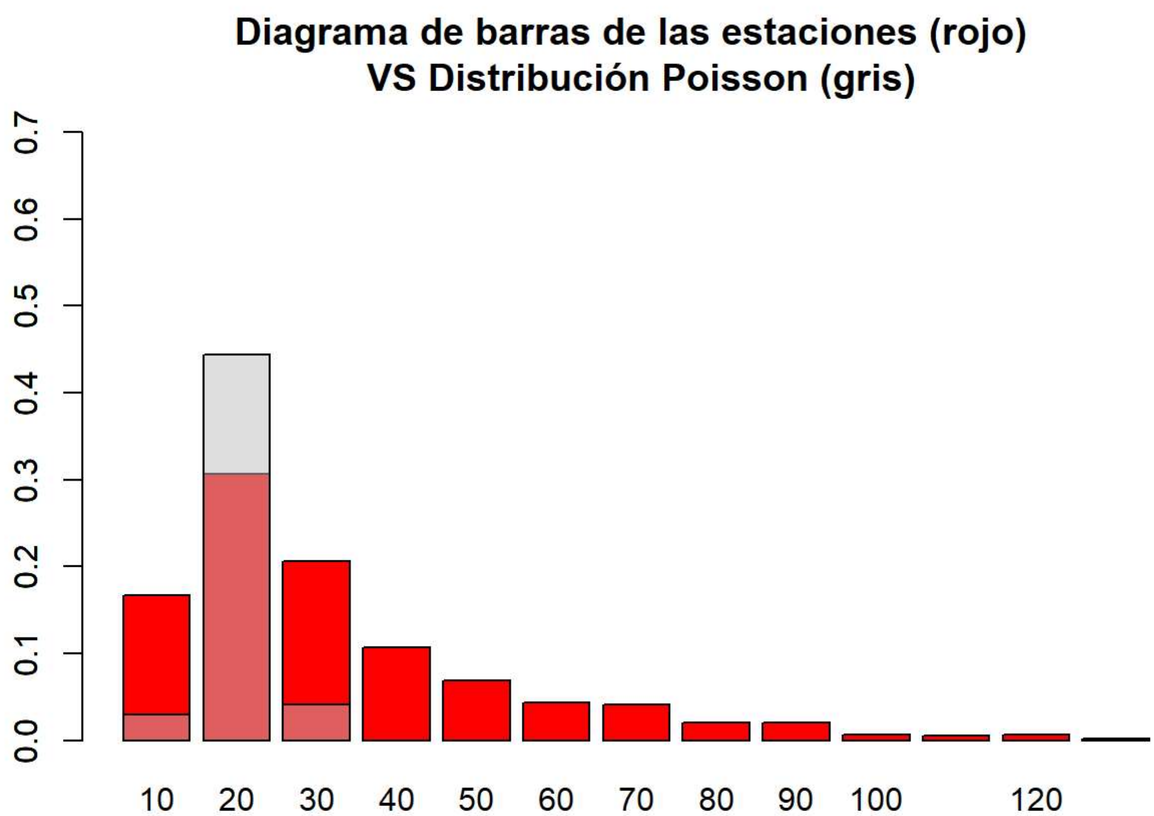
```
barplot(tablar_est, pch=20 ,main="Diagrama de barras de las estaciones (rojo) \n VS Distribución
Poisson (gris)", ylim = c(0,0.7), col = "red")

g<- seq(10,130, by=10)
modelo_estaciones<- dpois(g,lambda= 20)
rbind(tablar_est,modelo_estaciones)
```

```
##                           10         20          30           40           50
## tablar_est        0.167000000 0.30600000 0.206000000 1.070000e-01 6.900000e-02
## modelo_estaciones 0.005816307 0.08883532 0.008343536 2.777571e-05 7.630189e-09
##                            60           70           80           90
## tablar_est        4.400000e-02 4.100000e-02 2.000000e-02 2.000000e-02
## modelo_estaciones 2.855849e-13 2.031445e-18 3.481627e-24 1.717411e-30
##                           100          110          120          130
## tablar_est        7.000000e-03 5.000000e-03 6.000000e-03 2.000000e-03
## modelo_estaciones 2.799666e-37 1.684582e-44 4.095585e-52 4.338271e-60
```

```
library(grDevices)  # paquete basico
micolor<- adjustcolor( "gray", alpha.f = 0.5)  # alpha.f ajusta lo transparente

barplot(modelo_estaciones*5,col=micolor,add=T)  # encimamos la graficas
```



Mientras estaba probando mis datos simulados me di cuenta que son muy pequeños y que al multiplicarlos por 5 o por 10, se aproximan mejor

Probando con otras distribuciones discretas

```
(fitg <- fitdistr(x$st, densfun = "geometric"))
```

```
##        prob
##   0.0295508274
##  (0.0009205684)
```

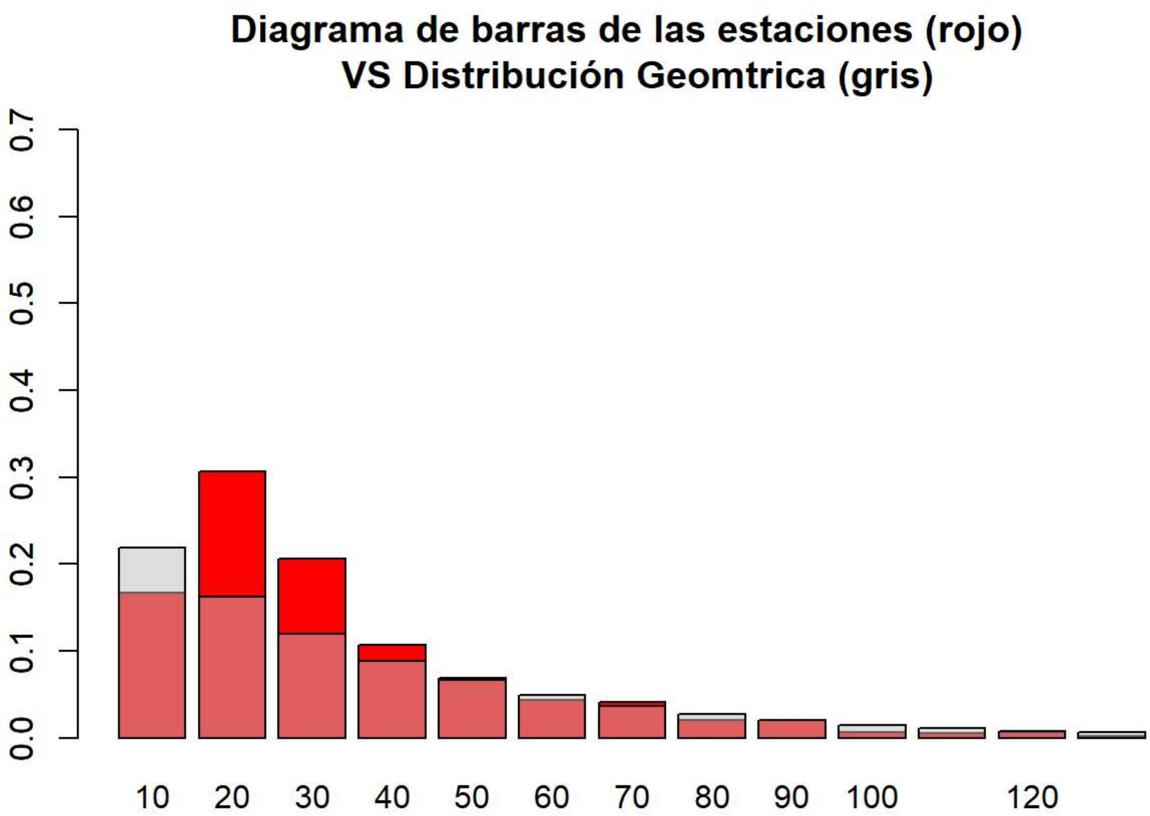
```
barplot(tablar_est, pch=20 ,main="Diagrama de barras de las estaciones (rojo) \n VS Distribución
Geomtrica (gris)", ylim = c(0,0.7), col = "red")

modelo_estaciones<- dgeom(g,fitg$estimate)
rbind(tablar_est,modelo_estaciones)
```

```
##                           10         20         30          40          50
## tablar_est        0.16700000 0.30600000 0.20600000 0.107000000 0.069000000
## modelo_estaciones 0.02189261 0.01621905 0.01201582 0.008901873 0.006594917
##                            60          70          80          90         100
## tablar_est        0.044000000 0.041000000 0.020000000 0.020000000 0.007000000
## modelo_estaciones 0.004885818 0.003619639 0.002681595 0.001986649 0.001471801
##                           110         120          130
## tablar_est        0.005000000 0.006000000 0.0020000000
## modelo_estaciones 0.001090378 0.000807802 0.0005984569
```

```
library(grDevices)  # paquete basico
micolor<- adjustcolor( "gray", alpha.f = 0.5)  # alpha.f ajusta lo transparente

barplot(modelo_estaciones*10,col=micolor,add=T)  # encimamos la grafica
```



Considero levemente mejor ajustada la distribución geometrica que la poisson. Así que me quedare con esta teniendo en cuenta que es $p * 10$

El parametro para la geometrica es:

$$p = 0.2955083$$

Continua

Considero que una distribución gamma puede aproximarse a mi variable de magnitud

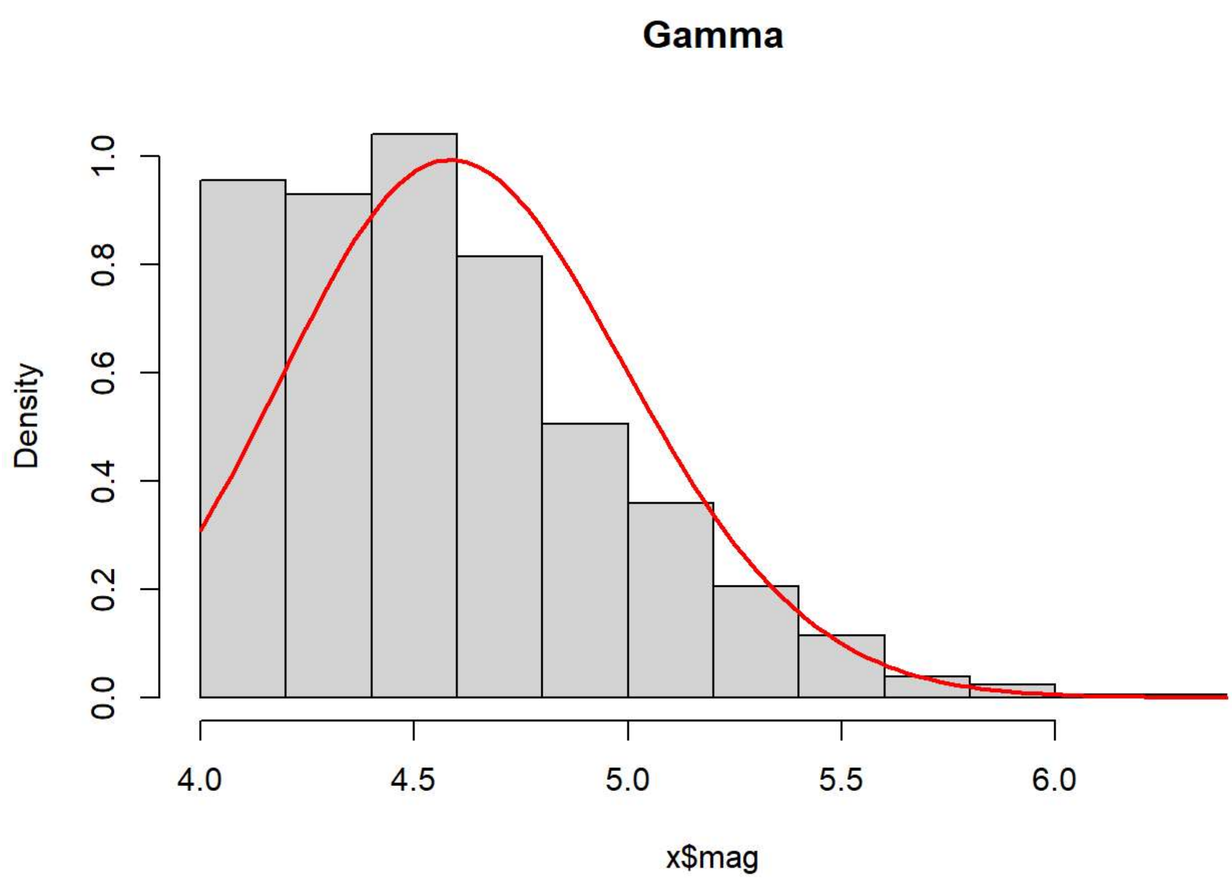
```
(fit<- fitdistr(x$mag, densfun = "gamma"))
```

```
##      shape         rate   
##   131.594743    28.481323 
##  (  5.877149) (  1.274423)
```

```
ee_MLE <- sqrt(fit$vcov); ee_MLE
```

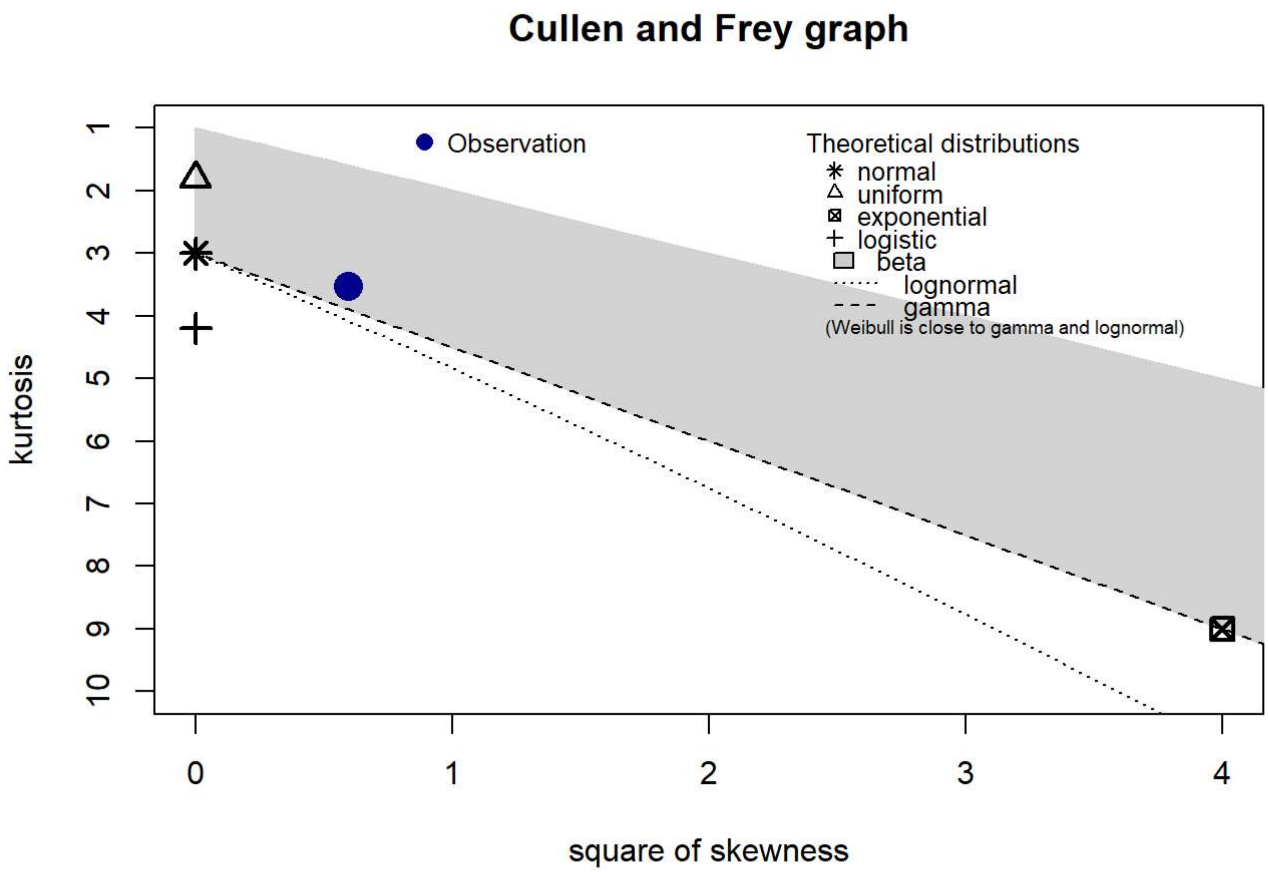
```
##          shape     rate
## shape 5.877149 2.734182
## rate  2.734182 1.274423
```

```
hist(x$mag, pch=20, prob=TRUE, main="Gamma")
curve(dgamma(x, fit$estimate[1], fit$estimate[2]), col="red", lwd=2, add=T)
```



Grafica Cullen-Frey

```
descdist(x$mag)
```




```
## summary statistics
## ------
## min:  4   max:  6.4
## median:  4.6
## mean:  4.6204
## estimated sd:  0.402773
## estimated skewness:  0.7697548
## estimated kurtosis:  3.518884
```

De la grafica cullen-frey observo que la distribución gamma es una buena opcion pero tambien podria probar con la distribucion weibull e incluso la lognormal.

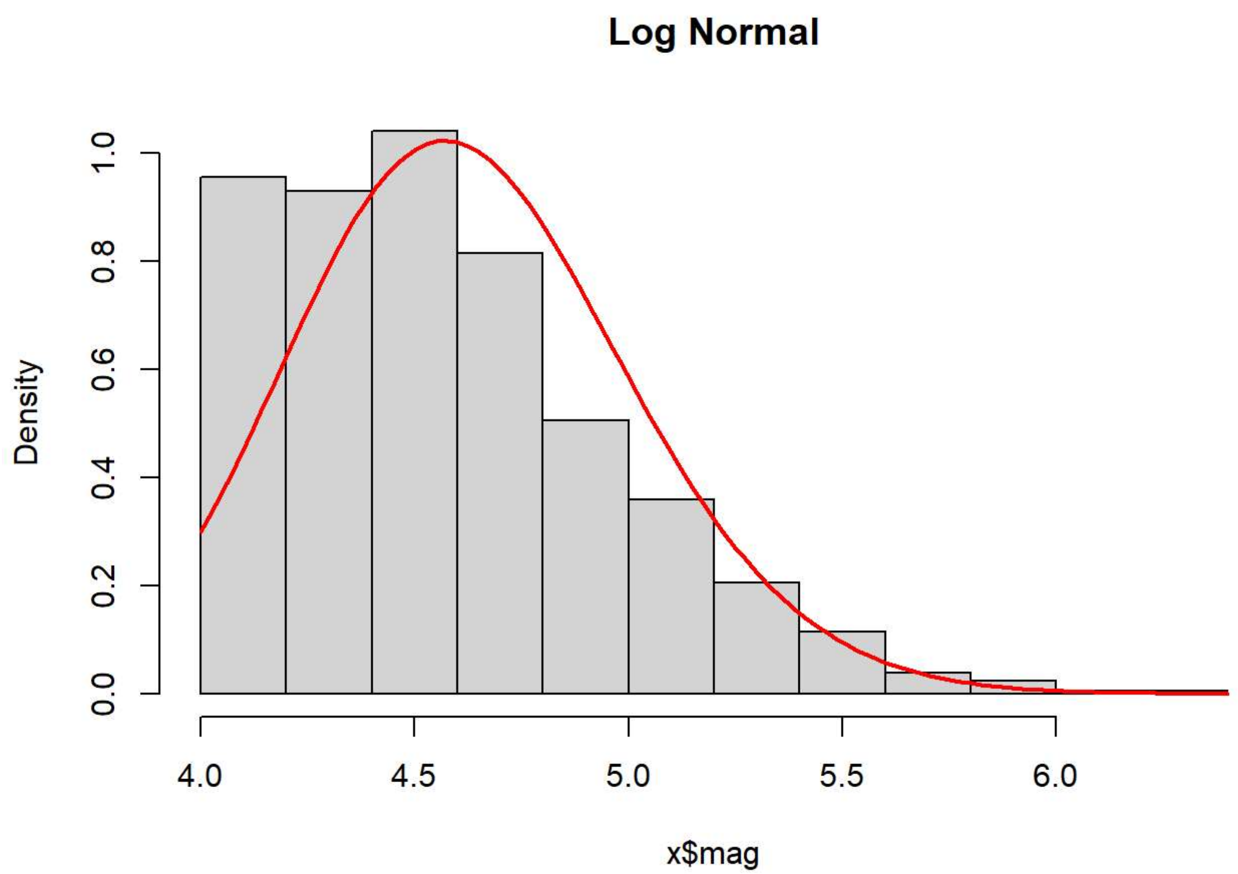
```
(fit<- fitdistr(x$mag, den = "lognormal"))
```

```
##      meanlog        sdlog
##   1.526810316   0.085035540
##  (0.002689060) (0.001901452)
```

```
ee_MLE <- sqrt(fit$vcov); ee_MLE
```

```
##             meanlog       sdlog
## meanlog 0.00268906 0.000000000
## sdlog   0.00000000 0.001901452
```

```
hist(x$mag, pch=20, prob=TRUE, main="Log Normal")
curve(dlnorm(x, fit$estimate[1], fit$estimate[2]), col="red", lwd=2, add=T)
```



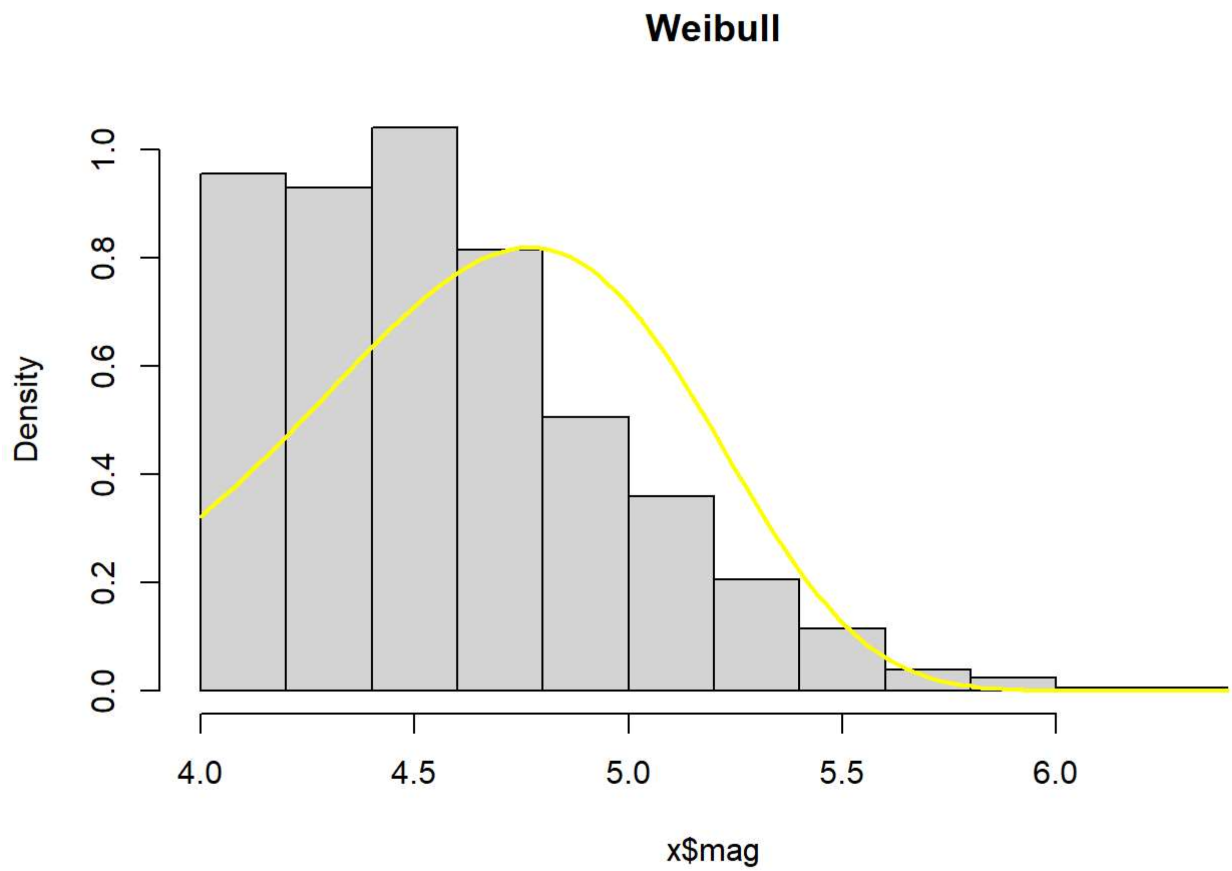

```
(fitw<- fitdistr(x$mag, den = "weibull"))
```

```
##       shape         scale
##   10.67255750    4.81257866
##  ( 0.23156634) ( 0.01515957)
```

```
ee_MLEw <- sqrt(fit$vcov); ee_MLEw
```

```
##             meanlog       sdlog
## meanlog 0.00268906 0.000000000
## sdlog   0.00000000 0.001901452
```

```
hist(x$mag, pch=20, prob=TRUE, main="Weibull")
curve(dweibull(x, fitw$estimate[1], fitw$estimate[2]), col="yellow", lwd=2, add=T)
```



Para el caso de la magnitud concluyo que la funcion LogNormal se acerca ligeramente más a mis datos que la Gamma.

```
(fitg<- fitdistr(x$mag, densfun = "gamma"))
```

```
##      shape         rate
##   131.594743    28.481323
##  (  5.877149) (  1.274423)
```

```
ee_MLEg <- sqrt(fit$vcov); ee_MLEg
```

```
##            meanlog       sdlog
## meanlog 0.00268906 0.000000000
## sdlog   0.00000000 0.001901452
```

```
(fitl<- fitdistr(x$mag, den = "lognormal"))
```

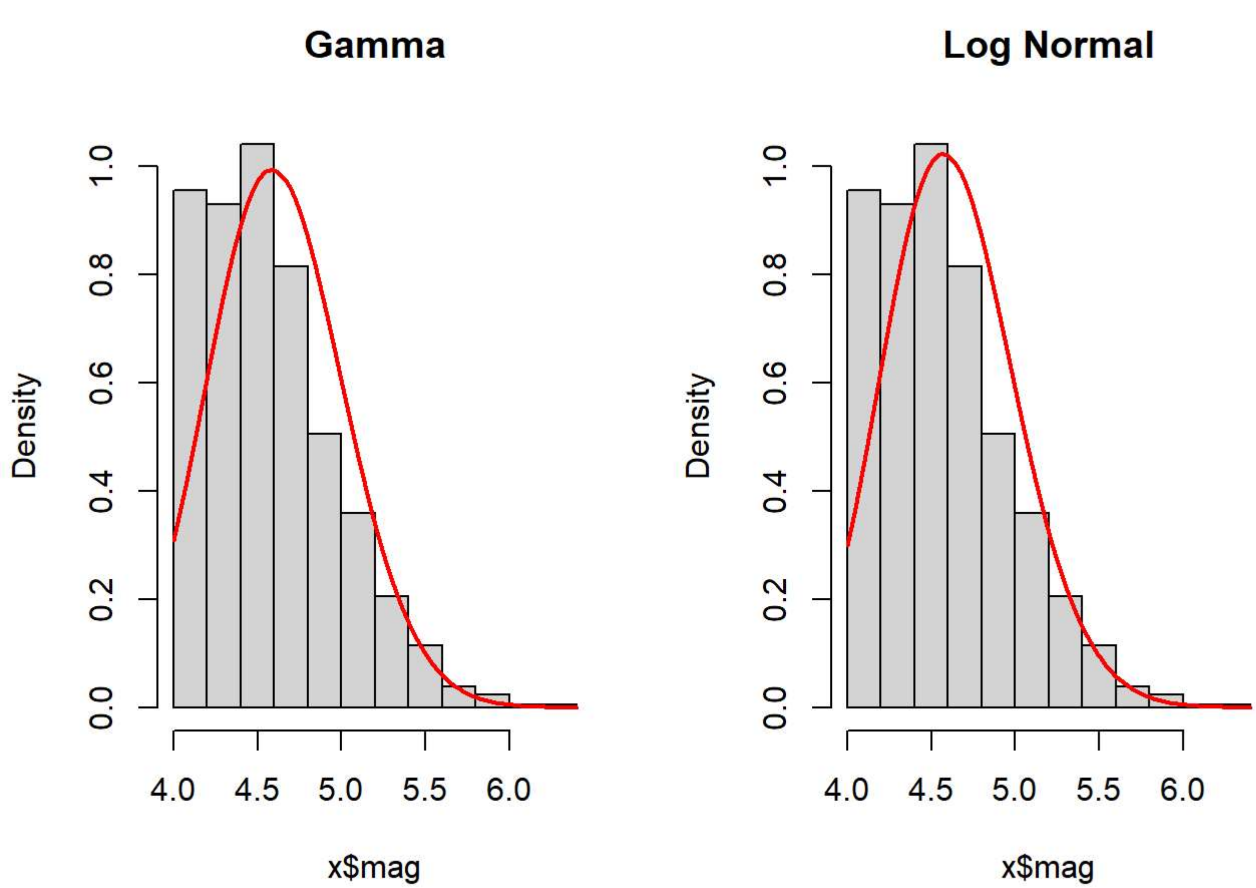
```
##      meanlog        sdlog
##   1.526810316   0.085035540
##  (0.002689060) (0.001901452)
```

```
ee_MLEl <- sqrt(fit$vcov); ee_MLEl
```

```
##            meanlog       sdlog
## meanlog 0.00268906 0.000000000
## sdlog   0.00000000 0.001901452
```

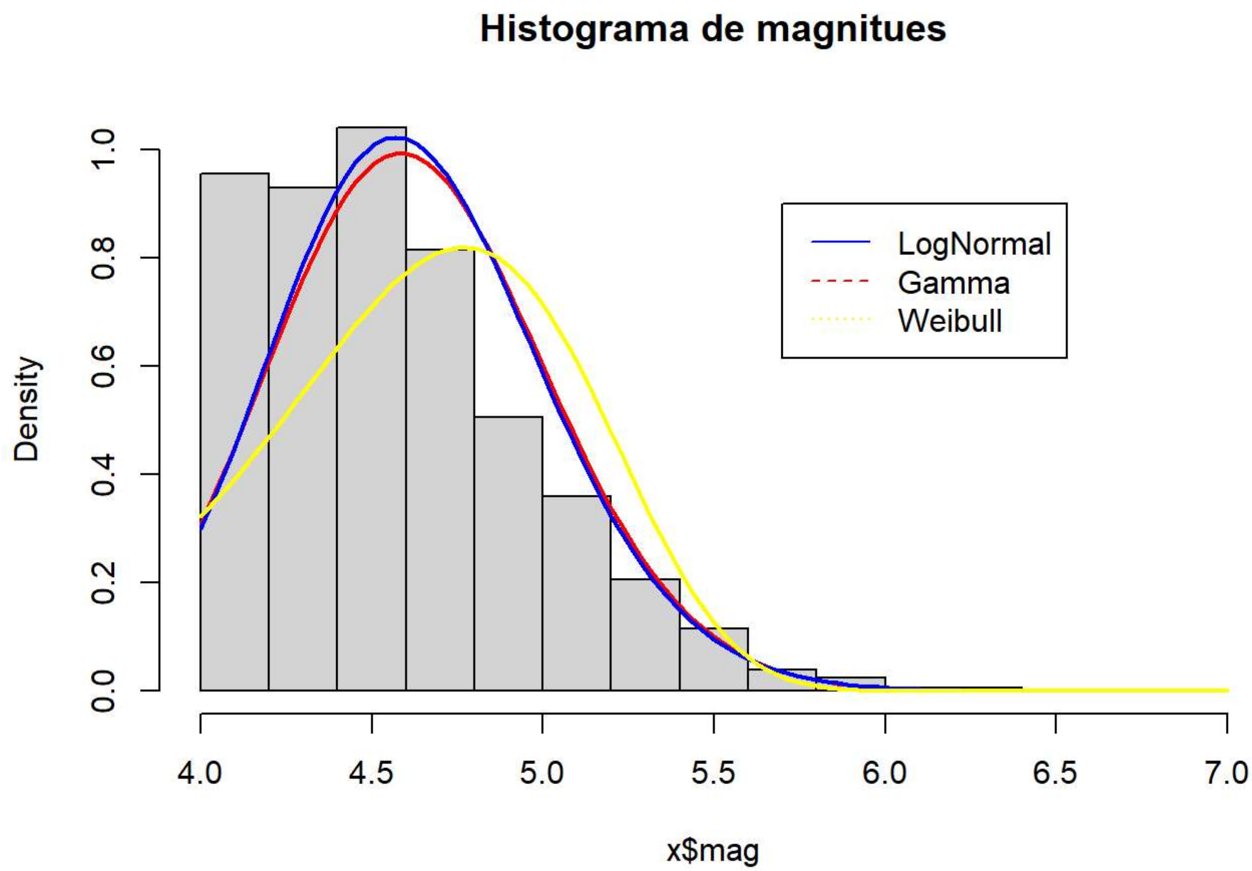
```
par(mfrow=c(1,2))
hist(x$mag, pch=20, prob=TRUE, main="Gamma")
curve(dgamma(x, fitg$estimate[1], fitg$estimate[2]), col="red", lwd=2, add=T)

hist(x$mag, pch=20, prob=TRUE, main="Log Normal")
curve(dlnorm(x, fitl$estimate[1], fitl$estimate[2]), col="red", lwd=2, add=T)
```



Comparandolas en una sola gráfica:

```
hist(x$mag, pch=20, prob=TRUE, main="Histograma de magnitues", xlim=c(4,7))
curve(dgamma(x, fitg$estimate[1], fitg$estimate[2]), col="red", lwd=2, add=T)
curve(dlnorm(x, fitl$estimate[1], fitl$estimate[2]), col="blue", lwd=2, add=T)
curve(dweibull(x, fitw$estimate[1], fitw$estimate[2]), col="yellow", lwd=2, add=T)
legend(x=5.7, y=0.9, legend = c("LogNormal", "Gamma", "Weibull"), col=c("blue", "red", "yello
w"), lty = 1:3)
```



La logrnormal es la que mejor se ajusta a mis datos.