

# Introducción a GAMLSS

Curso: Modelos lineales generalizados

2020

Presentado por: Carlos Andrés Pérez Aguirre

# Introducción

Modelo tradicional (obtenido con la función lm)

$$y_i = \beta_0 + \beta_1 x_i + e_i$$

$$e_i \sim N(0, \sigma^2)$$

Modelo equivalente reescrito

$$y_i \sim N(\mu_i, \sigma^2)$$

$$\mu_i = \beta_0 + \beta_1 x_i$$

Modelos GAMLSS

$$y_i \sim \mathcal{D}(\mu_i, \sigma_i, \nu_i, \tau_i)$$

$$g(\mu_i) = \beta_0 + \beta_1 x_i$$

$$g(\sigma_i) = \beta_2 + \beta_3 x_i$$

$$g(\nu_i) = \beta_4 + \beta_5 x_i$$

$$g(\tau_i) = \beta_6 + \beta_7 x_i$$

$\mathcal{D}$ : Puede ser cualquier distribución

Notas: No todas las distribuciones tienen cuatro parámetros

# Función de enlace

$$g(\theta_i) = \beta_0 + \beta_1 x_i$$

$$\theta_i \in \mathbb{R}$$

$$\theta_i = \beta_0 + \beta_1 x_i$$

$$\theta_i \in (0, 1)$$

$$\log\left(\frac{\theta_i}{1 - \theta_i}\right) = \beta_0 + \beta_1 x_i$$

$$\theta_i = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_i)}}$$

$$\theta_i \in \mathbb{R}^+$$

$$\log(\theta_i) = \beta_0 + \beta_1 x_i$$

$$\theta_i = e^{\beta_0 + \beta_1 x_i}$$

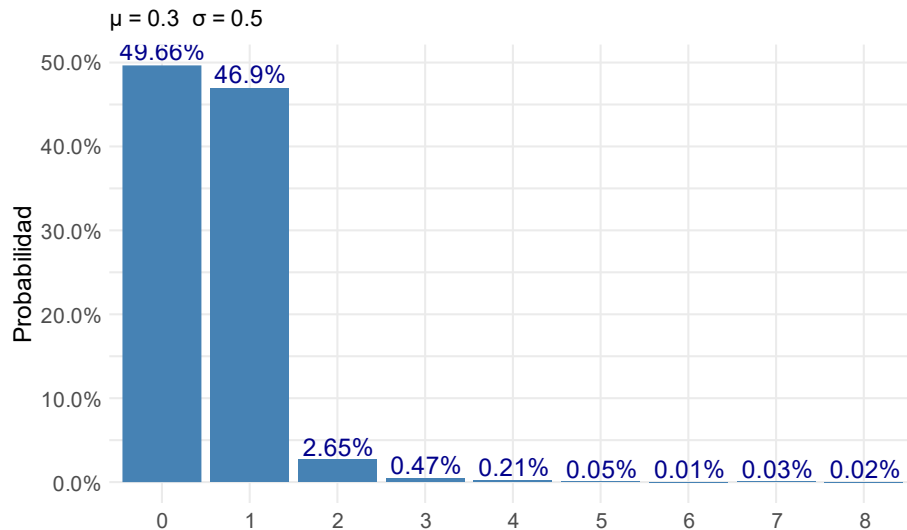
# Ajuste marginal

Distribución Pareto discreta inflada en cero (ZAZIPF)

$$P(Y = 0|\mu, \sigma) = \sigma$$

$$P(Y = y|\mu, \sigma) = (1 - \sigma) \frac{y^{-(\mu+1)}}{\zeta(\mu + 1)}$$

$$\mu \in \mathbb{R}^+ \quad \sigma \in (0, 1)$$



Donde  $\zeta()$  es la función zeta de Reimann

```
if (!require("gamlss")) install.packages("gamlss")
```

```
PARAM_MU <- 3
```

```
PARAM_SIGMA <- 0.5
```

```
set.seed(122)
```

```
y <- rZAZIPF(10000, mu = PARAM_MU,  
             sigma = PARAM_SIGMA)
```

```
estimation <- gamlssML(y, family=ZAZIPF)
```

```
print(paste("mu:", round(estimation$mu, 3)))
```

```
print(paste("sigma:", round(estimation$sigma, 3)) )
```

Salida:

mu: 3.137

sigma: 0.497

Ejemplo1.Rmd en el repositorio de github

# Simulación ajuste marginal

Simulación 1:

$$\mu_0 = 0.5$$

$$\sigma_0 = 0.3$$

Tamaños de muestra = {5, 10, 20, 30, 50, 100, 150, 200, 300, 500, 700, 1000}

Número de repeticiones = 1000

Resultados = {}

**Para cada  $n$  en** Tamaños de muestra:

**Repetir** Número de repeticiones **veces**:

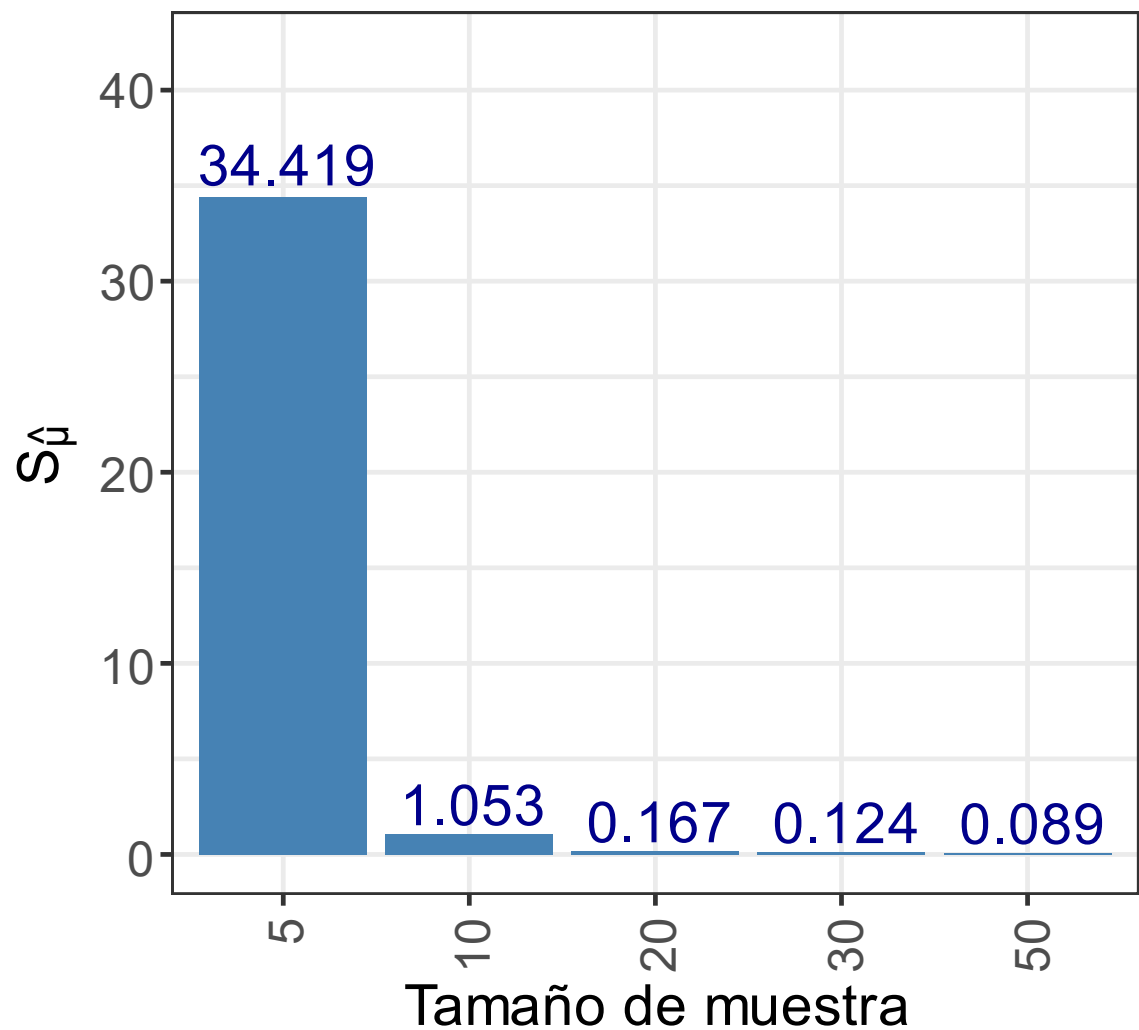
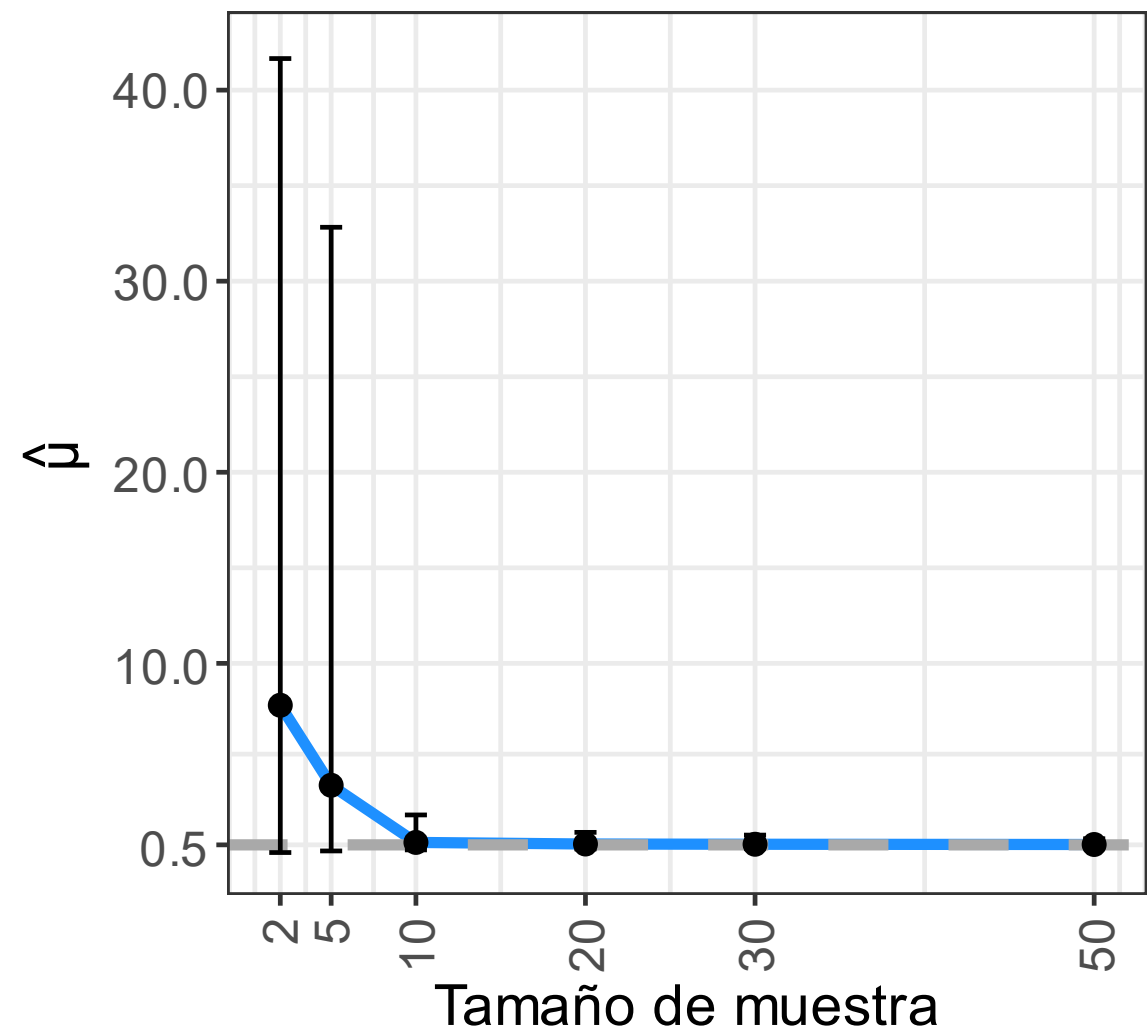
$y \leftarrow$  Tomar una muestra aleatoria de tamaño  $n$  de la distribución **ZAZIPF**( $\mu = \mu_0, \sigma = \sigma_0$ )

Estimaciones  $\leftarrow$  Estimar  $\{\mu, \sigma\}$  a partir de  $y$  usando **gamlssML**

Almacenar  $\{\mu, \sigma, n\}$  en Resultados.

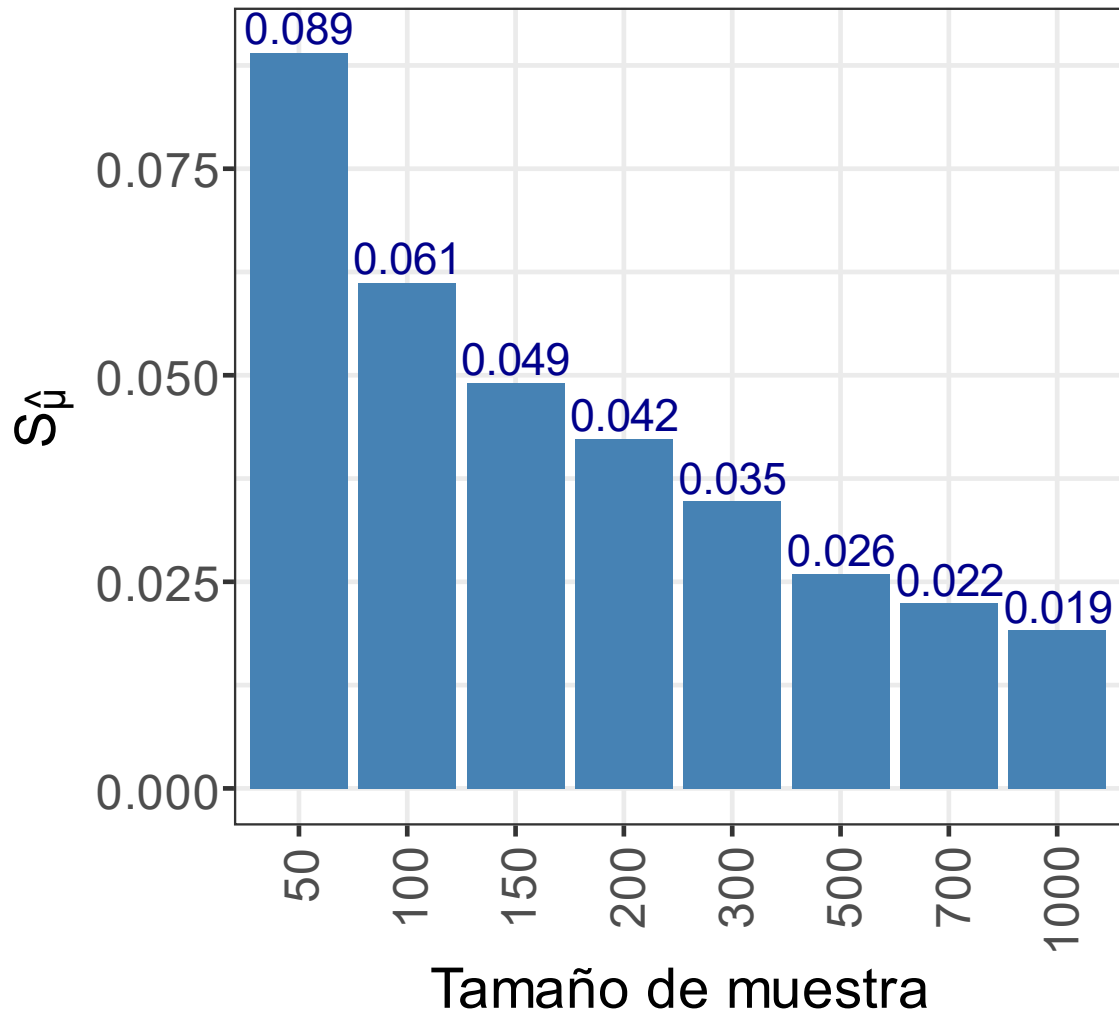
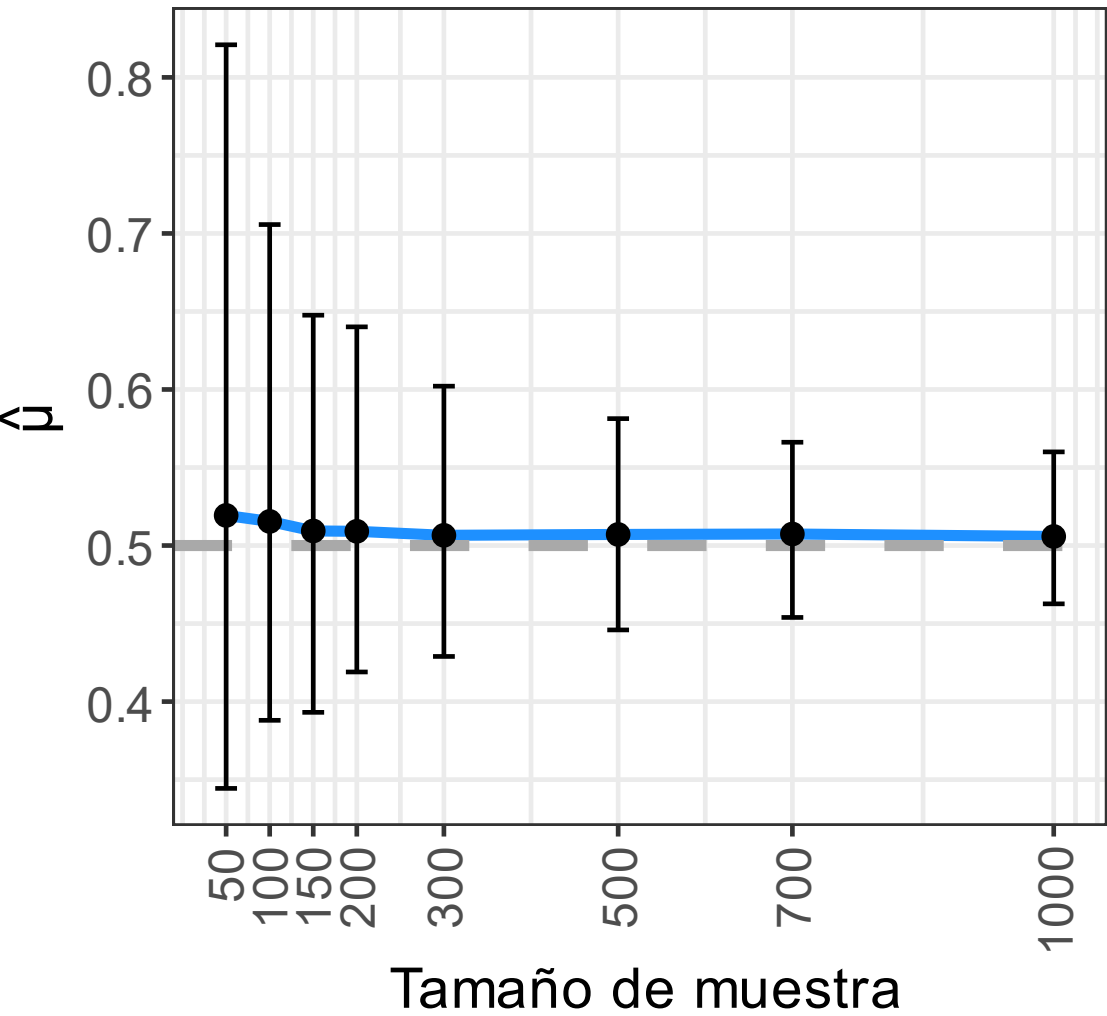
# Resultados

$\mu = 0.5$



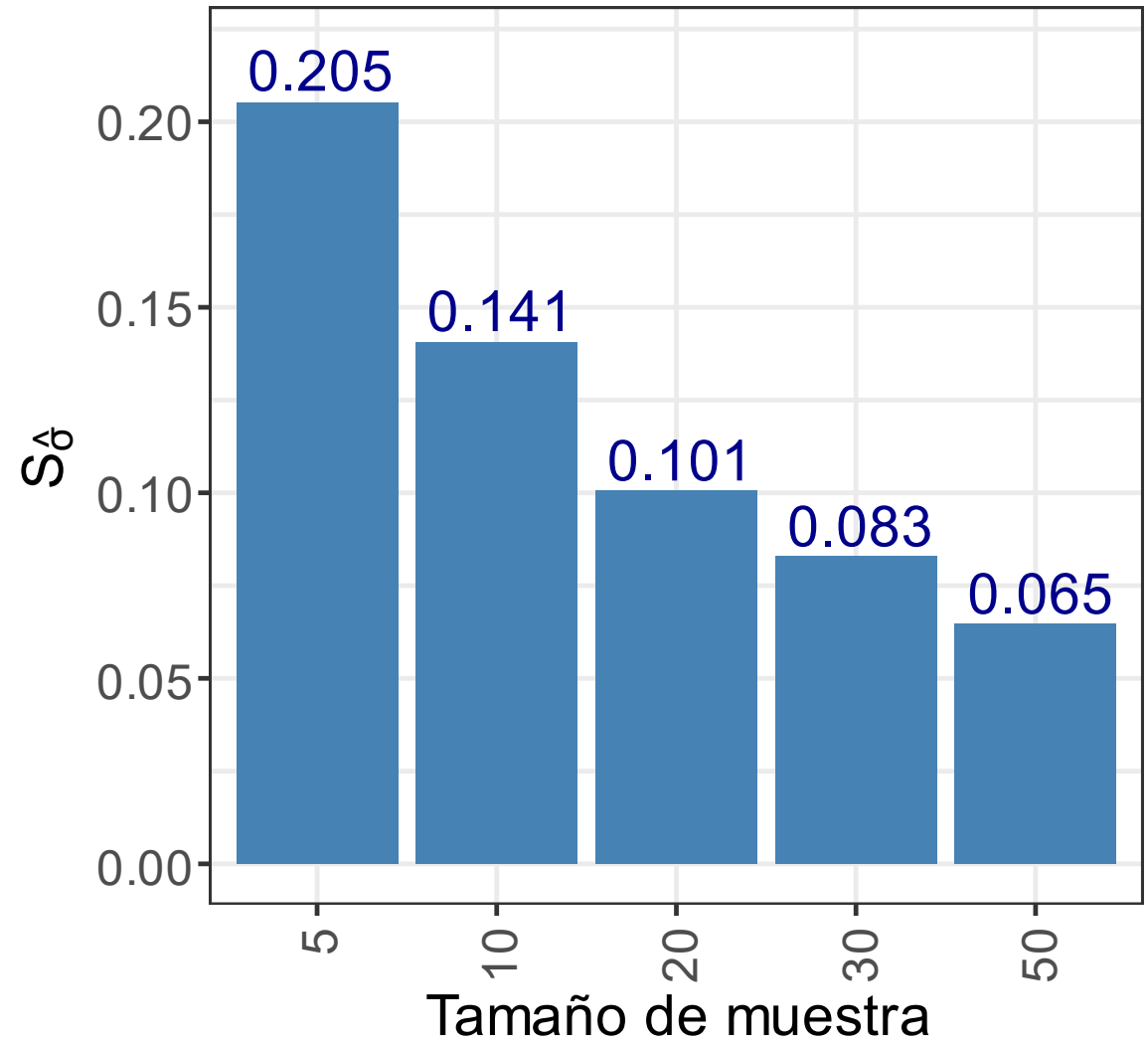
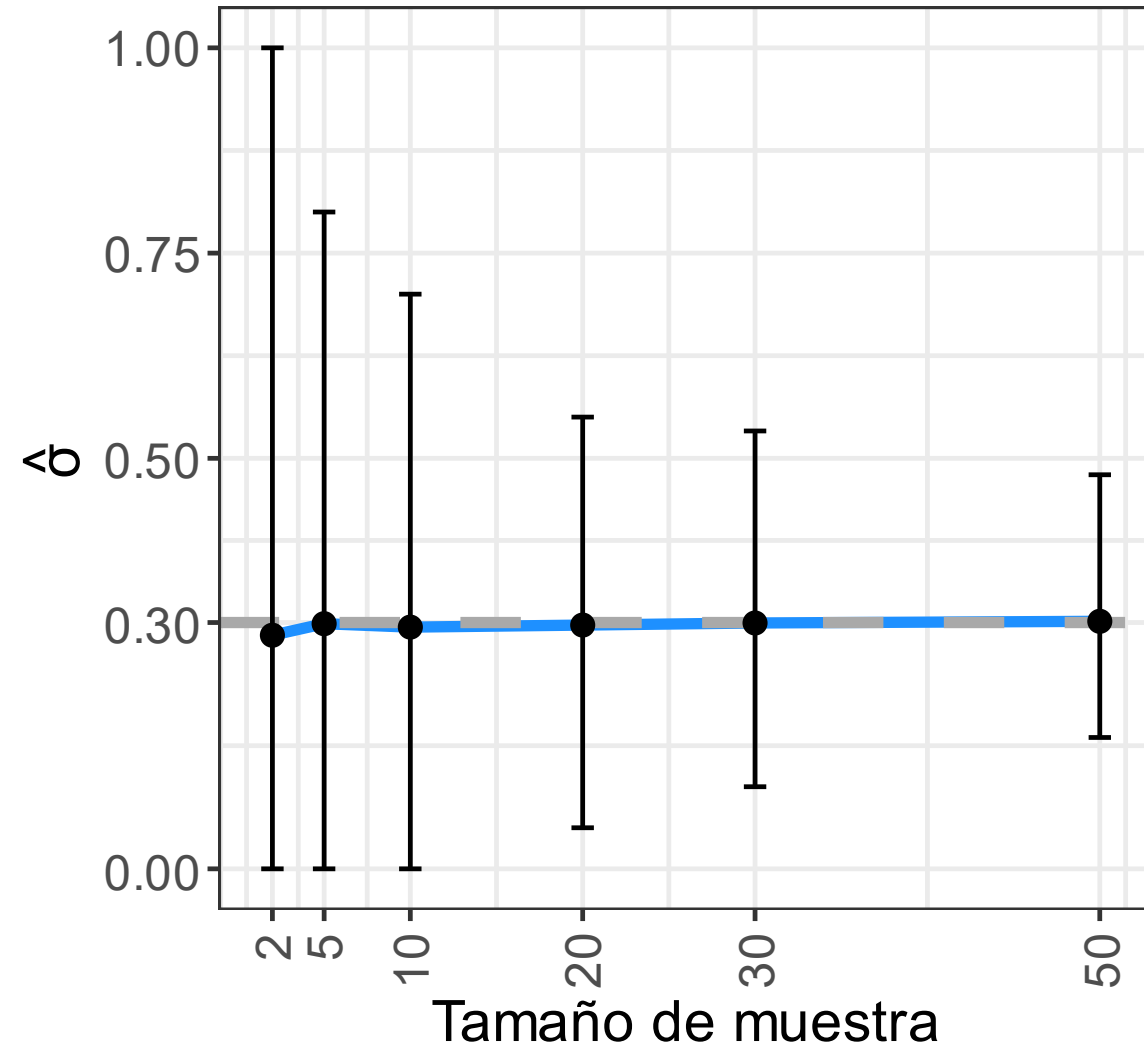
# Resultados

$\mu = 0.5$



# Resultados

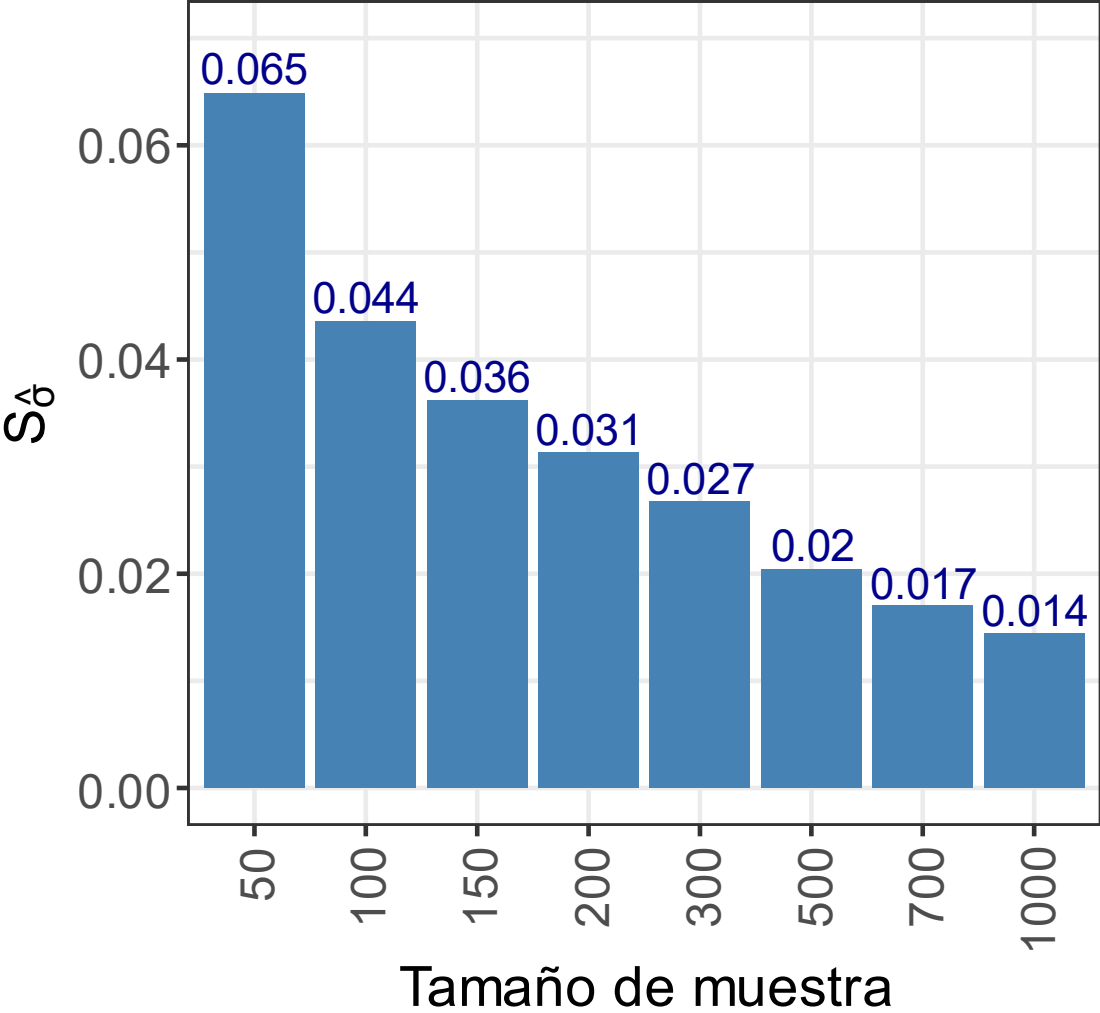
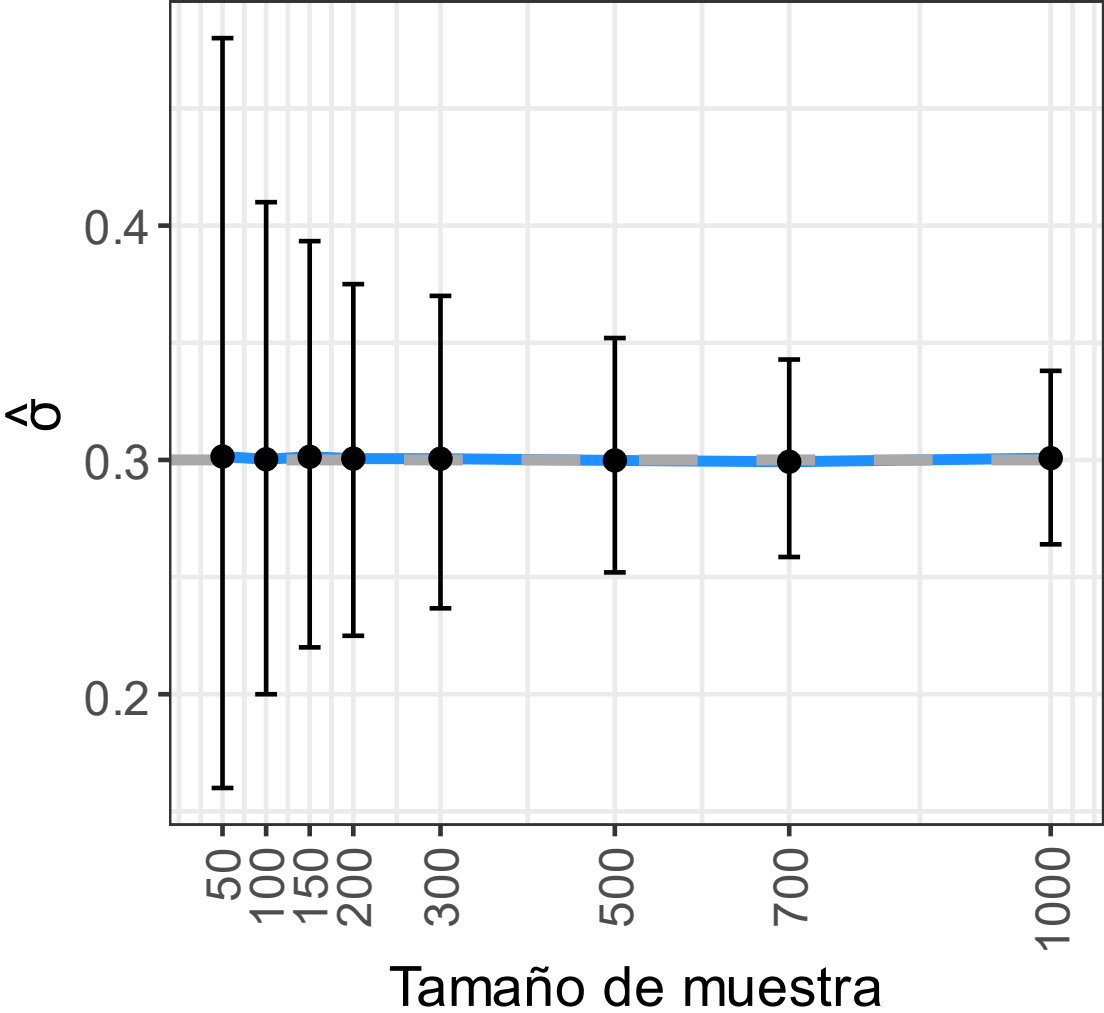
$$\sigma = 0.3$$





# Resultados

$\sigma = 0.3$

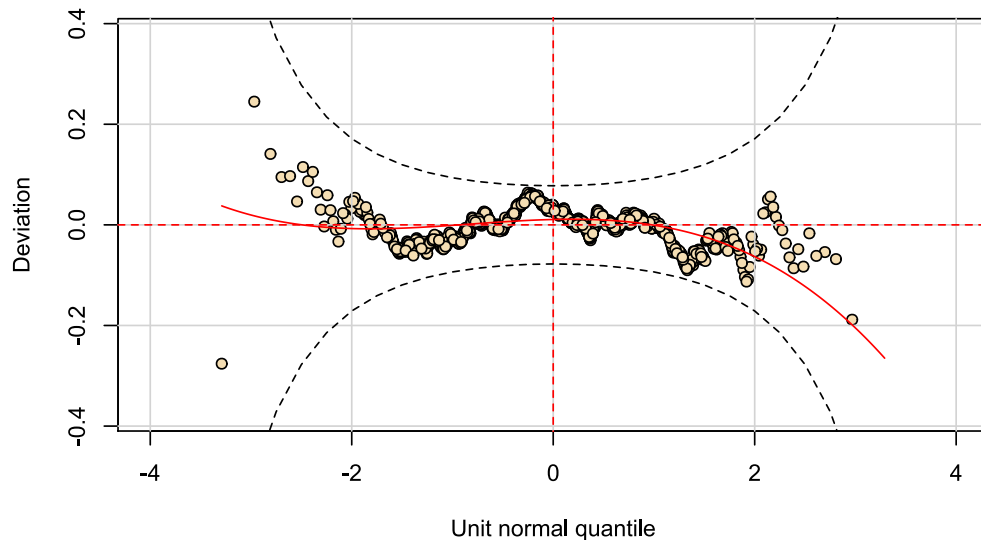


# Ajuste con covariables

Modelo a simular

$$\begin{aligned}y &\sim \text{ZAZIPF}(\mu, \sigma) \\ \log(\mu) &= 1 + 0.015x_1 \\ x_1 &\sim \text{Gamma}(\alpha = 2, \beta = 1) \\ \sigma &= 0.3\end{aligned}$$

Grafico de gusano



```
if (!require("gamlss")) install.packages("gamlss")
```

```
B0_MU <- 1  
B1_MU <- 0.015  
PARAM_SIGMA <- 0.3  
PARAM_SHAPE <- 2
```

```
set.seed(122)  
x <- rgamma(1000, shape = PARAM_SHAPE)
```

```
PARAM_MU <- exp(B0_MU + B1_MU*x)
```

```
y <- rZAZIPF(1000, mu = PARAM_MU, sigma = PARAM_SIGMA)
```

```
model <- gamlss(y ~ x, family = ZAZIPF)
```

```
print(paste(c("b0:", "b1:"), round(model$mu.coefficients, 3)))  
print(paste("sigma:", round(plogis(model$sigma.coefficients), 3)))  
wp(model)
```

Salida:

b0: 0.991	b1: 0.014
sigma: 0.315	

Ejemplo2.Rmd en el repositorio de github

# Simulación ajuste covariables

Modelo 1:

$$\begin{aligned}y &\sim \text{ZAZIPF}(\mu, \sigma) \\ \log(\mu) &= \beta_{0,\mu} + \beta_{1,\mu}x_1 + \beta_{2,\mu}x_2 \\ \log\left(\frac{\sigma}{1-\sigma}\right) &= \beta_{0,\sigma} + \beta_{1,\sigma}x_1 + \beta_{2,\sigma}x_2 \\ x_1 &\sim \text{Poisson}(\lambda = 3) \\ x_2 &\sim \text{Gamma}(\alpha = 2, \beta = 1)\end{aligned}$$

Donde:

$$\begin{aligned}\beta_{0,\mu} &= -1 \\ \beta_{1,\mu} &= -0.5 \\ \beta_{2,\mu} &= 2 \\ \beta_{0,\sigma} &= -2 \\ \beta_{1,\sigma} &= -4 \\ \beta_{2,\sigma} &= 5\end{aligned}$$

Tamaños de muestra = {5, 10, 20, 30, 50, 100, 150, 200, 300, 500, 700, 1000}

Número de repeticiones = 1000

Resultados = {}

Simulación 2:

**Para cada  $n$  en** Tamaños de muestra:

**Repetir** Número de repeticiones **veces**:

$y \leftarrow$  Tomar una muestra aleatoria de tamaño  $n$  del **modelo 1**

Estimaciones  $\leftarrow$  Estimar  $\{\mu, \sigma\}$  a partir de  $y$  usando **gamlss**

Almacenar  $\{\beta_{0,\mu}, \beta_{1,\mu}, \beta_{2,\mu}, \beta_{0,\sigma}, \beta_{1,\sigma}, \beta_{2,\sigma}, n\}$  en Resultados.

# Resultados

$$\beta_{0,\sigma} = -2 \beta_{1,\sigma} = -4 \beta_{2,\sigma} = 5$$

Tamaño de muestra	$\hat{\beta}_{0,\sigma}$	$\hat{\beta}_{1,\sigma}$	$\hat{\beta}_{2,\sigma}$
5	-31.7020	-10.2736	17.8802
10	-36.0437	-77.6747	117.5727
20	-9.9300	-283.5938	360.4604
30	-3.3659	-158.1165	156.3175
50	-2.9259	-5.8651	7.2319
100	-2.3273	-4.6546	5.7272
150	-2.1725	-4.3994	5.4628
200	-2.0798	-4.2846	5.3385
300	-2.1003	-4.1853	5.2419
500	-2.0756	-4.0970	5.0988
700	-2.0446	-4.0712	5.0940
1000	-2.0155	-4.0573	5.0527

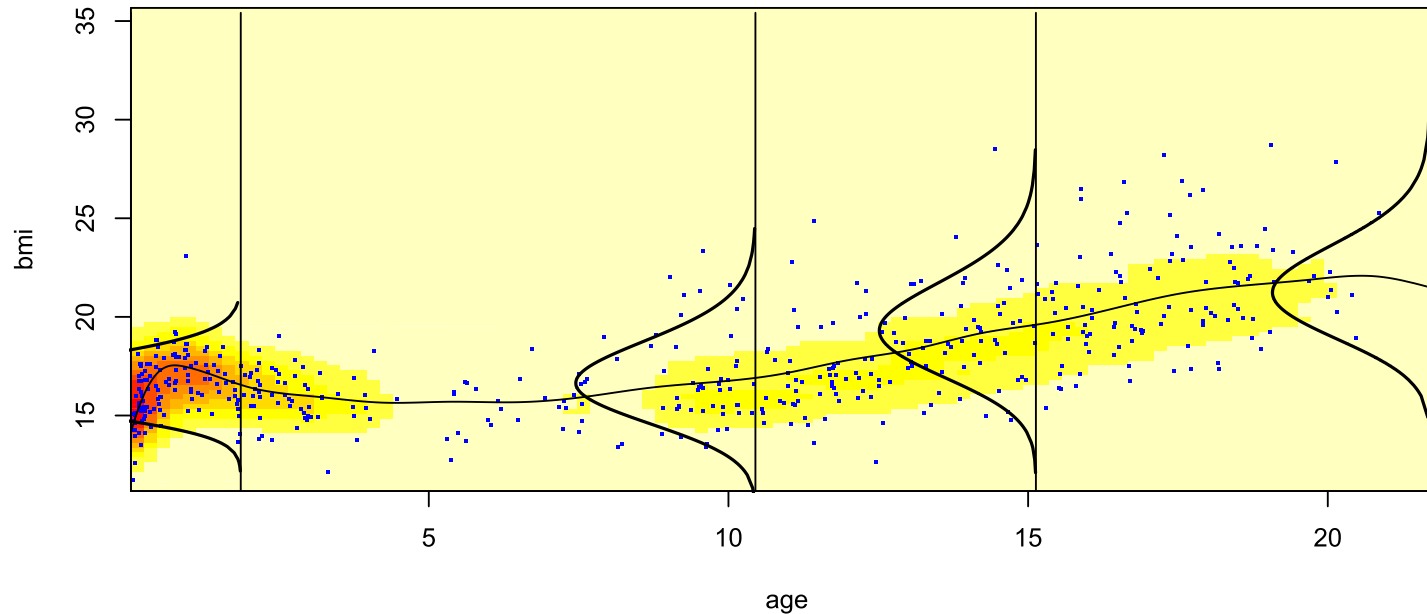
# Ejemplo splines

```
if (!require("gamlss")) install.packages("gamlss")  
if (!require("gamlss.util")) install.packages("gamlss.util")
```

```
data("dbbmi")
```

```
m1 <- gamlss( bmi~pbz(age), sigma.formula = ~pbz(age), data=dbbmi, family=GA )
```

```
plotSimpleGamlss(bmi, age, model=m1, data=dbbmi)
```



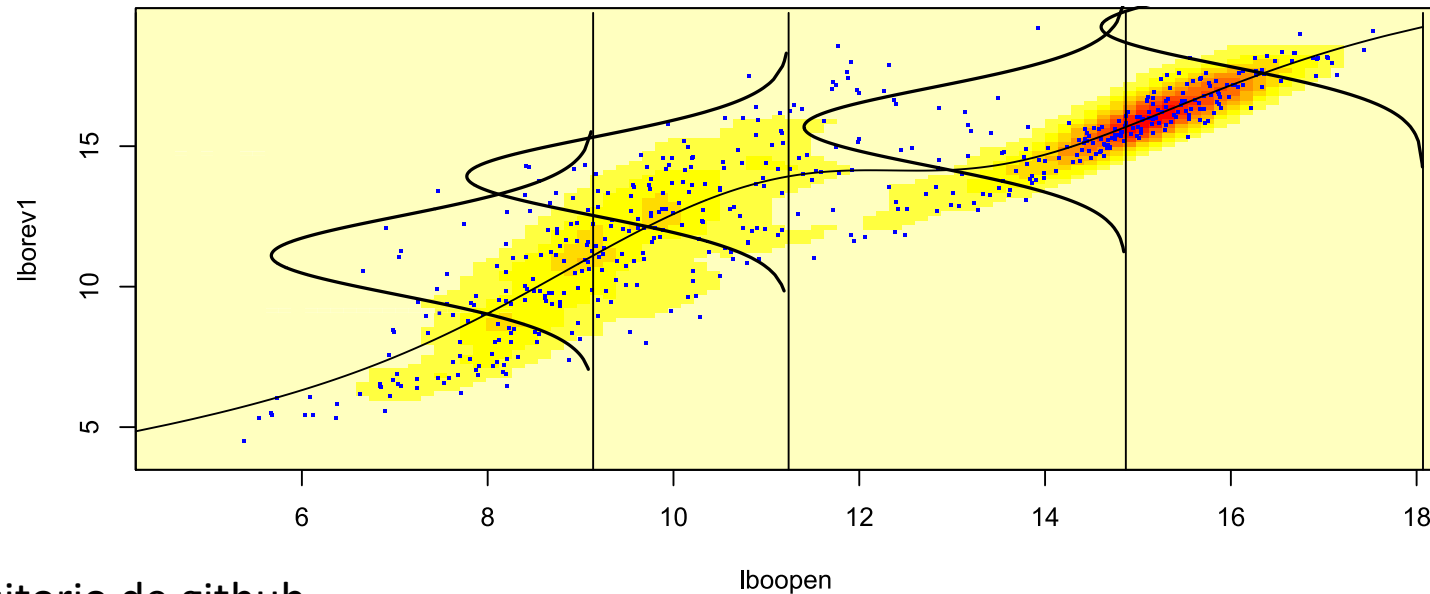
# Ejemplo redes neuronales

```
if (!require("gamlss")) install.packages("gamlss")  
if (!require("gamlss.util")) install.packages("gamlss.util")  
if (!require("gamlss.add")) install.packages("gamlss.add")
```

```
data(film90)
```

```
m1 <- gamlss(lborev1~nn(~lboopen,size=20,decay=0.1), data=film90,family=NO)
```

```
plotSimpleGamlss(lborev1, lboopen, model=m1, data=film90)
```



# Referencias

Rigby, R., Stasinopoulos, M., Heller, G., & Bastiani, F. D. (s. f.). *Distributions for Modelling Location, Scale and Shape: Using GAMLSS in R*. 378.

Stasinopoulos, M. D., Rigby, R. A., Heller, G. Z., Voudouris, V., & Bastiani, F. D. (s. f.). *Flexible Regression and Smoothing*. 572.

*plotSimpleGamlss: Plotting a simple GAMLSS model for demonstration purpose in gamlss.util: GAMLSS Utilities*. (s. f.). Recuperado 17 de agosto de 2020, de <https://rdrr.io/cran/gamlss.util/man/plotSimpleGamlss.html>