

Modelos GAM y GAMLSS

Humberto Vaquera Huerta

2022-07-12

Introducción

Muchos datos en las agropecuarias no se ajustan a modelos lineales simples y se describen mejor con "modelos ondulantes", también conocidos como **Modelos aditivos generalizados (GAM)**.

Suponga un modelo de regresión lineal simple:

$$y = \beta_0 + x_1\beta_1 + \varepsilon, \quad \varepsilon \sim N(0, \sigma^2)$$

Lo que cambia en un **GAM** es la presencia de un término de suavizado:

$$y = \beta_0 + f(x_1) + \varepsilon, \quad \varepsilon \sim N(0, \sigma^2)$$

Esto simplemente significa que la contribución al predictor lineal ahora es alguna función f . Esto parecido al uso de un término cuadrático (x_1^2) o cúbico (x_1^3) como su predictor.

la función f generalmente se suaviza con *splines*.

Puede tener combinaciones de términos lineales y suaves en su modelo, por ejemplo

$$y = \beta_0 + x_1\beta_1 + f(x_2) + \varepsilon, \quad \varepsilon \sim N(0, \sigma^2)$$

Introducción

En el modelo lineal generalizado la forma de introducir el efecto de las variables predictivas en el modelo se da usando la funciones, $\eta(x_i) = \sum_{j=1}^p \beta_j x_{ij}$, las cuales pueden ser restrictivas ya que no introducen efectos no lineales en la relacion.

El **modelo Generalized additive (GAM)**, introducido por **Hastie, T. y Tibshirani, R. (1986)**,., permite una mayor flexibilidad al modelar el predictor lineal de un modelo lineal generalizado como una suma de funciones más generales de cada variable:

$$\eta(x_i) = \sum_{j=1}^p f_j(x_{ij}),$$

donde f_j son funciones desconocidas, se supone que son suaves o de baja complejidad.

| Hastie, T. and Tibshirani, R. (1986). Generalized additive models. Statistical science 297–310.

Modelo GAM

Generalized Additive Model

$$y_i \sim \text{EF}(\mu_i, \phi)$$

donde:

$$g(\mu_i) = \mathbf{A}_i \gamma + \sum_j f_j(x_{ji}),$$

y_i es una variable de respuesta univariada, $\text{EF}(\mu_i, \phi)$ denota una distribución de **familia exponencial** con media μ_i y parámetro de escala ϕ , \mathbf{A}_i es la i -ésima fila de una matriz de modelo paramétrico, γ son parámetros de regresión, f_j son funciones suaves a estimar, y x_j es una covariable (normalmente, pero no necesariamente, univariante).

- Se relaja el supuesto de relación lineal entre variable dependiente y predictor
- Relación entre predictores individuales y dependientes (posiblemente transformados)
- La variable se estima mediante una función suave no lineal:

$$g(y) = s(x1) + s(x2, x3) + \beta_4 x_4 + \dots$$

El término $f_j(x_j)$ podrían ser regresiones tipo kernel o splines, pero algún término lineal en funciones de x_j . comp:

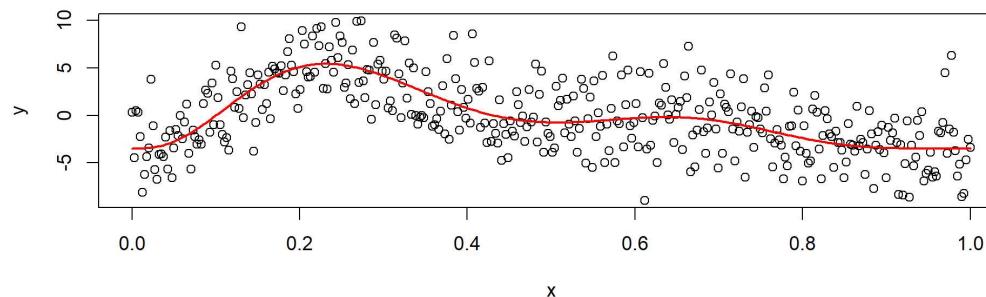
- polinomios creados por poly ();
- splines de regresión creados por bs ();
- splines cúbicos naturales creados por ns ();
- funciones de paso creadas por cut () o
- variables categóricas (ficticias) creadas por factor

Ejemplo Ilustrativo

Suponga una función

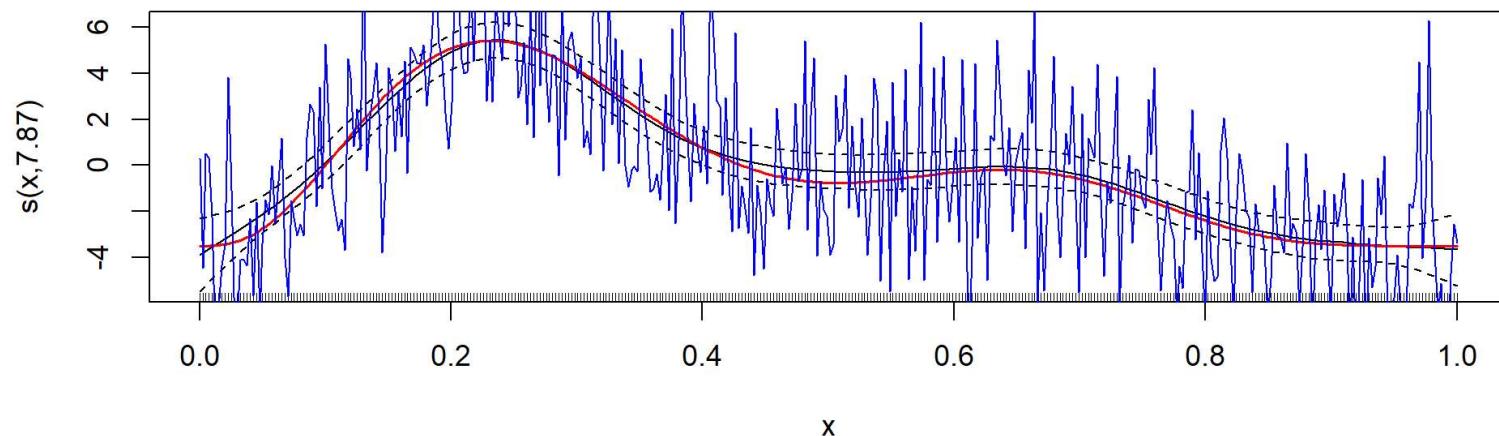
$$f(x) = -3.5 + 0.2x^{11}(10(1-x))^6 + 10(10x)^3(1-x)^{10}$$

```
set.seed(0);n <- 400; x <- 0:(n-1) / (n-1)
f <- -3.5+0.2*x^11*(10*(1-x))^6+10*(10*x)^3*(1-x)^10
y <- f + rnorm(n, 0, sd = 3)
plot(x,y);lines(x, f, col="red", lwd=2)
```



Estimación de la curva con GAM y splines

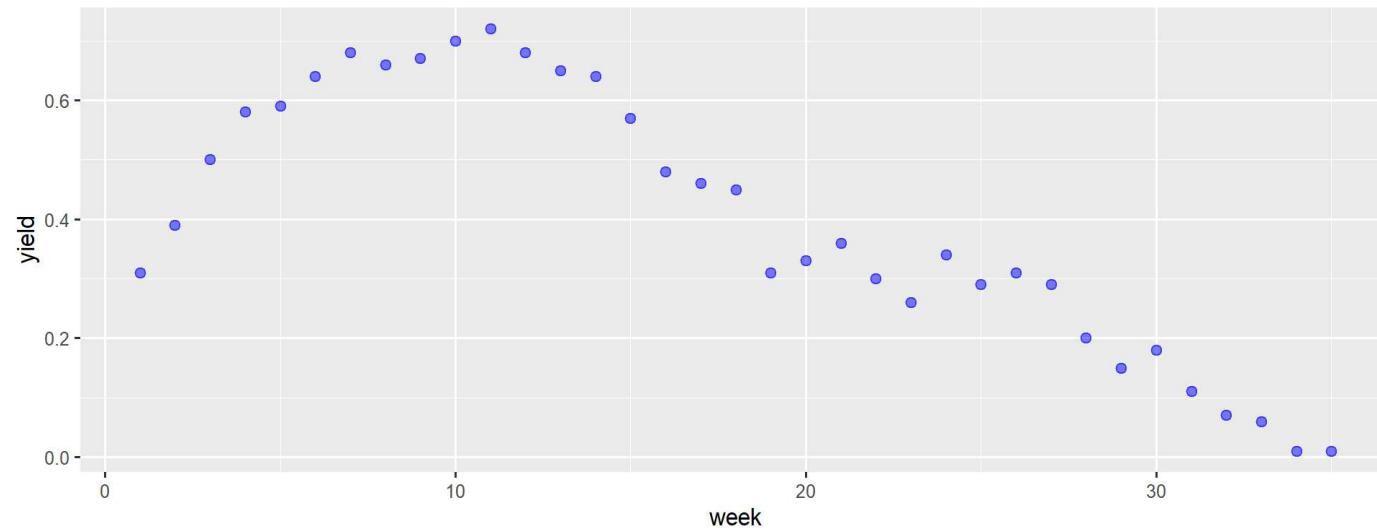
```
library(mgcv)
fit.gam <- gam(y~ s(x))
plot(fit.gam)
lines(x, f, col="red", lwd=1.5)
lines(x, y, col="blue")
```



Ejemplo: Curva de lactancia de una vaca

Rendimientos medios diarios de grasa ($kg/día$) de la leche de una sola vaca durante cada 35 semanas.

```
library(tidyverse)
library(agridat)
data(henderson.milkfat)
ggplot(henderson.milkfat, aes(x = week)) +
  geom_point(aes(y = yield), col="blue", size = 2, alpha = 0.5)
```



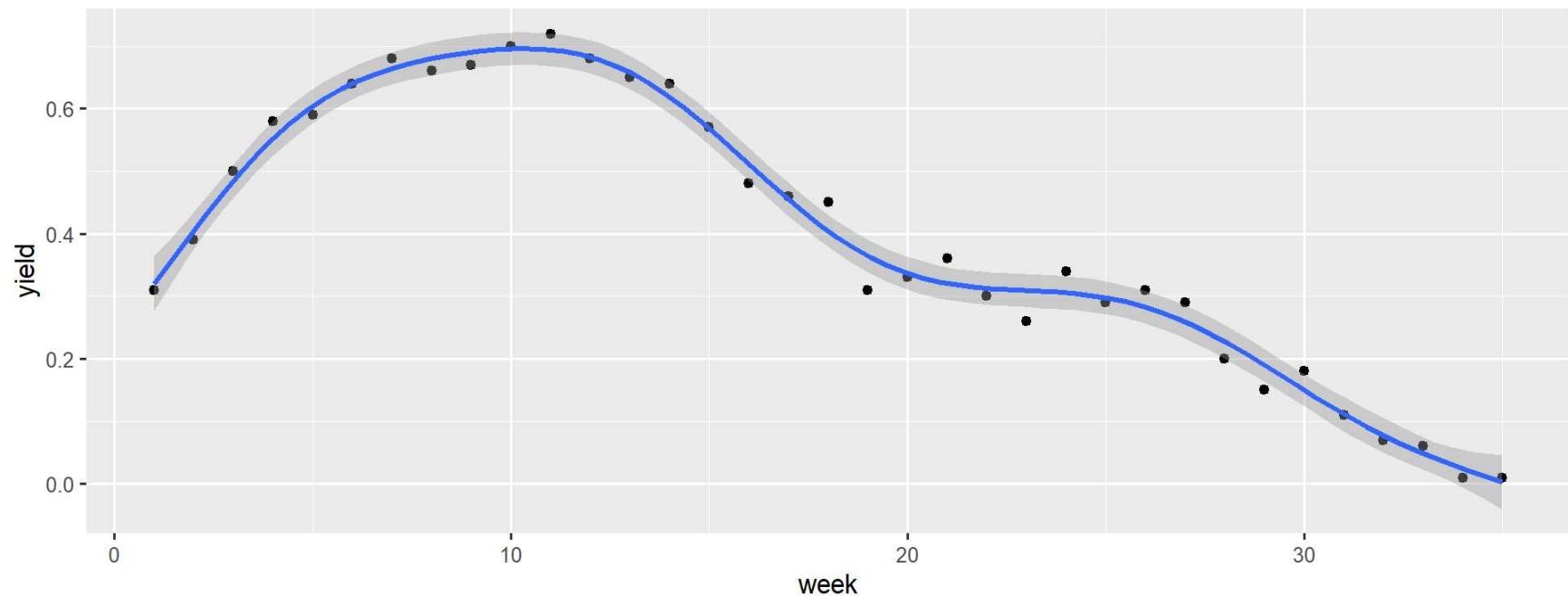
Ajuste GAM

```
dat=henderson.milkfat
libs("mgcv")
modelo_milk<- gam(yield ~ s(week), data=dat)
summary(modelo_milk)

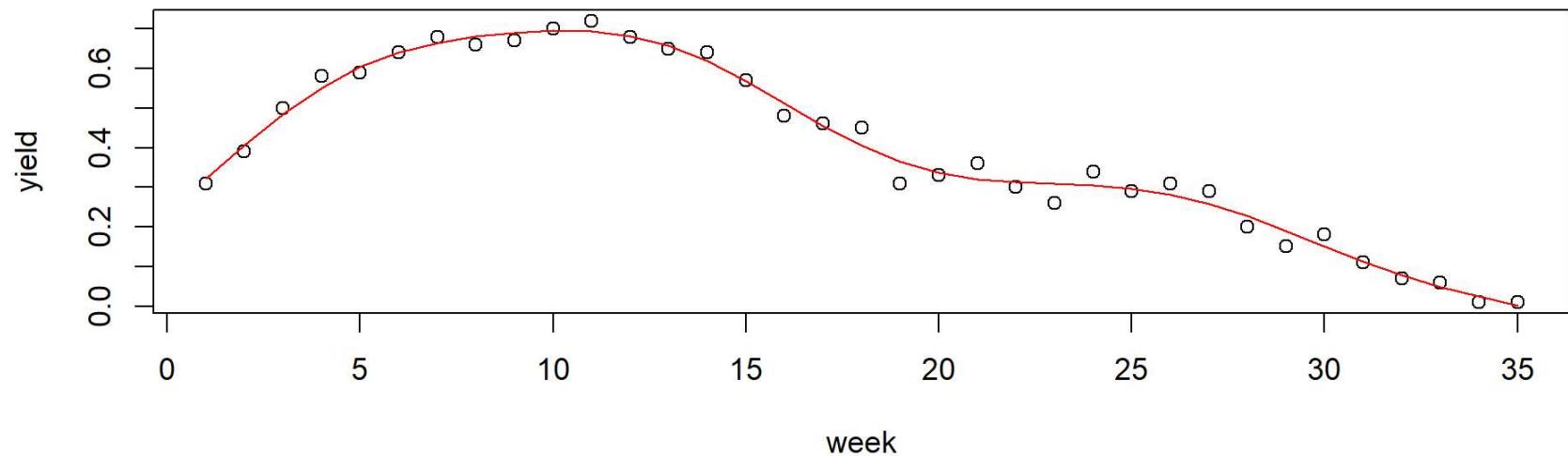
##
## Family: gaussian
## Link function: identity
##
## Formula:
## yield ~ s(week)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.398571  0.004763   83.69  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##             edf Ref.df      F p-value
## s(week) 7.79  8.623 235.5  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Curva ajustada con GAM

```
ggplot(henderson.milkfat, aes(x = week,y=yield)) +  
  geom_point() +  
  geom_smooth(method = "gam", formula = y ~ s(x))
```



```
plot(yield ~ week, data = dat)
lines(dat$week, predict(modelo_milk), lty = 1,col="red")
```



Curva ajustada con GAM

```
library(broom)
libs("mgcv")
modelo_milk<- gam(yield ~ s(week), data=dat)
knitr::kable(tidy(modelo_milk))
```

term	edf	ref.df	statistic	p.value
s(week)	7.790113	8.622723	235.4643	0

```
knitr::kable(glance(modelo_milk))
```

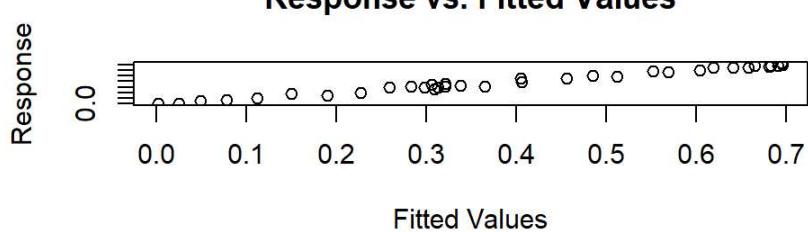
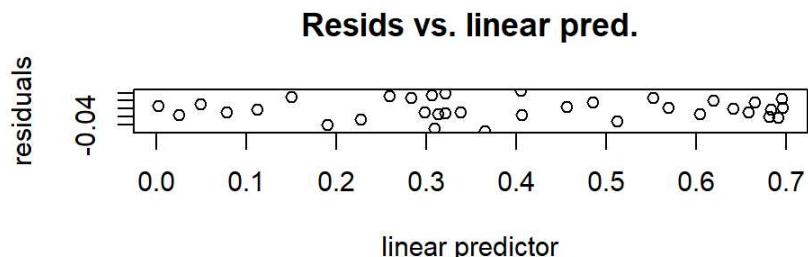
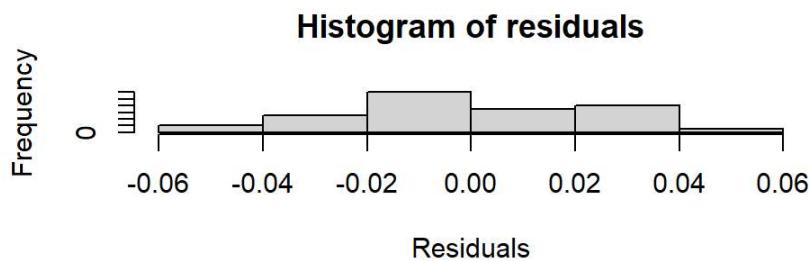
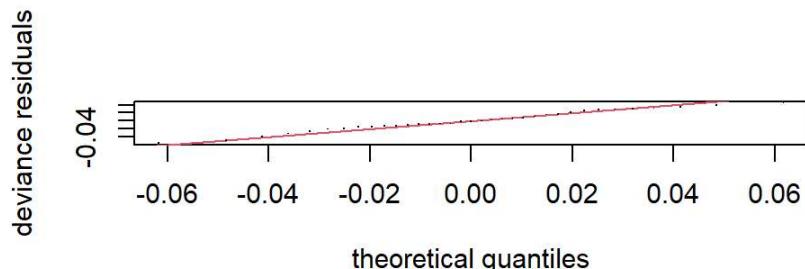
df	logLik	AIC	BIC	deviance	df.residual	nobs
8.790113	80.32304	-141.0659	-125.8388	0.020808	26.20989	35

```
knitr::kable(performance::performance(modelo_milk))
```

AIC	BIC	R2	RMSE	Sigma
-----	-----	----	------	-------

Diagnóstico GAM

```
libs("mgcv")
modelo_milk<- gam(yield ~ s(week), data=dat)
gam.check(modelo_milk)
```



Ejemplo: Biomasa

Datos:

Allometric Equations for Aboveground and Belowground Biomass Estimations in an Evergreen Forest in Vietnam
Nam VT, van Kuijk M, Anten NPR (2016) Allometric Equations for Aboveground and Belowground Biomass Estimations in an Evergreen Forest in Vietnam. PLOS ONE 11(6): e0156827.
<https://doi.org/10.1371/journal.pone.0156827>

Ajuste de curva alometrica para la especie *Endospermum chinensis*

Datos:

```
biomasa=read.csv("biomasa.csv")
biomasa
```

```
##      N_individual WD.class      Species_names DBH_cm   H_m WD_gcm3   AGB_kg
## 1            17       I Endospermum chinensis    3.2    6.0     0.45     1.46
## 2            18       I Endospermum chinensis    3.6    5.3     0.45     1.76
## 3            19       I Endospermum chinensis   8.3 12.4     0.45    13.96
## 4            20       I Endospermum chinensis   8.9 10.2     0.45    18.61
```

Modelos alométricos

Modelos alométricos que relacionan las medidas geométricas (DAP , H y WD) con la **biomasa aérea (AGB)**

Several equations are commonly used to develop allometric models for AGB and RB.

By Brown et al. [26] and Brown [1] (pan-tropical):

$$\ln(B) = a + b \ln(DBH) \quad (1)$$

$$\ln(B) = a + b \ln(DBH) + b_1 \ln(DBH)^2 \quad (2)$$

By Nelson et al. [27] (central Amazon):

$$\ln(B) = a + b \ln(DBH) + d \ln(H) \quad (3)$$

By Chave et al. [3] (pan-tropical):

$$\ln(B) = a + b \ln(DBH) + c \ln(WD) + d \ln(H) \quad (4)$$

$$\ln(B) = a + b \ln(DBH) + e(\ln(DBH))^2 + f(\ln(DBH))^3 + c \ln(WD) \quad (5)$$

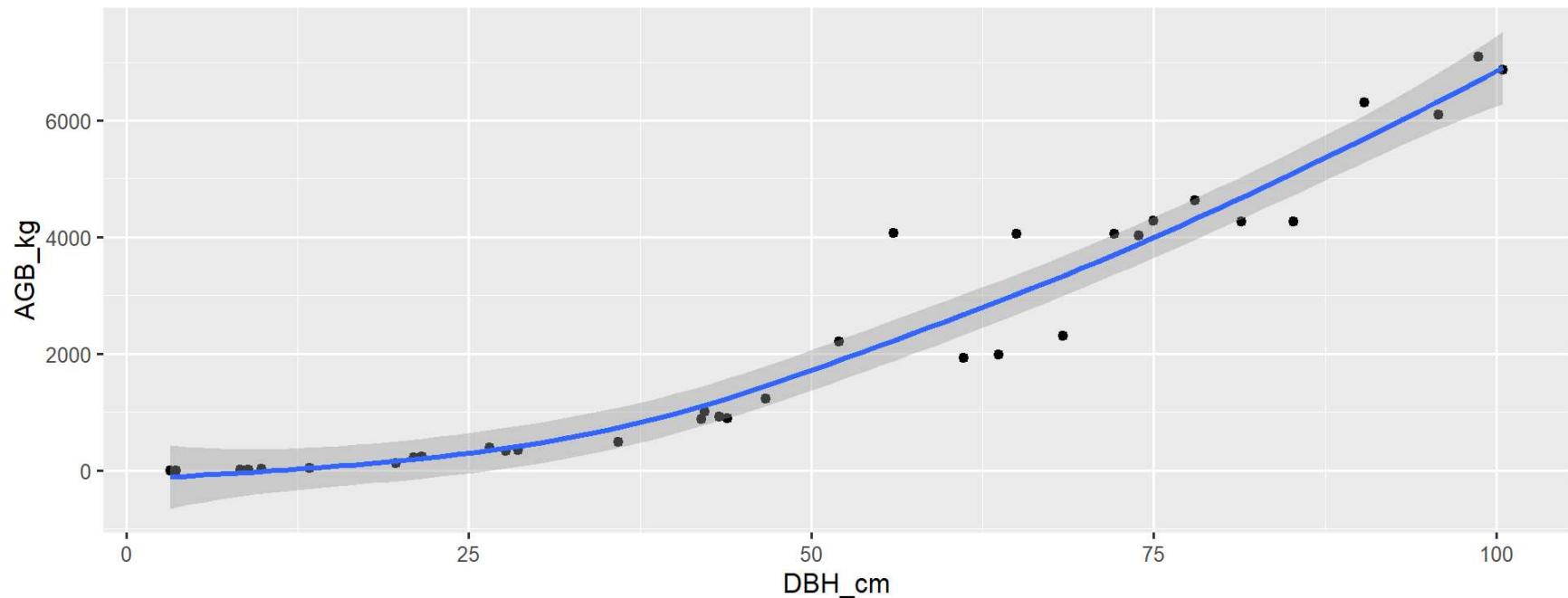
$$\ln(B) = a + g \ln(DBH^2 HWD) \quad (6)$$

By Djomo et al. [28] (tropical Africa):

$$\ln(B) = a + b \ln(DBH) + c \ln(WD)$$

Curva alometrica para la especie *Endospermum chinensis* con GAM

```
ggplot(biomasa, aes(x = DBH_cm,y=AGB_kg)) +  
  geom_point() +  
  geom_smooth(method = "gam", formula = y ~ s(x))
```



Ejemplo: Datos de propiedades del suelo

Varias propiedades químicas del suelo medidas en una cuadrícula regular con 10x25 puntos espaciados por 5 metros. 250 observaciones sobre las siguientes 22 variables:

- Coordenada x de Linha
- Coordenada y de Coluna
- Elevación de la cota
- AGrossa un vector numérico, una porción de arena de la muestra.
- Silte un vector numérico, porción de limo de la muestra.
- Argila un vector numérico, una porción de arena de la muestra.
- pHAgua un vector numérico, pH del suelo en agua
- pHKCl un vector numérico, pH del suelo por KCl
- Ca un vector numérico, contenido de calcio
- Mg a vector numérico, contenido de magnesio
- K un vector numérico, contenido de potasio
- Al un vector numérico, contenido de aluminio
- H un vector numérico, contenido de hidrógeno

Ejemplo: Datos de propiedades del suelo

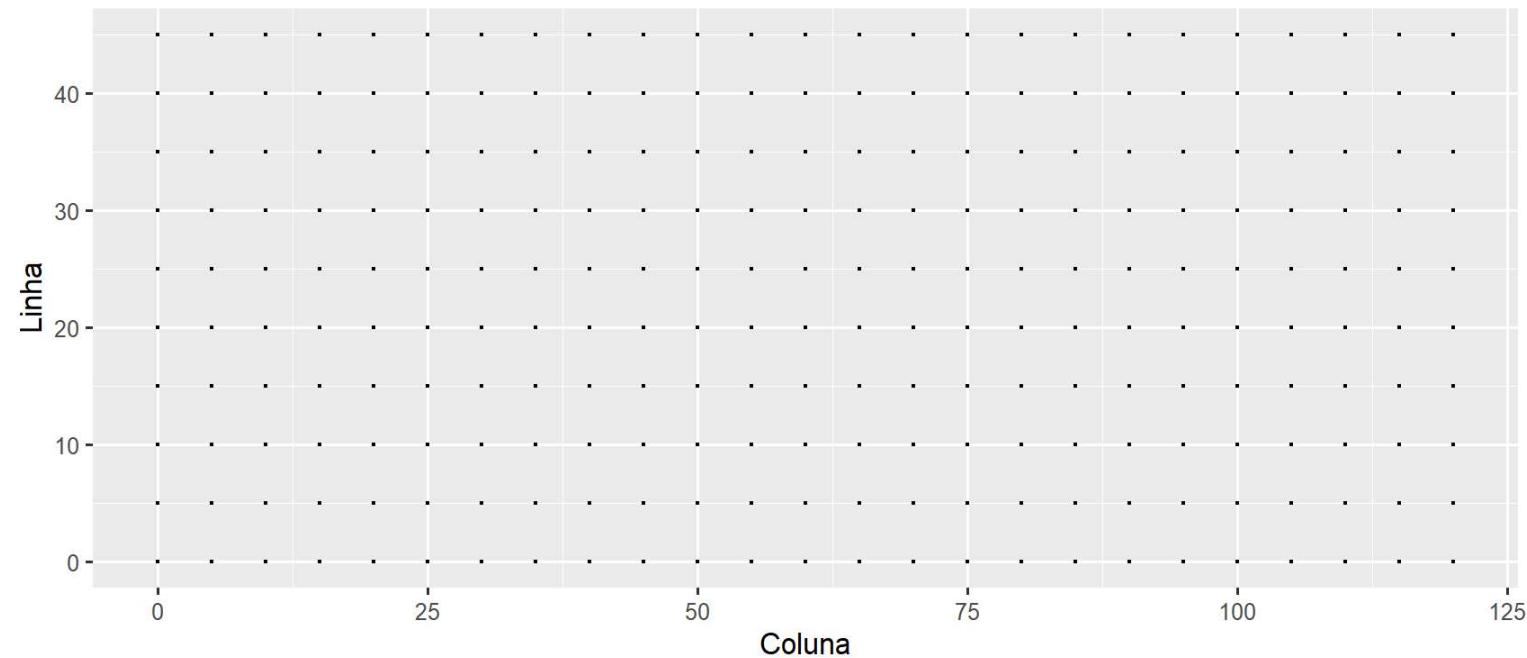
Varias propiedades químicas del suelo medidas en una cuadrícula regular con 10x25 puntos espaciados por 5 metros. 250 observaciones sobre las siguientes 22 variables:

cont...

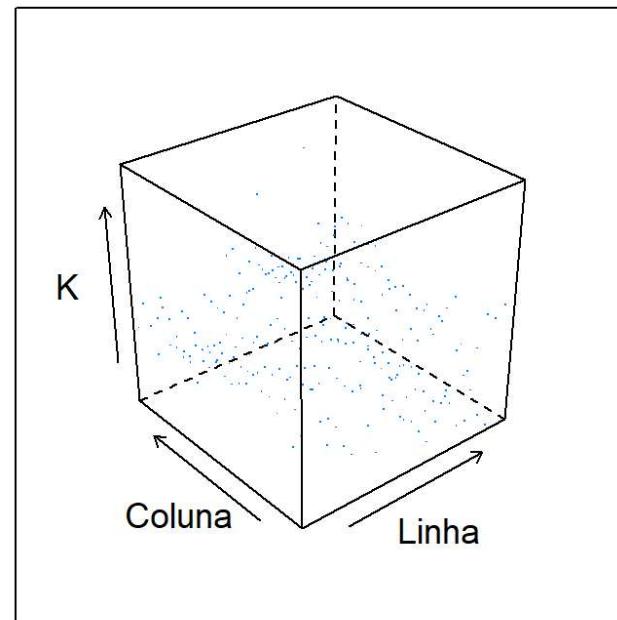
- C un vector numérico, contenido de carbono
- N un vector numérico, contenido de nitrógeno
- CTC un vector numérico, capacidad de intercambio de cationico
- S un vector numérico, contenido enxofrar
- V un vector numérico
- M un vector numérico
- NC un vector numérico
- CEC un vector numérico
- CN un vector numérico, relación carbono / nitrógeno _

Sitios de muestreo del suelo

```
soil250=read.csv("soil250.csv")
ggplot(soil250, aes(Coluna, Linha)) +
  geom_point(size = .25, show.legend = FALSE) +
  coord_quickmap()
```



```
library(lattice)
p <- cloud(K ~ Linha * Coluna, pch = ".", data = soil250)
print(p)
```



Grafica de distribución del Potasio.

```
lat=soil250$Linha  
long=soil250$Coluna  
K=soil250$K  
library(scatterplot3d)  
scatterplot3d(lat,long,K, pch=16, highlight.3d=TRUE,  
             type="h", main="3D Scatterplot")
```

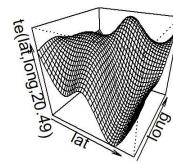
```

library(mgcv)
Ajuste_K <- gam(K~te(lat, long))
summary(Ajuste_K)

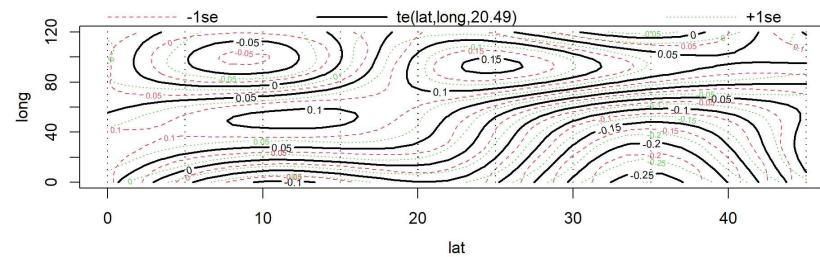
##
## Family: gaussian
## Link function: identity
##
## Formula:
## K ~ te(lat, long)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.469920  0.004897   95.96  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##             edf Ref.df      F p-value
## te(lat,long) 20.49  22.82 15.28  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.579  Deviance explained = 61.4%
## GCV = 0.0065595  Scale est. = 0.0059956 n = 250

```

```
library(mgcv)
Ajuste_K <- gam(K~te(lat, long))
plot(Ajuste_K, pers = TRUE)
```



```
plot(Ajuste_K)
```

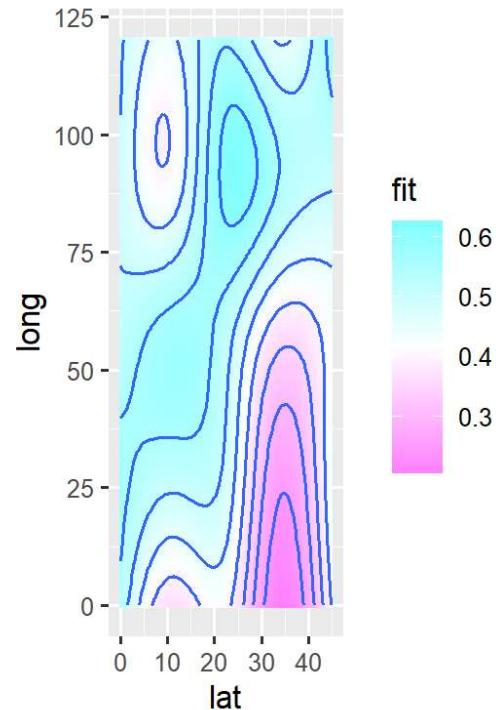


```

numero_div <- 100
lat=seq(min(lat),max(lat),length=numero_div)
long=seq(min(long), max(long),length=numero_div)
datos_malla=expand.grid(long = long, lat=lat)
# predicción de S04 en la malla
K_pred =matrix(predict(Ajuste_K, datos_malla),numero_div,numero_div)
#head(K_pred,3)
p <- persp(lat, long, K_pred, theta = 10, col = "yellow")
library(plot3D)
# Añadir observaciones de K
obs <- trans3d(lat, long, K, p)
pred <- trans3d(lat, long, fitted(Ajuste_K), p)
points(obs, col = "red", pch = 16)

```

```
library(ggplot2)
p <- ggplot(mutate(datos_malla, fit = as.numeric(K_pred)),
aes(x = lat, y = long, z = fit)) +
coord_fixed()
#p + geom_contour()
p + geom_tile(aes(fill = fit)) + geom_contour() +
scale_fill_gradientn(colors = rev(cm.colors(100)))
```

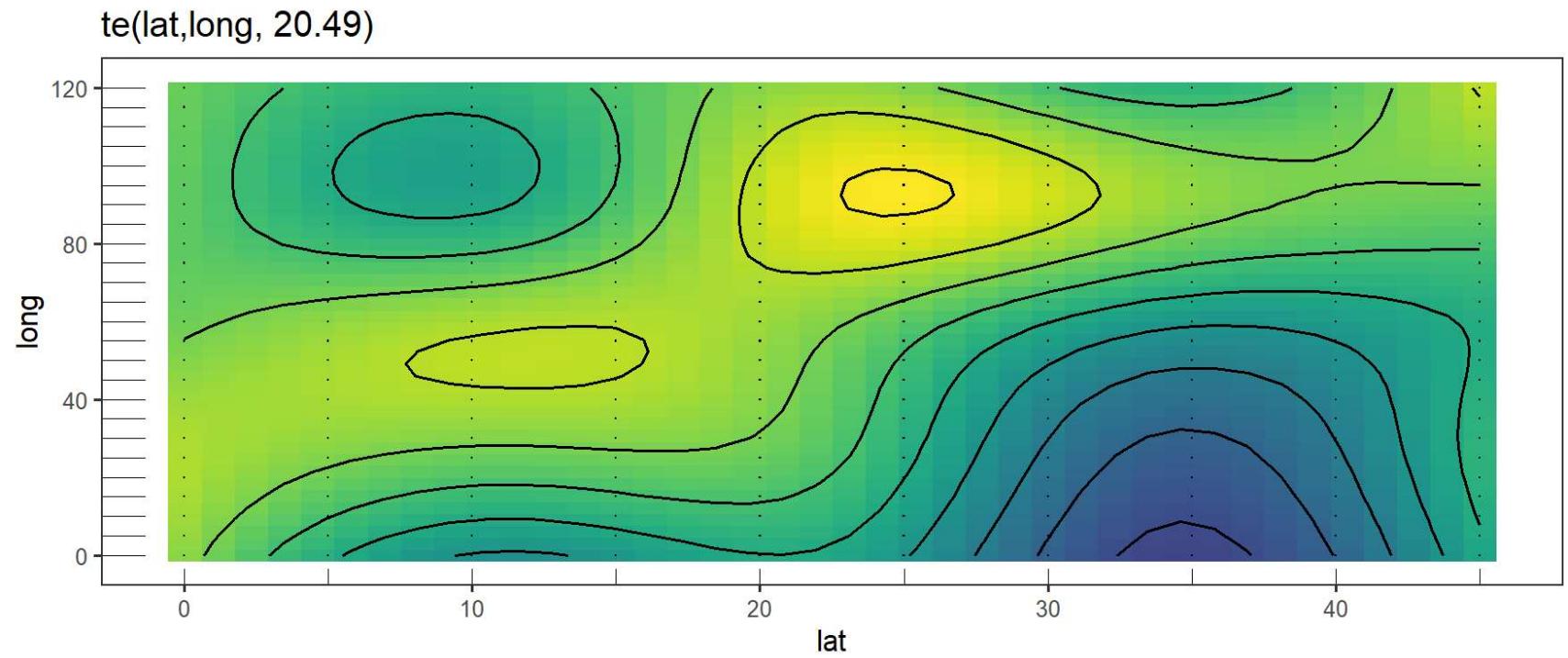


```

library(rgl)
#plot3d(Ajuste_K)
plot3d(Ajuste_K, image = TRUE, contour = TRUE)

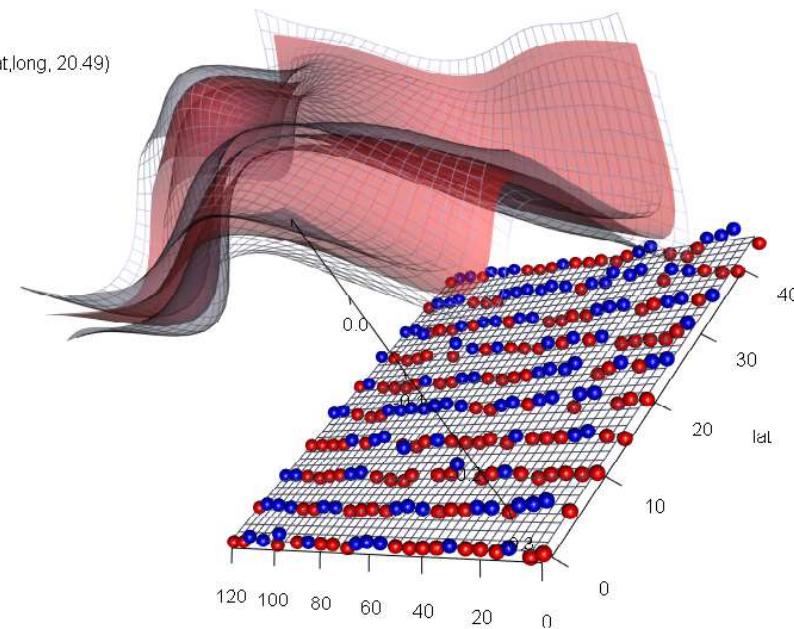
library(mgcViz)
b <- getViz(Ajuste_K)
plot(sm(b, 1))+l_fitRaster() + l_fitContour() + l_points() + l_rug()

```



Grafica ajuste GAM Potasio

```
library(knitr)
include_graphics("potasio1.png")
```



Rendimiento de Maiz en Mexico por Hectarea

Datos: Datos de FAO <http://www.fao.org/faostat/en/#data/QC>

Year	Rend_maiz	Year	Rend_maiz
1961	9934	1989	16929
1962	9946	1990	19942
1963	9867	1991	20515
1964	11332	1992	23450
1965	11578	1993	24401
1966	11188	1994	22255
1967	11304	1995	22883
1968	11806	1996	22387
1969	11840	1997	23840
1970	11935	1998	23429
1971	12723	1999	24720

Rendimiento de Maiz en Mexico Hg/Hectarea

```
maiz.rend <- read.csv("D:/cursos/no param/noparam/ejemplo gam/maiz-rend.csv")
rend_mex <- ts(maiz.rend$Rend_maiz,start=c(1961),end=c(2018),frequency=1)
plot(rend_mex,col="blue",lwd=2,main="Rendimiento por Ha de Maiz en Mexico 1961-1980")
```



Ajuste con GAM

```
gam_Maiz <- gam(Rend_maiz~ s(Year,bs="cr",k=55),
                  data =maiz.rend)
summary(gam_Maiz)

##
## Family: gaussian
## Link function: identity
##
## Formula:
## Rend_maiz ~ s(Year, bs = "cr", k = 55)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 20984.7      81.3    258.1   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##             edf Ref.df     F p-value
## s(Year) 23.61 29.03 350.9   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Prediccion

```
library(ggplot2)
layout(matrix(1:2, nrow = 1))
predic=data.frame(cbind(predic_rend=predict(gam_Maiz),maiz.rend))
ggplot(predic, aes(x=Year)) +
  geom_line(aes(y = Rend_maiz),color ="blue") +
  geom_line(aes(y = predic_rend), color="red", linetype="twodash")
```

prediccion rendimiento de Maiz Hg/Ha 2019-2020

```
a_2019_2020=data.frame(Year=c(2019,2020))
p_2019_2020=predict(gam_Maiz,a_2019_2020)
data.frame(cbind(a_2019_2020,p_2019_2020))

##    Year p_2019_2020
## 1 2019   38975.02
## 2 2020   39542.21

pred <- predict.gam(gam_Maiz,a_2019_2020 ,se = TRUE )$fit
se <- predict.gam(gam_Maiz,a_2019_2020 ,se = TRUE )$se.fit
# Intervalos de confianza
lcl_95 <- pred - 1.96 * se
ucl_95 <- pred + 1.96 * se
data.frame(cbind(a_2019_2020,p_2019_2020,lcl_95,ucl_95))

##    Year p_2019_2020 lcl_95 ucl_95
## 1 2019   38975.02 36801.44 41148.59
## 2 2020   39542.21 36164.68 42919.73
```

Ejemplo 2

Los GAM son similares a los modelos lineales generalizados (GLM), pero difieren al relajar el supuesto lineal, lo que potencialmente revela relaciones no lineales y una estructura importante en estos datos que de otro modo se perderían. En este sentido, un GAM puede mostrar relaciones lineales, monótonas o más complejas, dependiendo de la forma en que cada variable responda a los cambios en las variables dependientes. El GAM funciona de esta manera al extender un GLM para incluir una función de base de suavizado que puede medir relaciones arbitrariamente no paramétricas. Los datos categóricos se tratan como un término lineal sin suavizado. Este enfoque semiparamétrico hace que los GAM sean muy flexibles y se adapten fácilmente a diferentes tipos de datos.

Datos

Con el set de datos **Auto** (información sobre vehículos), trataremos de predecir mpg (consumo de combustible, en millas/galón) en función de la variable predictora displacement (desplazamiento del motor, en pulgadas cúbicas), Junto con la variable horsepower (potencia del motor).

Datos de Autos

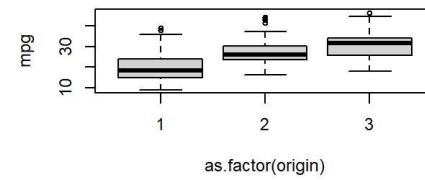
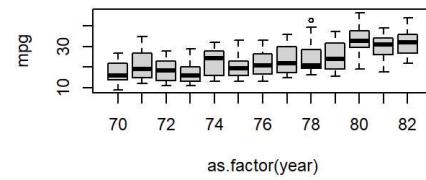
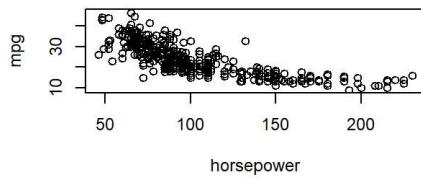
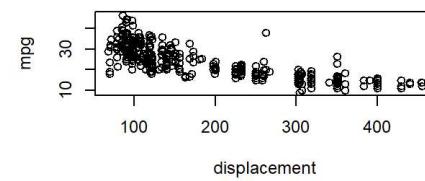
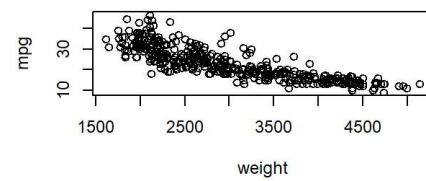
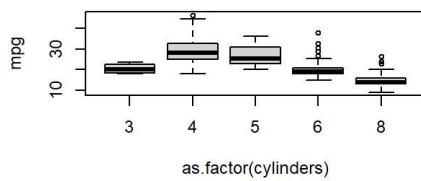
```
library(ISLR)
knitr::kable(head(Auto, 10))
```

mpg	cylinders	displacement	horsepower	weight	acceleration	year	origin	name
18	8	307	130	3504	12.0	70	1	chevrolet chevelle malibu
15	8	350	165	3693	11.5	70	1	buick skylark 320
18	8	318	150	3436	11.0	70	1	plymouth satellite
16	8	304	150	3433	12.0	70	1	amc rebel sst
17	8	302	140	3449	10.5	70	1	ford torino
15	8	429	198	4341	10.0	70	1	ford galaxie 500
14	8	454	220	4354	9.0	70	1	chevrolet impala
14	8	440	215	4312	8.5	70	1	plymouth fury iii
14	8	455	225	4425	10.0	70	1	pontiac catalina
15	8	390	190	3850	8.5	70	1	amc ambassador dpl

```

par(mfrow=c(2,3))
attach(Auto)
plot(mpg ~ as.factor(cylinders))
plot(mpg ~ weight)
plot(mpg ~ displacement)
plot(mpg ~ horsepower)
plot(mpg ~ as.factor(year))
plot(mpg ~ as.factor(origin))

```



```

library(caret)
set.seed(12345)
inTraining <- createDataPartition(Auto$name, p=.6, list = FALSE)
datosA.train <- Auto[inTraining,]
datosA.test <- Auto[-inTraining,]
datosA.test

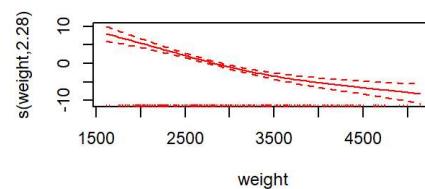
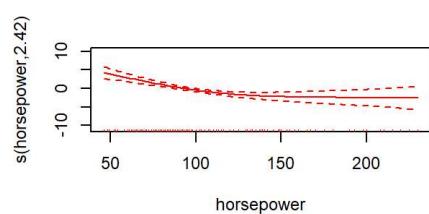
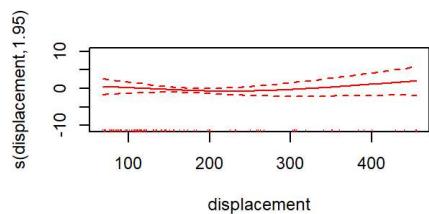
```

	##	mpg	cylinders	displacement	horsepower	weight	acceleration	year	origin
##	9	14.0	8	455	225	4425	10.0	70	1
##	16	22.0	6	198	95	2833	15.5	70	1
##	25	21.0	6	199	90	2648	15.0	70	1
##	38	18.0	6	232	100	3288	15.5	71	1
##	61	20.0	4	140	90	2408	19.5	72	1
##	63	13.0	8	350	165	4274	12.0	72	1
##	65	15.0	8	318	150	4135	13.5	72	1
##	66	14.0	8	351	153	4129	13.0	72	1
##	92	13.0	8	400	150	4464	12.0	73	1
##	128	19.0	6	232	100	2901	16.0	74	1
##	129	15.0	6	250	100	3336	17.0	74	1
##	131	26.0	4	122	80	2451	16.5	74	1
##	168	29.0	4	97	75	2171	16.0	75	3
##	172	24.0	4	134	96	2702	13.5	75	3
##	175	18.0	6	171	97	2984	14.5	75	1
##	186	26.0	4	98	79	2255	17.7	76	1
##	190	15.5	8	304	120	3962	13.9	76	1
##	191	14.5	8	351	152	4215	12.8	76	1
##	194	24.0	6	200	81	3012	17.6	76	1

```

library(leaps)
library(mgcv)
# Modelo GAM
modelo.gam <- gam(mpg ~ cylinders + year + origin + s(displacement) + s(horsepower) + s(weight), dat
par(mfrow = c(2, 3))
plot(modelo.gam, se = T, col = "red")

```



```

summary(modelo.gam)

##
## Family: gaussian
## Link function: identity
##
## Formula:
## mpg ~ cylinders + year + origin + s(displacement) + s(horsepower) +
##      s(weight)
##
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -38.48255   4.06060  -9.477 <2e-16 ***
## cylinders    0.23970   0.36425   0.658  0.5109
## year         0.78511   0.04694  16.725 <2e-16 ***
## origin        0.64628   0.27448   2.355  0.0191 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##             edf Ref.df     F p-value
## s(displacement) 1.948  2.484  2.255  0.0849 .
## s(horsepower)    2.421  3.076  9.984 2.45e-06 ***
## s(weight)        2.281  2.904 28.129  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##

```

Modelos aditivos generalizados para posición, escala y forma (GAMLSS)

Los modelos lineales generalizados (GLM) y los modelos generalizados aditivos (GAM) asumen que la variable respuesta Y sigue una distribución de la familia exponencial, cuya media μ puede ser modelada en función de otras variables (predictores) y cuya varianza σ se calcula mediante una constante de dispersión y una función $\nu(\mu)$. Esto último significa que, la varianza, skewness y kurtosis, no se modelan directamente en función de las variables predictoras sino de forma indirecta a través de su relación con la media.

Los modelos GAMLSS, introducidos por Rigby y Stasinopoulos en 2005, permiten, además de incorporar distribuciones que no son de la familia exponencial, modelar explícitamente cada uno de sus parámetros en función de las variables predictoras empleando funciones lineales y no lineales.

Rigby, R.A. and Stasinopoulos, D.M. (2005), Generalized additive models for location, scale and shape. Journal of the Royal Statistical Society: Series C (Applied Statistics), 54: 507-554.
<https://doi.org/10.1111/j.1467-9876.2005.00510.x>

Modelos aditivos generalizados para posición, escala y forma (GAMLSS)

Los términos empleados dentro del marco de los GAMLSS para referirse a los parámetros de localización, escala y forma son (μ, σ, ν, τ) .

$$\mathbf{Y} \sim D(\mu, \sigma, \nu, \tau), \text{ con } \mathbf{Y} = \mathbf{X}^T \boldsymbol{\beta}$$

$$\eta_1 = g_1(\mu) = \mathbf{X}^T \boldsymbol{\beta} + f_1(x_1) + f_2(x_2) + \dots + f_p(x_p)$$

$$\eta_2 = g_2(\sigma) = \mathbf{X}^T \boldsymbol{\beta} + f_1(x_1) + f_2(x_2) + \dots + f_p(x_p)$$

$$\eta_3 = g_3(\nu) = \mathbf{X}^T \boldsymbol{\beta} + f_1(x_1) + f_2(x_2) + \dots + f_p(x_p)$$

$$\eta_4 = g_4(\tau) = \mathbf{X}^T \boldsymbol{\beta} + f_1(x_1) + f_2(x_2) + \dots + f_p(x_p)$$

donde $\mathbf{Y} \sim D(\mu, \sigma, \nu, \tau)$ es la distribución de la variable respuesta (pueden ser menos parámetros), \mathbf{X} contiene los términos lineales del modelo, $\boldsymbol{\beta}$ son los coeficientes lineales y $f_i(x_i)$ son funciones de suavizado no lineales (smooth) de cada predictor.

GAMLSS

Los GAMLSS son una forma de superar las limitaciones de los modelos GLM y GAM. Como resultado del modelo, se consigue caracterizar la distribución completa, permitiendo generar intervalos probabilísticos y predicción de cuantiles.

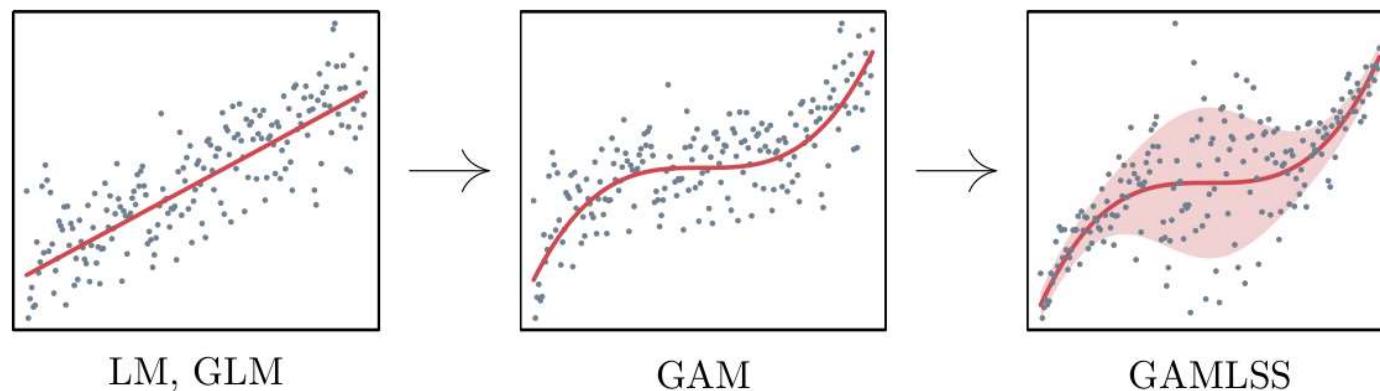


Imagen de: Schlosser, Lisa & Hothorn, Torsten & Stauffer, Reto & Zeileis, Achim. (2018). *Distributional Regression Forests for Probabilistic Precipitation Forecasting in Complex Terrain*. *The Annals of Applied Statistics*. 13. 10.1214/19-AOAS1247.

GAMLSS

Suponiendo una variable dependiente de una distribución con parámetros $\theta = (\theta_1, \dots, \theta_L)'$ y observaciones y_1, \dots, y_n , dadas las covariables z_1, \dots, z_K y x_1, \dots, x_Q , a *gamlss* se pueden describir con la siguiente especificación de modelo:

$$g_l(\theta_l) = \eta_l = X_l\beta_l + \sum_{j=1}^{K_l} Z_{jl}\gamma_{jl}$$

En la Ecuación, $g_l(\cdot)$ representa una función de enlace monótona conocida, que puede ser diferente para cada parámetro. X_l representa el subconjunto de todas las variables disponibles $x = (x_1, \dots, x_Q)'$ usado para modelar el parámetro θ_l , mientras que $Z_{jl}\gamma_{jl}$ sirve como la matriz de función básica para un efecto no paramétrico de la covariante z_j en el parámetro θ_l , tomado de un subconjunto de variables z_1, \dots, z_K . El subconjunto específico de covariantes z con efectos no paramétricos en el parámetro θ_l tiene una longitud de K_l variables.

Ajuste de los modelos GAMLSS

Los modelos GAMLSS se ajustan mediante una adaptación del algoritmo backfitting, un algoritmo típicamente empleado para el ajuste de modelos aditivos. Por ejemplo, para el modelo aditivo

$$g(\mu) = \beta_0 + f_1(x_1) + f_2(x_2) + \dots + f_p(x_p)$$

en términos generales, el algoritmo de ajuste por *backfitting* funciona de la siguiente forma:

- Se inicia con un valor para todos los términos f_i del modelo, por ejemplo poniéndolos todos a cero.
- Se estima el valor del primer término f_1 empleando los datos de entrenamiento, con el objetivo de predecir lo mejor posible la variable respuesta y .
- Se estima el valor del segundo término f_2 empleando residuos $y - f_1(x_1)$.
- Se repite el paso 3, ajustando cada término con los residuos del anterior.

Tras ajustar todos los términos, el valor del primer término se descarta y se reajusta empleando el residuo de todos los demás términos.

Repetir todos los pasos del 3 al 5 hasta que se alcance un criterio de parada (que los valores apenas cambien o que se alcance un número máximo de iteraciones).

Selección de modelos GAMLSS

Para identificar el mejor modelo GAMLSS es necesario comparar diferentes modelos candidatos, cada uno con una combinación de distribución para la variable respuesta, función link para cada uno de los parámetros, predictores e hiperparámetros. Existen varias estrategias para la selección del mejor modelo de entre los comparados:

- **Generalized Akaike information criterion (GAIC)**: esta métrica emplea el log likelihood del modelo multiplicado por -2 y añade una penalización por cada parámetro que incluye el modelo. Más detalles sobre el GAIC en el apartado métricas de ajuste.
- **Generalized cross-validation y cross-validation**

El segundo método es más recomendable, aunque supone mayor costo computacional.

GAMLSS

Por ejemplo, al asumir la distribución gaussiana para la variable dependiente y conectar μ a efectos paramétricos x_q usando la función de enlace de identidad ($g(\mu) = \mu$) y el parámetro de varianza σ^2 a una constante, llegamos a una especificación de modelo lineal.

Datos de Wage

collected by the United States Census Bureau (2011), includes 3000 male individuals living in the Mid-Atlantic region of the United States of America with records of the following variables:

- wage: Worker's raw wage (in \$1000)
- age: Age of worker
- year: Year that wage information was recorded
- race: A factor with levels 1. White, 2. Black, 3. Asian and 4. Other
- education: A factor with levels 1. < HS Grad, 2. HS Grad, 3. Some College, 4. College Grad and 5. Advanced Degree
- health: A factor with levels 1. <= Good and 2. >= Very Good, indicating the health level of worker.

Datos de Autos

```
library(ISLR)
knitr::kable(head(Wage))
```

	year	age	maritl	race	education	region	jobclass	health	health_ins	logwage	wage
231655	2006	18	1. Never Married	1. White	1. < HS Grad	2. Middle Atlantic	1. Industrial	1. <=Good	2. No	4.318063	75.04315
86582	2004	24	1. Never Married	1. White	4. College Grad	2. Middle Atlantic	2. Information	2. >=Very Good	2. No	4.255273	70.47602
161300	2003	45	2. Married	1. White	3. Some College	2. Middle Atlantic	1. Industrial	1. <=Good	1. Yes	4.875061	130.98218
155159	2003	43	2. Married	3. Asian	4. College Grad	2. Middle Atlantic	2. Information	2. >=Very Good	1. Yes	5.041393	154.68529

Datos Wage

```
library(gamlss)

modelo_gamlss <- gamlss(
  formula = wage ~ bs(age) + race + year + education + health,
  sigma.formula = ~ bs(age) + race + year + education + health,
  family = NO, data = Wage,)

## GAMLSS-RS iteration 1: Global Deviance = 29292.76
## GAMLSS-RS iteration 2: Global Deviance = 29277.64
## GAMLSS-RS iteration 3: Global Deviance = 29277.57
## GAMLSS-RS iteration 4: Global Deviance = 29277.57

summary(modelo_gamlss)

## ****
## Family: c("NO", "Normal")
##
## Call: gamlss(formula = wage ~ bs(age) + race + year + education + health,
##   sigma.formula = ~bs(age) + race + year + education + health,
##   family = NO, data = Wage)
##
## Fitting method: RS()
```

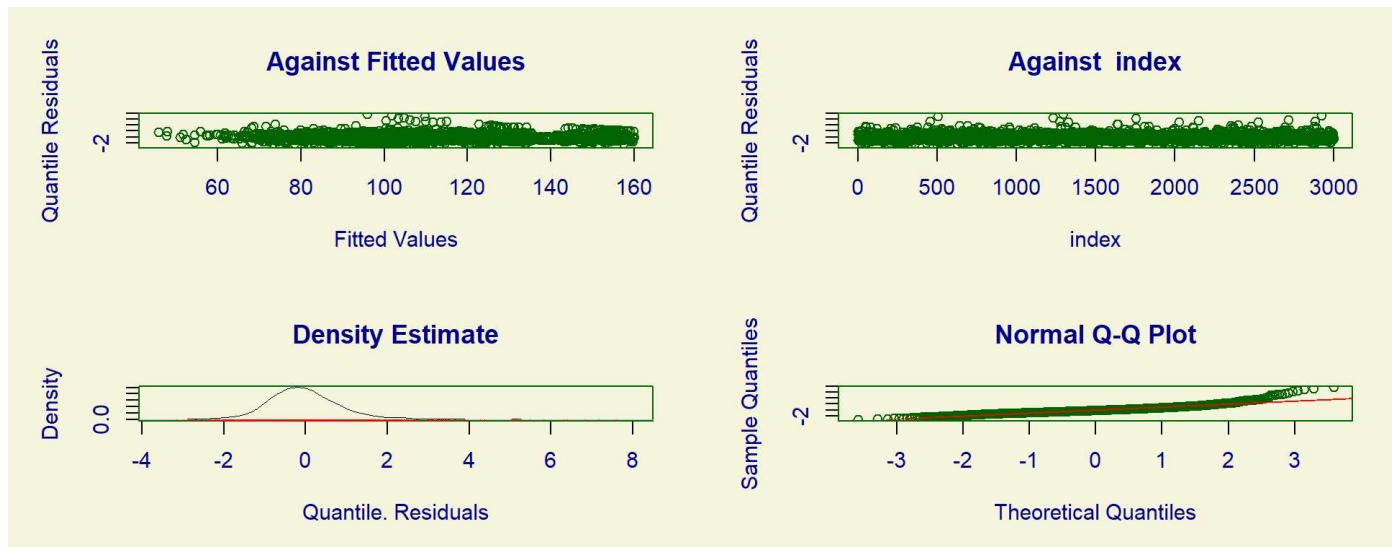
Datos Wage

```
summary(modelo_gamlss)
```

```
## ****
## Family: c("NO", "Normal")
##
## Call: gammelss(formula = wage ~ bs(age) + race + year + education + health,
##                 sigma.formula = ~bs(age) + race + year + education + health,
##                 family = NO, data = Wage)
##
## Fitting method: RS()
##
## -----
## Mu link function: identity
## Mu Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)           -2057.7691   528.9249 -3.890 0.000102 ***
## bs(age)1              63.8698    6.4926  9.837 < 2e-16 ***
## bs(age)2              42.4229    5.4234  7.822 7.14e-15 ***
## bs(age)3              23.9740    6.2488  3.837 0.000127 ***
## race2. Black          -2.7589    1.9174 -1.439 0.150295
## race3. Asian          -6.4240    2.4237 -2.650 0.008080 **
## race4. Other           -4.8280    4.0419 -1.194 0.232380
```

Datos Wage

```
plot(modelo_gamlss)
```



```
## ****
##      Summary of the Quantile Residuals
##          mean      =  0.002149396
##          variance =  1.00033
##          coef. of skewness =  1.407693
##          coef. of kurtosis =  9.270535
```

Datos Wage

```
performance::performance(modelo_gamlss)

## # Indices of model performance
##
## AIC      |      BIC |   RMSE | Sigma
## -----
## 29329.574 | 29485.740 | 1.000 | 3.131
```

Datos Wage

```
library(mgcv)

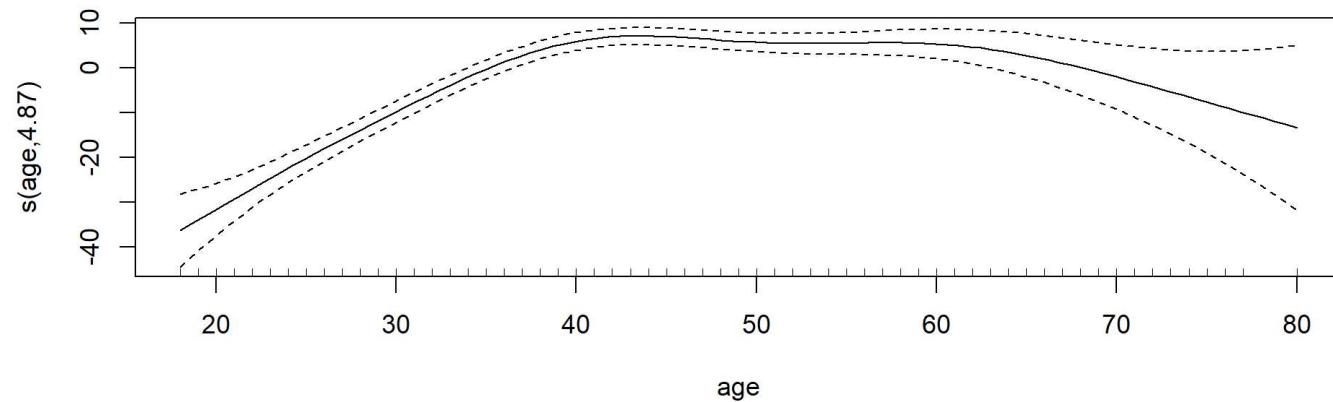
modelo_gam2 <- gam( wage ~ s(age) + race + year + education + health,data      = Wage,)

summary(modelo_gam2)

## 
## Family: gaussian
## Link function: identity
##
## Formula:
## wage ~ s(age) + race + year + education + health
##
## Parametric coefficients:
##                               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)           -2375.5642   633.9466  -3.747  0.000182 ***
## race2. Black          -6.0253    2.1768  -2.768  0.005676 **  
## race3. Asian          -3.1399    2.6706  -1.176  0.239808    
## race4. Other           -6.7914    5.8206  -1.167  0.243392    
## year                  1.2251    0.3161   3.876  0.000108 ***  
## education2. HS Grad   10.3323   2.4197   4.270  2.01e-05 ***
## education3. Some College 22.5689   2.5530   8.840  < 2e-16 ***
## education4. College Grad 36.1930   2.5543  14.169  < 2e-16 ***
```

Datos Wage

```
plot(modelo_gam2)
```



Datos Wage

```
performance::performance(modelo_gam2)

## # Indices of model performance
##
## AIC | BIC | R2 | RMSE | Sigma
## -----
## 29850.496 | 29945.831 | 0.299 | 34.843 | 34.954
```