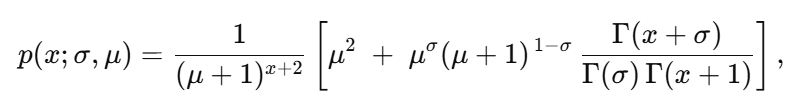
# Taller 2

# Implementación de distribuciones en gamlss

Nombre: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

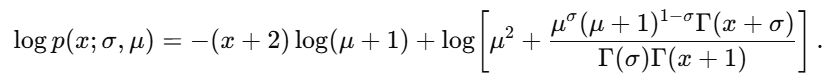
El objetivo de este taller es crear la familia PGL para la distribución Poisson-Generalized Lindley (PGL).

1. En el artículo Altun, E. (2021). A new two-parameter discrete Poisson-generalized Lindley distribution with properties and applications to healthcare data sets. *Computational Statistics*, *36*(4), 2841-2861, el autor propone una nueva distribución discreta y la función de masa para esta distribución es:





Al calcular el logaritmo de la función de masa se obtiene la siguiente expresión:



Calcule las siguientes dos derivadas parciales:

1. Calcule el logaritmo natural de la función .
2. Complete el siguiente código de R para crear la función dPGL. Lo que nos interesa es que usted escriba la parte central de la función, no escriba los bloques de código generales.

dPGL <- function(x, mu = 0.5, sigma = 0.5, log = FALSE) {

...

# pdf in log-scale

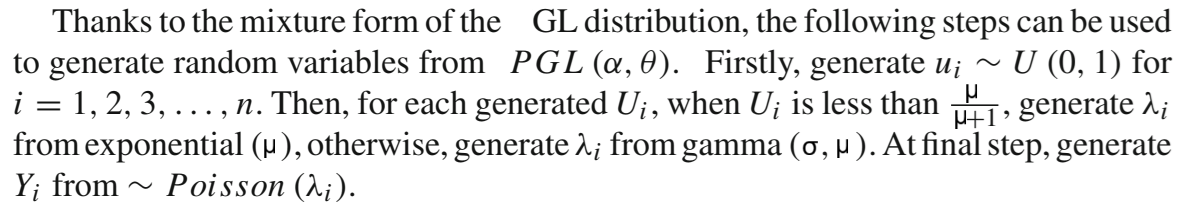
...

}

1. Escriba dentro del recuadro el código de R para replicar la siguiente figura del artículo.

|  |  |
| --- | --- |
|  |  |

1. En algunas distribuciones, debido a su construcción, la generación de los números aleatorios puede hacerse aprovechando aprovechando las propiedades sin hacer uso de la función de cuantiles. En la siguiente imagen aparece la recomendación del autor Altun para generar números aleatorios de la distribución GPL.



Complete el siguiente código de R para generar n observaciones aleatorias de la distribución usando la recomendación anterior.

rGPL <- function(n, mu, sigma) {

u <- runif(n=n, min= , =1)

lambda <- numeric(n)

id <- u < mu / ( + 1) # To identify the condition

lambda[id] <- rexp(n=sum(id), rate = )

lambda[!id] <- r (n=sum(!id), shape = sigma, rate = )

y <- rpois(n=n, =lambda)

return( )

}