

# Guia TP MLE

Agustin Muñoz González

26/6/2020

## Preparamos el entorno

```
rm(list=ls())
library(ggplot2)
library(tidyr)
library(gganimate)
datos=read.table("GRB_afterglow.dat.csv",header=T, skip=1)
flux=datos$f
```

## Estimación bajo modelo Gamma $\Gamma(\alpha, \lambda)$ :

Distribución Gamma de parámetros  $\alpha > 0, \lambda > 0, X_i \sim \Gamma(\alpha, \lambda)$ .

$$f(x) = \frac{\lambda^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\lambda x}.$$

Los estimadores de los parámetros  $\alpha$  y  $\lambda$  que son fáciles de computar son los obtenidos por el Método de Momentos y corresponden a:

$$\begin{aligned}\tilde{\alpha} &= \frac{n\bar{X}_n^{-2}}{\sum_i (X_i - \bar{X}_n)^2} \\ \tilde{\lambda} &= \frac{n\bar{X}_n}{\sum_i (X_i - \bar{X}_n)^2}.\end{aligned}$$

(esto sale de que

$$\begin{aligned}\frac{\tilde{\alpha}}{\tilde{\lambda}} &= E_{\alpha,\lambda}(X_1) = \frac{1}{n} \sum_i X_i = \bar{X}_n \\ Var_{\alpha,\lambda}(X_1) &= \frac{\alpha}{\lambda^2} \\ Var_{\alpha,\lambda}(X_1) &= E_{\alpha,\lambda}(X_1^2) - E_{\alpha,\lambda}(X_1)^2 \\ \frac{\tilde{\alpha}}{\tilde{\lambda}^2} - \frac{\tilde{\alpha}^2}{\tilde{\lambda}^2} &= E_{\alpha,\lambda}(X_1^2) = \frac{1}{n} \sum_i X_i^2\end{aligned}$$

y resolviendo el sistema obtengo las estimaciones de arriba)

Por otro lado, la verosimilitud, cuando  $x_i \geq 0 \forall i$ , resulta

$$\begin{aligned}
L(\alpha, \lambda; x) &= \prod_i f(x_i, \alpha, \lambda) \\
&= \prod_i \frac{\lambda^\alpha}{\Gamma(\alpha)} x_i^{\alpha-1} e^{-\lambda x_i} \\
&= \frac{\lambda^{n\alpha}}{\Gamma(\alpha)^n} e^{-\lambda \sum_i x_i} \left\{ \prod_i x_i \right\}^{\alpha-1}.
\end{aligned}$$

Por lo tanto, se obtiene que la log-verosimilitud es:

$$l(\alpha, \lambda; x) = n\alpha \log(\lambda) - n \log(\Gamma(\alpha)) + (\alpha - 1) \sum_i \log(x_i) - \lambda \sum_i x_i.$$

En este caso, no hay fórmula cerrada para obtener los valores que maximizan esta expresión y se utilizan métodos numéricos para la aproximación de los mismos.

1. Implementar una función MME.gamma que tenga por argumento un conjunto de datos  $x = (x_1, \dots, x_n)$  y devuelva los estimadores de momentos de  $\alpha$  y  $\lambda$ .

Resolución:

Definimos la función del estimador de momentos.

```
MME.gamma=function(datos){
  n=length(datos)
  promedio=mean(datos)
  aux=c()
  for(i in datos){
    aux=c(aux, (i-promedio)^2)
  }
  alpha=n*promedio^2/sum(aux)
  lambda=n*promedio/sum(aux)
  c(alpha,lambda)
}
```

2. Suponiendo que los datos de flux provienen de una población con distribución Gamma, estimar  $\alpha$  y  $\lambda$  usando la función MME.gamma.

Resolución:

Aproximamos los parámetros alpha y lambda de los datos usando la función anterior.

```
alpha_MME=MME.gamma(flux)[1]
alpha_MME
```

```
## [1] 1.387366
```

```
lambda_MME=MME.gamma(flux)[2]
lambda_MME
```

```
## [1] 0.04028012
```

3. Estimar  $P(X < 40)$  usando las estimaciones obtenidas y el método plug-in, es decir reemplazando cada parámetro por su valor estimado. Comparar con las estimaciones obtenidas en la Guía de clase.

Resolución:

Usamos el comando `pgamma(p, shape=alfa, rate=lambda)` o `pgamma(p, shape=alfa, scale=1/lambda)`.

```
pgamma(40, shape=alpha_MME,rate=lambda_MME)
```

```
## [1] 0.6785147
```

4. Graficar la empírica asociada a los datos flux y superponer la función de distribución acumulada Gamma con los parámetros que considere pertinentes. Comparar con los gráficos obtenidos en la Guía de clase.

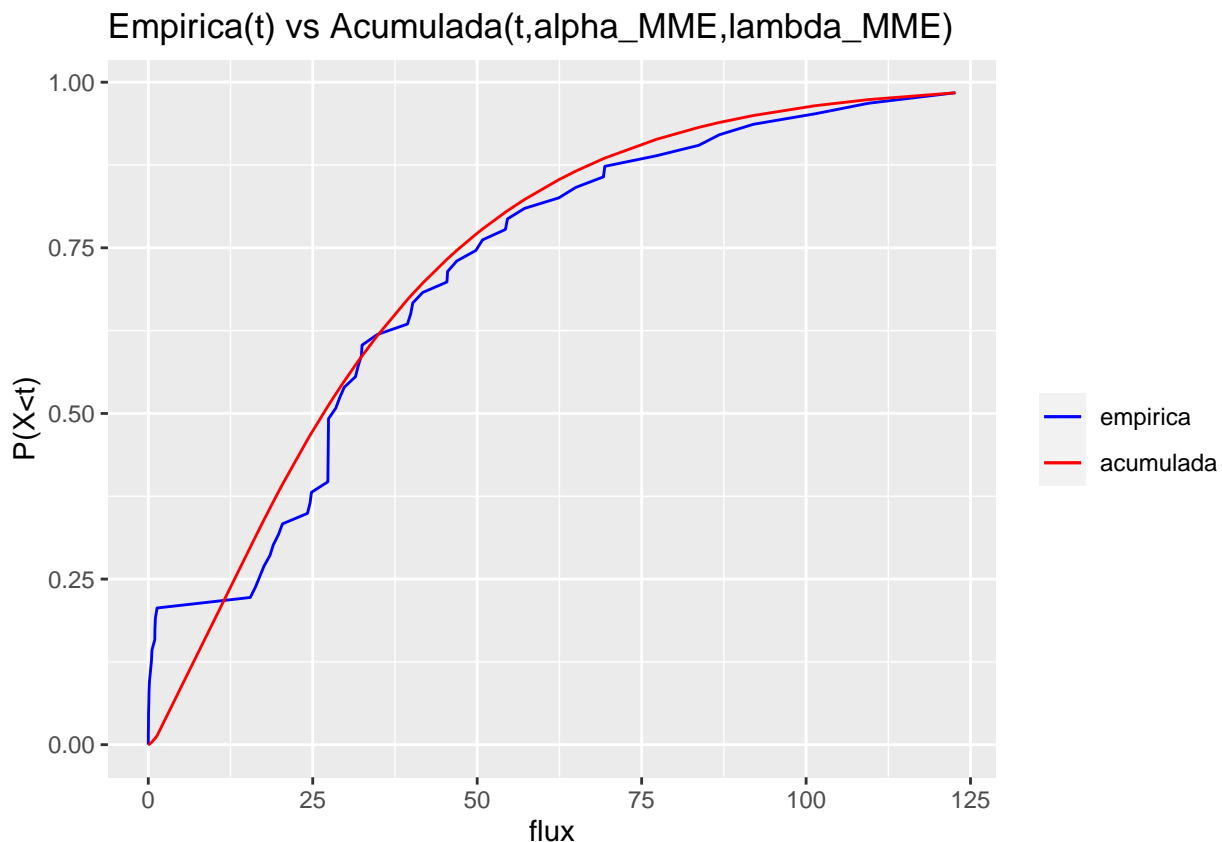
Resolución:

Definimos en primer lugar la empírica.

```
empirica=function(t){  
  mean(flux<t)  
}
```

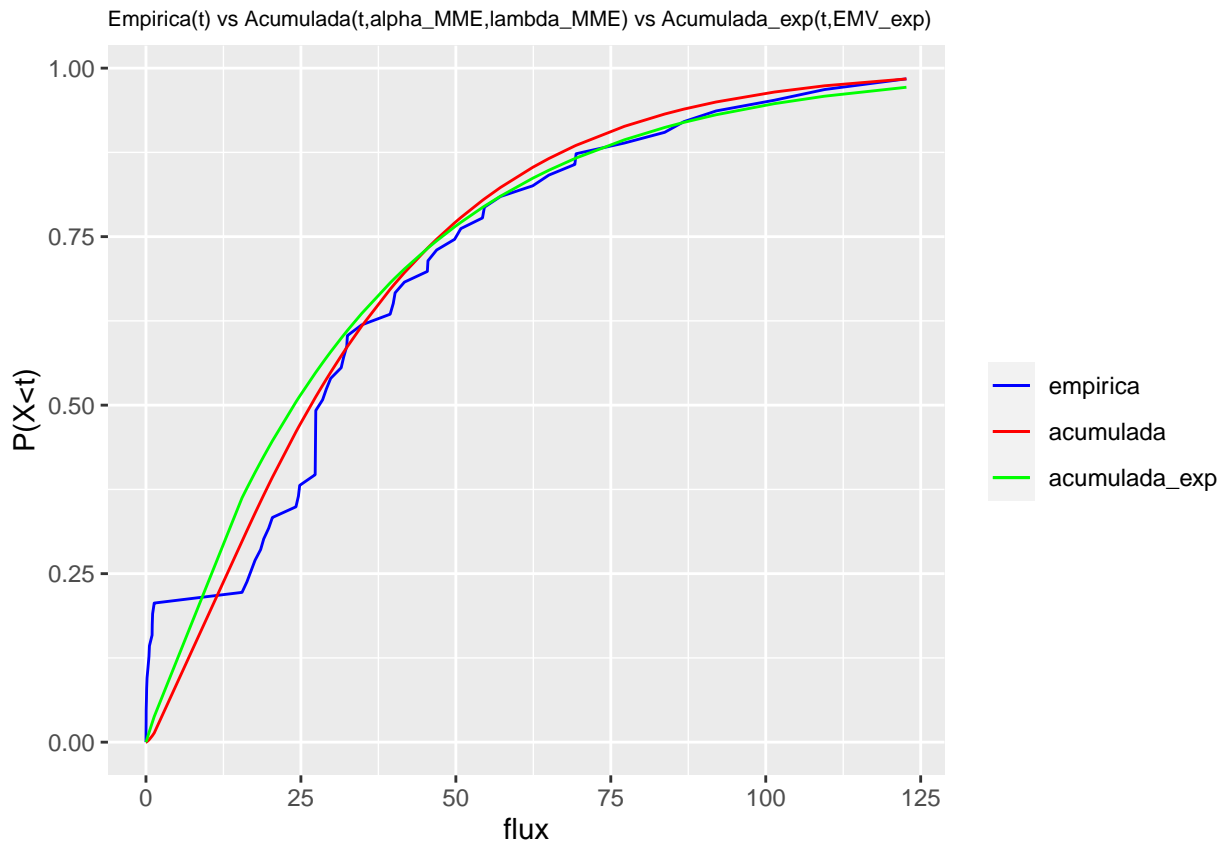
Ploteamos.

```
datos_plot=data.frame(cbind('flux'=flux,  
'empirica'=sapply(flux,empirica),  
'acumulada'=sapply(flux,pgamma,shape=alpha_MME,rate=lambda_MME)))  
ggplot(datos_plot)+  
  geom_line(aes(x=flux,y=empirica,color='empirica'))+  
  geom_line(aes(x=flux,y=acumulada,color='acumulada'))+  
  scale_colour_manual("",  
    breaks = c("empirica", "acumulada"),  
    values = c("blue", "red")) +  
  xlab("flux") +  
  scale_y_continuous("P(X<t)") +  
  labs(title="Empirica(t) vs Acumulada(t,alpha_MME,lambda_MME)")
```



Ahora graficamos lo mismo junto con la acumulada de la exponencial de la clase pasada para comparar todo. Para eso definimos la acumulada y la funcion EMV de la guia de la clase.

```
F_exp=function(t,lambda){
  1-exp(-lambda*t)
}
EMV_exp=function(datos){
  1/mean(datos)
}
#####
datos_plot=data.frame(cbind('flux'=flux,
'empirica'=sapply(flux,empirica),
'acumulada'=sapply(flux,pgamma,shape=alpha_MME,rate=lambda_MME),
'acumulada_exp'=sapply(flux,F_exp,lambda=EMV_exp(flux))))
ggplot(datos_plot)+
  geom_line(aes(x=flux,y=empirica,color='empirica'))+
  geom_line(aes(x=flux,y=acumulada,color='acumulada'))+
  geom_line(aes(x=flux,y=acumulada_exp,color='acumulada_exp'))+
  scale_colour_manual("",
    breaks = c("empirica", "acumulada", "acumulada_exp"),
    values = c("blue", "red", "green")) +
  xlab("flux") +
  scale_y_continuous("P(X<t)" ) +
  labs(title="Empirica(t) vs Acumulada(t,alpha_MME,lambda_MME) vs Acumulada_exp(t,EMV_exp)") +
  theme(plot.title = element_text(size=8))
```

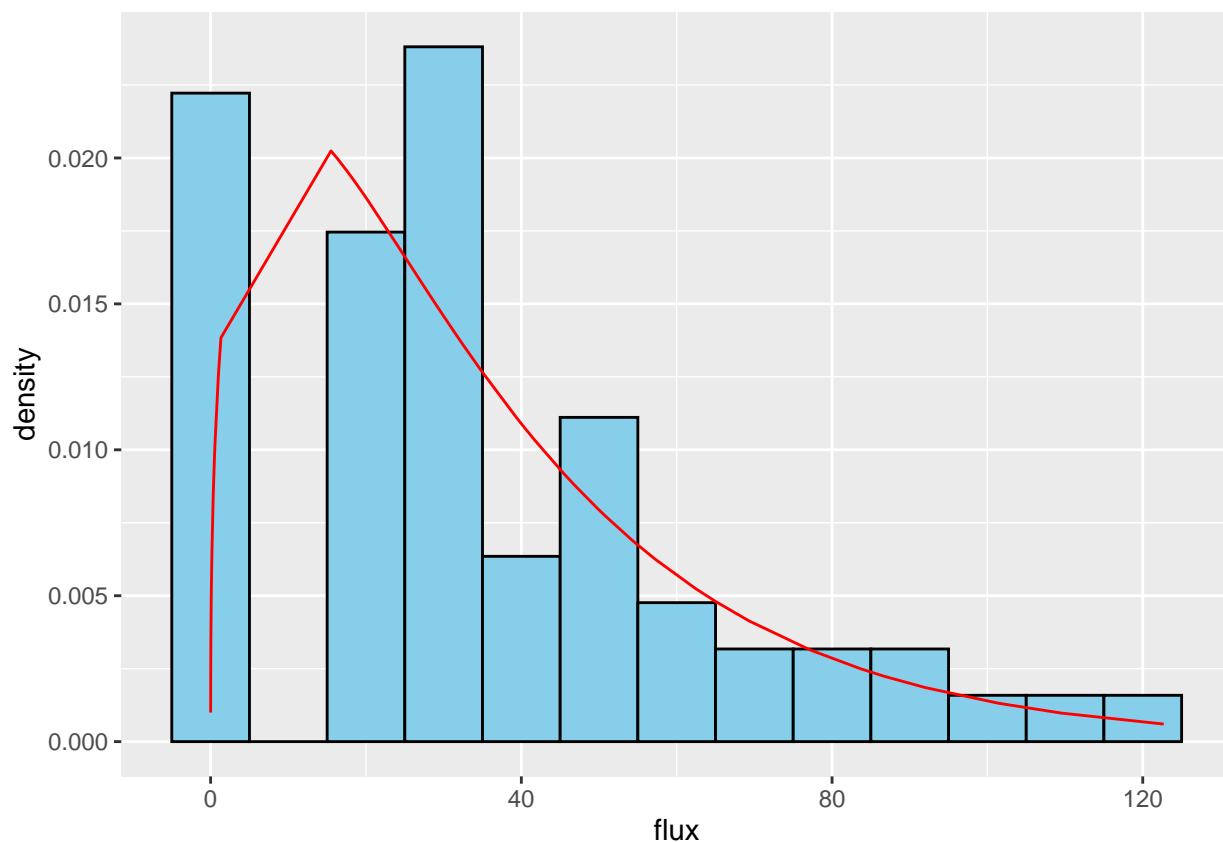


```
# Esto ultimo es para variar el tamaño del título.
```

5. Realizar un histograma para los datos de flux y superponer la función de densidad gamma con los parámetros que considere pertinentes.

Resolución:

```
datos_plot=data.frame(cbind(datos_plot,'densidad'=sapply(flux,dgamma,shape=alpha_MME,rate=lambda_MME)))
ggplot(datos_plot)+
  geom_histogram(aes(x=flux,y=..density..),
    binwidth=10,
    # tengo que ver que binwidth pasarle para que la
    # densidad quede linda.
    # Rdo: El area de cada barra es 1.
    fill="skyblue",color="black")+
  geom_line(aes(x=flux,y=densidad),color='red')
```



6. Bajo el mismo supuesto, calcular los estimadores de máxima verosimilitud de  $\alpha$  y  $\lambda$ . Explorar el comando `fitdistr` de la librería MASS. Comparar con los valores obtenidos en ítems anteriores. Repetir los tres ítems anteriores y comparar los resultados obtenidos.

Resolución:

```
library(MASS)
alpha_EMV=fitdistr(flux, 'gamma')$estimate[1]
```

```
## Warning in densfun(x, parm[1], parm[2], ...): NaNs produced
```

```
## Warning in densfun(x, parm[1], parm[2], ...): NaNs produced
```

```
## Warning in densfun(x, parm[1], parm[2], ...): NaNs produced
## Warning in densfun(x, parm[1], parm[2], ...): NaNs produced
## Warning in densfun(x, parm[1], parm[2], ...): NaNs produced
## Warning in densfun(x, parm[1], parm[2], ...): NaNs produced
## Warning in densfun(x, parm[1], parm[2], ...): NaNs produced
## Warning in densfun(x, parm[1], parm[2], ...): NaNs produced
## Warning in densfun(x, parm[1], parm[2], ...): NaNs produced
## Warning in densfun(x, parm[1], parm[2], ...): NaNs produced
alpha_EMV
```

```
##      shape
## 0.549351
```

```
alpha_MME
```

```
## [1] 1.387366
```

```
lambda_EMV=fitdistr(flux, 'gamma')$estimate[2]
```

```
## Warning in densfun(x, parm[1], parm[2], ...): NaNs produced
## Warning in densfun(x, parm[1], parm[2], ...): NaNs produced
## Warning in densfun(x, parm[1], parm[2], ...): NaNs produced
## Warning in densfun(x, parm[1], parm[2], ...): NaNs produced
## Warning in densfun(x, parm[1], parm[2], ...): NaNs produced
## Warning in densfun(x, parm[1], parm[2], ...): NaNs produced
## Warning in densfun(x, parm[1], parm[2], ...): NaNs produced
## Warning in densfun(x, parm[1], parm[2], ...): NaNs produced
## Warning in densfun(x, parm[1], parm[2], ...): NaNs produced
## Warning in densfun(x, parm[1], parm[2], ...): NaNs produced
lambda_EMV
```

```
##      rate
## 0.0159496
```

```
lambda_MME
```

```
## [1] 0.04028012
```

TIENE SENTIDO QUE NO DEN LO MISMO AMBOS METODOS, PORQUE DEPENDEN X EJ DE

DONDE ARRANCAN Y SE QUEDAN CON EL VALOR MAS CERCANO AL LUGAR DONDE ARRANCAN (fitdistr HACE ALGO ASI), ENTONCES CAPAZ UNO TENIA CIERTO VALOR Y EL OTRO TENIA CERCA OTRO VALOR Y POR ESO DAN DISTINTO.

**GENERAR DATOS GAMMA Y ESTIMAR CON AMBOS ESTIMADORES Y CALCULAR LOS ERRORES CUADRATICOS. VER TAMBIEN CUANDO DA MAS GRANDE EMV Y CUANDO MME (X EJ GENERANDO UNA LISTA DE BOOLEANOS DONDE EN CADA LUGAR TENEMOS LA COMPARACION  $EMV < MME$ ).**

Repetimos los items anteriores con ambas estimaciones.

3.

```
pgamma(40, shape=alpha_MME,rate=lambda_MME)
```

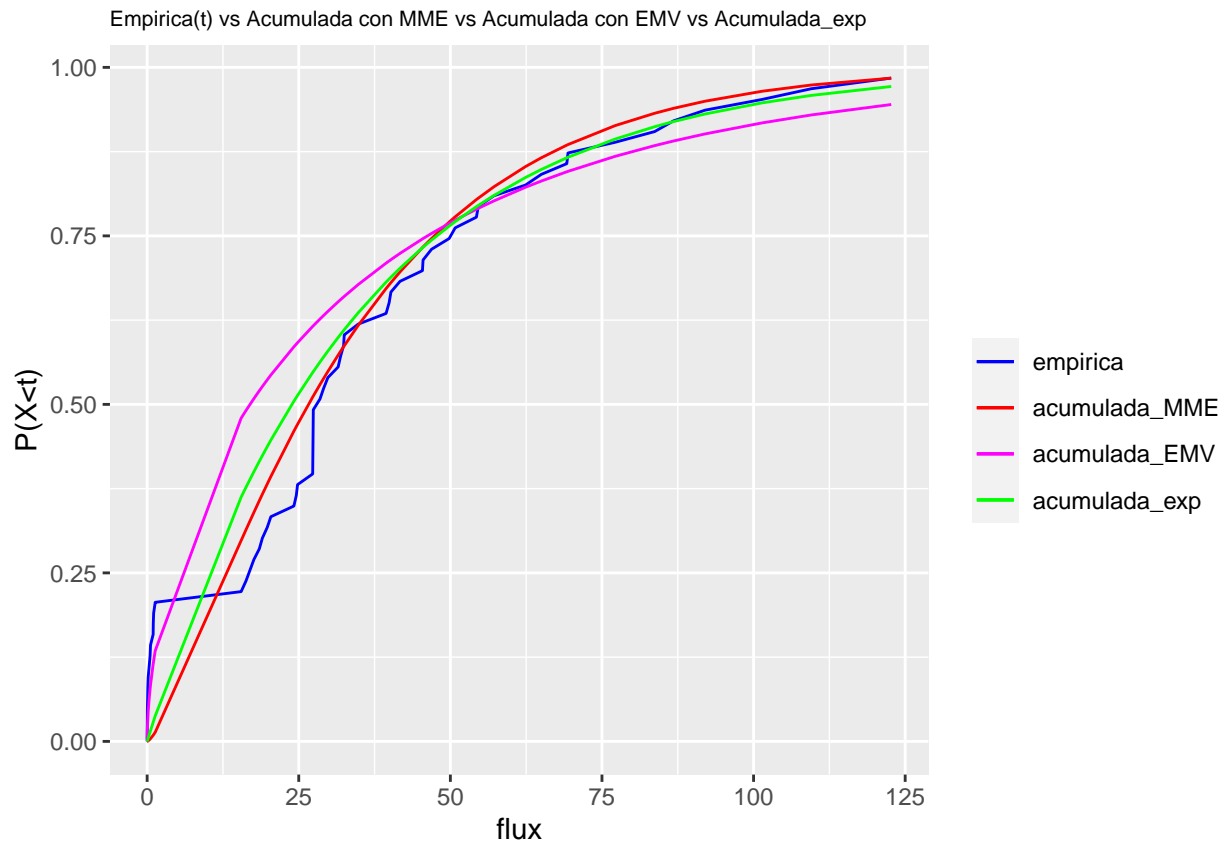
```
## [1] 0.6785147
```

```
pgamma(40, shape=alpha_EMV,rate=lambda_EMV)
```

```
## [1] 0.7134332
```

4.

```
datos_plot=data.frame(cbind('flux'=flux,
'empirica'=sapply(flux,empirica),
'acumulada_MME'=sapply(flux,pgamma,shape=alpha_MME,rate=lambda_MME),
'acumulada_EMV'=sapply(flux,pgamma,shape=alpha_EMV,rate=lambda_EMV),
'acumulada_exp'=sapply(flux,F_exp,lambda=EMV_exp(flux))))
ggplot(datos_plot)+
  geom_line(aes(x=flux,y=empirica,color='empirica'))+
  geom_line(aes(x=flux,y=acumulada_MME,color='acumulada_MME'))+
  geom_line(aes(x=flux,y=acumulada_EMV,color='acumulada_EMV'))+
  geom_line(aes(x=flux,y=acumulada_exp,color='acumulada_exp'))+
  scale_colour_manual("",
                      breaks = c("empirica", "acumulada_MME", "acumulada_EMV", "acumulada_exp"),
                      values = c("blue", "red", "magenta", "green")) +
  xlab("flux") +
  scale_y_continuous("P(X<t)") +
  labs(title="Empirica(t) vs Acumulada con MME vs Acumulada con EMV vs Acumulada_exp")+
  theme(plot.title = element_text(size=8))
```

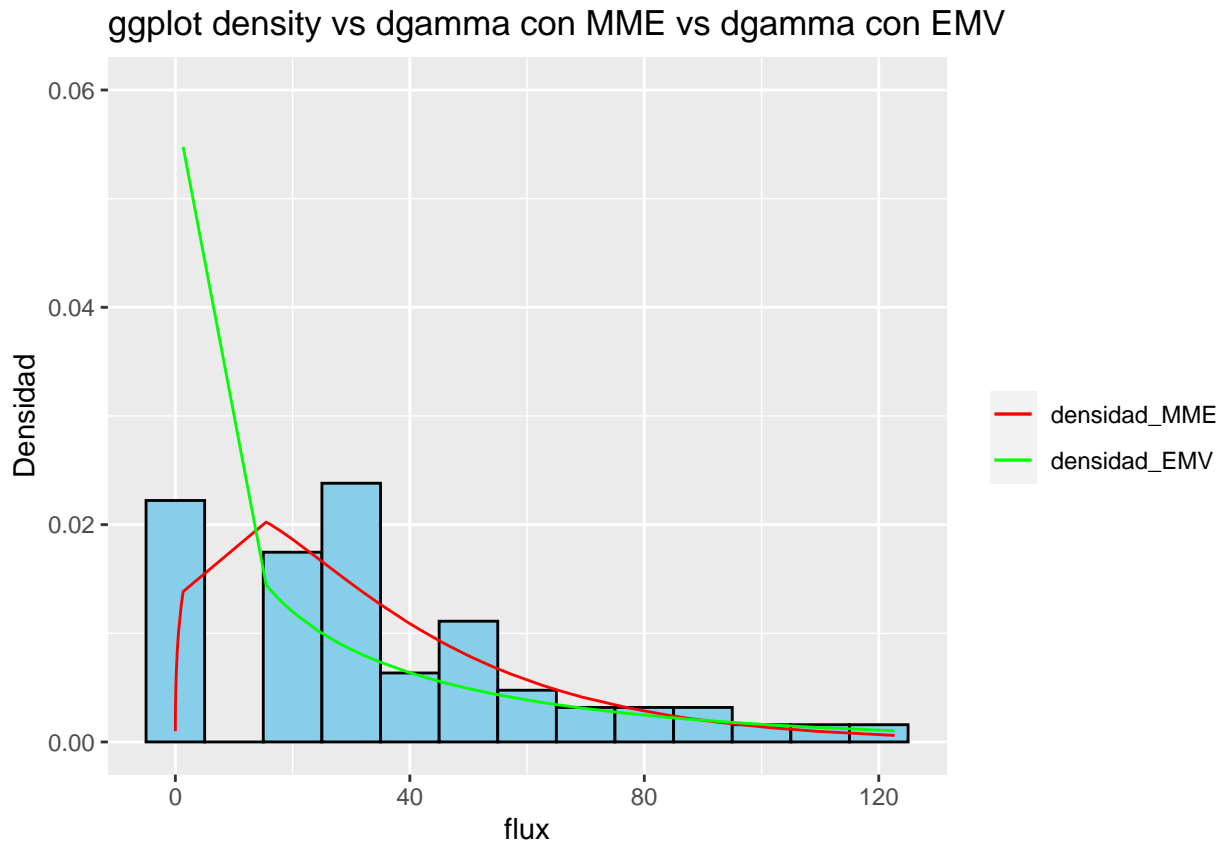


5.

```
datos_plot=data.frame(cbind(datos_plot,
                             'densidad_MME'=sapply(flux,dgamma,shape=alpha_MME,rate=lambda_MME),
                             'densidad_EMV'=sapply(flux,dgamma,shape=alpha_EMV,rate=lambda_EMV)))
ggplot(datos_plot)+
  geom_histogram(aes(x=flux,y=..density..),
                 binwidth=10,
                 # tengo que ver que binwidth pasarle para que la
                 # densidad quede linda.
                 # Rdo: El area de cada barra es 1.
                 fill="skyblue",color="black")+
  geom_line(aes(x=flux,y=densidad_MME,color='densidad_MME'))+
  geom_line(aes(x=flux,y=densidad_EMV,color='densidad_EMV'))+
  scale_colour_manual("",
                      breaks = c("ggplot density", "densidad_MME", "densidad_EMV"),
                      values = c("blue", "red", "green")) +
  xlab("flux") +
  scale_y_continuous("Densidad",limits = #c(0,min(max(sapply(flux,dgamma,shape=alpha_EMV,rate=lambda_EMV,
                                                             c(0,0.06))) +
  labs(title="ggplot density vs dgamma con MME vs dgamma con EMV")
```

```
## Warning: Removed 13 row(s) containing missing values (geom_path).
```





7. Estimar por el método de Máxima Verosimilitud el flux medio a partir de los datos. ¿Con qué valor podría comparar esta estimación?

Resolución:

Sabiendo que la esperanza de la gamma es  $E(X) = \frac{\alpha}{\lambda}$  podemos estimar con el metodo de plug-in en esa formula.

Podemos compararlo con el flux medio calculado usando el metodo de los momentos y con el `flux_medio=mean(flux)` calculado en la práctica de la clase. (digamos hay infinitos etimadores, por ej los que vimos en la guia 17, pasa que depender x ej de si sabes la distribución de tu v.a.)

```
alpha_EMV/lambda_EMV
```

```
## shape
## 34.44294
```

```
alpha_MME/lambda_MME
```

```
## [1] 34.44295
```

```
mean(flux)
```

```
## [1] 34.44295
```

8. Estimar por el método plug-in la mediana de flux. Explorar el comando `qgamma()`

Resolución:

Estimamos usando el comando `qgamma(0.5, shape=alpha, rate=lambda)` con el método plug-in usando los parámetros obtenidos con los métodos EMV y MME, y además compararemos con el comando `median(flux)`.

```
mediana_EMV=qgamma(0.5,shape=alpha_EMV,rate=lambda_EMV)
mediana_EMV
```

```
## [1] 16.95737
```

```
mediana_MME=qgamma(0.5,shape=alpha_MME,rate=lambda_MME)
mediana_MME
```

```
## [1] 26.61185
```

```
median(flux)
```

```
## [1] 27.4
```