



---

**Universidad de Valladolid**

Facultad de Ciencias

## **TRABAJO FIN DE GRADO**

Grado en Estadística

**Modelos aditivos  
y su implementación en R**

***Autor:***

*Óscar Alexis Paredes Pérez*

***Tutor:***

*Bonifacio Salvador González*

***Junio 2019***



# Índice general

<b>Resumen</b> .....	1
<b>Abstract</b> .....	1
<b>Capítulo 1 Introducción</b> .....	3
1.1 Modelos Lineales.....	3
1.2 Modelos no paramétricos de regresión.....	4
1.3 Suavizadores univariantes .....	5
1.3.1 Regresión polinómico local. Lineal local .....	5
1.3.2 Loess .....	7
1.3.3 Regresión con Splines. Spline cúbico suavizado .....	8
<b>Capítulo 2 Modelos Aditivos</b> .....	11
2.1 Algoritmo Backfitting .....	13
2.1.1 Backfitting para dos suavizadores .....	15
2.1.2 Convergencia del Backfitting .....	16
2.2 Mínimos cuadrados penalizados.....	17
2.3 Modelos aditivos ponderados.....	18
2.4 Inferencias .....	19
2.4.1 Bandas de variabilidad .....	19
2.4.2 Grados de libertad .....	19
2.4.3 Análisis de la Deviance.....	20
<b>Capítulo 3 Implementación en R</b> .....	21
3.1 Librería <i>gam</i> .....	21
3.2 Librería <i>mgcv</i> .....	25
3.3 Librería <i>gamlss</i> .....	27
3.4 Librería <i>gamm4</i> .....	28
3.5 Librería <i>sBF</i> .....	28
3.6 Librería <i>gamair</i> .....	29
3.7 Librería <i>sm</i> .....	29
<b>Capítulo 4 Aplicaciones</b> .....	31
4.1 Datos reales .....	31
4.2 Datos simulados .....	46

<b>Conclusiones</b> .....	55
<b>Bibliografía</b> .....	57
<b>Lista de figuras</b> .....	59
<b>Lista de tablas</b> .....	61
<b>Anexos. Código R</b> .....	63
Anexo 1.    Funciones .....	63
Anexo 2.    Datos reales .....	64
Anexo 3.    Datos simulados .....	67

# Resumen

Los modelos aditivos surgen dentro de la regresión no paramétrica para tratar el problema de la maldición de la dimensionalidad. Por analogía con el modelo de regresión múltiple se mantiene la aditividad del efecto de los predictores, pero se sustituye la relación lineal de cada uno por una función no paramétrica suave. La estimación de estos modelos se realiza mediante el algoritmo backfitting el cual ajusta cada componente del modelo mediante la suavización de los residuos parciales; esto hace que en la estimación de cada predictor se eliminen los efectos de los demás y nos permite estudiar el efecto de cada uno por separado. Además, nos permite obtener una representación visual del efecto de cada predictor que interviene en el ajuste. En cada paso del algoritmo se ajusta una regresión univariante en la que se pueden utilizar diferentes suavizadores. De esta forma se elimina el problema de la dimensionalidad.

A lo largo de este trabajo se explicarán las principales características de los modelos aditivos y del algoritmo backfitting. También se expondrán dos aplicaciones de estos modelos en datos reales y simulados en los que se compararan el ajuste que proporcionan distintos suavizadores lineales.

**Palabras clave:** modelos lineales, regresión no paramétrica, modelos aditivos, algoritmo backfitting, suavizadores lineales, deviance, R.

# Abstract

Additive models emerge from non-parametric regression in order to face the problem of the curse of dimensionality. By analogy to the multiple regression model, the additivity of the effect of the predictors is kept, but the lineal relation of each one of them is replaced by a non-parametric smoothed function. The estimation of these models is carried out by the backfitting algorithm, which adjusts each one of the model components by smoothing the partial residuals; as a consequence, the estimation of each predictor is made ignoring the effects of the rest of predictors, and allows us to study the effect of each one separately. Furthermore, it allows us to obtain a visual representation of the effects of each one of the predictors which take part in the fit. On each step of the algorithm, an univariate regression is fitted, which enables us to use different smoothing parameters. This way, the dimensionality problem is avoided.

Along the project, the main characteristics of the additive models and the backfitting algorithm will be explained. It also will include two applications of these models in real and simulated data, where the fit obtained as a consequence of the use of different lineal smoothers will be compared.

**Key words:** linear models, non-parametric regression, additive models, backfitting algorithm, linear smoothers, deviance, R.



# Capítulo 1

## Introducción

### 1.1 Modelos Lineales

En el problema de regresión múltiple tenemos  $n$  observaciones para una variable respuesta continua  $y_i$ , y  $p$  variables explicativas  $(x_{1i}, \dots, x_{pi})$  con  $i = 1, \dots, n$ , cuyo objetivo es modelar la dependencia de la media de la variable  $Y$  en función de las variables explicativas  $X_1, \dots, X_p$ , estas últimas también son llamadas variables predictoras o covariables.

Queremos describir dicha dependencia de forma que podamos evaluar la contribución relativa de cada una de las variables explicativas en la explicación de  $Y$ , y que nos permita predecir esta variable para algún conjunto de valores de  $X_1, \dots, X_p$ . Para esto suponemos un modelo de regresión múltiple de la siguiente forma

$$y_i = \alpha + \beta_1 x_{1i} + \beta_2 x_{2i} + \dots + \beta_p x_{pi} + \epsilon_i, \quad i = 1, \dots, n \quad (1.1)$$

donde  $\epsilon_i$  son errores normales independientes con media cero y varianza  $\sigma^2$ . Como se puede observar estamos suponiendo que la dependencia de  $E(Y|X_1, \dots, X_p)$  es lineal. En este tipo de modelos lo que se busca es una estimación para  $\alpha$  y  $\beta_j$ , siendo  $j = 1 \dots p$ , por mínimos cuadrados de forma que se minimice la suma de cuadrados de los errores.

Estos modelos son útiles y convenientes porque:

- Proporciona una descripción simple de los datos.
- Podemos describir la contribución de cada una de las variables explicativas mediante un coeficiente.
- Proporcionan un método simple para predecir nuevas observaciones.

Pero estos modelos no siempre son adecuados, ya que en muchos casos suponer que la variable respuesta se relaciona linealmente con las variables explicativas es demasiado restrictivo si la relación no es lineal.

## 1.2 Modelos no paramétricos de regresión

Como se puede observar la regresión no paramétrica ha surgido debido a los problemas que aparecen a la hora de formular los modelos paramétricos, ya que hay veces que no se encuentra ninguno que sea adecuado para los datos. Los modelos no paramétricos permiten una mayor flexibilidad al no establecer hipótesis sobre la estructura subyacente en los datos, y esto se traduce en soluciones más flexibles.

Sea  $(x_{1i}, \dots, x_{pi}, y_i)$ ,  $i = 1, \dots, n$ , un conjunto de  $n$  observaciones de un vector aleatorio multidimensional,  $(X_1, \dots, X_p, Y)$ , que satisface el siguiente modelo

$$y_i = m(x_{1i}, \dots, x_{pi}) + \epsilon_i, \quad i = 1, \dots, n \quad (1.2)$$

donde  $\epsilon_i$  son errores aleatorios independientes con media cero y varianza  $\sigma^2$ , y la función  $m$  es desconocida y se define como  $m(X_1, \dots, X_p) = E(Y|X_1 = x_1, \dots, X_p = x_p)$ . Este corresponde a un planteamiento de tipo multivariante, es decir, se tienen varias variables explicativas, basado en un diseño aleatorio, donde las observaciones constituyen una muestra aleatoria de una población  $(X_1, \dots, X_p, Y)$ .

Para la estimación de  $m(\cdot)$  se puede optar por dos opciones diferentes, asumir algún modelo paramétrico para la función de regresión y centrarnos en la estimación de los parámetros o no imponer tal restricción, asumiendo solo que  $m(\cdot)$  es suave en términos de derivabilidad y el problema se centra en la estimación de dicha función, esta opción corresponde a la estimación no paramétrica.

Ante este segundo planteamiento nos encontramos con la maldición de la dimensionalidad, asociada al hecho de que cuando estamos estimando en un entorno con un número fijo de datos y tenemos una superficie de gran dimensión, dicho entorno puede estar vacío o ser demasiado grande como para llamarlo local, es decir, se necesita un número exponencialmente mayor de observaciones para que dichos entornos contengan observaciones de la muestra.

Para solucionar el problema anterior se han introducido los modelos aditivos, que fueron desarrollados por Trevor Hastie y Robert Tibshirani en 1990, en los que se modela la media de la respuesta como una suma de funciones no lineales, una por cada predictor que está incluido en el modelo. Estos modelos son estimados mediante el algoritmo backfitting, el cual permite estimar cada una de las funciones de forma independiente y con técnicas de suavizado diferentes para cada una de las variables. La estimación se hace de forma que se minimice la suma de cuadrados del error penalizando por falta de suavidad. Una de las aplicaciones para este modelo es pensar en ellos como un método de estimación simultaneo para medir las variables en las unidades apropiadas, después podemos realizar el análisis lineal en las variables transformadas restringiendo dicha transformación en el sentido de que las funciones estimadas sean monótonas.

## 1.3 Suavizadores univariantes

Supongamos ahora el siguiente modelo con una sola variable explicativa

$$y_i = m(x_i) + \epsilon_i, \quad i = 1, \dots, n \quad (1.3)$$

donde  $\epsilon_i$  son errores independientes con media cero y varianza  $\sigma^2$ . Para la estimación de  $m(\cdot)$  nos centraremos en suavizadores lineales, que son aquellos suavizadores que se pueden expresar como un promedio de la variable respuesta por unos pesos, es decir

$$\hat{m}(x) = \sum_{i=1}^n l_i(x) y_i \quad (1.4)$$

donde  $l_i(x)$  son pesos conocidos que tienen la propiedad de que  $\forall x, \sum_{i=1}^n l_i(x) = 1$ .

### 1.3.1 Regresión polinómico local. Lineal local

Este método ajusta los datos mediante un polinomio de grado  $p$ , utilizando solo observaciones de un entorno de  $X$ , es decir, para cada punto  $x$  se define un entorno  $[a, b]$ , estimamos la función de regresión en este entorno y el ajuste local es la función ajustada evaluada en  $x$ . Se ha de suponer que  $m(\cdot)$  tiene  $p + 1$  derivadas continuas en dicho entorno. Al aumentar el grado del polinomio se mejora la estimación, pero hace falta suponer que la función es tan suave como para tener tantas derivadas continuas. Se suele incorporar una función de peso para dar mayor importancia a valores  $x_i$  que están próximos a  $x$  en el entorno.

La estimación en dichos entornos son promedios de las observaciones de la variable respuesta correspondientes a cada  $x$  que se encuentran en la ventana de la forma  $(x - h, x + h)$ , donde  $h \geq 0$  es el ancho de banda o parámetro de suavizado, ya que es el que determina la suavidad de la función estimada. También incluye en su definición una función núcleo,  $w(t) \geq 0$ , que está definida como una densidad simétrica y con varianza finita, es decir,  $\int w(t) dt = 1$ ,  $\int t w(t) dt = 0$ ,  $\int t^2 w(t) dt = k_2 < \infty$ . Podemos destacar el estimador *Nadaraya - Watson* que ajusta constantes localmente.

Supongamos un predictor  $X$  y  $m(X)$  con  $p + 1$  derivadas continuas en  $[a, b]$  lo que queremos es minimizar

$$\min_m \sum_{i=1}^n (y_i - m(x_i))^2 w(x_i - x, h) \quad (1.5)$$

donde  $w(x_i - x, h) = \frac{1}{h} w\left(\frac{x_i - x}{h}\right)$ , siendo  $w(\cdot)$  la función núcleo que determina el peso de  $(X_i, Y_i)$  en la estimación de  $x$ . Ahora fijado un  $x \in [a, b]$ , como  $x_i \in [a, b]$  podemos considerar el siguiente desarrollo de Taylor de  $m(x_i)$  en torno a  $x$ , que usa los primeros  $p$  términos

$$m(x_i) \approx m(x) + (x_i - x)m'(x) + \frac{(x_i - x)^2}{2!}m''(x) + \dots + \frac{(x_i - x)^p}{p!}m^{(p)}(x). \quad (1.6)$$

Sustituyendo y reescribiendo el polinomio en la expresión (1.5) en función de  $\beta_j(x) = \frac{m^{(j)}(x)}{j!}$ ,  $j = 0, \dots, p$ , ahora lo que queremos minimizar es

$$\min_{\beta} \sum_{i=1}^n \left( y_i - \beta_0(x) - (x_i - x)\beta_1(x) - \dots - (x_i - x)^p \beta_p(x) \right)^2 w(x_i - x, h) \quad (1.7)$$

dado que  $m(x_i)$  debería dar una buena aproximación cuando  $x_i$  esta cerca de  $x$ , los datos que están en el entono de  $x$  deben tener más importancia cuando se ajusta por mínimos cuadrados.

Para llevar a cabo la minimización de (1.7), lo escribiremos de forma matricial, donde  $\mathbf{X}_x$  es una matriz de diseño de tamaño  $n \times (p + 1)$  y es una función de  $x$  e  $\mathbf{y}$  es un vector de  $n$  elementos.

$$\mathbf{X}_x = \begin{pmatrix} 1 & (x_1 - x) & (x_1 - x)^2 & \dots & (x_1 - x)^p \\ 1 & (x_2 - x) & (x_2 - x)^2 & \dots & (x_2 - x)^p \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & (x_n - x) & (x_n - x)^2 & \dots & (x_n - x)^p \end{pmatrix}; \quad \mathbf{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}.$$

También definimos  $\mathbf{W}_x$  como una matriz diagonal de tamaño  $n \times n$

$$\mathbf{W}_x = \text{diag}(w(x_i - x, h)), \quad i = 1, \dots, n$$

por último, definimos  $\beta_x$  como

$$\beta_x = \begin{pmatrix} \beta_0(x) \\ \vdots \\ \beta_p(x) \end{pmatrix}.$$

Ahora, reescribimos (1.7) de la siguiente manera

$$\min_{\beta} (\mathbf{y} - \mathbf{X}_x \beta_x)^t \mathbf{W}_x (\mathbf{y} - \mathbf{X}_x \beta_x) \quad (1.8)$$

derivando respecto de  $\beta_x$  e igualando a cero obtenemos

$$\hat{\beta}_x = (\mathbf{X}_x^t \mathbf{W}_x \mathbf{X}_x)^{-1} \mathbf{X}_x^t \mathbf{W}_x \mathbf{y}. \quad (1.9)$$

El estimador de la regresión será la primera componente de este vector

$$\hat{m}(x) = \mathbf{e}_1^t (\mathbf{X}_x^t \mathbf{W}_x \mathbf{X}_x)^{-1} \mathbf{X}_x^t \mathbf{W}_x \mathbf{y} \quad (1.10)$$

con  $\mathbf{e}_1^t = (1, 0, 0, \dots, 0)^t$  donde el número de elementos es  $p + 1$ . Cuando  $p = 1$  obtenemos el estimado lineal local, en este caso la matriz de diseño,  $\mathbf{X}_x$ , será de tamaño  $n \times 2$  y  $\beta_x$  tendrá dos componentes

$$\mathbf{X}_x = \begin{pmatrix} 1 & x_1 - x \\ \vdots & \vdots \\ 1 & x_n - x \end{pmatrix}, \quad \beta_x = \begin{pmatrix} \beta_0(x) \\ \beta_1(x) \end{pmatrix}.$$

Por tanto, el estimador lineal local es

$$\hat{m}(x, 1) = \frac{\sum_{i=1}^n w(x_i - x, h)[S_2 - S_1(x_i - x)]y_i}{n(S_0S_2 - S_1^2)} \quad (1.11)$$

donde  $S_r = \frac{1}{n} \sum_{i=1}^n (x_i - x)^r w(x_i - x, h)$  para  $r = 0, 1, 2$ . Este estimador puede ser escrito como un promedio de la variable respuesta.

### 1.3.2 Loess

La regresión polinómico local utiliza un parámetro de suavizado constante para la estimación y esto es razonable en modelos de diseño fijo y con valores equiespaciados de las variables explicativas. Pero cuando estos valores no están equiespaciados es necesario utilizar parámetros de suavizado variables, para que el ancho de banda varié según el punto de estimación  $x$  teniendo en cuenta la densidad de los datos, es decir, un ancho de banda pequeño en regiones donde hay una mayor concentración de datos, mientras que un ancho de banda grande donde los datos son escasos.

Una forma de implementar esto es emplear un ancho de banda que refleje la densidad de los datos a través de la distancia al  $k$  vecino más cercano, es decir

$$h_k(x) = h * d_k(x) \quad (1.12)$$

donde  $d_k(x)$  representa la distancia al  $k$ -ésimo vecino más cercano de  $x$ . De esta forma el parámetro de suavizado se escala para aumentar el grado de suavizado en regiones donde los datos son escasos, que se representa con distancias grandes entre  $x$  y el  $k$ -ésimo vecino, y disminuir el grado de suavizado donde los datos son densos, que se representa con distancias pequeñas en  $x$  y el  $k$ -ésimo vecino. Esto nos va a proporcionar pesos que son cero para los datos que estén a una distancia mayor o igual al  $k$ -vecino más cercano, dicho de otra forma, las observaciones que estén a una distancia mayor no se incluirán para el ajuste local. De esta forma ahora el parámetro  $k$  determina el suavizado.

Ahora, fijado  $x$ , en lugar de minimizar (1.7) lo que se quiere minimizar es

$$\min_{\beta} \sum_{i=1}^n \left( y_i - \beta_0(x) - (x_i - x)\beta_1(x) - \dots - (x_i - x)^p \beta_p(x) \right)^2 w(x_i - x, h_k(x)). \quad (1.13)$$

En la estimación loess en lugar de especificar el  $k$ , se utiliza el span que es la proporción de observaciones que contribuyen con un peso positivo en la regresión lineal local, dicho de otra forma, las que queremos que intervengan en el ajuste local, es decir,  $k/n$  siendo  $n$  el número de datos de los que se dispone.

Este estimador incorpora robustez en el procedimiento de ajuste, para evitar que observaciones atípicas ejerzan influencia en el ajuste de la curva. Esto se implementa mediante el uso de pesos apropiados en el procedimiento iterativo de suavizado de manera que sea paralelo a la regresión lineal.

Aunque existe diversas funciones núcleo utilizadas para el ajuste, la más utilizada es el núcleo tricubo definido como  $w(z, h) = (1 - (|z|/h)^3)^3$ , donde  $z \in [-h, h]$ .

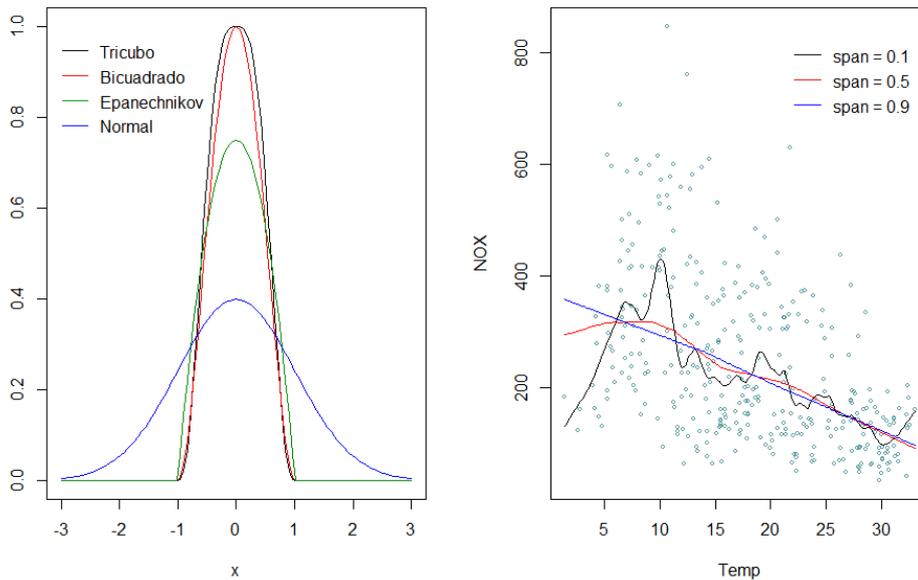


Figura 1.1: Funciones núcleo (izquierda) e influencia del span en la estimación (derecha).

### 1.3.3 Regresión con Splines. Spline cúbico suavizado

La regresión con splines representa el ajuste como un polinomio por partes, donde las regiones son separadas por puntos de ruptura o nodos,  $\xi_1, \dots, \xi_k$ . Además, se fuerza a que las uniones de los polinomios en estos puntos sean suaves, obligando a que las derivadas sean continuas. La función estimada,  $\hat{m}(\cdot)$ , se obtiene de forma que se minimice la suma de cuadrados del error.

Para la elección de las funciones tenemos diferentes alternativas, una es los splines naturales que tiene la restricción adicional de que la función es lineal. Para imponer esta condición hay que imponer dos restricciones en cada uno de los nodos:  $m''' = m'' = 0$ . En la práctica se proporciona un nodo adicional en cada uno de los extremos de los datos y se impone la linealidad más allá de los nodos que están en la frontera. Pero la alternativa más común es los splines cúbicos a trozos que tiene la condición de que son continuos y que la primera y la segunda derivada son continuas en los nodos.

El problema principal de los splines es la selección del número de nodos y su posición. Para esto hay diferentes procedimientos, un enfoque simple es seleccionar el número de nodos y distribuirlos uniformemente en el rango de los datos, una versión más adaptativa es colocarlos en los cuantiles apropiados de la variable predictora. Otro problema de los splines es que la suavidad no puede variarse fácilmente de manera continua en función de un único parámetro de suavizado.

Otro problema es la selección de las funciones para representar los splines para un conjunto de nodos. Llamemos a los nodos interiores  $\xi_1 < \dots < \xi_k$ , y aumentamos el conjunto con dos nodos frontera  $\xi_0$  y  $\xi_{k+1}$ . Una elección simple, para splines cúbicos a trozos, es la base de series de potencia truncada

$$s(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \sum_{j=1}^K \theta_j (x - \xi_j)_+^3 \quad (1.14)$$

Donde  $(a)_+ = a$  si  $a > 0$  y 0 en otro caso. Además, se puede ver que tiene las propiedades requeridas:

- (1)  $s$  es un polinomio cubico en cualquier subintervalo  $[\xi_j, \xi_{j+1})$ .
- (2)  $s$  tiene dos derivadas continuas.
- (3)  $s$  tiene una tercera derivada que es una función escalonada con saltos en  $\xi_1, \dots, \xi_k$ .

El spline Cubico Suavizado surge como la solución al problema de encontrar entre todas las funciones  $m(x)$  con dos derivadas continuas, una que minimice la suma de cuadrados del error, es decir

$$\min_m \sum_{i=1}^n \{y_i - m(x_i)\}^2 + \lambda \int_a^b \{m''(t)\}^2 dt \quad (1.15)$$

donde  $\lambda$  es una constante fija y  $a \leq x_1 \leq \dots \leq x_n \leq b$ . El primer término mide la cercanía de los datos, mientras que el segundo término penaliza la curvatura de la función. Destacamos que la estimación de los coeficientes se hace de manera restringida. El parámetro  $\lambda$  desempeña el mismo papel que el span en la regresión polinómico local. Valores grandes de  $\lambda$  producen curvas más suaves, es decir menos onduladas, que para valores pequeños. Si  $\lambda \rightarrow \infty$  el término de penalización fuerza a que  $m''(x) = 0$  en todas partes de esta forma la solución es la de mínimos cuadrados. Si  $\lambda \rightarrow 0$  el término de penalización deja de ser importante y la solución tiende a interpolar una función dos veces diferenciable.

Si partimos de que la solución a (1.15) es un spline cubico y usando otra base distinta a la de series de potencia truncada donde las funciones de la base son distintas de cero en un intervalo de como máximo 5 nodos distintos y escribiendo  $s(x) = \sum_{j=1}^{n+2} \gamma_j B_j(x)$ , donde  $\gamma_j$  son coeficientes y  $B_j(x)$  la base, podemos sustituir  $m(x)$  por  $s(x)$  y calcular la integral.

Definiendo  $\mathbf{B}$  de tamaño  $n \times (n + 2)$  y  $\mathbf{\Omega}$  de tamaño  $(n + 2) \times (n + 2)$  como

$$B_{ij} = B_j(x_i), \quad \Omega_{ij} = \int B_i''(x) B_j''(x) dx$$

podemos reescribir (1.15) como

$$(\mathbf{y} - \mathbf{B}\boldsymbol{\gamma})^t (\mathbf{y} - \mathbf{B}\boldsymbol{\gamma}) + \lambda \boldsymbol{\gamma}^t \mathbf{\Omega} \boldsymbol{\gamma}. \quad (1.16)$$

Derivando respecto a  $\boldsymbol{\gamma}$  e igualando a cero obtenemos

$$(\mathbf{B}^t \mathbf{B} + \lambda \mathbf{\Omega}) \hat{\boldsymbol{\gamma}} = \mathbf{B}^t \mathbf{y} \quad (1.17)$$

teniendo en cuenta que podemos obtener la descomposición de Cholesky de  $\mathbf{M} = (\mathbf{B}^t \mathbf{B} + \lambda \mathbf{\Omega})$  como  $\mathbf{M} = \mathbf{L}\mathbf{L}^t$ , resolviendo  $\mathbf{L}\mathbf{L}^t \hat{\boldsymbol{\gamma}} = \mathbf{B}^t \mathbf{y}$  por sustitución regresiva obtenemos  $\hat{\boldsymbol{\gamma}}$  y por tanto la  $\hat{s}(x)$ .

Sea  $N$  una matriz no singular de bases para representar la solución y  $\hat{\beta}$  la versión transformada de  $\hat{\gamma}$  correspondiente al cambio de base. Entonces podemos reescribir la solución de la siguiente manera

$$\hat{s} = N\hat{\beta} = N(N^tN + \lambda\Omega)^{-1}N^t\mathbf{y} = (I + \lambda K)^{-1}\mathbf{y} \quad (1.18)$$

donde  $K = N^{-t}\Omega N^{-1}$ .

En el **Capítulo 2** se explicarán de forma detallada los modelos aditivos. Se comentarán algunas de las características de estos modelos, así como la relación y las diferencias que tiene con los modelos lineales. Se explicará el algoritmo backfitting y algunos criterios de convergencia para el caso de suavizadores lineales para dos y  $p$  variables explicativas. Por último, se explicarán el método de ajuste de mínimos cuadrados penalizados y algunas inferencias sobre los modelos.

En el **Capítulo 3** se explicarán diversas librerías de R que implementan estos modelos. Se verán librerías que restringen el ajuste de cada componente a un solo tipo de suavizador univariante, así como librerías que implementan algunas variantes de estos modelos como los mixtos o los de localización y escala. Se explicará de manera más detallada la librería *gam* que es la que se utilizara para el ajuste de los modelos en distintos conjuntos de datos.

Por último, en el **Capítulo 4** veremos estos modelos aplicados a dos conjuntos de datos diferentes. El primero, un conjunto de datos reales correspondientes a la calidad del aire en una ciudad italiana. Y el segundo, correspondiente a datos simulados. En ambos casos se ajustarán los modelos con diferentes suavizadores univariantes y se compararán los resultados obtenidos.

# Capítulo 2

## Modelos Aditivos

Empecemos consideremos el siguiente modelo de regresión múltiple no paramétrico para  $n$  observaciones de una variable respuesta  $y_i$ , y  $p$  predictores  $(x_{1i}, \dots, x_{pi})$

$$y_i = m(x_{1i}, \dots, x_{pi}) + \epsilon_i, \quad i = 1, \dots, n \quad (2.1)$$

donde  $\epsilon_i$  son errores independientes con  $E(\epsilon_i) = 0$  y varianza  $\sigma^2$ . En el que  $m$ , es una función desconocida, que modeliza la relación de dependencia entre  $X_1, \dots, X_p$  e  $Y$ , y donde el objetivo es la estimación de  $m$  mediante algún criterio.

Por lo general este tipo de modelos suelen ser poco útiles en dimensiones altas ya que resulta difícil proporcionar información visual que ayude a la comprensión de los resultados y a que las propiedades del estimador se deterioran, debido a que requiere que los datos proporcionen una estimación de una superficie en dimensiones altas para cualquier tamaño de muestra. Esto es lo que se conoce como la maldición de la dimensionalidad, para un número fijo de datos y para un entorno de dicha superficie a estimar, dicho entorno puede ser demasiado grande como para llamarlo local, tener pocas observaciones de la muestra o estar vacío, por lo que se necesita un número exponencialmente mayor de observaciones para que los entornos contengan observaciones de la muestra.

Consideremos ahora un modelo de regresión lineal múltiple, en el que la variable respuesta está relacionada con  $p$  predictores, dado por la siguiente expresión

$$y_i = \beta_0 + \beta_1 x_{1i} + \dots + \beta_p x_{pi} + \epsilon_i, \quad i = 1, \dots, n$$

cuando la suposición de linealidad es insostenible podemos sustituir dicha relación lineal por funciones no paramétricas. Esta es la idea del Modelo Aditivo, permitir que cada una de las componentes tome esta forma no paramétrica

$$y_i = \beta_0 + m_1(x_{1i}) + \dots + m_p(x_{pi}) + \epsilon_i, \quad i = 1, \dots, n.$$

Con esto vemos que los efectos de los predictores son aditivos para explicar la respuesta, es decir, cada predictor tiene efectos separados en la explicación de la respuesta, sin interacciones entre ellos. Además, se puede ver como esta suposición es análoga a la suposición realizada en el modelo lineal y esto permite una estimación más eficiente de cada componente y de la superficie de regresión.

Dado que estamos representando cada variable por separado, podemos representar las  $p$  funciones y examinar cómo influyen cada predictor en el modelado de la respuesta y también obtener una estimación global como la suma de  $p$  funciones estimadas univariantes.

De esta forma el modelo aditivo queda definido como

$$y_i = \alpha + \sum_{j=1}^p m_j(x_{ji}) + \epsilon_i, \quad i = 1, \dots, n \quad (2.2)$$

donde los errores  $\epsilon_i$  son independientes de  $X_j$  con  $E(\epsilon_i) = 0$  y  $var(\epsilon_i) = \sigma^2$ , y las  $m_j(\cdot): \mathbb{R} \rightarrow \mathbb{R}$  son funciones cuya forma no están restringidas, solo tienen un supuesto de suavidad entendida en términos de derivabilidad de la función. Además, se impone la condición de  $\sum_{i=1}^n m_j(x_{ji}) = 0$ ,  $\forall j$  para hacer que las funciones sean únicas. Dichas funciones son estimadas individualmente y  $\hat{m}(x_i) = \sum_{j=1}^p \hat{m}_j(x_{ji})$ . Esto proporciona un modelo general útil para modelar datos que no se ajustan a supuestos lineales.

Estos modelos tienen como ventaja que se pueden emplear diversos métodos de estimación unidimensional, llamados suavizadores univariantes, para la estimación de las funciones.

Teniendo en cuenta la restricción de que la suma de los valores observados es cero para cada una de las componentes del modelo, se puede ver como  $\hat{\alpha} = \bar{y}$ , con esto y reordenando el modelo se puede obtener una estimación no paramétrica para cada una de las componentes en función de todas las demás, basada en el suavizado de los residuos de los datos después de ajustar las otras componentes, es decir

$$\hat{\mathbf{m}}_k = \mathcal{S}_k \left( y - \hat{\alpha} - \sum_{j \neq k} \hat{m}_j \right) \quad (2.3)$$

donde  $\mathcal{S}_k$  es una matriz de suavizado para estimar la función  $\mathbf{m}_k$  y representa un estimador no paramétrico univariante de  $E(y|x_k)$ . Esta expresión es la que lleva a proponer el algoritmo backfitting desarrollado por *Buja* en 1989, que es una forma análoga al método de *Gauss – Seidel* para el problema de mínimos cuadrados en los problemas de regresión lineal, el cual nos permite estimar cada función mediante el suavizado de los residuos parciales, utilizando cualquier suavizador univariante en las componentes.

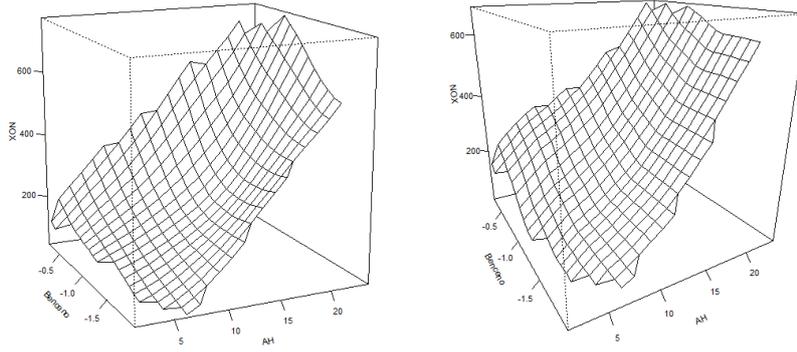


Figura 2.1: Ajuste no paramétrico con dos variables (izquierda) y ajuste aditivo con dos variables (derecha).

Observando la Figura 2.1 vemos, en el caso de dos variables, como la superficie estimada tiene una característica especial, en cada dirección los cortes trasversales en la otra dirección son idénticos y esto es lo que nos permite hablar de las componentes del modelo. Esto no pasa con los ajustes no paramétricos.

## 2.1 Algoritmo Backfitting

Partiendo del modelo (2.2) y para cualquier  $k$ ,  $E(Y - \alpha - \sum_{j \neq k} m_j(X_j) | X_k) = m_k(X_k)$  esto sugiere un algoritmo iterativo para calcular todas las  $m_j$  el cual hace la estimación a partir de las  $n$ -observaciones disponibles. Para la implementación es necesario proporcionar funciones iniciales. Sin información previa de las funciones una estimación inicial podría ser una regresión lineal de  $y$  en cada uno de los predictores, esto lo podemos hacer ya que el resultado final no depende de la elección de los valores iniciales para cada una de las funciones.

De esta forma podemos implementar el algoritmo de la siguiente manera

- (I) Inicializamos  $\hat{\alpha} = \text{media}(y)$ .  
Damos estimaciones iniciales para las funciones  $\hat{m}_1^0, \dots, \hat{m}_p^0$ .
- (II) Para cada  $j = 1, \dots, p$  calculamos las nuevas estimaciones de la siguiente forma, reordenamos el modelo suponiendo conocidas todas las funciones excepto una y aplicamos el suavizador univariante a cada modelo

$$y - \hat{\alpha} - \sum_{j \neq 1}^p \hat{m}_j^0(x_{ji}) = m_1(x_{1i}) + \epsilon_i \rightarrow$$

$$\hat{m}_1^1(x_{1i}) = \mathcal{S}_1 \left( y - \hat{\alpha} - \sum_{j \neq 1}^p \hat{m}_j^0(x_{ji}) \right)$$

$$y - \hat{\alpha} - \hat{m}_1^1(x_{1i}) - \sum_{j \neq 1,2}^p \hat{m}_j^0(x_{ji}) = m_2(x_{2i}) + \epsilon_i \rightarrow$$

$$\hat{m}_2^1(x_{2i}) = \mathcal{S}_2 \left( y - \hat{\alpha} - \hat{m}_1^1(x_{1i}) - \sum_{j \neq 1,2}^p \hat{m}_j^0(x_{ji}) \right)$$

$$y - \hat{\alpha} - \hat{m}_1^1(x_{1i}) - \hat{m}_2^1(x_{2i}) - \sum_{j \neq 1,2,3}^p \hat{m}_j^0(x_{ji}) = m_3(x_{3i}) + \epsilon_i \rightarrow$$

$$\hat{m}_3^1(x_{3i}) = \mathcal{S}_3 \left( y - \hat{\alpha} - \hat{m}_1^1(x_{1i}) - \hat{m}_2^1(x_{2i}) - \sum_{j \neq 1,2,3}^p \hat{m}_j^0(x_{ji}) \right)$$

.....

$$y - \hat{\alpha} - \sum_{j=1}^{p-1} \hat{m}_j^1(x_{ji}) = m_p(x_{pi}) + \epsilon_i \rightarrow$$

$$\hat{m}_p^1(x_{pi}) = \mathcal{S}_p \left( y - \hat{\alpha} - \sum_{j=1}^{p-1} \hat{m}_j^1(x_{ji}) \right)$$

(III) Repetir (II) reemplazando  $\hat{m}_j^0 = \hat{m}_j^1, \forall j$  hasta que la diferencia de las funciones individuales sea pequeña o no cambien.

Como ya se ha comentado antes  $\mathcal{S}_j$  representa un estimador no paramétrico. Además, se puede ver que cada vez que se reajusta una de las funciones,  $\hat{m}_j$ , se eliminan los efectos de todas las otras variables de  $\mathbf{y}$  antes suavizar el residuo parcial.

Por otra parte, el algoritmo es deliberadamente vago ya que depende del tipo de suavizador que se use y del contexto en el cual se esté usando. Sin embargo, esto es una característica importante ya que puede ser ajustado según las preferencias de las personas:

- Si es apropiado, cualquiera de las funciones puede ser ajustada linealmente  $m_j(x) = x\beta_j$ .
- Las variables categóricas pueden ser incorporadas al modelo. Si  $X_j$  tiene  $K$  categorías, podemos crear un factor de nivel  $K$  y ajustar cada nivel por una constante.
- Se pueden considerar también interacciones entre variables categóricas y continuas. Supongamos que  $X_j$  tiene efectos diferentes para  $M$  y  $H$ , podemos estimar una función para cada una, es decir,  $m_{jM}$  y  $m_{jH}$ .
- La interacción entre dos variables continuas puede ser modelada de varias maneras, podemos crear unas variables compuestas que sean un producto de pares y usarlo de forma lineal o no paramétricamente. Otra alternativa es modelar la interacción por pares como una superficie general, esto implica tratar el par como una sola variable y usar una superficie bidimensional.

En muchos casos estamos asumiendo implícitamente que se utiliza el mismo suavizador para cada una de las variables, pero esto es solo para facilitar la explicación. Los resultados son de carácter general y se pueden aplicar a cualquier

procedimiento backfitting en el que se utilice un suavizador lineal para cada una de las variables. Incluso de forma más general podemos pensar en  $\mathcal{S}_1, \dots, \mathcal{S}_p$  como un conjunto de transformaciones lineales.

### 2.1.1 Backfitting para dos suavizadores

Consideremos un modelo aditivo que involucra dos variables explicativas con suavizadores lineales  $\mathcal{S}_1$  y  $\mathcal{S}_2$ . A partir de ahora consideraremos, sin pérdida de generalidad, las observaciones  $y_i$  centradas con respecto a su media  $y_i = y_i - \bar{y}$ . De esta forma nos podemos olvidar de la constante  $\alpha$  ya que esto no afecta a las estimaciones. En este caso podemos obtener la solución del algoritmo backfitting de forma explícita partiendo de que buscamos la solución al siguiente sistema

$$\begin{aligned} \mathbf{m}_1 &= \mathcal{S}_1(\mathbf{y} - \mathbf{m}_2) \\ \mathbf{m}_2 &= \mathcal{S}_2(\mathbf{y} - \mathbf{m}_1) \end{aligned} \tag{2.4}$$

Sean  $\hat{\mathbf{m}}_1^{(h)}$  y  $\hat{\mathbf{m}}_2^{(h)}$  las estimaciones en la etapa  $h$ -ésima del algoritmo, donde  $h = 0$  indican las funciones iniciales. El backfitting consiste en alternar los siguientes pasos

$$\begin{aligned} \hat{\mathbf{m}}_1^{(h)} &= \mathcal{S}_1(\mathbf{y} - \hat{\mathbf{m}}_2^{(h-1)}) \\ \hat{\mathbf{m}}_2^{(h)} &= \mathcal{S}_2(\mathbf{y} - \hat{\mathbf{m}}_1^{(h)}). \end{aligned}$$

Usando inducción podemos ver que para  $h \geq 1$

$$\begin{aligned} \hat{\mathbf{m}}_1^{(h)} &= \mathbf{y} - \sum_{j=0}^{h-1} (\mathcal{S}_1 \mathcal{S}_2)^j (\mathbf{I} - \mathcal{S}_1) \mathbf{y} - (\mathcal{S}_1 \mathcal{S}_2)^{h-1} \mathcal{S}_1 \hat{\mathbf{m}}_2^{(0)} \\ \hat{\mathbf{m}}_2^{(h)} &= \mathcal{S}_2 \sum_{j=0}^{h-1} (\mathcal{S}_1 \mathcal{S}_2)^j (\mathbf{I} - \mathcal{S}_1) \mathbf{y} + \mathcal{S}_2 (\mathcal{S}_1 \mathcal{S}_2)^{h-1} \mathcal{S}_1 \hat{\mathbf{m}}_2^{(0)} \end{aligned}$$

Sea  $\|\mathcal{C}\|$  cualquier norma de la matriz  $\mathcal{C}$ , entonces una condición suficiente para que  $\hat{\mathbf{m}}_1^{(h)}$  y  $\hat{\mathbf{m}}_2^{(h)}$  converjan es  $\|\mathcal{S}_1 \mathcal{S}_2\| < 1$ . Si este es el caso podemos obtener la solución como

$$\begin{aligned} \hat{\mathbf{m}}_1^{(\infty)} &= \{\mathbf{I} - (\mathbf{I} - \mathcal{S}_1 \mathcal{S}_2)^{-1} (\mathbf{I} - \mathcal{S}_1)\} \mathbf{y} \\ \hat{\mathbf{m}}_2^{(\infty)} &= \mathcal{S}_2 (\mathbf{I} - \mathcal{S}_1 \mathcal{S}_2)^{-1} (\mathbf{I} - \mathcal{S}_1) \mathbf{y} = \{\mathbf{I} - (\mathbf{I} - \mathcal{S}_2 \mathcal{S}_1)^{-1} (\mathbf{I} - \mathcal{S}_2)\} \mathbf{y} \end{aligned} \tag{2.5}$$

Donde  $\mathbf{I} - (\mathbf{I} - \mathcal{S}_1 \mathcal{S}_2)^{-1} (\mathbf{I} - \mathcal{S}_1)$  y  $\mathbf{I} - (\mathbf{I} - \mathcal{S}_2 \mathcal{S}_1)^{-1} (\mathbf{I} - \mathcal{S}_2)$  son matrices de suavizado, y el ajuste de  $\hat{\mathbf{y}}$  será

$$\hat{\mathbf{y}} = \hat{\mathbf{m}}_1^{(\infty)} + \hat{\mathbf{m}}_2^{(\infty)} = \{\mathbf{I} - (\mathbf{I} - \mathcal{S}_2)(\mathbf{I} - \mathcal{S}_1 \mathcal{S}_2)^{-1} (\mathbf{I} - \mathcal{S}_1)\} \mathbf{y} \tag{2.6}$$

Con esto probamos que si  $\|\mathcal{S}_1 \mathcal{S}_2\| < 1$  existe una solución única.

## 2.1.2 Convergencia del Backfitting

Respecto a la convergencia del algoritmo, si todos los suavizadores son operadores de proyección la convergencia está asegurada y podría ser reemplazado por una proyección global donde no sería necesaria ninguna iteración. Algunos suavizadores, como los splines suavizados no son proyecciones, pero poseen la propiedad de proyección que requiere la convergencia. Sin embargo, existen otros suavizadores, como los suavizadores núcleo de los cuales no hay pruebas de convergencia para el algoritmo.

Ahora nos centraremos en la estimación de los modelos aditivos con suavizadores lineales  $\mathcal{S}_1, \dots, \mathcal{S}_p$ , y supongamos que las funciones de inicio son todas cero. En cada etapa del algoritmo  $\hat{m}_1$  tendrá la misma media que  $y$ , pero las otras  $\hat{m}_j$  tendrán media cero. Si el algoritmo empieza en  $j = 2$  la media de  $y$  será la misma que la de  $\hat{m}_2$  y las de todas la demás cero. Esto nos lleva a una dependencia de las iteraciones finales en los valores de las funciones de inicio, para funciones iniciales distintas de cero. Una solución para esto es sustituir  $\mathcal{S}_j$  por una matriz de suavizado centrada que se obtiene como  $(I - \mathbf{1}\mathbf{1}^T/n)\mathcal{S}_j$ , con esto aseguramos que en cada etapa del algoritmo las  $\hat{m}_j$  tengan media cero.

Como ya se ha visto para el caso bivalente la condición de convergencia viene dada por

$$\|\mathcal{S}_1\mathcal{S}_2\| < 1. \quad (2.7)$$

Si las matrices de suavizado no están centradas tendríamos que  $\|\mathcal{S}_1\mathcal{S}_2\| = 1$  que es un ejemplo de concurvidad, que es el análogo de la colinealidad en los modelos lineales y por tanto no deseable, en el que la función relativa a una variable es una combinación lineal de cierto número de funciones del modelo. Pero el centrado hace que  $\mathcal{S}_1\mathcal{S}_2\mathbf{1} = \mathbf{0}$ , es decir,  $\|\mathcal{S}_1\mathcal{S}_2\| < 1$ .

En el caso  $p$ -variante, Opsomer propuso en el 2000 un criterio de convergencia para el caso de suavizadores lineales, en el que un modelo aditivo con matrices de suavizado  $\mathcal{S}_1, \dots, \mathcal{S}_p$  converge a una solución única si, para algún  $j$

$$\|\mathcal{S}_j\mathbf{W}^{-j}\| < 1 \quad (2.8)$$

donde  $\mathbf{W}^{-j} = \mathbf{W}_1 + \dots + \mathbf{W}_{j-1} + \mathbf{W}_{j+1} + \dots + \mathbf{W}_p$  es una matriz de suavizado para la función  $m_{-j} = \sum_{k=1, k \neq j}^p m_k$ , que se puede considerar el suavizador de un modelo aditivo donde los datos provienen del siguiente modelo

$$y_i = m_1(x_{1i}) + \dots + m_{j-1}(x_{j-1,i}) + m_{j+1}(x_{j+1,i}) + \dots + m_p(x_{pi}) + \epsilon_i.$$

En este caso el suavizador con respecto a la  $j$ -ésima covariable puede ser escrito como

$$\mathbf{W}_j = \mathbf{I} - (\mathbf{I} - \mathcal{S}_j\mathbf{W}^{-j})^{-1}(\mathbf{I} - \mathcal{S}_j) \quad (2.9)$$

## 2.2 Mínimos cuadrados penalizados

El modelo aditivo  $E(Y|X) = \sum_{j=1}^p m_j(X_j)$ , en el cual suponemos que  $\bar{y} = 0$ , puede ser estimado mediante el algoritmo backfitting por mínimos cuadrados, pero es necesario introducir una penalización para la suavidad que proporcionan los estimadores, es decir, si intentemos minimizar

$$\sum_{i=1}^n \left\{ y_i - \sum_{j=1}^p m_j(x_{ji}) \right\}^2 \quad (2.10)$$

se debe añadir un término a (2.10) que penalice por falta de suavidad.

Para solucionar esto partimos del criterio de minimización para mínimos cuadrados penalizados del spline cubico para un modelo con una sola variable predictora, es decir, para el modelo  $y_i = m(x_i) + \epsilon_i$ ,  $i = 1, \dots, n$ .

$$\sum_{i=1}^n \{y_i - m(x_i)\}^2 + \lambda \int_a^b \{m''(x)\}^2 dx. \quad (2.11)$$

Podemos escribir la expresión anterior de forma matricial

$$(\mathbf{y} - \mathbf{m})^t (\mathbf{y} - \mathbf{m}) + \lambda \mathbf{m}^t \mathbf{K} \mathbf{m} \quad (2.12)$$

donde  $\mathbf{K}$  es una matriz de penalización cuadrática. Se puede ver que (2.12) tiene un mínimo dado por

$$\hat{\mathbf{m}} = (\mathbf{I} - \lambda \mathbf{K})^{-1} \mathbf{y}. \quad (2.13)$$

También se puede argumentar en el sentido contrario, dado un suavizador lineal simétrico basado en la matriz de suavizado  $\mathbf{S}$ ,  $\hat{\mathbf{m}} = \mathbf{S} \mathbf{y}$  minimiza

$$(\mathbf{y} - \mathbf{m})^t (\mathbf{y} - \mathbf{m}) - \mathbf{m}^t (\mathbf{S}^- - \mathbf{I}) \mathbf{m} \quad (2.14)$$

con  $\mathbf{m} \in \mathcal{R}(\mathbf{S})$  (rango de  $\mathbf{S}$ ), donde  $\mathbf{S}^-$  es una inversa generalizada de  $\mathbf{S}$ .

Extendiendo esto a la estimación del modelo aditivo, generalizamos (2.11) de la siguiente manera

$$\sum_{i=1}^n \left\{ y_i - \sum_{j=1}^p m_j(x_{ji}) \right\}^2 + \sum_{j=1}^p \lambda_j \int \{m_j''(t)\}^2 dt. \quad (2.15)$$

Se observa como cada función es penalizada por una constante  $\lambda_j$ , esto a su vez determina la suavidad de la función en la solución. Hay que tener en cuenta que si  $\lambda_j = 0, \forall j$ , es decir, no hay penalización de suavidad, la solución es cualquier conjunto de funciones cuyas evaluaciones satisfacen  $\sum_{j=1}^p m_j(x_{ji}) = y_i$ ,  $i = 1, \dots, n$ . Por otro lado, si cada  $\lambda_j$  tiende a infinito, el termino de penalización tiende a infinito a menos que cada  $m_j$  sea lineal, es decir, que  $m_j''(t) = 0, \forall j$ . Por lo tanto, el problema se reduce al problema de mínimos cuadrados.

Al igual que en el caso anterior podemos reescribir (2.15) de forma matricial

$$\left(\mathbf{y} - \sum_{j=1}^p \mathbf{m}_j\right)^t \left(\mathbf{y} - \sum_{j=1}^p \mathbf{m}_j\right) + \sum_{j=1}^p \lambda_j \mathbf{m}_j^t \mathbf{K}_j \mathbf{m}_j \quad (2.16)$$

donde las  $\mathbf{K}_j$  son matrices de penalización para cada predictor. Ahora si diferenciamos (2.16) con respecto a la función  $\mathbf{m}_k$ , e igualamos a cero obtenemos

$$\hat{\mathbf{m}}_k = (\mathbf{I} + \lambda_k \mathbf{K}_k)^{-1} (\mathbf{y} - \sum_{j \neq k} \hat{\mathbf{m}}_j) \quad (2.17)$$

como ya se ha mencionado anteriormente  $(\mathbf{I} + \lambda_k \mathbf{K}_k)^{-1}$  es la matriz de suavizado. También podemos ver esto en la dirección contraria, dado un estimador vemos que es solución de

$$\left(\mathbf{y} - \sum_{j=1}^p \mathbf{m}_j\right)^t \left(\mathbf{y} - \sum_{j=1}^p \mathbf{m}_j\right) + \sum_{j=1}^p \mathbf{m}_j^t (\mathcal{S}_j^- - \mathbf{I}) \mathbf{m}_j. \quad (2.18)$$

Podemos interpretar cada uno de los términos de penalización como una reducción del peso de cada una de las componentes de  $\mathbf{m}_j$ .

## 2.3 Modelos aditivos ponderados

Consideremos el siguiente criterio de mínimos cuadrados penalizados ponderados, que consiste en añadir una matriz de pesos al criterio de mínimos cuadrados penalizados

$$\left(\mathbf{y} - \sum_j \mathbf{m}_j\right)^t \mathbf{W} \left(\mathbf{y} - \sum_j \mathbf{m}_j\right) + \sum_j \lambda_j \mathbf{m}_j^t \mathbf{K}_j \mathbf{m}_j \quad (2.19)$$

donde  $\mathbf{W}$  es una matriz diagonal de pesos,  $\lambda_j$  es un parámetro de suavizado y  $\mathbf{K}_j$  es una matriz de penalización para el  $j$ -ésimo predictor. Estos pesos pueden representar la posición relativa de cada observación o pueden surgir como parte de otro procedimiento iterativo, como el algoritmo de puntuación local que es el que implementan los paquetes de R para los modelos aditivos generalizados.

Las soluciones para este problema tienen la misma forma que para el caso no ponderado, teniendo en cuenta que los suavizadores son splines ponderados, que vienen dados por  $\mathcal{S}_j = (\mathbf{W} + \lambda_j \mathbf{K}_j)^{-1} \mathbf{W}$ . Podríamos generalizar todos los resultados anteriores para tratar el caso ponderado calculando las normas y los productos interiores en la métrica de  $\mathbf{W}$ . Sin embargo, es más fácil hacer las siguientes transformaciones y pasar al problema no ponderado

$$\begin{aligned} \mathbf{y}' &= \mathbf{W}^{1/2} \mathbf{y} \\ \mathbf{m}'_j &= \mathbf{W}^{1/2} \mathbf{m}_j \\ \mathbf{K}'_j &= \mathbf{W}^{-1/2} \mathbf{K}_j \mathbf{W}^{-1/2} \end{aligned} \quad (2.20)$$

Hay que tener en cuenta que  $S_j$  no es simétrica, pero  $W^{1/2}SW^{-1/2}$  es simétrica con autovalores en  $[0,1]$ , y con valores propios unitarios correspondientes a funciones lineales de la  $j$ -ésima variable. Con esto, los resultados de convergencia para el caso no ponderado se pueden aplicar directamente al caso ponderado.

## 2.4 Inferencias

### 2.4.1 Bandas de variabilidad

Partimos de que la matriz de covarianzas del ajuste  $\hat{\mathbf{m}} = \mathbf{S}\mathbf{y}$  es  $cov(\hat{\mathbf{m}}) = \mathbf{S}\mathbf{S}^t\sigma^2$  y dado un estimador de  $\sigma^2$  podemos construir las bandas de variabilidad para cada suavizador. Si las observaciones tienen errores independientes e igualmente distribuidos y además expresamos  $\hat{\mathbf{m}}_j = \mathbf{R}_j\mathbf{y}$  con  $\mathbf{R}_j$  de tamaño  $n \times n$ , entonces

$$cov(\hat{\mathbf{m}}_j) = \mathbf{R}_j\mathbf{R}_j^t\sigma^2 \quad (2.21)$$

donde  $\sigma^2 = var(Y_i)$ . Para el cálculo de  $\mathbf{R}_j$  el mejor enfoque es aplicar el backfitting a cada una de las columnas de  $\mathbf{I}_n$  (matriz identidad de tamaño  $n \times n$ ). El resultado aplicado al  $i$ -ésimo vector produce vectores ajustados  $\hat{\mathbf{m}}_j^i, j = 1, \dots, p$ , donde  $\hat{\mathbf{m}}_j^i$  es la  $i$ -ésima columna de  $\mathbf{R}_j$ , extendiendo esto,  $\hat{\mathbf{m}}_+^i$  es la  $i$ -ésima columna de  $\mathbf{R}$ .

Las bandas de variabilidad se construyen sumando y restando dos veces la raíz cuadrada de los elementos de la diagonal de  $\hat{\sigma}^2\mathbf{R}_j\mathbf{R}_j^t$ , es decir

$$\left(\hat{\mathbf{m}}_j \pm 2\sqrt{diag(\hat{\sigma}^2\mathbf{R}_j\mathbf{R}_j^t)}\right). \quad (2.22)$$

Para las bandas de variabilidad globales, que el procedimiento es similar al caso univariante, supongamos que el modelo aditivo es correcto y un estimador para  $\mathbf{R}\mathbf{m}_+ = \mathbf{g}_+$  es  $\hat{\mathbf{m}}_+ = \mathbf{R}\mathbf{y}$ , entonces una aproximación para  $\mathbf{g}_+$  es

$$v(\mathbf{g}_+) = (\hat{\mathbf{m}}_+ - \mathbf{g}_+)^t(\mathbf{R}\mathbf{R}^t\hat{\sigma}^2)^{-1}(\hat{\mathbf{m}}_+ - \mathbf{g}_+). \quad (2.23)$$

Si asumimos que tenemos una estimación para  $\hat{\sigma}^2$  y que la distribución  $v$  es  $G$ , entonces podemos construir unas bandas de variabilidad simultaneas de confianza  $1 - \alpha$  para todos los predictores

$$C(\mathbf{g}_1, \dots, \mathbf{g}_p) = \{\mathbf{g}_1, \dots, \mathbf{g}_p; v(\mathbf{g}_+) \leq G_{1-\alpha}\}. \quad (2.24)$$

### 2.4.2 Grados de libertad

Definimos los grados de libertad para un suavizador lineal  $\mathbf{S}$  como  $df = tr(\mathbf{S})$ , así los grados de libertad son la suma de los autovalores de  $\mathbf{S}$ . Para los modelos aditivos la definición es análoga, los grados de libertad son  $tr(\mathbf{R})$ , donde  $\mathbf{R}$  es la matriz de suavizado que produce  $\hat{\mathbf{m}}_+ = \mathbf{R}\mathbf{y}$ , siendo  $\hat{\mathbf{m}}_+ = \hat{\mathbf{m}}_1 + \dots + \hat{\mathbf{m}}_p$ .

Para la comparación de modelos se utilizará los grados de libertad del error que se define como  $df^{err} = n - tr(2\mathbf{R} - \mathbf{R}\mathbf{R}^t)$ , definimos  $\Delta df^{err}$  como la variación en los

grados de libertad del error. Sea  $\mathbf{R}_{-j}$  el operador que produce el ajuste eliminando el  $j$ -ésimo término. Entonces definimos  $df_j^{err}$  como los grados de libertad para el error para  $j$ -ésimo término como  $df_j^{err} = tr(2\mathbf{R} - \mathbf{R}\mathbf{R}^t) - tr(2\mathbf{R}_{-j} - \mathbf{R}_{-j}\mathbf{R}_{-j}^t)$ , que se puede ver como el aumento esperado en la suma de cuadrados del error si el  $j$ -ésimo predictor es eliminado del modelo, asumimos que esto no incrementa el sesgo.

La suma de las varianzas de los valores ajustados es definida como  $df^{var} = tr(\mathbf{R}\mathbf{R}^t)$ , además, la suma de las varianzas de la función ajustadas  $\hat{\mathbf{m}}_j$  es  $\sigma^2 tr(\mathbf{R}_{-j}\mathbf{R}_{-j}^t)$ .

En la práctica se utiliza  $tr(\mathbf{S}_j) - 1$  o  $tr(2\mathbf{S}_j - \mathbf{S}_j\mathbf{S}_j^t) - 1$  como una aproximación de los grados de libertad para la comparación de los modelos. Restamos 1 porque hay una constante redundante en  $p - 1$  de los  $p$  términos en el modelo.

### 2.4.3 Análisis de la Deviance

La deviance para un modelo  $\mathbf{y}$  está definida como

$$D(\mathbf{y}; \boldsymbol{\mu}) = 2\{l(\boldsymbol{\mu}_{max}; \mathbf{y}) - l(\boldsymbol{\mu}; \mathbf{y})\} \quad (2.25)$$

donde  $\boldsymbol{\mu}_{max}$  es el valor del parámetro que maximiza  $l(\boldsymbol{\mu}; \mathbf{y})$  en el modelo saturado, siendo  $l(\cdot)$  la log-verosimilitud. La deviance desempeña el papel de la suma de cuadrados del error y se utiliza para evaluar la bondad del ajuste y para comparar modelos. Para la comparación de modelos, supongamos  $\widehat{\mathcal{M}}_1$  y  $\widehat{\mathcal{M}}_2$  dos modelos ajustados con  $\widehat{\mathcal{M}}_1$  anidado dentro de  $\widehat{\mathcal{M}}_2$ , bajo algunas condiciones de regularidad

$$D(\widehat{\mathcal{M}}_2; \widehat{\mathcal{M}}_1) = D(\mathbf{y}; \widehat{\mathcal{M}}_1) - D(\mathbf{y}; \widehat{\mathcal{M}}_2). \quad (2.26)$$

Para los modelos no paramétricos y aditivos, la deviance tiene sentido para evaluar los modelos y ver sus diferencias. Sin embargo, la teoría de la distribución no está desarrollada, no obstante, se pueden realizar contrastes con una justificación heurística. Si la distribución de la deviance fuera  $\chi^2$  entonces  $df^{err}$  sería sus grados de libertad. Aunque la deviance no sigue esta distribución, ni siquiera asintóticamente, las simulaciones han demostrado que la distribución  $\chi^2$  es una aproximación útil para los modelos de detección. Por lo tanto, usamos esta distribución para la comparación de modelos.

Dado que en los modelos aditivos estamos suponiendo que  $y$  sigue una distribución normal, la cual pertenece a la familia exponencial y en su definición se incluye un parámetro de dispersión,  $\phi$ . Cuando el parámetro de dispersión es desconocido es más apropiado utilizar un test  $F$  aproximado.

De esta forma y con lo visto hasta este punto, podemos definir el estadístico AIC (criterio de información de Akaike) como

$$AIC = D(\mathbf{y}; \boldsymbol{\mu})/n + 2df\phi/n. \quad (2.27)$$

# Capítulo 3

## Implementación en R

Aunque existen diversas librerías, las cuales se explicaran a continuación, que implementan los modelos aditivos y aditivos generalizados, y variantes de estos como los mixtos o los de localización, escala y forma, la mayoría se basan o implementan funciones propias en la *librería gam*. Teniendo en cuenta lo anterior y que algunas de las librerías, como la *mgcv*, restringen el ajuste a la utilización de solo un tipo de suavizador univariante, consideraremos la librería *gam* como la librería principal para el ajuste de los modelos aditivos y aditivos generalizados, la cual se utilizara más adelante para el modelado de diferentes conjuntos de datos. Por ello se hará una explicación más en detalle de esta librería respecto a las demás.

### 3.1 Librería *gam*

Esta librería contiene funciones para ajustar y trabajar con modelos aditivos generalizados. Dentro de estas podemos destacar aquellas que efectúan el ajuste de la parte no paramétrica de los predictores, principalmente loess y splines. También se pueden encontrar funciones para construir modelos paso a paso o para hacer predicciones, así como diferentes conjuntos de datos reales y simulados para poder realizar ajuste sobre ellos.

A continuación, describiremos las principales funciones para el estudio de los modelos, así como aquellas funciones que sean de interés, aunque no se utilicen en el análisis de los datos. Hay que tener en cuenta que será una descripción breve, para más detalle consultar la guía de R.

(<https://cran.r-project.org/web/packages/gam/gam.pdf>).

**Función *gam*** (ajuste de modelos aditivos generalizados):

Ajusta modelos especificando una descripción de los predictores y una descripción de la distribución de los errores. Utiliza el algoritmo backfitting para diferentes métodos de

suavizado o ajuste. Los métodos actualmente implementados para el ajuste son la regresión polinómico local y splines suavizados. El modelo se ajusta utilizando el algoritmo de puntuación local, que se ajusta de manera iterativa a los modelos aditivos ponderados mediante backfitting.

```
gam(formula, family = gaussian, data, weights, subset, na.action, start,
     etastart, mustart, control = gam.control(...), model = TRUE,
     method, x = FALSE, y = TRUE, ...).
```

#### Argumentos:

**formula:** expresión de la formula, respuesta~predictores. Los términos de suavizado no paramétricos se indican con *lo* para loess y *s* para los splines suavizados.

**family:** descripción de la distribución de errores y la función de enlace que se utilizará en el modelo.

**data:** marco de datos opcional que contiene las variables del modelo. Si no se encuentran los datos, las variables se toman del entorno.

**subset:** vector opcional que especifica un subconjunto de observaciones que se utilizaran en el proceso de ajuste.

**control:** lista de parámetros para controlar el proceso de ajuste (mirar guía para más detalle).

**model:** valor lógico que indica si el marco del modelo debe devolverse como una componente del valor devuelto. El valor por defecto es TRUE.

**method:** especifica el método que se utilizara para ajustar la parte paramétrica del modelo.

valores devueltos: (devuelve un objeto que hereda de la clase *glm* y *lm*)

Esta función nos devuelve diferentes valores como los valores del ajuste del modelo, los coeficientes de la parte paramétrica y los valores medios ajustados. Así como los residuos, aunque estos no se pueden interpretar. Los grados de libertad residual y la deviance del modelo y del modelo nulo. Información relacionada con el algoritmo de puntuación local y del backfitting. Por último, también nos devuelve información relacionada con el aspecto no paramétrico del modelo como la matriz de suavizado, sus grados de libertad aproximados y su varianza. También obtenemos un test de puntuación para la eliminación de los términos de suavizado.

#### **Función lo** (ajuste mediante loess):

Indicamos que queremos ajustar la parte no paramétrica mediante regresión local, más concretamente mediante un polinomio que se ajusta localmente. Esta función no hace normalización de los datos.

```
lo(..., span = 0.5, degree = 1)
```

### Argumentos:

... : con esto se indica que puede ser un vector, una matriz o expresiones que evalúen cualquiera de estos. Si es una lista de vectores han de tener la misma longitud.

span: número (porcentaje) de observaciones que queremos que entren en el ajuste. Es el parámetro de suavizado para el ajuste

degree: grado del polinomio para el ajuste. Restringido a 1 ó 2.

### valores devueltos:

Devuelve una matriz numérica. Según el grado del polinomio devuelve una versión normalizada del vector o una base polinomial del grado 2. Dicha matriz tiene una serie de atributos que son necesarios para los algoritmos de adaptación.

### **Función s** (ajuste mediante splines suavizados):

`s(x, df = 4, spar = 1)`

### Argumentos:

x: predictor univariante, que se evalúa como un vector numérico.

df: grados equivalentes de libertad usado como parámetro de suavizado. El parámetro de suavizado real es  $df = tr(S) - 1$ . Los valores deben ser mayores que 1, con  $df = 1$  implica un ajuste lineal. Este valor no es necesariamente un entero. Si se da df y spar, el primero tiene prioridad.

spar: se puede usar como parámetro de suavizado, con valores entre (0,1].

### valores devueltos:

Devuelve una matriz con una con una serie de atributos que se utilizan en los algoritmos de adaptación. Como los splines reproducen ajustes lineales, la parte lineal se ajusta de manera eficiente con las otras partes lineales paramétricas del modelo.

### **Función plot.Gam** (gráficos para cada componente de un objeto GAM):

Método que realiza gráficos adecuados para cada termino de un objeto GAM centrándose en los efectos principales, también puede usarse con objetos GLM y LM.

`plot(x, residuals = FALSE, se = FALSE, ask = FALSE, ...)`

`preplot(object, newdata, ...)`

### Argumentos:

x: objeto GAM o preplot.Gam. esta función comprueba si el elemento x tiene un componente preplot, en caso de no tenerlo lo calcula, ya que es este el que se requiere para hacer los graficos del objeto GAM.

residuals: si el valor es TRUE se trazan los residuos de la desviación parcial junto a las estimaciones.

ask: si es TRUE, funciona de modo interactivo. En vez se producen los gráficos de forma secuencial.

se: si es verdadero se incluyen las bandas de variabilidad, que son dos veces el error estándar.

object: igual que x.

newdata: si se suministra en la función preplot, el cálculo se basa en estos datos y no en los originales.

#### valores devueltos:

devuelve un gráfico adecuado para cada uno de los predictores del objeto GAM, x. solo traza las funciones de efecto principal de uno o dos predictores, así que las interacciones no se trazan. Los gráficos se basan en la función gplot() que tiene métodos para diferentes clases. Si x es numérico, una categoría o factor, una matriz o una lista, se produce un gráfico diferente.

#### **Función anova.Gam** (análisis de la *Deviance* para un modelo aditivo generalizado):

Genera una tabla ANODEV (análisis de la Deviance) para un conjunto de modelos, o un resumen para un solo modelo GAM, en este último caso nos devuelve tablas ANOVA especiales. Nos da un desglose de los grados de libertad para todos los términos del modelo separando la parte de proyección de la parte no paramétrica de cada una y devolviéndola como una lista de dos objetos anova.

#### **Función step.Gam** (constructor de modelos stepwise para los GAM):

Construye el modelo de manera escalonada. Para cada término hay una lista ordenada de alternativas. Devuelve el modelo seleccionado con hasta dos componentes adicionales. Tiene un componente ANOVA correspondiente a los pasos dados en la búsqueda.

#### **Función predict.Gam** (método de predicción para los modelos ajustados):

Obtiene predicciones y estima los errores estándar de esas predicciones de los modelos GAM ajustados. Nos devuelve una matriz con las predicciones y los errores estándar, también puede devolver dentro de la matriz una columna con los términos ajustados

#### **Función na.gam.replace** (filtro para valores ausentes en los datos):

Método para tratar con valores ausentes y que devuelve un modelo o los datos con las observaciones faltantes reemplazadas. Para un factor se reemplaza este por otro factor con un nivel más que etiqueta los NA. Los factores ordenados se tratan igual solo que devuelve un factor no ordenado. Para un vector numérico los datos faltantes son reemplazados por la media de los datos no ausentes.

## 3.2 Librería *mgcv*

Esta librería contiene funciones para ajustar modelos aditivos generalizado y modelos aditivos generalizados mixtos, así como funciones especiales para el ajuste de grandes conjuntos de datos. También se pueden encontrar funciones para obtener ajustes mediante diferentes métodos de estimación como el suavizado por probabilidad marginal, validación cruzada generalizada y similares. Se puede encontrar también funciones específicas para el ajuste de diversas distribuciones y diversos modelos de regresión. La función principal de ajuste de los modelos se baja en un ajuste con splines con estimación automática de la suavidad basada en diferentes métodos. Respecto al ajuste hay funciones específicas para la configuración de los ajustes por defecto del backfitting así como una función para la convergencia. (<https://cran.r-project.org/web/packages/mgcv/mgcv.pdf>).

**Función gam** (modelos aditivos generalizados con estimación de suavidad integrada):

Ajusta modelos aditivos generalizados a los datos y el grado de suavidad es estimado como parte del ajuste. También incluyen una variedad de modelos estimados por un enfoque de penalización cuadrática (GLM). El grado de suavidad de los términos se estima como parte de `fitting.gam()`, también puede ajustarse a cualquier GLM sujeto a múltiples penalizaciones cuadráticas. Los términos suaves se representan mediante splines de regresión penalizados (o similares) con diferentes alternativas para la selección del parámetro de suavizado (GCV/UBRE/AIC/REML), o con splines con grados de libertad fijos. También se pueden introducir en el ajuste términos lineales.

```
gam(formula,family=gaussian(),data=list(),weights=NULL,subset=NULL,
     na.action,offset=NULL,method="GCV.Cp",optimizer=c("outer","newton"),control=list(),scale=0,select=FALSE,knots=NULL,sp=NULL,min.sp=NULL,H=NULL,gamma=1,fit=TRUE,paraPen=NULL)
```

Si `fit = FALSE`, la función devuelve una lista de elementos necesarios para ajustar un GAM, pero en realidad no lo hace.

**Función s** (definición del suavizado para cada componente en el ajuste aditivo):

Es la función que se utiliza en la definición del suavizador para el ajuste de las componentes en los modelos ajustados. No evalúa un spline suave, solo sirve para ayudar a configurar un modelo utilizando suavizados basados en spline y genera bases y penalizaciones para realizar el ajuste.

**Función gamm** (modelos aditivos generalizados mixtos):

Esta función ajusta modelos mixtos a los datos, mediante la función `lme` para una función link identidad de errores normales, o mediante la función `gammPQL` (de la librería MASS) para los otros casos, en este caso las estimaciones son MLEs. Los efectos aleatorios y las estructuras de correlación se emplean principalmente para modelar la correlación residual en los datos. Los suavizados se especifican como en una llamada a `gam` como parte de la fórmula del modelo de efectos fijos, pero los

componentes de suavizado se tratan como efectos aleatorios. Devuelve dos términos, un objeto de la clase gam que contiene información para utilizar algunos métodos, y un objeto lme.

**Función bam** (modelos aditivos generalizados para conjuntos de datos muy grandes):

Esta función devuelve un objeto de la clase Gam y ajusta un modelo aditivo generalizado a un conjunto de datos muy grande, al igual que para un conjunto de datos pequeño incluyen modelos GLM u otras familias con penalización cuadrática en el que el grado de suavidad es estimado como parte del ajuste. Esta función es muy parecida a la anterior, pero los métodos están diseñados para conjuntos con varias decenas de miles de datos. Se puede realizar un ajuste alternativo discretizando los valores de las covariables.

**Función anova.gam**: contraste de hipótesis aproximados para modelos GAM. Para un solo objeto se realizan contrastes de Wald para cada término paramétrico y para cada término no paramétrico. Con dos o más objetos se hace una comparación en función del análisis de la deviance. Los modelos a comparar deben utilizar el mismo método de selección del parámetro de suavizado.

**Función predict.gam**: método de predicción para un nuevo conjunto de valores dado que produce predicciones para las covariables del modelo o para los valores originales utilizados para el ajuste. También existe una función para los modelos bam, **predict.bam()**.

**Función residuals.gam**: residuales para modelos aditivos generalizados en la que se puede seleccionar el tipo de residuales que se quiere (deviance, Pearson, scaled.pearson, working o response).

**Función gam.check**: diagnóstico para un gam ajustado. Genera información de diagnóstico sobre el procedimiento, gráficos para los residuales e información sobre el parámetro de suavizado.

**Función gam.scale**: estimación del parámetro de escala para GAMs. Es una función interna que utiliza la función **gam.control()**, que se utiliza para configurar los valores por defecto del ajuste en los gam.

**Función logLik.gam**: AIC y log-verosimilitud para un ajuste gam. Esta función extrae la log-verosimilitud de un modelo gam ajustado que es utilizado por el AIC.

**Función summary.gam y print.gam**: proporcionan resúmenes del ajuste de un objeto gam dado y generan una impresión predeterminada para dicho modelo.

**Función plot.gam**: traza las funciones componentes en la escala del predictor lineal. También produce trazados para los componentes del modelo paramétrico si los hubiese. Devuelven una lista de los datos utilizados para realizar los gráficos de manera que se puedan utilizar para generar gráficos personalizados.

### 3.3 Librería *gamlss*

Esta librería ajusta modelos lineales generalizados de localización, escala y forma donde la estimación se hace mediante máxima verosimilitud penalizada para los modelos no paramétricos, que utiliza un algoritmo de puntuación de Newton – Raphson/Fisher para maximizar la verosimilitud. Los términos aditivos en el modelo se ajustan utilizando el algoritmo backfitting.

(<https://cran.r-project.org/web/packages/gamlss/gamlss.pdf>)

**Función *gamlss*** : (Modelos aditivos generalizados para localización, escala y forma):

Devuelve un objeto de la clase *gamlss*, esta es muy similar a la función *gam()* de la librería *gam*, pero se puede adaptar a más distribuciones y puede modelar todos los parámetros de la distribución como funciones de las variables explicativas. Nos permite modelar hasta 4 parámetros de una familia.

```
gamlss(formula = formula(data), sigma.formula = ~1, nu.formula = ~1,
        tau.formula = ~1, family = NO(), data = sys.parent(), weights = NULL,
        contrasts = NULL, method = RS(), start.from = NULL, mu.start = NULL,
        sigma.start = NULL, nu.start = NULL, tau.start = NULL, mu.fix = FALSE,
        sigma.fix = FALSE, nu.fix = FALSE, tau.fix = FALSE,
        control = gamlss.control(...), i.control = glim.control(...), ...)
```

**Función *lo*** (ajuste mediante loess):

Esta función permite especificar el ajuste mediante loess en el modelo GAMLSS. Es similar a la que se implementa en la librería *gam*. Los controles del ajuste se pueden modificar mediante la función ***lo.contronl()*** y con ***vis.lo()*** podemos ver los resultados.

**Función *cs*** (ajuste mediante spline cubico suavizado):

La función utiliza ***smooth.spline ()*** para realizar el suavizado que a su vez utiliza la función *additive.fit()*. Toman un vector y lo devuelven con varios atributos. Estas funciones no realizan el suavizado, pero asignan los atributos al vector para ayudar a los *gamlss* en el suavizado. La función que realiza el suavizado es ***gamlss.cs ()***. Esta función permite.

**Función *ps***: ajuste mediante P-splines. Hay varias funciones que utilizan esta procedimiento en las que todas devuelven un vector con atributos que son necesarios para el ajuste con *additive.fit()*.

**Función *residuals.gamlss*** : residuales de un modelo GAMLSS ajustado. Con la función ***plot.gamlss()*** se obtienen gráficos de diagnóstico para los residuales.

**Función *predict.gamlss***: predicciones y errores para nuevos datos para un parámetro específico a partir de objetos GAMLSS. La función *predict.gamlss* se puede usar para extraer los predictores lineales, los valores ajustados y los términos específicos en el modelo con los nuevos valores de datos.

**Función *gamlssML***: estimación de máxima verosimilitud para un GAMLSS simple.

**Función gen.likelihood**: genera la función de verosimilitud desde un objeto GAMLSS.

**Función gamlss.control**: configuración de los ajustes en la estimación del gamlss.

**Función IC**: calcula el GAIC (criterio de información de Akaike generalizado) para una penalización  $k$  dada para un objeto GAMLSS ajustado.

**Función stepGAIC**: realiza una selección de modelos mediante stepwise utilizando el GAIC para la selección.

**Función ri**: especifica un ajuste mediante regresión ridge o lazo en el GAMLSS.

**Función deviance.gamlss**: deviance global para un modelo GAMLSS ajustado.

**Función edf**: grados de libertad de un modelo GAMLSS ajustado.

**Función fitDist**: ajuste para diferentes distribuciones paramétricas.

**Función additive.fit**: implementación del algoritmo backfitting para GAMLSS.

**Función LR.test**: test de razón de verosimilitud para GAMLSS anidados.

**Función coef.gamlss**: estimación de los coeficientes del modelo GAMLSS ajustado.

**Función summary.gamlss**: resumen para un modelo GAMLSS ajustado.

**Función term.plot**: gráficos de la estimación de la regresión para un parámetro específico de un objeto GAMLSS ajustado.

### 3.4 Librería *gamm4*

Esta librería solo consta de una función la cual ajusta modelos aditivos generalizados mixtos utilizando dos funciones de librerías diferentes lme4 y mgcv.

(<https://cran.r-project.org/web/packages/gamm4/gamm4.pdf>)

**Función gamm4**: Ajusta un modelo aditivo generalizado mixto a los datos, haciendo uso de las funciones de ajuste modular que proporciona lme4. La selección de suavidad es por REML en el caso de aditivo gaussiano y Laplace aproximado de lo contrario. Esta función se basa en gamm() del paquete mgcv, pero usa lme4 en lugar de nlme como el motor de ajuste subyacente. Su principal desventaja es que no puede manejar la mayoría de los suavizadores de penalización, pero gamm4() es útil cuando los efectos aleatorios no son i.i.d, o cuando hay un gran número de coeficientes aleatorios

### 3.5 Librería *sBF*

Esta librería es una alternativa para la estimación backfitting para modelos aditivos, en la que se encuentran funciones que obtienen una estimación para una mejora de este algoritmo, el smooth backfitting basado en el estimador Nadaraya – Watson. (<https://cran.r-project.org/web/packages/sBF/sBF.pdf>)

**Función *k***: función de ponderación del núcleo. Devuelve los pesos de la función núcleo utilizados en la estimación, se pueden elegir diferentes funciones: gaussian, unifrom, epanechnikov, biweight y triweight. El dominio de las funciones está centrado en el origen.

**Función *sBF***: estimador smooth backfitting. Obtiene el estimado en el que se puede elegir el ancho de banda de diferentes maneras. La entrada es una matriz de datos en la que se indica cual es la columna de la variable respuesta.

### 3.6 Librería *gamair*

Esta librería se caracteriza porque es una librería en la que solo se encuentran diversos conjuntos de datos reales y simulados, tanto para modelos aditivos como para modelos aditivos generalizados para diferentes familias para el ajuste. Los conjuntos de datos son de dimensiones pequeñas.

(<https://cran.r-project.org/web/packages/gamair/gamair.pdf>)

### 3.7 Librería *sm*

(<https://cran.r-project.org/web/packages/sm/sm.pdf>)

**Función *sm.regression*** (regresión no paramétrica con uno o dos predictores):

Esta función obtiene una estimación de la regresión no paramétrica para una variable respuesta y una o dos variables explicativas. Se basa en la estimación polinómica local y no permite hacer algunos contrastes sobre dicha regresión, más concretamente, el contraste de no efecto y el de linealidad. Tras el ajuste nos devuelve una lista que contiene los valores de la estimación en los puntos de evaluación, el parámetro de suavizado y las ponderaciones de los parámetros de suavizado, una estimación de la desviación estándar del error y el error estándar de la estimación.

```
sm.regression(x, y, h, model = "none", weights = NA, group = NA, ...)
```



# Capítulo 4

## Aplicaciones

Una vez explicada la teoría referente a los modelos aditivos y el ajuste mediante el algoritmo backfitting, y explicada la librería principal en la cual nos centraremos, ajustaremos este modelo a dos conjuntos de datos diferentes. Un conjunto de datos reales en el cual miraremos si todas las variables ayudan a la explicación de la respuesta y un conjunto de datos simulados en el que se incluirá un predictor que no aportará nada y que quedará excluido del modelo final.

### 4.1 Datos reales

Los datos que se utilizarán para la implementación de los modelos aditivos corresponden a la calidad del aire de una ciudad italiana. El conjunto de datos contiene las respuestas medias por hora de distintos sensores químicos de óxido metálico incrustados en un dispositivo de calidad del aire, la concentración real de algunos agentes contaminantes y datos correspondientes a la temperatura y a la humedad.

Los datos han sido registrados desde el 11 de marzo de 2004 hasta el 3 de abril de 2005. Aunque el conjunto de datos tiene mediciones para distintos gases nos centraremos en la concentración real de *Óxidos de Nitrógeno (NOX)* que se intentará explicar mediante las siguientes variables:

1. NOX: concentración media real por hora en ppb (partes por billón).
2. Benceno: concentración media real por hora en  $\text{microg}/\text{m}^3$ .
3. Temp: temperatura en  $^{\circ}\text{C}$ .
4. RH: humedad relativa en %.
5. AH: humedad absoluta.

La humedad relativa es la cantidad de vapor de agua que contiene el aire respecto al máximo que podría contener a esa misma temperatura. Para una misma cantidad de vapor de agua se produce una mayor humedad relativa en aire frío que en aire

caliente. La humedad absoluta es la cantidad de vapor de agua que contiene el aire por unidad de volumen.

El conjunto de datos consta de 9336 observaciones. Debido a que muchas observaciones presentan NA o valores ausentes en algunas de las variables, se hará un procesamiento de los datos haciéndose una media por día sin tener en cuenta los datos ausentes, de esta forma logramos reducir el número de observaciones ausentes y el conjunto de datos a 389 observaciones. Como sigue habiendo observaciones con algún campo con NA, estas se eliminarán para el análisis y se trabajara con un conjunto de datos que consta de 347 observaciones sin ningún campo con NA. De esta forma las variables con las que trabajaremos son:

1. NOX: concentración media por día en ppb (partes por billón).
2. Benceno: concentración media por día en  $\mu\text{g}/\text{m}^3$ .
3. Temp: temperatura media por día en  $^{\circ}\text{C}$ .
4. RH: media por día de humedad relativa en %.
5. AH: media por día de humedad absoluta.

Aunque en este trabajo nos centraremos en el análisis y comparación de los modelos aditivos, se comenzara haciendo una breve descripción de cada una de las variables que intervienen en el ajuste de los modelos, así como la relación que tienen entre ellas.

	<i>n</i>	<i>Min</i>	<i>Max</i>	<i>Media</i>	<i>Desviación típica</i>
<b>NOX</b>	347	33.609	847.435	241.265	151.216
<b>Benceno</b>	347	1.431	23.841	10.144	4.111
<b>Temp</b>	347	1.433	32.996	18.053	8.040
<b>RH</b>	347	21.326	81.108	49.171	13.166
<b>AH</b>	347	0.240	1.998	1.008	0.392

Tabla 4.1: Descriptivos para las variables de Calidad del Aire.

Observando la Figura 4.1 en la que se pueden ver los gráficos de dispersión y los coeficientes de correlación, vemos como la variable respuesta, NOX, tiene una correlación positiva y moderada en torno a 0.5 con las variables Benceno y RH, siendo mayor con la primera variable. También se puede ver que tiene una correlación negativa con Temp y AH, siendo la más baja para esta última. Podemos destacar también que en los gráficos de dispersión de NOX con las variables Benceno y RH tenemos una concentración de observaciones para valores bajos y medios de dichas variables, mientras que para la variable AH las observaciones están concentradas para los valores medios y para la variable Temp se ve como esta concentración se produce para los valores altos.

Respecto a la correlación entre los predictores se puede ver como la variable Temp es la que presenta la correlación más alta con AH. También destacamos que la correlación más baja la presenta la variable RH con AH. Se puede apreciar como la mayoría de las correlaciones entre los predictores son positivas, excepto la correlación entre Temp y RH que es negativa. Por último, podemos ver como Benceno tienes correlaciones bajas con todos los predictores, siendo muy parecida para Temp y para RH.

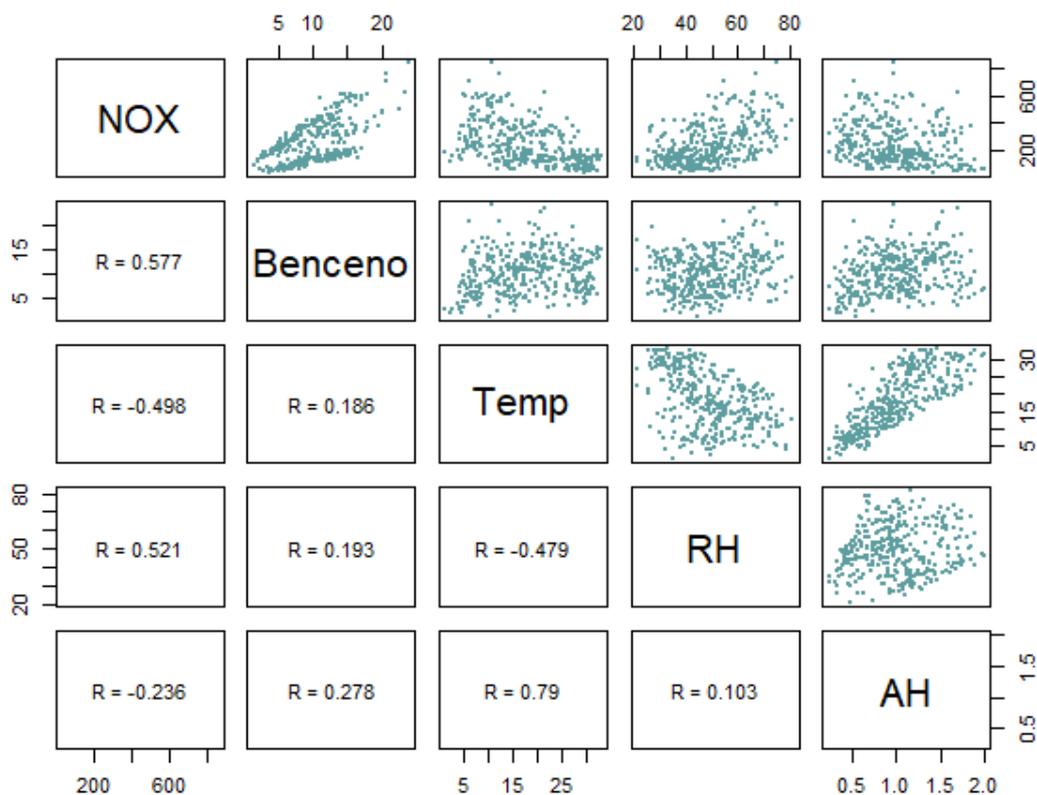


Figura 4.1: Matriz de dispersión y correlaciones para las variables de Calidad del Aire.

Ahora se realizará un análisis más detallado de la relación que tiene cada uno de los predictores con la variable respuesta, para ello se obtendrá un estimador no paramétrico de la regresión para cada uno de los casos, esto se hará con un ajuste polinómico local mediante la función `sm.regression()` de la librería `sm`, que utiliza un núcleo normal con parámetro de suavizado  $h$ , siendo este la desviación estándar de la función núcleo. También se realizarán contrastes sobre dicha regresión para poder asumir ciertas características que nos servirán después para el ajuste de los modelos aditivos, más concretamente se contrastará si el predictor tiene algún efecto sobre la variable respuesta y de ser así, si podemos asumir que dicho efecto sea lineal o no.

En los gráficos que se muestran a continuación, se pueden ver tanto el estimador no paramétrico de la regresión como las bandas de variabilidad asociadas a dicha estimación, que representan la variabilidad del estimador, pero no el sesgo. Estas bandas son, punto a punto, un intervalo de confianza para el valor esperado de la regresión con una confianza fija, en este caso del 95%. Se puede destacar que en los puntos donde hay una mayor concentración de observaciones las bandas son más estrechas.

Fijándonos en la Figura 4.2 –superior izquierda– vemos como el estimador de Benceno es una función creciente, teniendo un tramo al principio casi lineal hasta la mitad de la función y otro tramo con un crecimiento más lento en la segunda mitad. Se puede apreciar cómo es una estimación regular sin grandes cambios en ella.

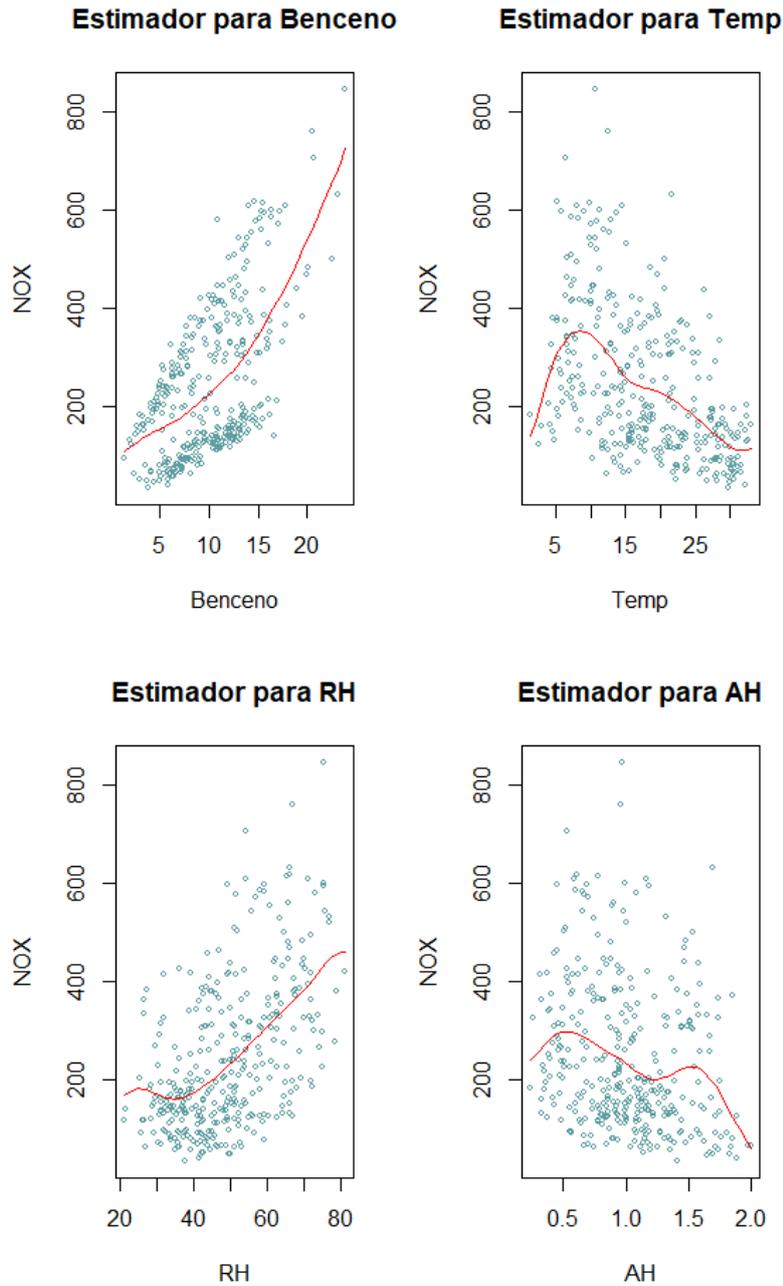


Figura 4.2: Estimadores no paramétricos de las regresiones univariantes obtenidas con regresión lineal local para calidad del aire.

Para la temperatura, Figura 4.2 –superior derecha– se puede apreciar como la función en su primer tramo es creciente en donde alcanza un máximo alrededor de 8 (aproximadamente) y después decrece, con esto se puede decir que para temperaturas altas la concentración NOX es baja con respecto a la que se consigue con temperaturas intermedias. Para la humedad relativa, Figura 4.2 –inferior izquierda– vemos como tiene un comportamiento parecido al benceno siendo una función creciente en su mayor parte y destacando que es casi lineal en el tramo intermedio y final de la estimación, lo que implica que para concentraciones altas de óxido nítrico tendremos valores altos para la humedad relativa. En este caso se puede apreciar un máximo y un mínimo para los valores bajos, esto puede ser debido a la

variabilidad que presentan los datos y a que son escasos para estos valores haciendo que la estimación oscile en estos valores.

Para la humedad absoluta, Figura 4.2 –inferior derecha– vemos como presenta dos máximos, entorno a valores de 0.5 y 1.5, esto hace que se tengan zonas de crecimiento y decrecimiento en la estimación. Al contrario que pasa con la humedad relativa, para concentraciones altas de NOX obtenemos por lo general valores bajos para la humedad absoluta. Por último, se puede apreciar que para los valores intermedios tenemos una gran dispersión de los datos.

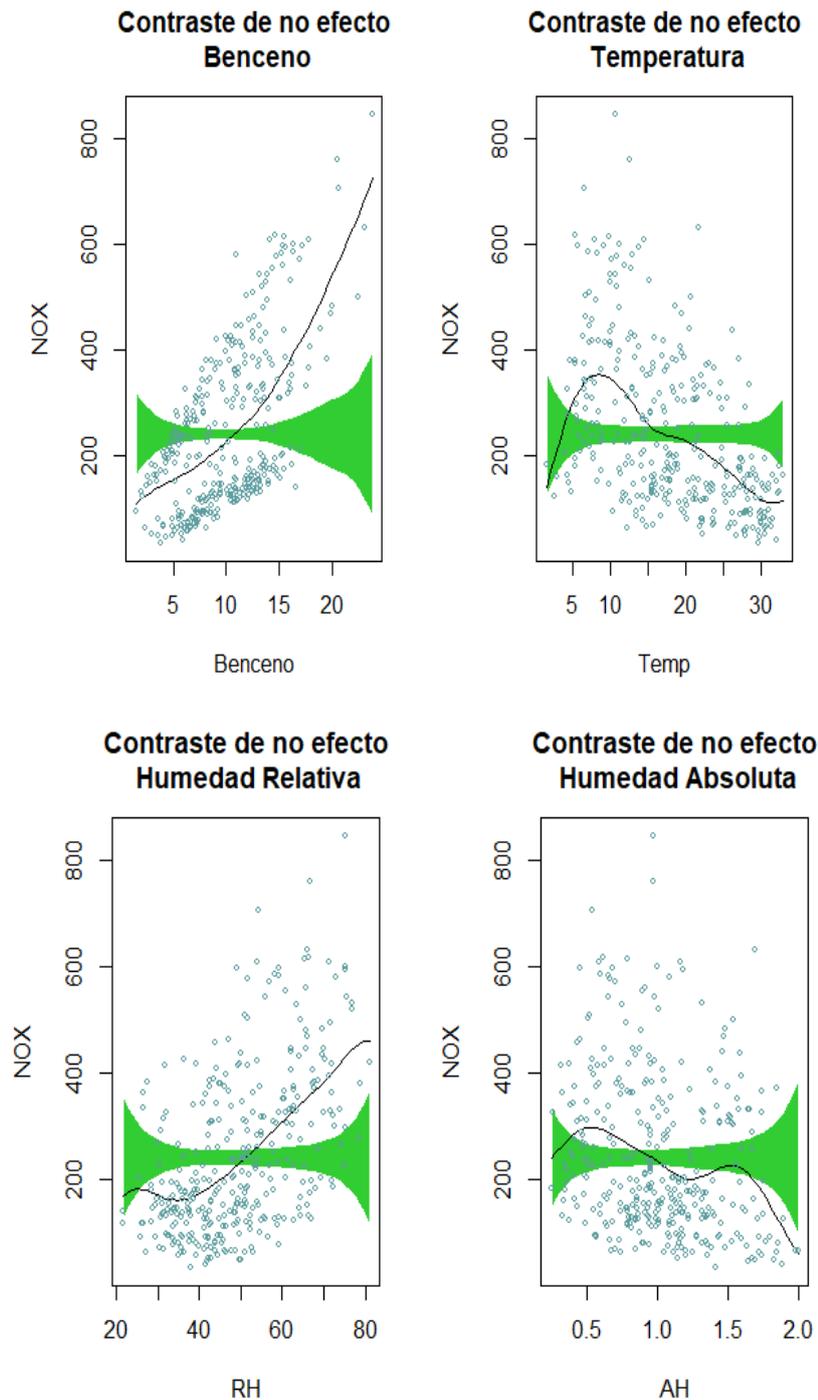


Figura 4.3: Contrastes de no efecto para cada predictor.

Teniendo en cuenta las estimaciones obtenidas de cada predictor, se puede ver que todas las variables tienen algún efecto sobre la variable respuesta, es decir, no podemos suponer para ninguna que la regresión es constante. De esta forma, siendo

$$H_0: E(Y|X = x) = cte \text{ (no hay efecto);}$$

$$H_1: E(Y|X = x) = m(X) \text{ (existe una relación suave),}$$

las hipótesis del contraste de no efecto y fijándonos en la Figura 4.3 en la que se representan las bandas de referencia a la hipótesis de no efecto para cada predictor, rechazamos la hipótesis nula. Partiendo de esto contrastaremos si dicha relación se puede suponer lineal o no para cada uno de los predictores, es decir

$$H_0: E(Y|X = x) = a + bX \text{ (relación lineal);}$$

$$H_1: E(Y|X = x) = m(X) \text{ (relación suave no lineal).}$$

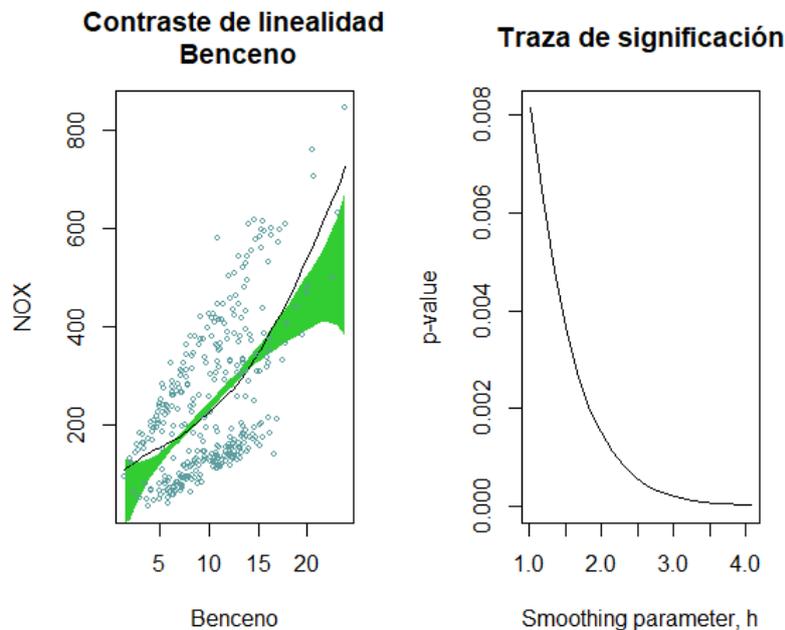


Figura 4.4: Contraste de linealidad para Benceno.

En la Figura 4.4 vemos como la estimación no paramétrica de la regresión sale de las bandas de referencia a la linealidad y observando la traza de significación, en la que para un entorno de  $h$  razonable obtenemos p-valores significativos, rechazamos la hipótesis de linealidad, es decir, vamos a considerar que el efecto que tiene el Benceno sobre NOX es una función no lineal suave.

En la Figura 4.5 se puede observar como la estimación sale de la banda de referencia para valores bajos de la variable y además los p-valores obtenidos para un entorno del  $h$  utilizado son todos significativos, con lo cual rechazamos la hipótesis de linealidad para esta variable.

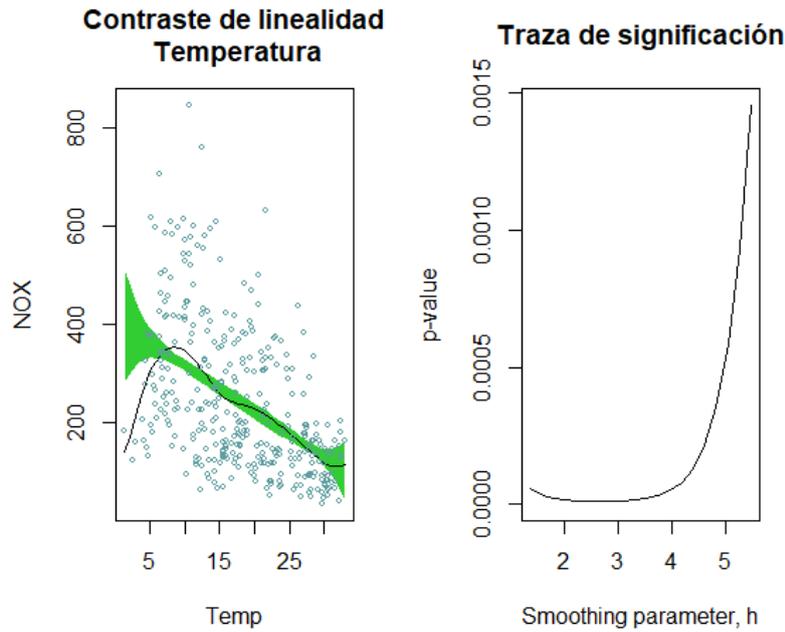


Figura 4.5: Contraste de linealidad para Temperatura.

En la Figura 4.6 podemos ver que, al igual que en el caso anterior, para valores bajos de la variable la estimación sale de las bandas de referencia a la linealidad, esto puede ser debido a que hay poca concentración de datos. Fijándonos en la traza de significación rechazamos la hipótesis de linealidad.

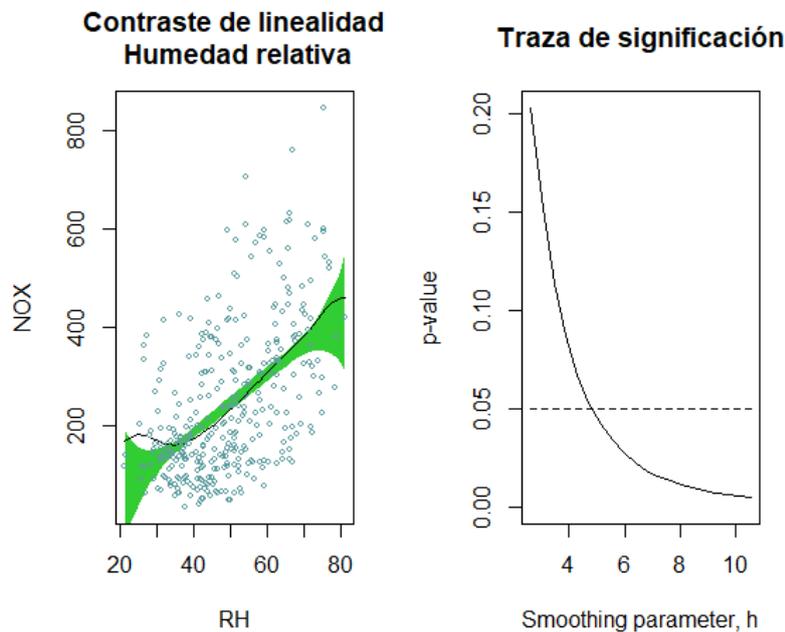


Figura 4.6: Contraste de linealidad para Humedad relativa.

En la Figura 4.7 se puede observar como para los valores intermedios hay una mayor variabilidad en los datos haciendo que la estimación de la regresión ondule y salga de las bandas de referencia, además, vemos como para un entorno de  $h$  razonable

rechazamos la hipótesis, con lo que para esta variable también consideraremos que el efecto que tiene sobre la variable respuesta es no lineal.

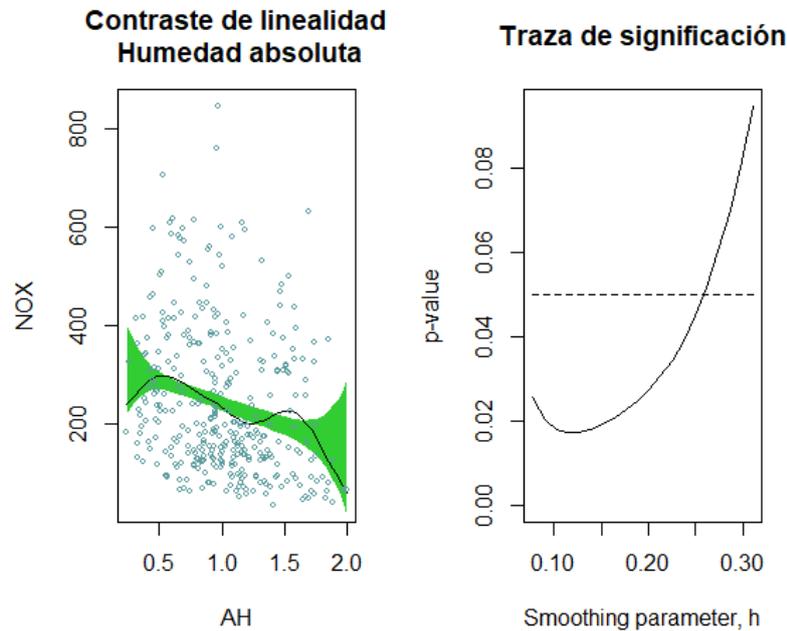


Figura 4.7: Contraste de linealidad para Humedad absoluta.

Teniendo en cuenta los resultados obtenidos anteriormente, es decir, los efectos que tienen los predictores sobre la variable respuesta, y ajustando cada una de las componentes del modelo aditivo mediante Loess, con polinomios de grado 1, empezaremos ajustando y comparando los siguientes modelos, asumiendo errores normales

$$\text{Modelo L1: } NOX = \alpha + m(\text{Benceno}) + \epsilon,$$

$$\text{Modelo L2: } NOX = \alpha + m(\text{Temp}) + \epsilon,$$

$$\text{Modelo L3: } NOX = \alpha + m(\text{RH}) + \epsilon,$$

$$\text{Modelo L4: } NOX = \alpha + m(\text{AH}) + \epsilon.$$

Como se puede ver se comenzarán ajustando modelos con solo una variable explicativa para ver cuál de estas es la que aporta más a la explicación de la respuesta, aunque esto se puede mirar con las correlaciones que tiene cada predictor con la variable respuestas lo haremos mediante el ajuste de modelos. La comparación entre los modelos la haremos mediante la deviance residual, seleccionando aquel que presente un menor valor, y mediante la comparación del AIC de cada modelo.

Con el fin de evitar un sobresuavizado de las funciones utilizaremos un span de 0.35, es decir, utilizaremos en 35% de las observaciones para el ajuste local de las funciones, excepto para la variable RH que se utilizara el 50% de las observaciones, es decir, un span de 0.5, debido a que con valores más bajo obtenemos estimaciones muy poco suaves.

	<b>Df residual</b>	<b>Dev residual</b>	<b>AIC</b>
<b>Modelo L1</b>	339.84	4975956	4322.128
<b>Modelo L2</b>	340.90	5512543	4355.543
<b>Modelo L3</b>	342.27	5586155	4357.418
<b>Modelo L4</b>	340.67	7110255	4444.318

Tabla 4.2: Deviance residual y AIC para modelos con un predictor.

Fijándonos en la Tabla 4.2, vemos como el modelo L1 es el que menor deviance residual presenta. Además, con la comparación del AIC vemos como es este el mejor modelo son una variable explicativa. Con esto vemos que la variable Benceno es la que más aporta a la explicación de la respuesta. Ahora, dejando esta variable fija añadiremos un segundo predictor y veremos si dicho predictor aporta algo al modelo.

Habiendo seleccionado ya un predictor, a partir de ahora haremos una comparación de modelos anidados, en el que miraremos el cambio que se produce en la deviance al añadir distintos predictores al modelo prefijado con un predictor. Dicho de otra forma, lo que queremos es contrastar si un nuevo predictor tiene algún efecto en el modelo o no, y esto se hará mediante el estadístico F y su correspondiente p-valor.

$$F = \frac{dev\ residual_{M1} - dev\ residual_{M2} / df\ residual_{M1} - df\ residual_{M2}}{dev\ residual_{M2} / df\ residual_{M2}} \quad (4.1)$$

En la formula anterior M1 y M2 hacen referencia al modelo 1 y al modelo 2 respectivamente, en el que M2 corresponde al modelo con el nuevo predictor. Para que quede más claro cada vez que se añada un predictor al modelo se calculara el estadístico F para el primer modelo que se compara, que servirá de ejemplo del cálculo. Los modelos con dos predictores son los siguientes

$$\text{Modelo L12: } NOX = \alpha + m_1(\text{Benceno}) + m_2(\text{Temp}) + \epsilon,$$

$$\text{Modelo L13: } NOX = \alpha + m_1(\text{Benceno}) + m_2(\text{RH}) + \epsilon,$$

$$\text{Modelo L14: } NOX = \alpha + m_1(\text{Benceno}) + m_2(\text{AH}) + \epsilon.$$

	<b>Df residual</b>	<b>Dev residual</b>	<b>ΔDf</b>	<b>ΔDev</b>	<b>F</b>	<b>p-valor</b>
<b>Modelo L1</b>	339.84	4975956				
<b>Modelo L12</b>	334.74	1987039	5.0978	2988916	98.774	2.2e-16
<b>Modelo L13</b>	336.11	3666793	3.7337	1309163	32.14	<2.2e-16
<b>Modelo L14</b>	334.52	3427226	5.3271	1548730	28.376	2.2e-16

Tabla 4.3: ANODEV para modelos con dos predictores.

$$F = \frac{dev\ residual_{L1} - dev\ residual_{L12} / df\ residual_{L1} - df\ residual_{L12}}{dev\ residual_{L12} / df\ residual_{L12}}$$

$$= \frac{4975956 - 1987039 / 339.84 - 334.74}{1987039 / 334.74} = 98.7$$

Con el análisis de la deviance, Tabla 4.3, vemos como los tres nuevos predictores tiene algún efecto en el modelo, ya que los p-valores son todos significativos y con cualquiera de los tres predictores la reducción de la deviance es significativa, y esto hace que se reduzcan los errores del modelo. Ahora, para seleccionar el segundo predictor a introducir en el modelo podemos destacar que el modelo L14 es el que menor grados de libertad residual presenta y el modelo L12 el que menos deviance residual tiene, que es lo que se pretende minimizar en el ajuste de los modelos aditivos, por lo que estos dos predictores son los que más aportan para la explicación de la respuesta. Pero para fijar el segundo predictor compararemos el AIC de los modelos.

	<b>AIC</b>
<b>Modelo L12</b>	4013.787
<b>Modelo L13</b>	4223.656
<b>Modelo L14</b>	4203.397

Tabla 4.4: AIC para modelos con dos predictores.

Observando el AIC en la Tabla 4.4, vemos como con la introducción de Temp se produce una reducción significativa del error con respecto a los otros modelos, con esto podemos decir que el modelo L12, que es el que introduce la variable temperatura, es el que mejor ajusta a los datos con dos predictores.

Ahora dejaremos fijas estas dos variables introduciremos una tercera y repetiremos lo anterior para ver si se consigue mejorar el modelo. Los modelos con tres predictores son

$$\text{Modelo L123: } NOX = \alpha + m_1(\text{Benceno}) + m_2(\text{Temp}) + m_3(\text{RH}) + \epsilon,$$

$$\text{Modelo L124: } NOX = \alpha + m_1(\text{Benceno}) + m_2(\text{Temp}) + m_3(\text{AH}) + \epsilon$$

	<b>Df residual</b>	<b>Dev residual</b>	<b>ΔDf</b>	<b>ΔDev</b>	<b>F</b>	<b>p-valor</b>
<b>Modelo L12</b>	334.74	1987039				
<b>Modelo L123</b>	331.01	1804425	3.7337	182614	8.9722	1.4e-6
<b>Modelo L124</b>	329.42	1792498	5.3271	194541	6.7113	3.2e-6

Tabla 4.5: ANODEV para modelos con tres predictores.

$$F = \frac{\text{dev residual}_{L12} - \text{dev residual}_{L123} / \text{df residual}_{L12} - \text{df residual}_{L123}}{\text{dev residual}_{L123} / \text{df residual}_{L123}}$$

$$= \frac{1987039 - 1804425 / 334.74 - 331.01}{1804425 / 331.01} = 8.9$$

En la Tabla 4.5 vemos como la reducción de la deviance es significativa para ambos modelos, con lo que ambos predictores tiene algún efecto en el modelo. La introducción de los nuevos predictores el modelo mejora y proporcionará mejor resultados que un modelo con dos predictores.

	AIC
<b>Modelo L123</b>	3987.802
<b>Modelo L124</b>	3988.688

Tabla 4.6: AIC para modelos con tres predictores.

Observando la Tabla 4.5 se puede ver como es el modelo con la variable Humedad absoluta es el que más reduce la deviance y el que menos grado de libertad residual presenta, pero si los comparamos con los del otro modelo vemos como la diferencia es muy pequeña entre ellos. Haciendo la comparación mediante el AIC vemos, en la Tabla 4.6, como es el modelo que tiene como nuevo predictor la Humedad relativa el que mejor ajuste proporciona, y al igual que pasa con la deviance y los grados de libertad se puede ver como los valores difieren poco entre sí.

Por último, ajustaremos un modelo introduciendo todos los predictores y lo compararemos con el mejor modelo con 3 predictores y veremos si la nueva variable tiene algún efecto en el modelo.

$$\text{Modelo LC: } NOX = \alpha + m_1(\text{Benceno}) + m_2(\text{Temp}) + m_3(\text{RH}) + m_4(\text{AH}) + \epsilon$$

	Df residual	Dev residual	$\Delta Df$	$\Delta Dev$	F	p-valor
<b>Modelo L123</b>	331.01	1804425				
<b>Modelo LC</b>	325.68	1702336	5.3271	102090	3.6664	0.0024

Tabla 4.7: ANODEV para el modelo con cuatro predictores.

$$F = \frac{\text{dev residual}_{L123} - \text{dev residual}_{LC} / df \text{ residual}_{L123} - df \text{ residual}_{LC}}{\text{dev residual}_{LC} / df \text{ residual}_{LC}}$$

$$= \frac{1804425 - 1702336 / 331.01 - 325.68}{1702336 / 325.68} = 3.6$$

En la Tabla 4.7 se puede observar como la reducción de la deviance es significativa con lo que la introducción de los cuatro predictores mejora el modelo ya que todos tienen algún efecto en este modelo. De esta forma podemos ver como el mejor modelo que se ajusta a los datos es aquel que tiene cuatro predictores

$$NOX = \alpha + m_1(\text{Benceno}) + m_2(\text{Temp}) + m_3(\text{RH}) + m_4(\text{AH}) + \epsilon.$$

En los gráficos de la Figura 4.8 se representan las funciones componentes estimadas en el modelo completo, LC, para cada una de las variables explicativas, ajustadas mediante el algoritmo backfitting. En la Figura 4.8 –superior izquierda– vemos como la estimación de la función componente para la variable Benceno es una función creciente, con lo que fijadas las otras variables, para concentraciones altas o bajas de benceno tendremos concentraciones altas o bajas para NOX respectivamente.

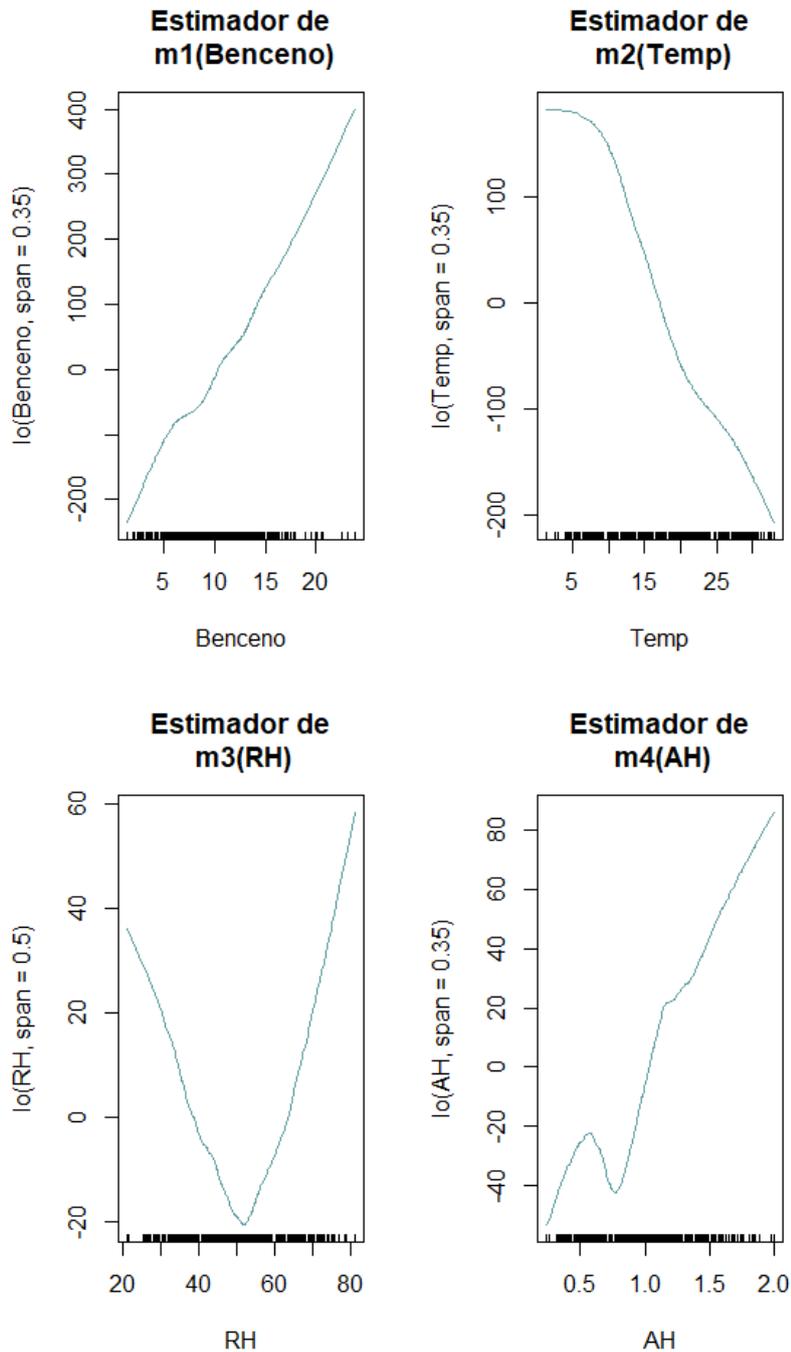


Figura 4.8: Estimación de las funciones componentes mediante Backfitting con loess.

Para la variable Temp, Figura 4.8 –superior derecha– vemos el efecto contrario que el Benceno, tenemos una curva decreciente que representa la influencia de esta variable sobre la respuesta, además esta estimación tiene pocas ondulaciones pudiéndose observar que el cambio más apreciable se encuentra en los valores bajos de Temp. Con esto podemos decir que, fijadas las otras variables, las concentraciones bajas de NOX se obtiene con temperaturas altas. Para la variable RH, Figura 4.8 –inferior izquierda– vemos como la estimación presenta forma de parábola, teniendo un mínimo en torno a 52, aproximadamente, con esto podemos ver que, fijadas las otras

variables, para valores bajos y altos de humedad relativa obtenemos valores altos para la concentración de óxidos de nitrógeno. También se puede apreciar como la estimación presenta pocas ondulaciones. Para AH, Figura 4.8 –inferior derecha– se podría decir que es la más irregular respecto a las otras 3, presentando zonas de crecimiento y decrecimiento, con un máximo y un mínimo. Destaca que en la zona central el crecimiento es prácticamente lineal. Al ser una función en su mayor parte creciente, fijadas las otras variables, a medida que aumenta la concentración de NOX aumenta los valores de la Humedad absoluta.

Comparando las estimaciones de la Figura 4.2: Estimadores no paramétricos de las regresiones univariantes obtenidas con regresión lineal local para calidad del aire. Figura 4.2 con las estimaciones obtenidas con el modelo aditivo en la Figura 4.8 se puede apreciar como la estimación mediante backfitting es bastante diferente a la estimación que se obtiene con modelos en el que la variable respuesta está afectada solo por un predictor, la diferencia de las estimaciones es debido al número de predictores que afecta a la variable respuesta.

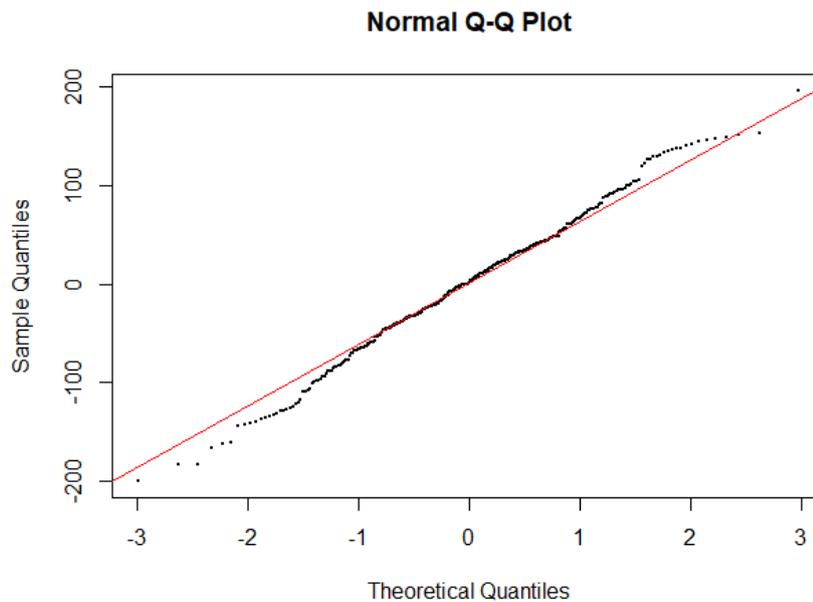


Figura 4.9: Q-Q plot para los residuos del modelo seleccionado ajustado con loess.

Observando el Q-Q plot de la Figura 4.9 vemos como los residuos se ajustan bastante a la normalidad, aunque se puede destacar que en los extremos del grafico los residuos tienen comportamientos ondulatorios alejándose un poco de la recta, aun así, podemos decir que este modelo es un buen modelo para los datos.

Ahora realizaremos el procedimiento anterior, pero ajustaremos cada predictor mediante un spline cubico en el que usaremos 4 grados de libertad en cada uno de los ajustes, siendo este el valor por defecto de la función  $s()$  de la librería `gam`. Al igual

que antes comenzaremos ajustando los modelos con un solo predictor e iremos añadiendo uno a uno hasta llegar a un modelo con todos los predictores. Después realizaremos una comparación de los dos métodos de ajuste, Loess y Spline, con el mejor modelo obtenido en cada caso.

Como esto ya se ha hecho paso a paso en el caso anterior, solo compararemos los modelos que mejor resultados proporcionen con un número de predictores fijos, es decir, el mejor con dos predictores, el mejor con tres y así sucesivamente.

Los modelos seleccionados para la comparación son los siguientes, la selección se ha hecho comparando el AIC en cada caso:

$$\text{Modelo S1: } NOX = \alpha + m(\text{Benceno}) + \epsilon,$$

$$\text{Modelo S12: } NOX = \alpha + m_1(\text{Benceno}) + m_2(\text{Temp}) + \epsilon,$$

$$\text{Modelo S123: } NOX = \alpha + m_1(\text{Benceno}) + m_2(\text{Temp}) + m_3(\text{RH}) + \epsilon,$$

$$\text{Modelo SC: } NOX = \alpha + m_1(\text{Benceno}) + m_2(\text{Temp}) + m_3(\text{RH}) + m_4(\text{AH}) + \epsilon.$$

	<b>Df residual</b>	<b>Dev residual</b>	<b><math>\Delta Df</math></b>	<b><math>\Delta Dev</math></b>	<b>F</b>	<b>P-valor</b>
<b>Modelo S1</b>	342	5006213				
<b>Modelo S12</b>	338	2019737	4.0003	2986476	124.94	<2.2e-16
<b>Modelo S123</b>	334	1841583	4.0000	178154	8.0777	3.2e-6
<b>Modelo SC</b>	<b>330</b>	<b>1770570</b>	4.0003	71013	3.3087	0.0112

Tabla 4.8: ANODEV para modelos anidados ajustados mediante spline.

Como se puede apreciar en la Tabla 4.8 vemos como al ir introduciendo cada predictor al modelo la reducción de la deviance es significativa, es decir, todos los predictores tiene algún efecto en el modelo que ayudan a explicar la variable respuesta. Al igual que pasa con el ajuste con Loess, el mejor modelo con splines que se ajusta a los datos es el que incorpora todos los predictores.

En los gráficos de la Figura 4.10 se representan las estimaciones no paramétricas de cada una de las funciones componentes de cada variable explicativa, ajustadas mediante el algoritmo backfitting. Ahora, fijándonos en la Figura 4.8 y en la Figura 4.10, que se encuentra a continuación, vemos como en el modelo en el que ajustamos cada componente con un spline obtenemos estimaciones más suaves que utilizando Loess. Ahora, respecto a cada una de las estimaciones vemos como son muy parecidas a las que se obtienen con loess solo que, con menos ondulaciones a lo largo de la estimación por lo que se llegara a las mismas conclusiones a las que se han llegado anteriormente.

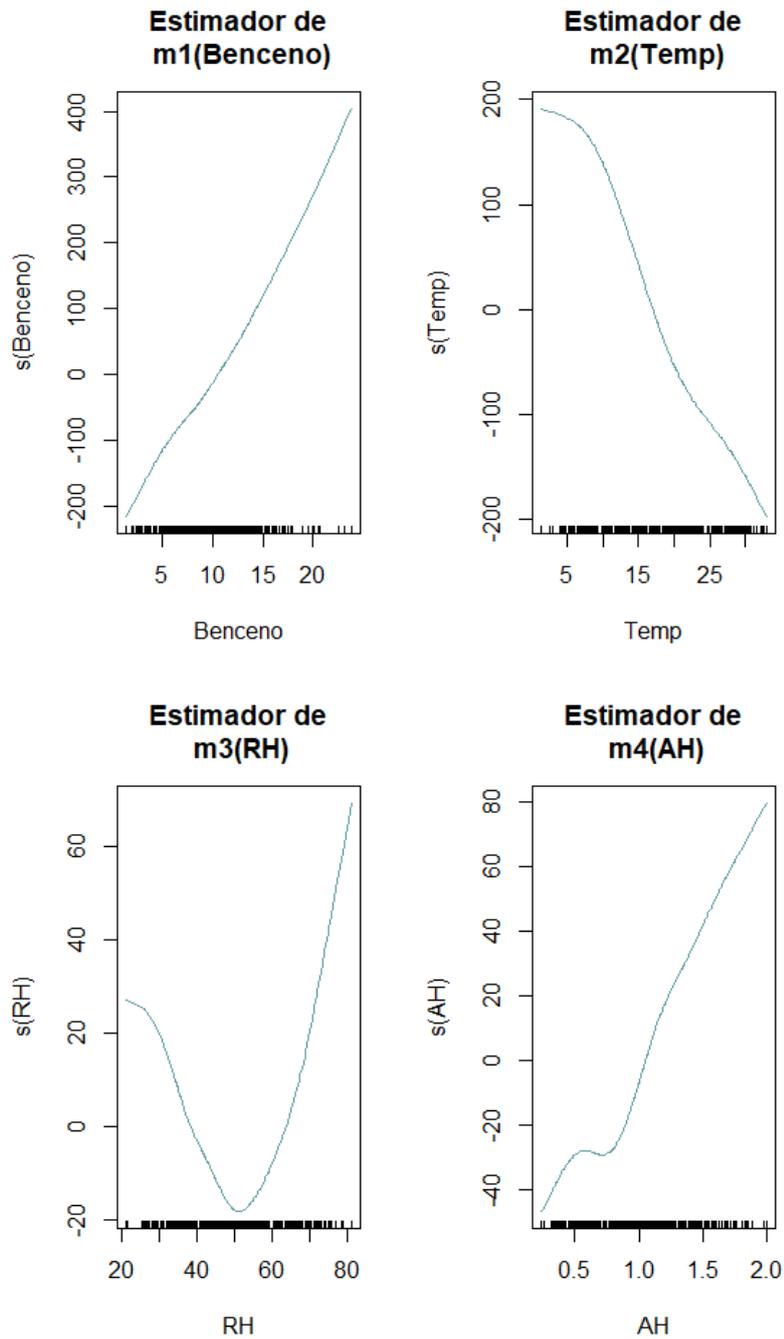


Figura 4.10: Estimación de las funciones componente mediante Backfitting con spline.

Observando el Q-Q plot de la Figura 4.11, que se encuentra a continuación, vemos que al igual que en el caso del modelo anterior los residuos se ajustan bastante a la normalidad. Con lo que este modelo es un buen ajuste para los datos.

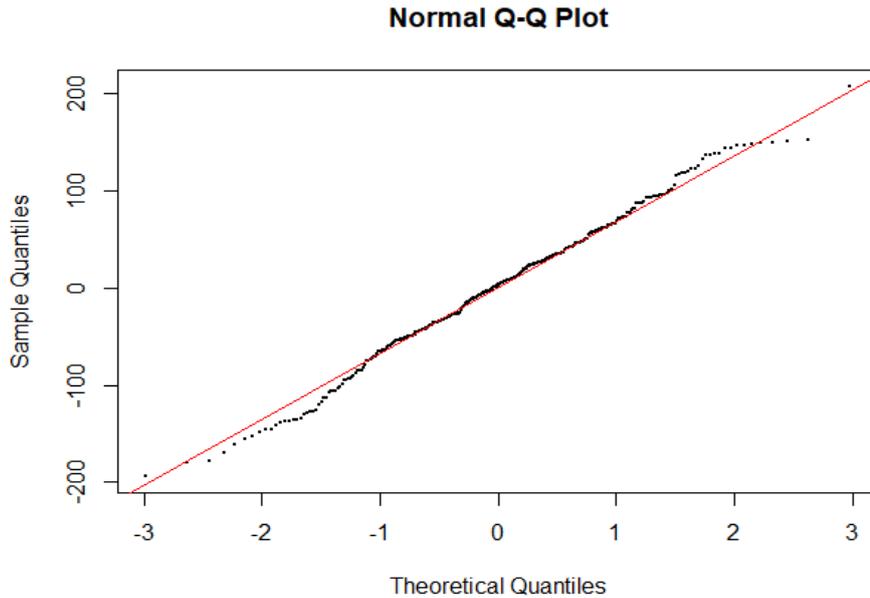


Figura 4.11: Q-Q plot para los residuos del modelo completo ajustado con spline.

Por último, compararemos los dos modelos obtenidos con cada método de ajuste y veremos cual nos proporciona un mejor ajuste para los datos

	<i>Deviance residual</i>	<i>AIC</i>
<b>Modelo Loess</b>	1702336	3978.247
<b>Modelo Spline</b>	1770570	3983.253

Tabla 4.9: AIC y Deviance residual para los modelos seleccionados para Calidad del aire.

Observando la Tabla 4.9 podemos ver como el modelo que se ajusta con loess, en cada una de las componentes, es el que menor deviance residual presenta y el que tiene un AIC más pequeño, es decir, es el que más va a reducir los errores. Con esto podemos concluir que para estos datos se obtiene una mejor estimación con loess que con spline.

## 4.2 Datos simulados

Al igual que en el caso de datos reales realizaremos una comparación del backfitting utilizando los dos suavizadores univariantes sobre unos datos simulados. Con lo que se realizara el mismo procedimiento anterior, se ajustaran modelos con distinto número de predictores para ver la influencia de cada uno sobre la respuesta y se buscara el mejor modelo que se ajuste a los datos.

Consideraremos un tamaño de muestras de  $n = 200$  y las siguientes funciones para los predictores

$$m_1(x_{1i}) = -\sqrt{x_{1i}} + \sin(x_{1i}) * x_{1i} \text{ con valores para } x_{1i} \text{ de una uniforme, } \mathcal{U}(0,4).$$

$$m_2(x_{2i}) = \cos(x_{2i}^2) * x_{2i}^2 \text{ con valores para } x_{2i} \text{ de una uniforme, } \mathcal{U}(-2,2).$$

$$m_3(x_{3i}) = x_{3i}^3 - 3 * x_{3i} \text{ con valores para } x_{3i} \text{ de una uniforme, } \mathcal{U}(-2,2).$$

$m_4(x_{4i}) = 3$  con valores para  $x_{4i}$  de una uniforme,  $\mathcal{U}(4,8)$

La variable respuesta será entonces la suma de los predictores más un error aleatorio, es decir

$$y_i = m_1(x_{1i}) + m_2(x_{2i}) + m_3(x_{3i}) + m_4(x_{4i}) + \epsilon_i, \quad i = 1, \dots, n$$

donde  $\epsilon_i$  son los errores aleatorios independiente con  $E(\epsilon_i) = 0$  y  $\sigma = 0.4$ . Ahora supondremos que las funciones  $m_i$  son desconocidas y las estimaremos mediante el algoritmo backfitting y haremos una comparación con las regresiones reales de cada predictor suponiendo los otros dos fijos para poder ver el comportamiento y las diferencias que produce cada suavizador univariante en la estimación.

Comenzaremos haciendo una descripción de cada una de las variables y un análisis de las interacciones que tienen entre si

	<i>n</i>	<i>Min</i>	<i>Max</i>	<i>Media</i>	<i>Desviación típica</i>
<b>Y</b>	200	-6.788	5.627	1.431	2.309
<b>X<sub>1</sub></b>	200	0.024	3.997	2.039	1.184
<b>X<sub>2</sub></b>	200	-1.998	1.980	-0.068	1.170
<b>X<sub>3</sub></b>	200	-1.999	1.953	-0.667	1.117
<b>X<sub>4</sub></b>	200	4.025	7.985	6.057	1.169

Tabla 4.10: Descriptivos para las variables de la simulación.

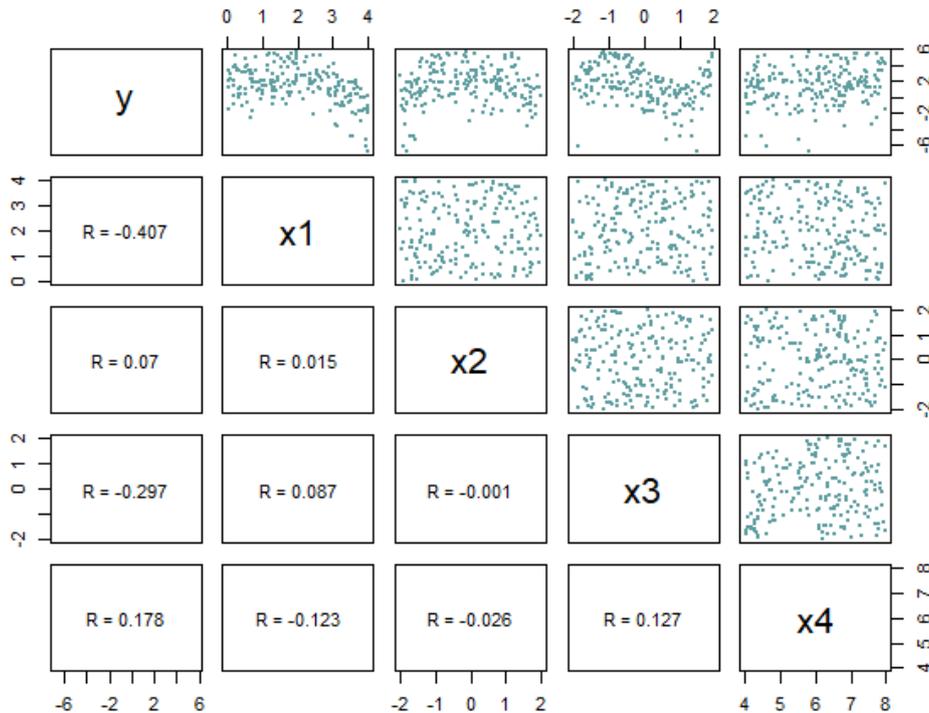


Figura 4.12: Matriz de dispersión y correlaciones para las variables de la simulación.

Observando la Figura 4.12 vemos como la variable respuesta tiene correlaciones bajas con los cuatro predictores, siendo negativa con  $X_1$  y  $X_3$ , presentando el valor más alto

con  $X_1$ . Con respecto a  $X_2$  y  $X_4$  vemos como la correlación es positiva, pero estas son más bajas, en valor absoluto que las otras dos, siendo la más baja la que presenta con  $X_2$ . También se puede ver como la correlación entre los predictores son muy bajas, estando cercanas a cero en la mayoría de los casos. Se puede destacar que el predictor  $X_4$  es que tiene las correlaciones más altas con los otros predictores, en torno a 0.1. Esto se puede corroborar observando los gráficos de dispersión en los cuales vemos como los puntos no siguen ningún patrón y están dispersos de manera aleatoria sobre el gráfico. Al contrario de lo que pasa con la variable respuesta que se pueden intuir ciertas formas en los gráficos con los predictores.

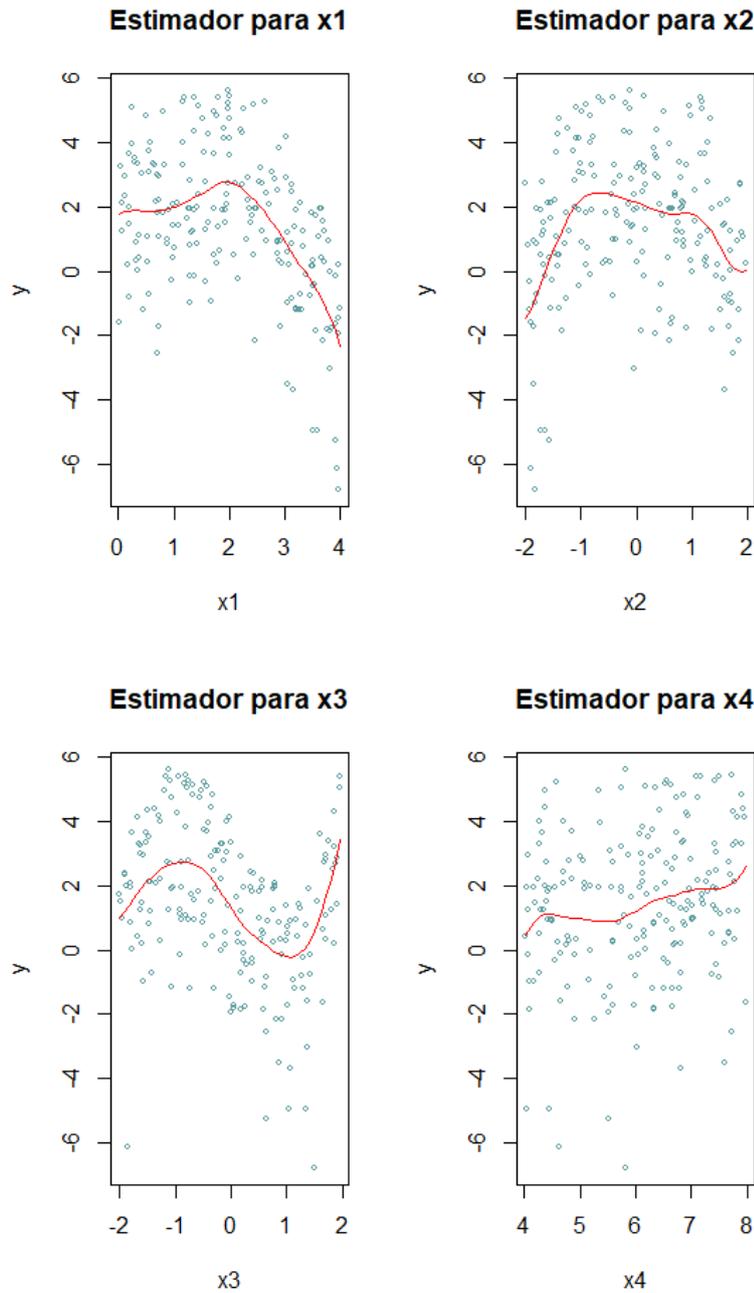


Figura 4.13: Estimadores no paramétricos de las regresiones univariantes obtenidas con regresión lineal local para la simulación.

Ahora, considerando solamente un predictor en el ajuste, realizaremos un análisis más detallado de la relación que tiene cada uno de los predictores con la variable respuesta, para ello se obtendrá un estimador no paramétrico de la regresión para cada uno de los predictores esto se hará con un ajuste lineal local mediante la función `sm.regression()`, con esto pretendemos hacernos una idea de la influencia de cada predictor en la variable.

En la Figura 4.13 podemos ver las estimaciones de la regresión. Para  $X_1$  –superior izquierda– vemos como la estimación tiene un primer tramo en el cual crece muy lentamente, tras el cual alcanza un máximo y después hay una zona de decrecimiento suave. Para  $X_2$  –superior derecha– vemos como la mayor parte de la estimación es decreciente, aunque con un tramo de crecimiento al principio en donde se alcanza un máximo también presenta un tramo que permanece constante tras el cual se puede apreciar un decrecimiento más rápido de la función. Para  $X_3$  –inferior izquierda– la función estimada presenta zonas de crecimiento y decrecimiento esto hace que se encuentren un máximo y un mínimo a lo largo de la estimación, podemos destacar que el primer crecimiento y el decrecimiento son suaves respecto al segundo tramo de crecimiento el cual presenta una pendiente más pronunciada. Para  $X_4$  –inferior derecha– se puede ver como la estimación es creciente en su mayor parte, con pequeñas ondulaciones suaves a lo largo de ella.

Para este supuesto práctico no realizaremos los contrastes de ‘no efecto’ y de ‘linealidad’ para las primera tres variables ya que observando las gráficas de las estimaciones se puede ver como los tres predictores tienen efecto sobre la variable respuesta y ninguno de estos efectos se puede suponer lineal. Para la variable  $X_4$  como los puntos en el gráfico de dispersión no presentan ninguna forma concreta y además hay gran variabilidad entre ellos, contrastaremos si esta variable tiene algún efecto sobre la variable respuesta, y en caso de tenerlo miraremos si dicho efecto se puede suponer lineal.

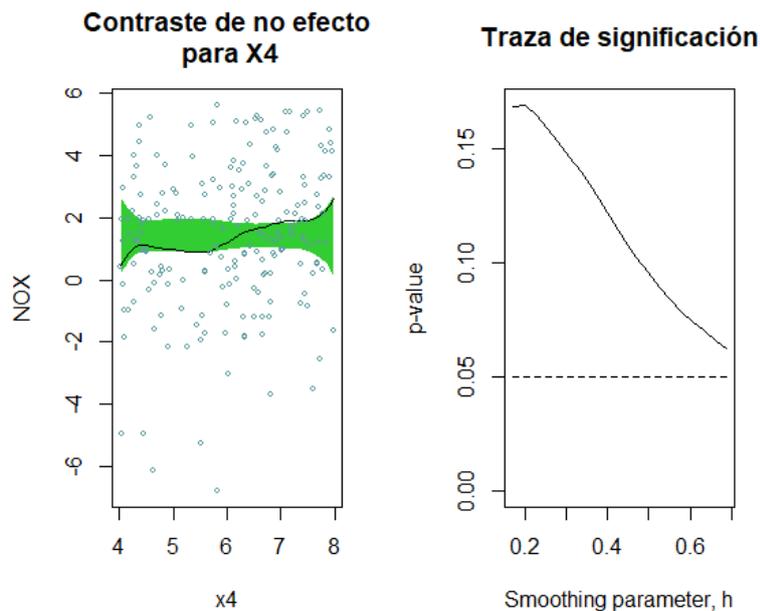


Figura 4.14: Contraste de linealidad para  $X_4$ .

En la Figura 4.14 vemos que, aunque la estimación de la regresión salga de las bandas de referencia al contraste de no efecto observando la traza de significación, no rechazamos la hipótesis nula del contraste de no efecto, es decir, vamos a suponer que esta regresión es una constante, con lo cual esta variable no influirá para la explicación de la variable respuesta.

Teniendo en cuenta el análisis hecho y los resultados obtenidos hasta ahora comenzaremos ajustando modelos con un predictor, después con dos y así sucesivamente hasta llegar al modelo en cual incluya todos los predictores, esto lo haremos tanto para un ajuste con loess, con un span de 0.35, como con splines. Para la comparación de los modelos no haremos una comparación exhaustiva entre los modelos ya que es el mismo procedimiento que se ha hecho con el conjunto de datos anterior, sino que seleccionaremos en cada caso el mejor modelo y haremos la comparación de estos modelos, anidados, para ver la influencia de cada uno de los predictores.

Los modelos que compararemos son los siguientes, en este caso destacamos que la elección se ha hecho comparando el AIC y la deviance residual.

$$\text{Modelo L1: } Y = \alpha + lo(X_1) + \epsilon,$$

$$\text{Modelo L13: } Y = \alpha + lo_1(X_1) + lo_2(X_3) + \epsilon,$$

$$\text{Modelo L132: } Y = \alpha + lo_1(X_1) + lo_2(X_3) + lo_3(X_2) + \epsilon.$$

Destacamos que en el ajuste con un solo predictor no se ha ajustado el correspondiente a la variable  $X_4$ , ya que se ha visto que esta variable sola no tiene ningún efecto en la variable respuesta. En los modelos en los que se incluyen dos y tres, tras realizar el contraste correspondiente el cual se basa en la deviance, esta variable sigue sin aportar nada al modelo. Respecto al modelo con todos los predictores se comparará junto a los otros tres seleccionados anteriormente.

$$\text{Modelo LC: } Y = \alpha + lo_1(X_1) + lo_2(X_3) + lo_3(X_2) + lo_4(X_4) + \epsilon.$$

	<b>Df residual</b>	<b>Dev residual</b>	<b><math>\Delta Df</math></b>	<b><math>\Delta Dev</math></b>	<b>F</b>	<b>P-valor</b>
<b>Modelo L1</b>	194.08	687.75				
<b>Modelo L13</b>	189.08	359.28	4.9950	328.47	34.608	<2.2e-16
<b>Modelo L132</b>	184.12	47.84	4.6995	311.45	241.22	<2.2e-16
<b>Modelo LC</b>	179.16	46.86	4.9560	0.9793	0.7555	0.582

Tabla 4.11: ANODEV para modelos de la simulación ajustados con LOESS

La comparación de los modelos se hace justo con el modelo anterior y la comparación se basa en si cada predictor tiene o no efecto en el modelo y esto se contrasta con la reducción de la deviance mediante el estadístico F, que se calcula de la misma forma que en el caso anterior.

$$F = \frac{dev\ residual_{L1} - dev\ residual_{L13} / df\ residual_{L1} - df\ residual_{L13}}{dev\ residual_{L13} / df\ residual_{L13}}$$

$$= \frac{687.75 - 359.28 / 194.08 - 189.08}{359.28 / 189.08} = 34.6$$

Observando los p-valores obtenidos en la Tabla 4.11 vemos como el correspondiente al modelo LC es no significativo, es decir, que la introducción de la variable  $X_4$  no tienen ningún efecto en el modelo, ya sea solo o en la presencia de algún otro predictor, con lo que esta variable no se incluirá en el modelo final. Comparando los modelos en los que vemos que la reducción de la deviance es significativa, lo que implica que los predictores tienen algún efecto en la explicación de la respuesta, vemos como es el modelo que tiene tres predictores el que mejores resultados proporciona, ya que es el que presenta una menor deviance residual y un menor número de grados de libertad residual.

Las estimaciones no paramétricas de la regresión obtenidas con el backfitting en el modelo seleccionado con tres predictores se pueden ver en la Figura 4.15 que se encuentra a continuación, junto a ellas también se pueden ver la verdadera regresión de cada predictor, suponiendo los otros dos fijos.

Como se puede observar en este caso las estimaciones se encuentra por encima de la verdadera regresión. Se puede destacar que las bandas de variabilidad son estrechas a lo largo de toda la estimación en los tres predictores.

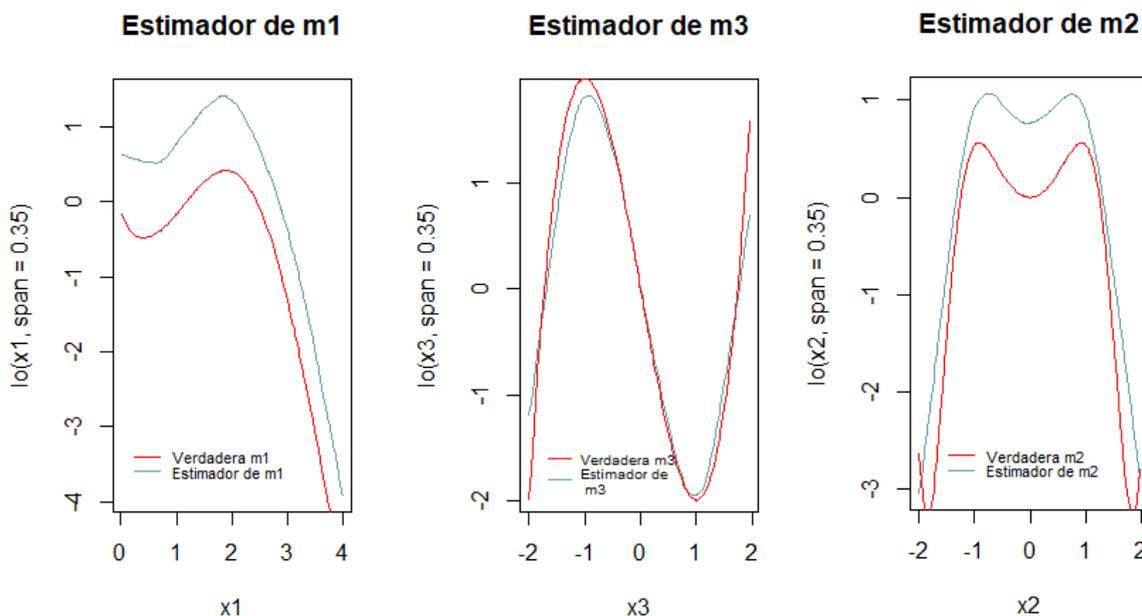


Figura 4.15: Estimación de las funciones componentes mediante Backfitting con loess.

Respecto a la comparación con la verdadera regresión vemos en la Figura 4.15 como para  $X_3$  –centro– se obtiene una muy buena estimación, siendo prácticamente la misma en la zona de decrecimiento de la función, variando poco en las zonas de

crecimiento. Para  $X_1$  –izquierda– podemos ver como la estimación con el backfitting consigue una estimación muy parecida en forma a la verdadera, pero esta se encuentra sobreestimada. Por último, para  $X_2$  –derecha– al igual que en la estimación anterior, se consigue una función que se parece bastante a la verdadera regresión, pero sobreestimada también. Por otra parte, se puede ver no estima bien los mínimos que presenta la función en los extremos y suaviza el que se encuentra entre los dos máximos.

Ahora, para el ajuste de cada una de las componentes mediante splines nos fijaremos en los siguientes modelos. En este caso ajustaremos cada spline con 7 grados de libertad, ya que las estimaciones que se consiguen con  $fd = 4$  son demasiado suaves.

$$\text{Modelo S1: } Y = \alpha + s(X_1) + \epsilon,$$

$$\text{Modelo S13: } Y = \alpha + s_1(X_1) + s_2(X_3) + \epsilon,$$

$$\text{Modelo S132: } Y = \alpha + s_1(X_1) + s_2(X_3) + s_3(X_2) + \epsilon,$$

$$\text{Modelo SC: } Y = \alpha + s_1(X_1) + s_2(X_3) + s_3(X_2) + s_4(X_4) + \epsilon.$$

	<b>Df residual</b>	<b>Dev residual</b>	<b><math>\Delta Df</math></b>	<b><math>\Delta Dev</math></b>	<b>F</b>	<b>P-valor</b>
<b>Modelo S1</b>	192	676.62				
<b>Modelo S13</b>	185	339.02	7.0001	337.6	26.317	<2.2e-16
<b>Modelo S132</b>	178	33.719	7.0004	305.3	230.22	<2.2e-16
<b>Modelo SC</b>	171	31.681	7.0004	2.034	1.5716	0.1468

Tabla 4.12: ANODEV para modelos de la simulación ajustados con Splines.

Con la Tabla 4.12 vemos como para este ajuste pasa lo mismo que en el ajuste anterior para la variable  $X_4$ , además, podemos ver el orden de introducción de los predictores es el mismo que se consigue con Loess. Observando los p-valoros significativos de los modelos vemos como los tres primeros predictores influyen en la respuesta, es decir, tienen algún efecto ella. También se puede destacar que con la introducción de cada predictor se consigue una reducción de la deviance residual bastante significativa respecto al anterior modelo. El modelo S132 es el que mejor se ajusta a estos datos.

Las estimaciones no paramétricas de la regresión obtenidas con los spline mediante el algoritmo bckfitting se pueden ver en la Figura 4.16. Ajustando cada componente con 7 grados de libertad vemos como conseguimos funciones parecidas, en forma, a la regresión verdadera, ya que cuanto mayor es este parámetro, menos suaves son las funciones estimadas. Comparando estas estimaciones con las obtenidas con Loess, vemos que para  $X_3$  –centro– se consigue una muy buena estimación ya que en general la función estimada es prácticamente igual a la verdadera regresión, la diferencia más apreciable se encuentra en la estimación del máximo y el mínimo. Para  $X_2$  –derecha– vemos como conseguimos una estimación mejor ya que se consigue que el mínimo no se suavice demasiado con respecto a la verdadera regresión, aunque se encuentre sobreestimada. Respecto a  $X_1$  –izquierda– las estimaciones no varían mucho de un ajuste a otro.

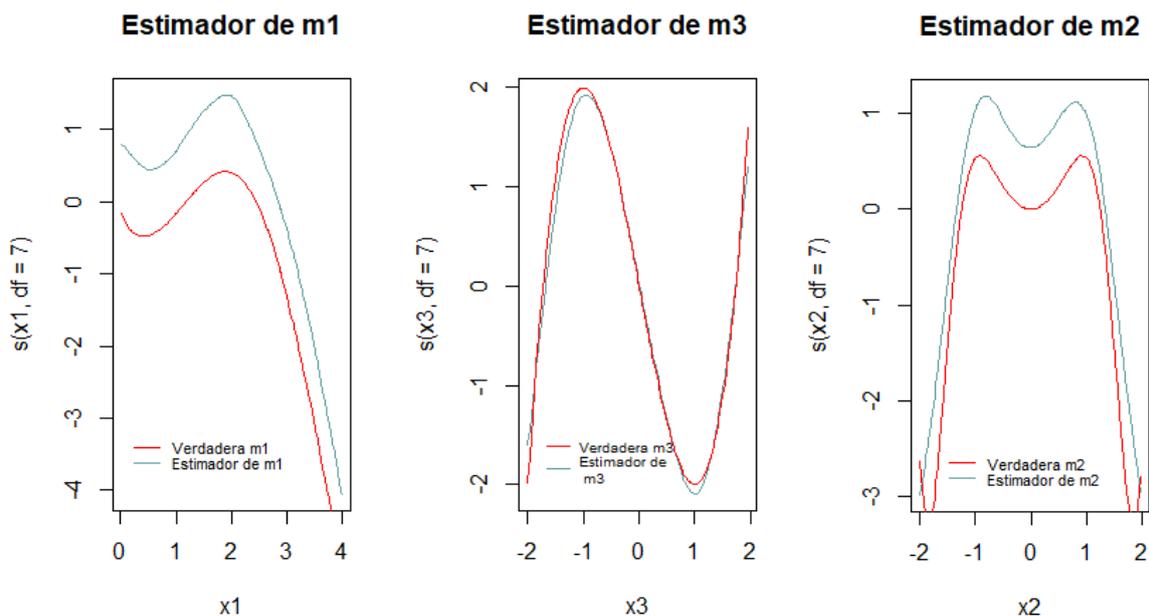


Figura 4.16: Estimación de las funciones componentes mediante Backfitting con spline.

Por último, compararemos mediante los modelos obtenidos que suavizador univariante se ajusta mejor a los datos simulados comparando la deviance residual y el AIC de cada uno

	<i>Deviance residual</i>	<i>AIC</i>
<b>Modelo Loess</b>	41.3383	286.7473
<b>Modelo Spline (df = 7)</b>	27.1389	214.1048

Tabla 4.13: AIC y Deviance residual para los modelos con Loes y Spline de la simulación.

Comparando los resultados obtenido de la Tabla 4.13 se puede ver como el modelo que ajusta mediante Loess se podría considerar un buen modelo ya que como se ha visto se obtiene buenas estimaciones para cada variable explicativa, pero el que mejor resultados proporciona es el ajuste con spline con 7 grados de libertad en cada una de las componentes, ya que la reducción de la deviance que se obtiene con este es bastante significativa respecto al ajuste con loess, además, es el que menor AIC presenta.

Al contrario de lo que pasa con los datos reales, vemos como en este caso el que mejor ajuste proporciona a los datos es la estimación con spline, en el cual se han ido variando los grados de libertad con los que se ajusta a cada componente para reducir la suavidad de las estimaciones y a su vez reducir la deviance residual para obtener un mejor modelo.



# Conclusiones

La regresión no paramétrica es una alternativa para la modelización de datos en los que la suposición de linealidad que imponen los modelos lineales no es adecuada, ya que tal restricción proporcionaría resultados poco fiables. Cuando nos encontramos en dimensiones altas la regresión no paramétrica tampoco es adecuada, ya que se necesitarían conjuntos de datos extremadamente grandes para que las propiedades de las estimaciones no se deterioren, y en muchos casos no se dispone de tal cantidad de datos para el ajuste.

A lo largo de este trabajo se ha comprobado como los modelos aditivos son una alternativa que solucionan los problemas anteriores, ya que permiten estimar las funciones componentes univariantes de forma no paramétrica para cada uno de los predictores mediante el algoritmo backfitting. El ajuste se hace por mínimos cuadrados penalizados para tener en cuenta la suavidad de las funciones de la solución.

También se ha visto que para el caso de suavizadores lineales existe una condición de convergencia para el caso de dos suavizadores y un criterio que asegura la convergencia para el caso de  $p$  suavizadores. Y que con el centrado de las matrices de suavizado la solución del backfitting no depende de las funciones iniciales que se proporcionan.

Uno de los principales problemas que presentan estos modelos es que en algunos aspectos la teoría está poco desarrollada. Como la distribución de la deviance que es desconocida incluso asintóticamente, pero mediante simulaciones se ha visto que la distribución  $\chi^2$  es útil. Otro problema no resuelto es la selección de los parámetros de suavizado, que se hace de manera subjetiva y como se ha visto, en este trabajo, para el ajuste con loess está determinado por el  $k$ -ésimo vecino más cercano, es decir, el número de observaciones que queremos que intervengan en el ajuste local una vez fijado un  $x$ . Para el caso de los spline este está determinado por los grados de libertad, que se calculan como  $df = tr(S) - 1$ .

Por último, se ha visto mediante la implementación de estos modelos en R que, en diferentes conjuntos de datos, reales y simulados, estos modelos proporcionan buenos ajustes independientemente del suavizador lineal que se utilice. En los datos de calidad del aire se ha visto como el mejor modelo, para el caso de ambos suavizadores lineales, es aquel que incorpora todos los predictores en el ajuste, es decir, todos los predictores que se han utilizado tienen algún efecto en la respuesta. En el que para las variables benceno y humedad absoluta, estando las otras tres fijas, presentan una tendencia creciente. Una tendencia decreciente con la variable temperatura, estando las otras tres fijas, y para la variable humedad relativa se ve como tiene una tendencia decreciente hasta la mitad del rango y después una tendencia creciente. Para ambos suavizadores obtenemos estimaciones muy parecidas siendo más suaves con el spline.

En los datos simulados que se han introducido cuatro predictores, en el que uno de ellos no aporta a la explicación de la respuesta, vemos como el mejor modelo es el que incorpora en el ajuste tres predictores. Con esto se aprecia como el algoritmo backfitting detecta aquellas variables que no son significativas para el ajuste. Respecto a las variables significativas vemos, como para ambos ajustes, se obtienen buenas estimaciones, ya que las funciones obtenidas con el backfitting son muy parecidas a las funciones reales.

# Bibliografía

Buja, A., Hastie, T.J. and Tibshirani, R. (1989). *Linear smoothers and additive models*. The Annals of Statistics, Vol. 17.

Bowman, A.W. and Azzalini, A. (1997) *Applied Smoothing Techniques for Data Analysis: The Kernel Approach with S-Plus Illustrations*. Oxford Science Publications.

Hastie, T.J. and Tibshirani, R.J. (1990) *Generalized Additive Models*. Chapman & Hall.

<https://cran.r-project.org/web/packages/gam/gam.pdf>

<https://cran.r-project.org/web/packages/gamlss/gamlss.pdf>

<https://cran.r-project.org/web/packages/gamm4/gamm4.pdf>

<https://cran.r-project.org/web/packages/mgcv/mgcv.pdf>

<https://cran.r-project.org/web/packages/sBF/sBF.pdf>

<https://cran.r-project.org/web/packages/sm/sm.pdf>

Myers, R.H., Montgomery D.C., Geoffrey, G. and Robinson T.J. (2012) *Generalized Linear Models with Applications in Engineering and the Sciences*. Wiley-Interscience.

Opsomer, J.D. (2000) *Asymptotic properties of backfitting estimators*. Journal of Multivariate Analysis, Vol. 73.

Takezawa, K. (2005) *Introduction to Nonparametric Regression*. Wiley-Interscience

Wasserman, L. (2006) *All of Nonparametric Statistics*. Springer Science.



# Lista de figuras

<i>Figura 1.1: Funciones núcleo (izquierda) e influencia del span en la estimación (derecha).....</i>	<i>8</i>
<i>Figura 2.1: Ajuste no paramétrico con dos variables (izquierda) y ajuste aditivo con dos variables (derecha).....</i>	<i>13</i>
<i>Figura 4.1: Matriz de dispersión y correlaciones para las variables de Calidad del Aire.....</i>	<i>33</i>
<i>Figura 4.2: Estimadores no paramétricos de las regresiones univariantes obtenidas con regresión lineal local para calidad del aire.....</i>	<i>34</i>
<i>Figura 4.3: Contrastes de no efecto para cada predictor. ....</i>	<i>35</i>
<i>Figura 4.4: Contraste de linealidad para Benceno. ....</i>	<i>36</i>
<i>Figura 4.5: Contraste de linealidad para Temperatura. ....</i>	<i>37</i>
<i>Figura 4.6: Contraste de linealidad para Humedad relativa.....</i>	<i>37</i>
<i>Figura 4.7: Contraste de linealidad para Humedad absoluta.....</i>	<i>38</i>
<i>Figura 4.8: Estimación de las funciones componentes mediante Backfitting con loess.....</i>	<i>42</i>
<i>Figura 4.9: Q-Q plot para los residuos del modelo seleccionado ajustado con loess.....</i>	<i>43</i>
<i>Figura 4.10: Estimación de las funciones componente mediante Backfitting con spline. ....</i>	<i>45</i>
<i>Figura 4.11: Q-Q plot para los residuos del modelo completo ajustado con spline. ....</i>	<i>46</i>
<i>Figura 4.12: Matriz de dispersión y correlaciones para las variables de la simulación. ....</i>	<i>47</i>
<i>Figura 4.13: Estimadores no paramétricos de las regresiones univariantes obtenidas con regresión lineal local para la simulación. ....</i>	<i>48</i>
<i>Figura 4.14: Contraste de linealidad para X4.....</i>	<i>49</i>
<i>Figura 4.15: Estimación de las funciones componentes mediante Backfitting con loess.....</i>	<i>51</i>
<i>Figura 4.16: Estimación de las funciones componentes mediante Backfitting con spline. ....</i>	<i>53</i>



# Lista de tablas

<i>Tabla 4.1: Descriptivos para las variables de Calidad del Aire. ....</i>	<i>32</i>
<i>Tabla 4.2: Deviance residual y AIC para modelos con un predictor. ....</i>	<i>39</i>
<i>Tabla 4.3: ANODEV para modelos con dos predictores. ....</i>	<i>39</i>
<i>Tabla 4.4: AIC para modelos con dos predictores. ....</i>	<i>40</i>
<i>Tabla 4.5: ANODEV para modelos con tres predictores. ....</i>	<i>40</i>
<i>Tabla 4.6: AIC para modelos con tres predictores. ....</i>	<i>41</i>
<i>Tabla 4.7: ANODEV para el modelo con cuatro predictores. ....</i>	<i>41</i>
<i>Tabla 4.8: ANODEV para modelos anidados ajustados mediante spline. ....</i>	<i>44</i>
<i>Tabla 4.9: AIC y Deviance residual para los modelos seleccionados para Calidad del aire. ....</i>	<i>46</i>
<i>Tabla 4.10: Descriptivos para las variables de la simulación. ....</i>	<i>47</i>
<i>Tabla 4.11: ANODEV para modelos de la simulación ajustados con LOESS ....</i>	<i>50</i>
<i>Tabla 4.12: ANODEV para modelos de la simulación ajustados con Splines. ....</i>	<i>52</i>
<i>Tabla 4.13: AIC y Deviance residual para los modelos con Loes y Spline de la simulación. ....</i>	<i>53</i>



# Anexos. Código R

## Anexo 1. Funciones

```
### Instalacion de librerias
# install.packages("sm")
# install.packages("gam")
# install.packages("xlsx")
library(sm)
library(gam)
library(xlsx)

### Matriz de dispersion y correlaciones
# Panel inferior: correlaciones
panel.cor <- function(x,y){
  usr <- par("usr"); on.exit(par(usr))
  par(usr = c(0, 1, 0, 1))
  r <- round(cor(x, y), 3)
  txt <- paste0("R = ", r); text(0.5, 0.5, txt)
}
# Panel superior: graficos de dispersion
upper.panel <- function(x,y){
  points(x, y, pch = 16, col = "cadetblue", cex = 0.5)
}

### Calculo de los descriptivos para unos datos
descriptivos <- function(datos){
  tabla <- vector()
  min <- apply(datos, 2, min)
  max <- apply(datos, 2, max)
  media <- apply(datos, 2, mean)
  sd <- apply(datos, 2, sd)
  tabla <- t(rbind(min, max, media, sd))
  return(round(tabla,3))
}

### Estimador de la funcion de regresion
# nv: nombre del predictor
# nr: nombre de la respuesta
estimador <- function (x,y,nv,nr){
  sm.regression(x, y, col.points = "cadetblue", ylab = nr, xlab = nv,
               cex = 0.5)
  sm.regression(x, y, col = "red1", add = T)
  title(main = paste("Estimador para", nv, sep = " "))
}

### Contraste de linealidad y no efecto
# test: contraste que se quiere realizar
contraste <- function(x,y,nv,test){
  aux <- sm.regression(x, y, model = test, col.band = "limegreen",
                    xlab = nv, col.points = "cadetblue", ylab = "NOX",cex = 0.5)
  return(aux$h)
}
```

```

### Funciones nucleo
bicuadrado <- function(x) {
  y <- (1-x^2)^2 * ((x< 1)+0)*((x > -1)+0)
  return(y)
}
epanechnikov <- function(x) {
  y <- 0.75*(1-x^2)*((x< 1)+0)*((x > -1)+0)
  return(y)
}
tricubo <- function(x) {
  y <- (1-abs(x^3))^3 * ((x< 1)+0)*((x > -1)+0)
  return(y)
}

```

## Anexo 2. Datos reales

```

### Procesamiento de los datos
# Media por dia de cada variable
cont <- 0
nox <- vector()
aux <- vector()
for(i in 1:length(Benceno[,1])){
  cont <- cont + 1
  aux <- rbind(aux, Benceno[i,])
  if(cont == 24){
    cont <- 0
    nox <- rbind(nox, colMeans(aux, na.rm = T))
    aux <- vector()
  }
}
# Eliminamos observaciones con algun campo nulo
aire <- na.omit(data.frame(round(nox,3)))
rm(nox, i, aux, cont, Benceno)

### Representacion de superficies con ajuste no parametrico y
### ajuste aditivo
par(mfrow = c(1,2))
par(cex = 0.7)
# Superficie ajuste no parametrico
modelo1 <- sm.regression(cbind(aire$Benceno,AH = -aire$AH), aire$NOX,
  zlab = "NOX", xlab = "AH", ylab = "Benceno", col = "white",
  theta = -25, phi = 15)
# Superficie ajuste aditivo
modelo2 <- gam(NOX ~ lo(Benceno) + lo(AH), data = aire)
ex <- modelo1$eval.points[,1] # Benceno
ey <- modelo1$eval.points[,2] # AH
ngrid <- length(ex)
grid <- data.frame(cbind(Benceno = rep(ex,ngrid),
  AH = rep(-ey, rep(ngrid, ngrid))))
superficie <- predict(modelo2, grid)
mask <- modelo1$estimate
mask[!is.na(mask)] <- 1
persp(ex,ey, matrix(superficie * mask, ncol = ngrid), xlab = "AH",
  ylab = "Benceno", zlab = "NOX", theta = -30, phi = 20,
  ticktype = "detailed")
par(cex = 1)
par(mfrow = c(1,1))

```

```

### Representacion de las funciones nucleo
par(mfrow = c(1,2))
curve(tricubo(x), from = -3, to = 3, col = 1, ylab = "")
curve(bic cuadrado(x), add = T, col = 2)
curve(epanechnikov(x), add = T, col = "green4")
curve(dnorm(x), add = T, col = 4)
legend(-3.2, 1, c("Tricubo", "Bic cuadrado", "Epanechnikov", "Normal"),
      col = c(1,2, "green4", 4), lty = 1, bty = "n")

### Influencia del span
plot(NOX ~ Temp, data = aire, xlab="Temp", ylab="NOX", col="cadetblue",
     cex = 0.5)
lines(loess.smooth(aire$Temp, aire$NOX, span = 0.1, evaluation = 200))
lines(loess.smooth(aire$Temp, aire$NOX, span = 0.5, evaluation = 200),
     col = 2)
lines(loess.smooth(aire$Temp, aire$NOX, span = 0.9, evaluation = 200),
     col = 4)
legend(21, 840, c("span = 0.1", "span = 0.5", "span = 0.9"), bty = "n",
     col = c(1,2,4), lty = 1)
par(mfrow = c(1,1))

### Descripcion de las variables
descriptivos(aire)

### Matriz de dispersion y correlaciones
pairs(aire, lower.panel = panel.cor, upper.panel = upper.panel)

### Estimador de la funcion de regresion
par(mfrow = c(1,2))
estimador(aire$Benceno, aire$NOX, "Benceno", "NOX")
estimador(aire$Temp, aire$NOX, "Temp", "NOX")
estimador(aire$RH, aire$NOX, "RH", "NOX")
estimador(aire$AH, aire$NOX, "AH", "NOX")

### Contraste de no efecto
contraste(aire$Benceno, aire$NOX, "Benceno", "no.effect")
title(main = "Contraste de no efecto\n Benceno")
contraste(aire$Temp, aire$NOX, "Temp", "no.effect")
title(main = "Contraste de no efecto\n Temperatura")
contraste(aire$RH, aire$NOX, "RH", "no.effect")
title(main = "Contraste de no efecto\n Humedad Relativa")
contraste(aire$AH, aire$NOX, "AH", "no.effect")
title(main = "Contraste de no efecto\n Humedad Absoluta")

### Contraste de linealidad
# Benceno
ben <- contraste(aire$Benceno, aire$NOX, "Benceno", "linear")
title(main = "Contraste de linealidad\n Benceno")
sig.trace(sm.regression(aire$Benceno, aire$NOX, model = "linear",
                      display = "none"), hvec = seq(ben*0.5, ben*2, leng = 20))
title(main = "Traza de significaci\u00f3n/span>)

# Temperatura
temp <- contraste(aire$Temp, aire$NOX, "Temp", "linear")
title(main = "Contraste de linealidad\n Temperatura")
sig.trace(sm.regression(aire$Temp, aire$NOX, model = "linear",
                      display = "none"), hvec = seq(temp*0.5, temp*2, leng = 20))
title(main = "Traza de significaci\u00f3n/span>)

```

```

# Humedad relativa
rh <- contraste(aire$RH, aire$NOX, "RH", "linear")
title(main = "Contraste de linealidad\n Humedad relativa")
sig.trace(sm.regression(aire$RH, aire$NOX, model = "linear",
  display = "none"), hvec = seq(rh*0.5, rh*2, leng = 20))
title(main = "Traza de significaci□/span>)

# Humedad absoluta
ah <- contraste(aire$AH, aire$NOX, "AH", "linear")
title(main = "Contraste de linealidad\n Humedad absoluta")
sig.trace(sm.regression(aire$AH, aire$NOX, model = "linear",
  display = "none"), hvec = seq(ah*0.5, ah*2, leng = 20))
title(main = "Traza de significaci□/span>)

### Ajuste mediante Loess
# 1 variable
l1 <- gam(NOX ~ lo(Benceno, span = 0.35), data = aire); summary(l1)
l2 <- gam(NOX ~ lo(Temp, span = 0.35), data = aire); summary(l2)
l3 <- gam(NOX ~ lo(RH, span = 0.5), data = aire); summary(l3)
l4 <- gam(NOX ~ lo(AH, span = 0.35), data = aire); summary(l4)

# 2 variables
l12 <- gam(NOX ~ lo(Benceno, span = 0.35) + lo(Temp, span = 0.35),
  data = aire); summary(l12)
l13 <- gam(NOX ~ lo(Benceno, span = 0.35) + lo(RH, span = 0.5),
  data = aire); summary(l13)
l14 <- gam(NOX ~ lo(Benceno, span = 0.35) + lo(AH, span = 0.35),
  data = aire); summary(l14)
anova(l1, l12, test = "F")
anova(l1, l13, test = "F")
anova(l1, l14, test = "F")

# 3 variables
l123 <- gam(NOX ~ lo(Benceno, span = 0.35) + lo(Temp, span = 0.35) +
  lo(RH, span = 0.5), data = aire); summary(l123)
l124 <- gam(NOX ~ lo(Benceno, span = 0.35) + lo(Temp, span = 0.35) +
  lo(AH, span = 0.35), data = aire); summary(l124)
anova(l12, l123, test = "F")
anova(l12, l124, test = "F")

# Todas las variables
lc <- gam(NOX ~ lo(Benceno, span = 0.35) + lo(Temp, span = 0.35) +
  lo(RH, span = 0.5) + lo(AH, span = 0.35), data = aire)
summary(lc); anova(l123, lc, test = "F")

### Estimaciones del Backfitting con loess para Modelo Seleccionado
par(mfrow = c(1,2))
plot.Gam(lc, residuals = F, se = F, col = "cadetblue", ask = T)
title(main = "Estimador de\n m1 (Benceno)")
title(main = "Estimador de\n m2 (Temp)")
title(main = "Estimador de\n m3 (RH)")
title(main = "Estimador de\n m4 (AH)")

### QQ-plot para el Modelo Seleccionado con loess
par(mfrow = c(1,1))
qqnorm(lc$residuals, pch = 16, cex = 0.1)
qqline(lc$residuals, col = "red")

```

```

### Ajuste mediante splines
# 1 variable
s1 <- gam(NOX ~ s(Benceno), data = aire); summary(s1)
s2 <- gam(NOX ~ s(Temp), data = aire); summary(s2)
s3 <- gam(NOX ~ s(RH), data = aire); summary(s3)
s4 <- gam(NOX ~ s(AH), data = aire); summary(s4)
# 2 variables
s12 <- gam(NOX ~ s(Benceno) + s(Temp), data = aire); summary(s12)
s13 <- gam(NOX ~ s(Benceno) + s(RH), data = aire); summary(s13)
s14 <- gam(NOX ~ s(Benceno) + s(AH), data = aire); summary(s14)
anova(s1, s12, test = "F")
anova(s1, s13, test = "F")
anova(s1, s14, test = "F")

# 3 variables
s123 <- gam(NOX ~ s(Benceno) + s(Temp) + s(RH), data = aire);
summary(s123)
s124 <- gam(NOX ~ s(Benceno) + s(Temp) + s(AH), data = aire);
summary(s124)
anova(s12, s123, test = "F")
anova(s12, s124, test = "F")

# Todas las variables
sc <- gam(NOX ~ s(Benceno) + s(Temp) + s(RH) + s(AH) , data = aire)
summary(sc)
anova(s123, sc, test = "F")

### Comparacion de modelos ajustados con spline
anova(s1, s12, s123, sc, test = "F")

### Estimaciones del Backfitting con spline para Modelo Seleccionado
par(mfrow = c(1,2))
plot.Gam(sc, residuals = F, se = F, col = "cadetblue", ask = T)
title(main = "Estimador de\n m1 (Benceno)")
title(main = "Estimador de\n m2 (Temp)")
title(main = "Estimador de\n m3 (RH)")
title(main = "Estimador de\n m4 (AH)")

### QQ-plot para el Modelo Seleccionado con spline
par(mfrow = c(1,1))
qqnorm(sc$residuals, pch = 16, cex = 0.1)
qqline(sc$residuals, col = "red")

```

## Anexo 3. Datos simulados

```

### Datos simulados
# Predictores uniformes
x1 <- runif(200,0,4)
x2 <- runif(200,-2,2)
x3 <- runif(200,-2,2)
x4 <- runif(200,4,8)
# Error normal
e <- rnorm(200,0,0.4)
# Funciones elegidas para los predictores
m1 <- -sqrt(x1) + sin(x1)*x1
m2 <- cos(x2^2) * x2^2
m3 <- x3^3 - 3*x3
m4 <- 3
# Varibale respuesta
y <- m1 + m2 + m3 + m4 + e

```

```

# Guardamos los datos simulados en un excel
simulacion <- data.frame(cbind(y,x1,x2,x3,x4,e))
write.xlsx(simulacion, "simulacion.xlsx", row.names = F,
           sheetName = "simul")
simulacion <- simulacion[,-6] # eliminamos el error
attach(simulacion)
# detach(simulacion)

### Descriptivos de las variables
descriptivos(simulacion)

### Matriz de dispersion y correlaciones
pairs(simulacion, lower.panel = panel.cor, upper.panel = upper.panel)

### Estimadores de la regresion para cada predictor
par(mfrow = c(1,2))
estimador(x1, y, "x1", "y")
estimador(x2, y, "x2", "y")
estimador(x3, y, "x3", "y")
estimador(x4, y, "x4", "y")

#### Contraste de no efecto para x4
par(mfrow = c(1,2))
h <- contraste(x4, y, "x4", "no.effect")
title(main = "Contraste de no efecto\n para X4")
sig.trace(sm.regression(x4, y, model = "no.effect", display = "none"),
          hvec = seq(h*0.5, h*2, leng = 20))
title(main = "Traza de significaci□/span>)

### Ajuste mediante loess
# 1 varibale
l1 <- gam(y ~ lo(x1, span = 0.35, degree = 1), data = simulacion)
summary(l1)
l2 <- gam(y ~ lo(x2, span = 0.35, degree = 1), data = simulacion)
summary(l2)
l3 <- gam(y ~ lo(x3, span = 0.35, degree = 1), data = simulacion)
summary(l3)
l4 <- gam(y ~ lo(x4, span = 0.35, degree = 1), data = simulacion)
summary(l4)

# 2 variables
l12 <- gam(y ~ lo(x1, span = 0.35) + lo(x2, span = 0.35),
          data = simulacion); summary(l12)
l13 <- gam(y ~ lo(x1, span = 0.35) + lo(x3, span = 0.35),
          data = simulacion); summary(l13)
l14 <- gam(y ~ lo(x1, span = 0.35) + lo(x4, span = 0.35),
          data = simulacion); summary(l14)
anova(l1, l12, test = "F")
anova(l1, l13, test = "F")
anova(l1, l14, test = "F")

# 3 variables
l132 <- gam(y ~ lo(x1, span = 0.35) + lo(x3, span = 0.35) +
           lo(x2, span = 0.35), data = simulacion); summary(l132)
l134 <- gam(y ~ lo(x1, span = 0.35) + lo(x3, span = 0.35) +
           lo(x4, span = 0.35), data = simulacion); summary(l134)
anova(l13, l132, test = "F")
anova(l13, l134, test = "F")

```

```

# Modelo completo
lc <- gam(y ~ lo(x1, span = 0.35) + lo(x3, span = 0.35) +
          lo(x2, span = 0.35) + lo(x4, span = 0.35),
          data = simulacion); summary(lc)
anova(l132, lc, test = "F")

### Comparacion de los modelos
anova(l1, l13, l132, lc, test = "F")

### Estimaciones del backfittin para el modelo seleccionado
par(mfrow = c(1,2))
plot.Gam(l132, se = F, col = "cadetblue", rugplot = F, ask = T)

title(main = "Estimador de m1")
x <- sort(x1); m <- -sqrt(x) + sin(x)*x; lines(x, m, col = 2)
legend(0, -3.2, col = c(2,"cadetblue"), cex = .7, lty = c(1,1),
       legend = c("Verdadera m1","Estimador de m1"), bty = "n")

title(main = "Estimador de m3")
x <- sort(x3); m <- x^3 - 3*x; lines(x, m, col = 2)
legend(-1.85, -1.4, col = c(2,"cadetblue"), cex = .6, lty = c(1,1),
       legend = c("Verdadera m3","Estimador de\n m3"), bty = "n")

title(main = "Estimador de m2")
x <- sort(x2); m <- cos(x^2) * x^2; lines(x, m, col = 2)
legend(-1.7, -2.5, col = c(2,"cadetblue"), cex = .7, lty = c(1,1),
       legend = c("Verdadera m2","Estimador de m2"), bty = "n")

### Ajuste mediante splines
# 1 varibale
s1 <- gam(y ~ s(x1, df = 7), data = simulacion); summary(s1)
s2 <- gam(y ~ s(x2, df = 7), data = simulacion); summary(s2)
s3 <- gam(y ~ s(x3, df = 7), data = simulacion); summary(s3)
s4 <- gam(y ~ s(x4, df = 7), data = simulacion); summary(s4)

# 2 variables
s12 <- gam(y ~ s(x1, df = 7) + s(x2, df = 7), data = simulacion)
summary(s12)
s13 <- gam(y ~ s(x1, df = 7) + s(x3, df = 7), data = simulacion)
summary(s13)
s14 <- gam(y ~ s(x1, df = 7) + s(x4, df = 7), data = simulacion)
summary(s14)
anova(s1, s12, test = "F")
anova(s1, s13, test = "F")
anova(s1, s14, test = "F")

# 3 variables
s132 <- gam(y ~ s(x1, df = 7) + s(x3, df = 7) + s(x2, df = 7),
            data = simulacion); summary(s132)
s134 <- gam(y ~ s(x1, df = 7) + s(x3, df = 7) + s(x4, df = 7),
            data = simulacion); summary(s134)
anova(s13, s132, test = "F")
anova(s13, s134, test = "F")

# Modelo completo
sc <- gam(y ~ s(x1, df = 7) + s(x3, df = 7) + s(x2, df = 7) +
          s(x4, df = 7), data = simulacion); summary(sc)
anova(s132, sc, test = "F")

### Comparacion de los modelos
anova(s1, s13, s132, sc, test = "F")

```

```

### Estimaciones del backfittin para los modelos
par(mfrow = c(1,2))
plot.Gam(s132, se = F, col = "cadetblue", rugplot = F, ask = T)

title(main = "Estimador de m1")
x <- sort(x1); m <- -sqrt(x) + sin(x)*x; lines(x, m, col = 2)
legend(0, -3.2, col = c(2,"cadetblue"), cex = .7, lty = c(1,1),
      legend = c("Verdadera m1","Estimador de m1"), bty = "n")

title(main = "Estimador de m3")
x <- sort(x3); m <- x^3 - 3*x; lines(x, m, col = 2)
legend(-1.85, -1.4, col = c(2,"cadetblue"), cex = .6, lty = c(1,1),
      legend = c("Verdadera m3","Estimador de\n m3"), bty = "n")

title(main = "Estimador de m2")
x <- sort(x2); m <- cos(x^2) * x^2; lines(x, m, col = 2)
legend(-1.7, -2.5, col = c(2,"cadetblue"), cex = .7, lty = c(1,1),
      legend = c("Verdadera m2","Estimador de m2"), bty = "n")

```