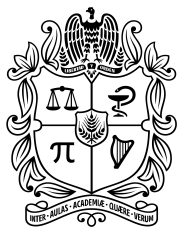
Clase 10 - Módulo 2: Introducción a la analítica

Mauricio Alejandro Mazo Lopera

Universidad Nacional de Colombia
Facultad de Ciencias
Escuela de Estadística
Medellín



UNIVERSIDAD
NACIONAL
DE COLOMBIA

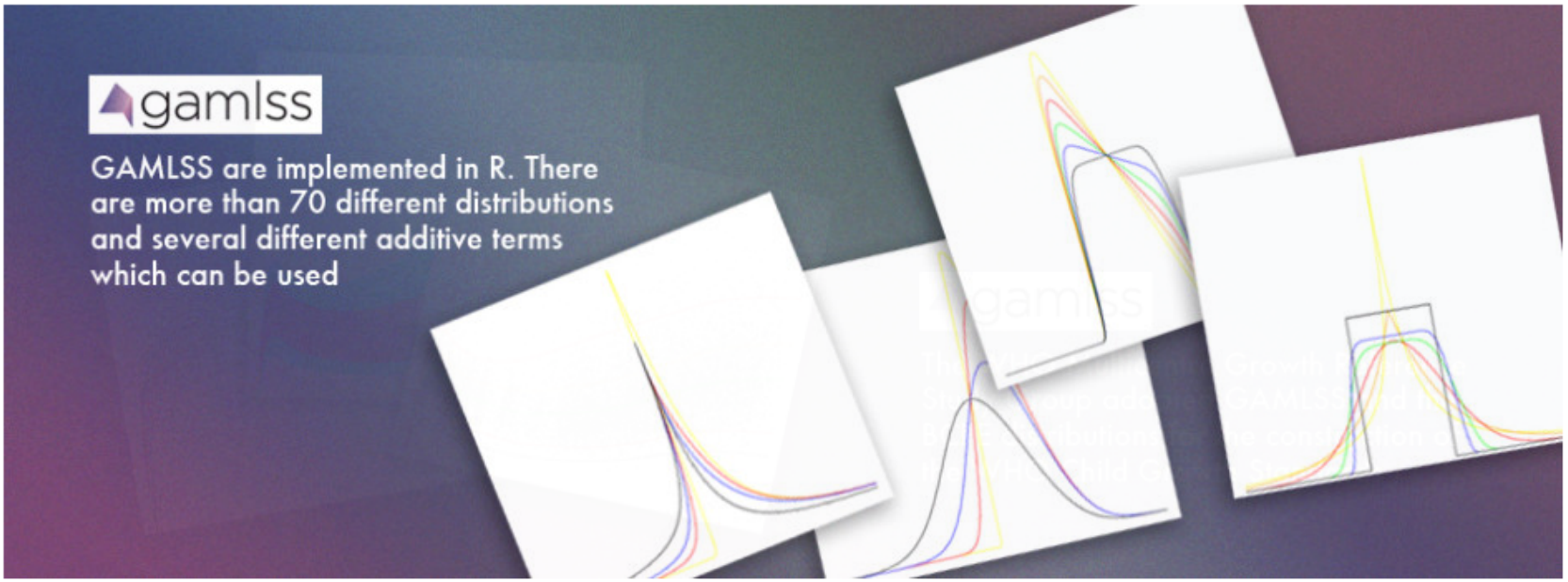
Modelos aditivos generalizados para ubicación escala y forma (GAMLSS)

gamlss

HOME MORE ON GAMLSS THE TEAM INFORMATION BLOG CONTACT

Generalized Additive Models for Location, Scale and Shape

Statistical modelling at its best



Planteamiento de los modelos GAMLSS

Considere los siguientes vectores de parámetros:

μ : Parámetro relacionados con la ubicación, por ejemplo, media

σ : Parámetro relacionados con la escala, por ejemplo, varianza

ν : Parámetro relacionados con la forma, por ejemplo, asimetría

τ : Parámetro relacionados con la forma, por ejemplo, curtosis

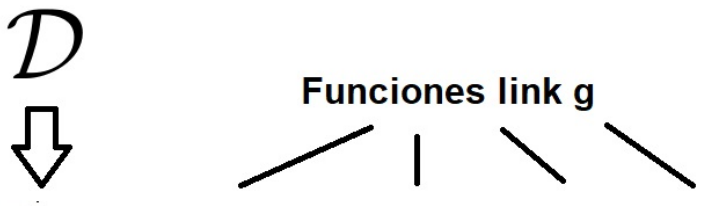
Asumiendo que Y es la variable respuesta y $\mathcal{D}$ es alguna distribución, el modelo GAMLSS se puede plantear como:

$$
\begin{aligned}
\boldsymbol{Y} &\sim \mathcal{D}(\mu, \sigma, \nu, \tau) \\
g_1(\mu) &= \beta_{01} + f_{11}(X_1) + f_{12}(X_2) + \ldots + f_{1p}(X_p) \\
g_2(\sigma) &= \beta_{02} + f_{21}(X_1) + f_{22}(X_2) + \ldots + f_{2p}(X_p) \\
g_3(\nu) &= \beta_{03} + f_{31}(X_1) + f_{32}(X_2) + \ldots + f_{3p}(X_p) \\
g_4(\tau) &= \beta_{04} + f_{41}(X_1) + f_{42}(X_2) + \ldots + f_{4p}(X_p)
\end{aligned}
$$

Opciones para las f_{rj} para $r = 1, 2, 3, 4$ y $j = 1, 2, \ldots, p$:

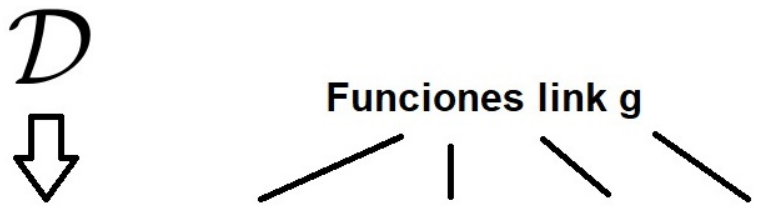
Additive term	**gamlss** function name
cubic splines	`cs()`, `scs()`
decision trees	`tr()`
fractional and power polynomials	`fp()`, `pp()`
free knots (break points)	`fk()`
`loess`	`lo()`
neural networks	`nn()`
nonlinear fit	`nl()`
P-splines	`pb()`, `pb0()`, `ps()`,
P-splines cyclic	`pbc()`, `cy()`
P-splines monotonic	`pbm()`
P-splines shrinking to zero	`pbz()`
P-splines varying coefficient	`pvc()`
penalized categorical	`pcat()`
random effects	`random()`, `re()`
ridge regression	`ri()`
Simon Wood's `gam`	`ga()`
Stephen Milborrow's `earth`	`ma()`

Opciones para las $\mathcal{D}$ y $g_1, g_2, g_3 g_4$:



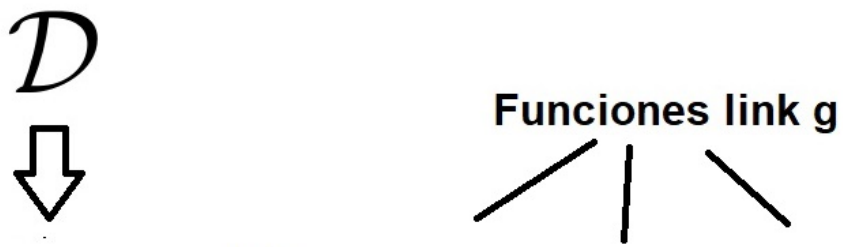
Distributions	R Name	μ	σ	ν	τ
beta	BE()	logit	logit	-	-
Box-Cox Cole and Green	BCCG()	identity	log	identity	-
Box-Cox power exponential	BCPE()	identity	log	identity	log
Box-Cox t	BCT()	identity	log	identity	log
exponential	EXP()	log	-	-	-
exponential Gaussian	exGAUS()	identity	log	log	-
exponential gen. beta type 2	EGB2()	identity	identity	log	log
gamma	GA()	log	log	-	-
generalized beta type 1	GB1()	logit	logit	log	log
generalized beta type 2	GB2()	log	identity	log	log
generalized gamma	GG()	log	log	identity	-
generalized inverse Gaussian	GIG()	log	log	identity	-
generalized t	GT()	identity	log	log	log
Gumbel	GU()	identity	log	-	-
inverse Gaussian	IG()	log	log	-	-
Johnson's SU (μ the mean)	JSU()	identity	log	identity	log
Johnson's original SU	JSUo()	identity	log	identity	log
logistic	LO()	identity	log	-	-
log normal	LOGNO()	log	log	-	-
log normal (Box-Cox)	LNO()	log	log	fixed	-
NET	NET()	identity	log	fixed	fixed
normal	NO()	identity	log	-	-
normal family	NOF()	identity	log	identity	-
power exponential	PE()	identity	log	log	-

Opciones para las $\mathcal{D}$ y $g_1, g_2, g_3 g_4$:



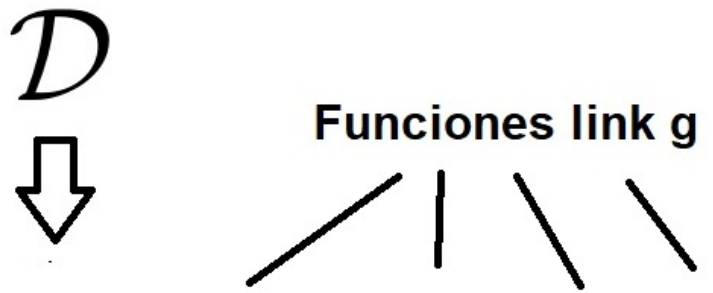
reverse Gumbel	RG()	identity	log	-	-
skew power exponential type 1	SEP1()	identity	log	identity	log
skew power exponential type 2	SEP2()	identity	log	identity	log
skew power exponential type 3	SEP3()	identity	log	log	log
skew power exponential type 4	SEP4()	identity	log	log	log
sinh-arcsinh	SHASH()	identity	log	log	log
skew t type 1	ST1()	identity	log	identity	log
skew t type 2	ST2()	identity	log	identity	log
skew t type 3	ST3()	identity	log	log	log
skew t type 4	ST4()	identity	log	log	log
skew t type 5	ST5()	identity	log	identity	log
t Family	TF()	identity	log	log	-
Weibull	WEI()	log	log	-	-
Weibull (PH)	WEI2()	log	log	-	-
Weibull (μ the mean)	WEI3()	log	log	-	-

Opciones para las $\mathcal{D}$ y $g_1, g_2, g_3 g_4$:



Distributions	R Name	μ	σ	ν
beta binomial	BB()	logit	log	-
binomial	BI()	logit	-	-
logarithmic	LG()	logit	-	-
Delaporte	DEL()	log	log	logit
negative binomial type I	NBI()	log	log	-
negative binomial type II	NBII()	log	log	-
Poisson	PO()	log	-	-
Poisson inverse Gaussian	PIG()	log	log	-
Sichel	SI()	log	log	identity
Sichel (μ the mean)	SICHEL()	log	log	identity
zero altered beta binomial	ZABB()	logit	log	logit
zero altered binomial	ZABI()	logit	logit	-
zero altered logarithmic	ZALG()	logit	logit	-
zero altered neg. binomial	ZANBI()	log	log	logit
zero altered poisson	ZAP()	log	logit	-
zero inflated beta binomial	ZIBB()	logit	log	logit
zero inflated binomial	ZIBI()	logit	logit	-
zero inflated neg. binomial	ZINBI()	log	log	logit
zero inflated poisson	ZIP()	log	logit	-
zero inflated poisson (μ the mean)	ZIP2()	log	logit	-
zero inflated poisson inv. Gaussian	ZIPIG()	log	log	logit

Opciones para las $\mathcal{D}$ y $g_1, g_2, g_3 g_4$:



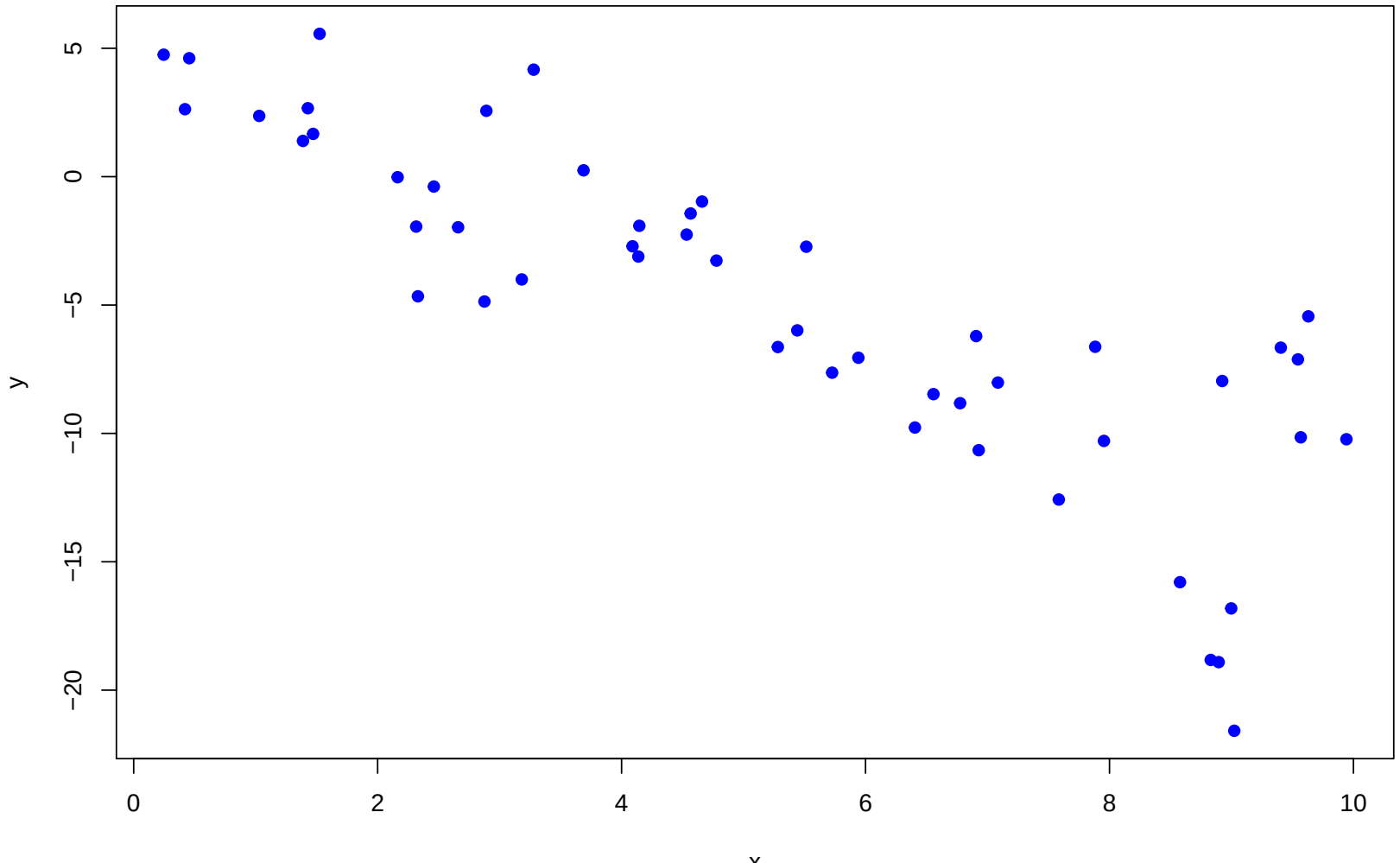
beta inflated (at 0)	BEOI()	logit	log	logit	-
beta inflated (at 0)	BEINF0()	logit	logit	log	-
beta inflated (at 1)	BEZI()	logit	log	logit	-
beta inflated (at 1)	BEINF1()	logit	logit	log	-
beta inflated (at 0 and 1)	BEINF()	logit	logit	log	log
zero adjusted GA	ZAGA()	log	log	logit	-
zero adjusted IG	ZAIG()	log	log	logit	-

Simulación 1 para ver cómo funciona GAMLSS:

```
set.seed(123)
n <- 50
x <- runif(n, 0, 10)
beta_0<-5
beta_1<--2
alpha_0<-1
alpha_1<-0.5
sigma <- alpha_0 + alpha_1 * x
error <- rnorm(n, mean=0, sd=sigma)
y<- beta_0+ beta_1*x+ error
```

Simulación 1 para ver cómo funciona GAMLSS:

```
plot(x,y, col="blue", pch=19)
```



Simulación 1 para ver cómo funciona GAMLSS:

```
require(gamlss)
mod1 <- gamlss(y~x, sigma.fo=~x,
               family=NO(mu.link = "identity",
                         sigma.link="identity"))
```

```
## GAMLSS-RS iteration 1: Global Deviance = 251.7552
## GAMLSS-RS iteration 2: Global Deviance = 251.7128
## GAMLSS-RS iteration 3: Global Deviance = 251.7128
```

Simulación 1 para ver cómo funciona GAMLSS:

```
summary(mod1)
```

```
## ******************************************************************
## Family:  c("NO", "Normal")
##
## Call:
## gamlss(formula = y ~ x, sigma.formula = ~x, family = NO(mu.link = "identity",
##     sigma.link = "identity"))
##
## Fitting method: RS()
##
## ------------------------------------------------------------------
## Mu link function:  identity
## Mu Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.6325     0.5524   8.386 8.02e-11 ***
## x            -1.8612     0.1418 -13.126  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## ------------------------------------------------------------------
## Sigma link function:  identity
## Sigma Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.1732     0.4434   2.646  0.01111 *
## x             0.3939     0.1129   3.490  0.00108 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## ------------------------------------------------------------------
## No. of observations in the fit:  50
```

Simulación 1 para ver cómo funciona GAMLSS:

```
c(beta_0,beta_1) # Betas reales
```

```
## [1]  5 -2
```

```
coef(mod1,what="mu") # Betas estimados por gamlss
```

```
## (Intercept)           x
##    4.632484   -1.861152
```

```
c(alpha_0,alpha_1) # Alphas reales
```

```
## [1] 1.0 0.5
```

```
coef(mod1,what="sigma") # Alphas estimados por gamlss
```

```
## (Intercept)           x
##   1.1732435   0.3939464
```

Criterios de información:

En los resumenes que presenta la función **gamlss** aparecen dos medidas que se conocen como criterio de información de Akaike (AIC) y criterio de información Bayesiano (SBC o también denotado BIC). Estas medidas dependen del valor que se alcanza al optimizar la función Global Deviance. Así, diremos que

$$
\begin{aligned}
n &: \text{Número de individuos} \\
k &: \text{Denota el número de parámetros del modelo} \\
L &: \text{Función de verosimilitud} \\
\text{Global Deviance} &= -2log(L) \\
AIC &= \text{Global Deviance} + 2k \\
SBC \text{ ó } BIC &= \text{Global Deviance} + k\log(n)
\end{aligned}
$$

El AIC y el SBC penalizan la Global Deviance con respecto al número de parámetros. Estas medidas no tienen una interpretación, solo se usan para comparar modelos y el mejor es el que tenga el menor AIC o el menor BIC.

Actividad 1 para realizar en clase:

Considere la base de datos **DATOS_A**.

1. Seleccione las variables más importantes entre $X_1, X_2, \ldots, X_{14}$ para explicar Y.
2. Realice gráficos descriptivos con las variables seleccionadas en el item anterior.
3. Ajuste al menos 3 modelos GAMLSS donde considere distintas funciones de suavizamiento, además de un ajuste para la variabilidad.
4. Seleccione el mejor modelo entre los 3 anteriores.

Actividad 2 para realizar en clase:

La base de datos **DATOS_B** contiene registros de accidentalidad de 728 días en una de las avenidas principales de cierta ciudad. Las variables se codifican como:

- Y: Número de accidentes díarios.
- X_1: Número de vehículos que transitan por día.
- X_2: Llueve (1) o No llueve (0).
- X_3: Día de la semana.

1. Ajuste un modelo gamlss a los datos considerando la distribución Poisson.
2. Interprete el modelo ajustado.
3. Escriba la ecuación del modelo.