REGRESIÓN “FLEXIBLE”:

1. ***Regresión polinómica***

2. ***Regresión polinómica “local”: Regresión spline***

Carmen María Cadarso Suárez

Dado el modelo de respuesta continua Y, y una sola covariable continua X:

$$E\left[Y / X\right] = \beta_0 + f\left(X\right)$$

donde f es una función "suave" desconocida. ¿Cómo obtener flexibilidad en la modelización?

1. ***Regresión polinómica.***

2. ***Regresión polinómica "local".***

   Splines de Regresión (regression splines)

   - *B-splines* (bs)
   - *Natural Splines* (ns)

REGRESIÓN POLINÓMICA

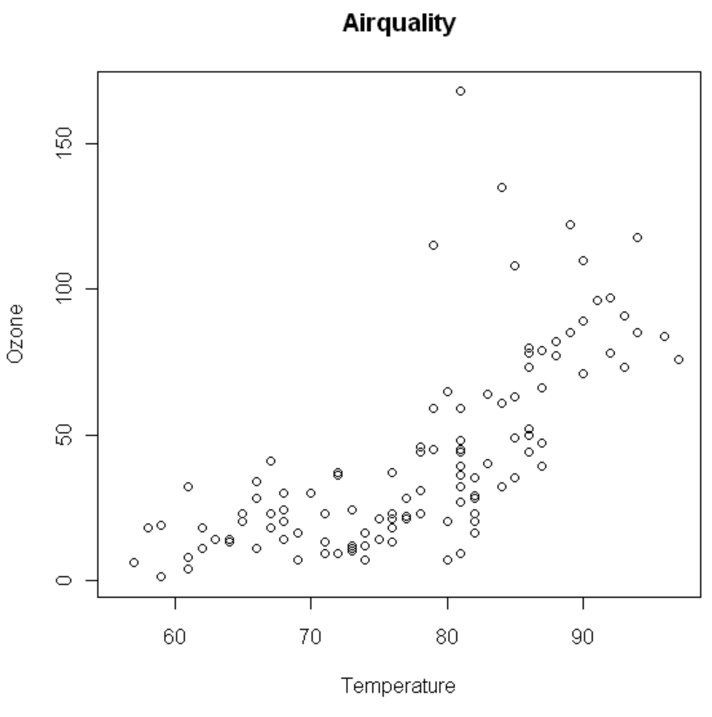
Realizamos un ajuste mínimo-cuadrático a la función polinómica de grado *p*

$$E[Y/X] = \beta_0 + \beta_1 X + \beta_2 X^2 + \ldots + \beta_p X^p$$

- Grados de libertad del modelo: *df* = *p+1*.
- Cuanto mayor es la potencia *p* (i.e., *df*) mayor flexibilidad.

Ejemplo: Relación entre Ozono y Temperatura.

```
plot(airquality$Temp,airquality$Ozone, xlab="Temperature", ylab="Ozone",
      main="Airquality")
```



Ajuste cuadrático

$$E\left[Ozone / Temp\right] = \beta_0 + \beta_1 Temp + \beta_2 Temp^2$$

airquality$Temp2<- airquality$Temp*airquality$Temp

air.fit2<- lm(Ozone~Temp+Temp2,data=airquality)
summary(air.fit2)

```
Coefficients:
              Estimate   Std. Error   t value   Pr(>|t|)
(Intercept)  305.48577   122.12182     2.501    0.013800 *
Temp          -9.55060     3.20805    -2.977    0.003561 **
Temp2          0.07807     0.02086     3.743    0.000288 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 22.47 on 113 degrees of freedom
Multiple R-squared: 0.5442,     Adjusted R-squared: 0.5362
F-statistic: 67.46 on 2 and 113 DF,  p-value: < 2.2e-16
```

El modelo también se puede formular así:

air.fit22<- lm(Ozone~poly(Temp,2,raw=T), data=airquality)
summary(air.fit22)

```
Coefficients:
                          Estimate    Std. Error   t value    Pr(>|t|)
(Intercept)              305.48577    122.12182     2.501     0.01380
poly(Temp, 2, raw = T)1   -9.55060      3.20805    -2.977     0.003561
poly(Temp, 2, raw = T)2    0.07807      0.02086     3.743     0.000288

Residual standard error: 22.47 on 113 degrees of freedom
Multiple R-squared: 0.5442,     Adjusted R-squared: 0.5362
F-statistic: 67.46 on 2 and 113 DF,  p-value: < 2.2e-16
```

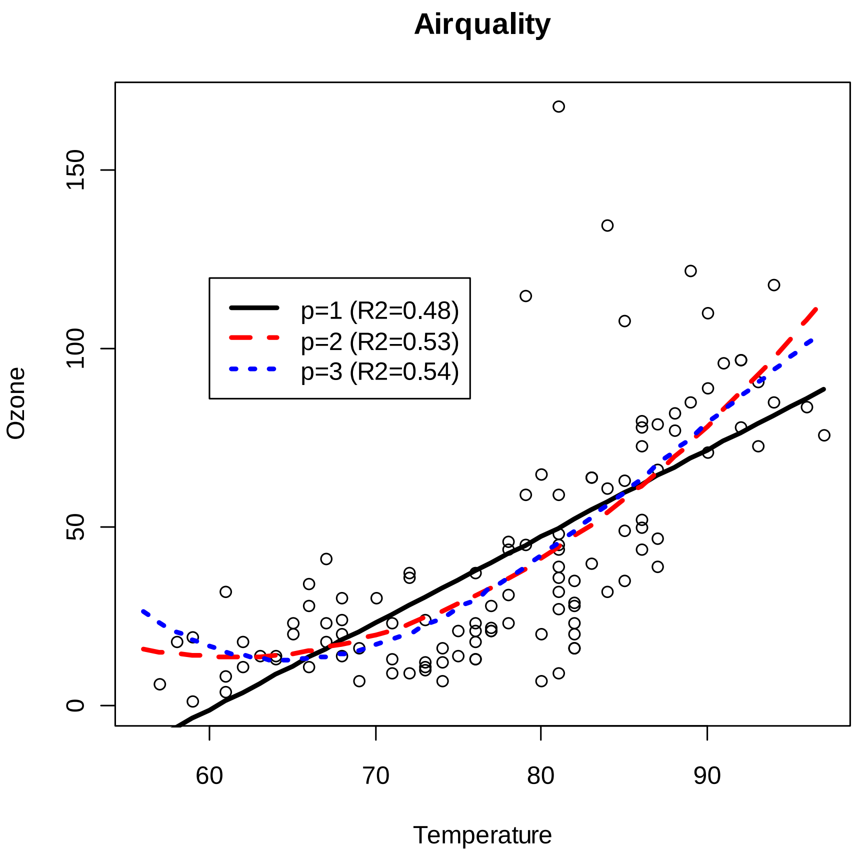
Ajuste lineal; ajuste cuadrático; ajuste cúbico

```
air.fit1<-lm(Ozone~Temp,data=airquality)

new <- data.frame(Temp =
seq(min(airquality$Temp),max(airquality$Temp),1))
plot(airquality$Temp,airquality$Ozone, xlab="Temperature",
ylab="Ozone", main="Airquality")
lines(new$Temp,predict(air.fit1,new),lty=1,col="black",lwd=3)

air.fit2<-lm(Ozone~poly(Temp,2,raw=T),data=airquality)
lines(new$Temp,predict(air.fit2,new),lty=2,col="red",lwd=3)

air.fit3<-lm(Ozone~poly(Temp,3),data=airquality)
lines(new$Temp,predict(air.fit3,new,raw=T),lty=3, col="blue",lwd=3)
legend(60,120,c("p=1 (R2=0.48)", "p=2 (R2=0.53)","p=3 (R2=0.54)"),
col=c("black","red","blue"), lty=c(1,2,3),lwd=3)
```



ANOVA

a) Comparación lineal versus cuadrático

anova(air.fit1,air.fit2,test="Chi")

Analysis of Variance Table:
Model 1: Ozone ~ Temp;
Model 2: Ozone ~ poly(Temp, 2,raw=T)

	Res.Df	RSS	Df	Sum of Sq	P(>|Chi|)
1	114	64110			
2	113	57038	1	7072	**0.0001818**

b) Comparación cuadrático versus cúbico

anova(air.fit2,air.fit3,test="Chi")

Analysis of Variance Table:
Model 1: Ozone ~ poly(Temp, 2,raw=T);
Model 2: Ozone ~ poly(Temp, 3,raw=T)

	Res.Df	RSS	Df	Sum of Sq	P(>|Chi|)
1	113	57038			
2	112	56440	1	598	**0.2759**

VENTAJAS /DESVENTAJAS DE LA REGRESIÓN POLINÓMICA

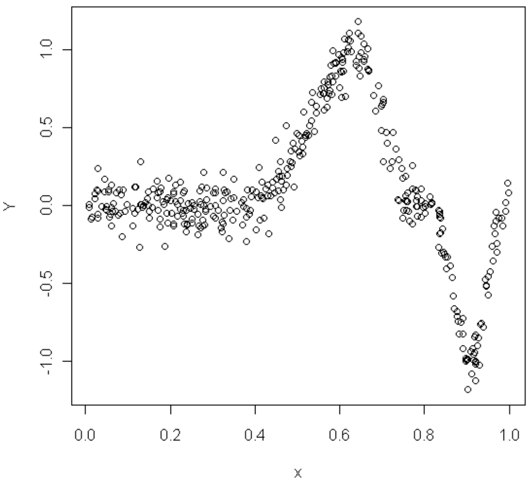
Ventajas:

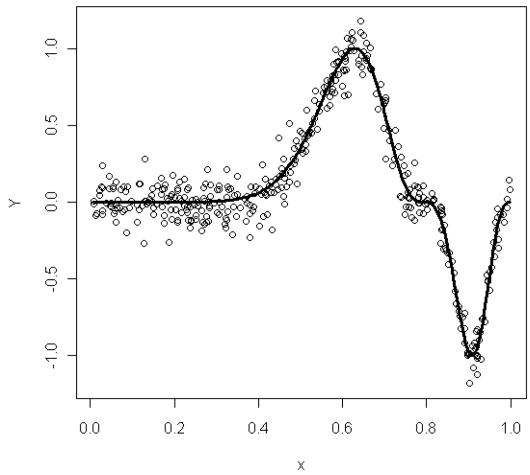
- Los parámetros son fácilmente interpretables.
- Puede realizarse con cualquier paquete estadístico (con R, usando **lm** ó **glm**, pudiéndose aplicar a cualquier tipo de respuesta (continua, binaria, poisson).

Desventajas:

- Es una regresión GLOBAL: los parámetros se ajustan utilizando TODOS los datos muestrales.

- Esto hace que, en ocasiones, no permita capturar relaciones con comportamientos locales diferenciados.





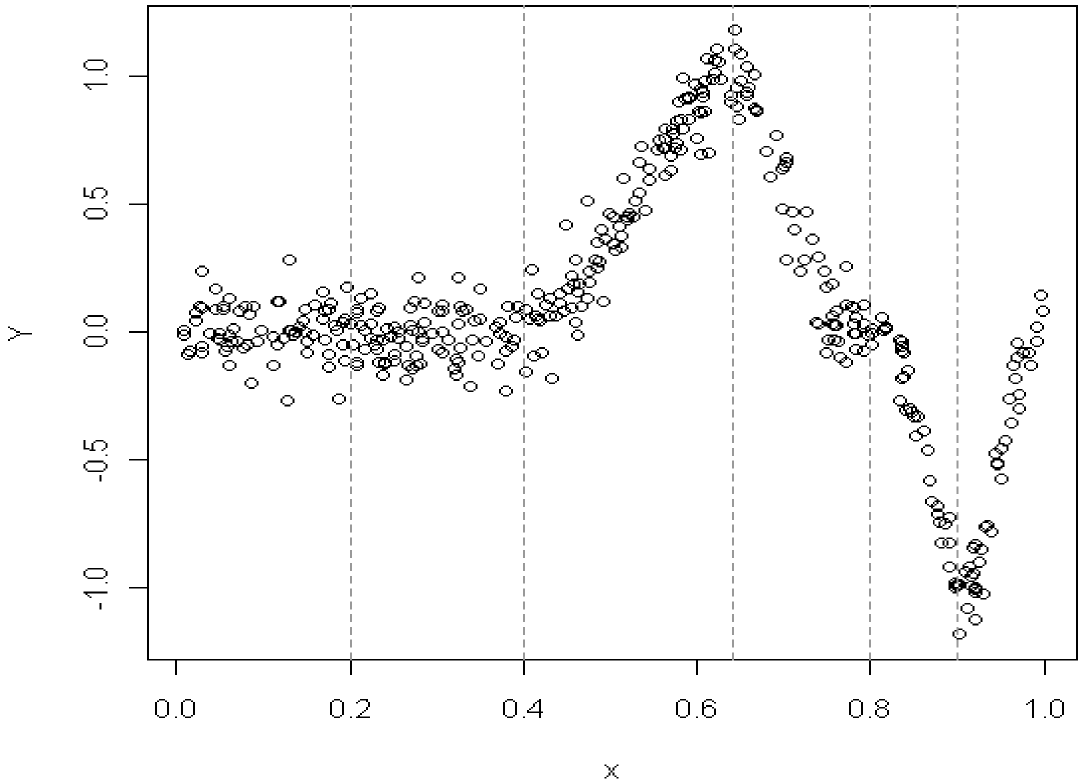
Los datos se basan en un modelo de regresión “simulado”

$$f(x_i) = sen^3\left(2\pi x_i^3\right) + \varepsilon_i \qquad , \qquad i = 1, \ldots, 400$$

$$x_i \sim U[0,1] \quad \text{y} \qquad \varepsilon_i \sim N(0, 0.1)$$

```
set.seed(90)
eps<-rnorm(400,0,0.1); X<-runif(400,0,1)
X<-X[order(X)]
yteor<-(sin(2*3.141516*X**3))**3
yobs<-(sin(2*3.141516*X**3))**3+eps
plot(X,yobs,pch=1,ylab="Y")
```

Como posible solución, puede utilizarse la **regresión polinómica local:**



1. Elegimos K nodos (knots) interiores.
2. Dividimos el rango de la covariable X, en k+1 regiones disjuntas.
3. Regresión polinómica (p.e. cúbica), en cada región.
4. **Restricción:** Los polinomios han de unirse “suavemente” en los nodos (asegurando que la función resultante sea continua).
5. El número de nodos, K, indica el grado de flexibilidad (a mayor K, mayor flexibilidad).

REGRESIÓN SPLINE

- B-Splines (bs)

- Natural Splines (ns)

(R library : splines)

De Boor C. *A practical guide to splines*. Revised Edition. New York, NY: Springer Verlag, 1987.

De Boor C. *A practical guide to splines.* Revised Edition. New York, NY: Springer Verlag, 2001.

Regresión B-spline (bs)

- Parte de una base de $(K+p)$ polinomios B-splines (de Boor, 1978)

$$\left\{B_1(x),\dots\dots\dots,B_{p+K}(x)\right\}$$

donde

K = número de nodos interiores.

p = grado de los polinomios (p.e. cúbicos, $p=3$).

- **Modelo de regresión B-spline**:

$$E\left[Y/X\right]=\beta_0+\sum_{i=1}^{K+p}\beta_j B_j(x)$$

Con este modelo, se ha convertido el problema original en un problema de regresión lineal múltiple.

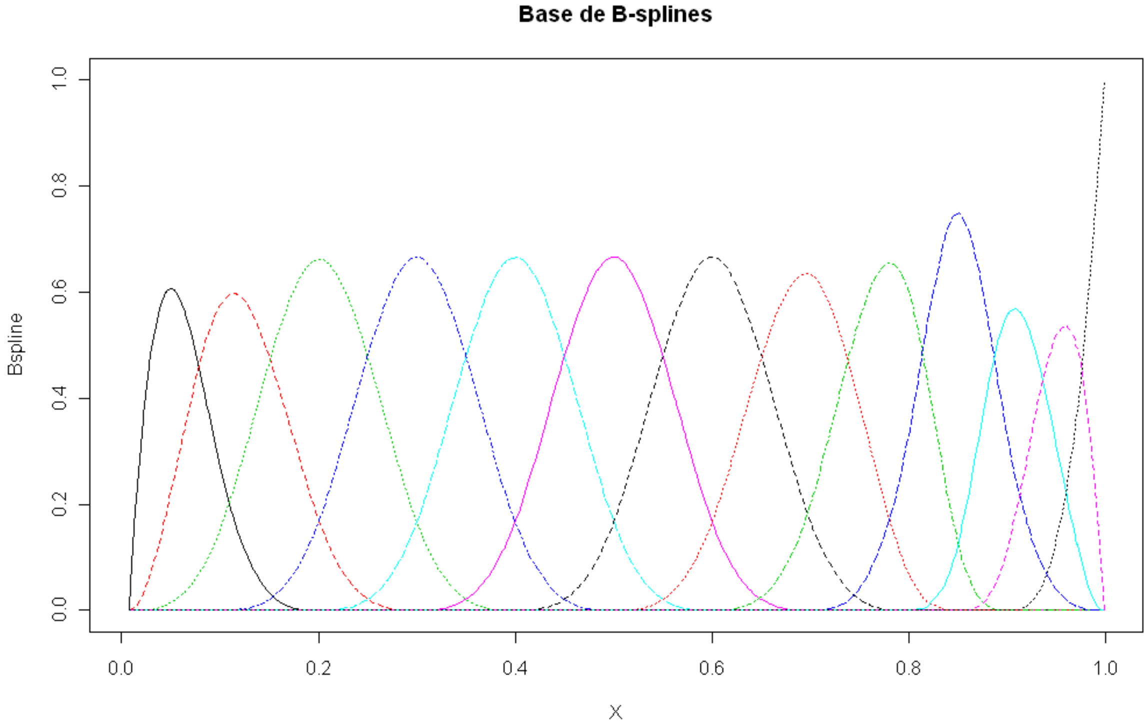
- **Grado de flexibilidad del modelo:**

K = número de nodos interiores.

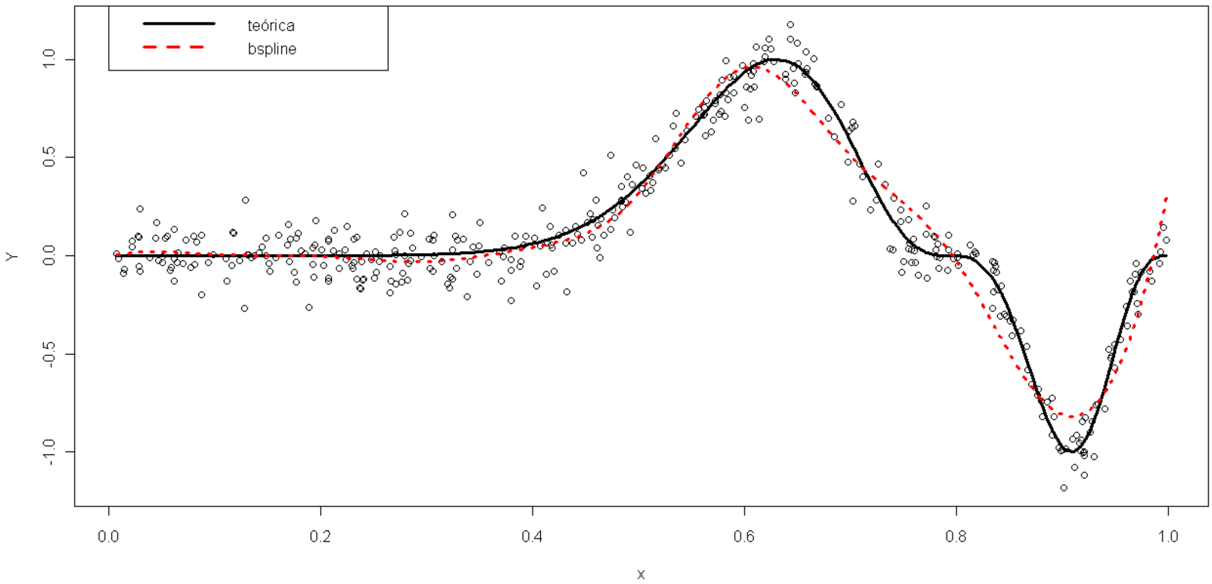
$df = K+p+1$ (ó $K+p$ sin la constante).

En nuestro ejemplo, proponemos una regresión b-spline con 9 knots localizados en los siguientes valores de X:

(0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9)



Como resultado obtenemos la siguiente estimación B-spline:



¿Cómo se obtiene la base de B-splines en R?

```
library(splines)

Bspline<-bs(X,knots=c(0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9))
```

Bspline es un objeto que guarda la evaluación de cada valor de X en la base de los 12 B-splines. Por ejemplo, para el valor X=0.1167016

```
X[40]
        [1] 0.1167016
```

Bspline[40,] nos ofrece la evaluación de la base en dicho valor:

```
Bspline[40,]

       1            2           3            4            5
0.1558194980 0.5964778668 0.2469261731 0.0007764621 0.0000000000
       6            7           8            9           10
0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
      11           12
0.0000000000 0.0000000000
```

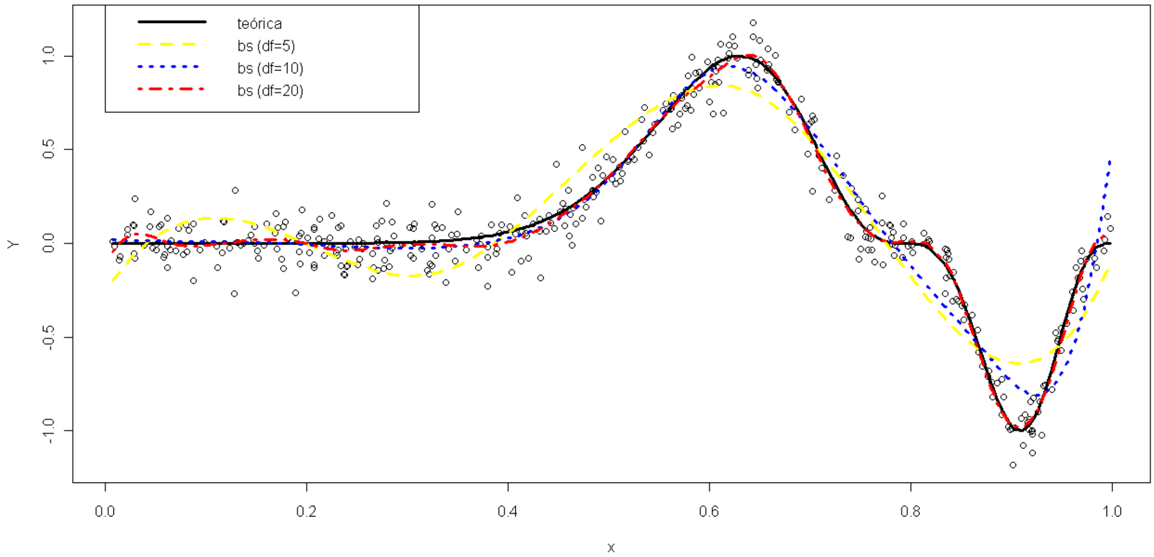
La representación gráfica de la base se obtiene así:

```
matplot(X,Bspline,col="white", main="Base de B-splines")
matlines(X,Bspline)
```

El objeto Bspline también también guarda otras informaciones:

```
attr(,"degree")
[1] 3
attr(,"knots")
[1] 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9
attr(,"Boundary.knots")
[1] 0.007405288 0.998299962
attr(,"intercept")
[1] FALSE
attr(,"class")
[1] "bs"    "basis"
```

a) Flexibilidad variando los gl (los knots los coloca en los cuantiles)

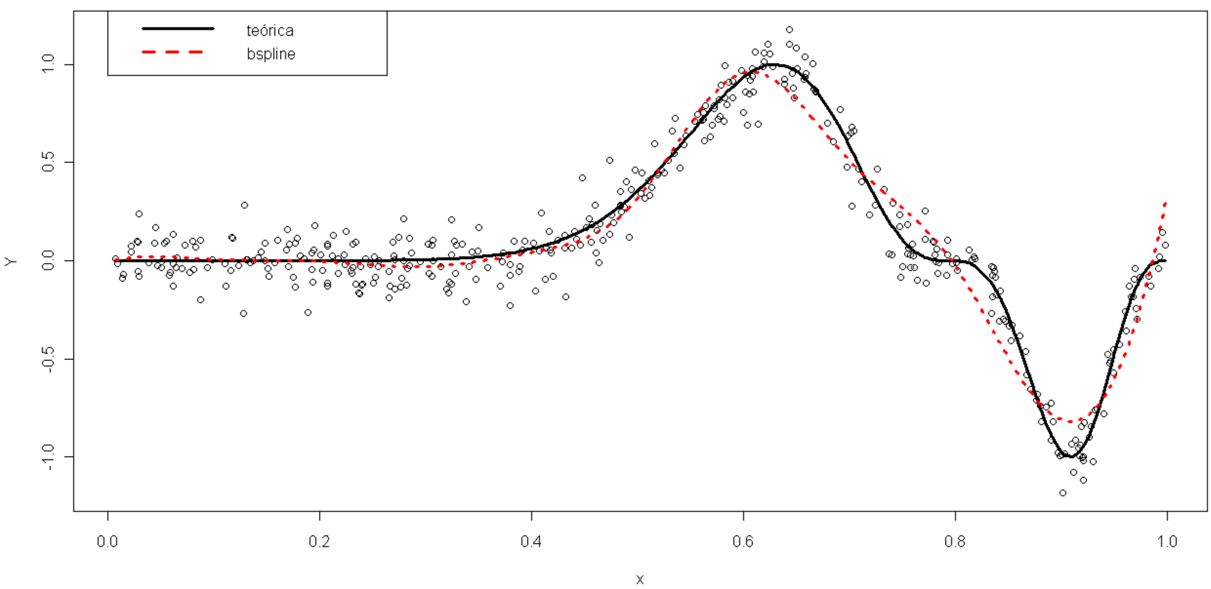


b) Flexibilidad fijando los knots.

Consideramos la siguiente secuencia de knots interiores (K=9)

(0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9)

Lo que nos lleva a un modelo b-spline con $df=k+p+1$ = 13 grados de libertad.



B-Spline Basis for Polynomial Splines

Description

Generate the B-spline basis matrix for a polynomial spline.

Usage

bs(x, df = NULL, knots = NULL, degree = 3, intercept = FALSE, Boundary.knots = range(x))

Arguments

x	the predictor variable. Missing values are allowed.
df	degrees of freedom; one can specify df rather than knots; bs() then chooses df-degree-1 knots at suitable quantiles of x (which will ignore missing values).
knots	the *internal* breakpoints that define the spline. The default is NULL, which results in a basis for ordinary polynomial regression. Typical values are the mean or median for one knot, quantiles for more knots. See also Boundary.knots.
degree	degree of the piecewise polynomial—default is 3 for cubic splines.
intercept	if TRUE, an intercept is included in the basis; default is FALSE.
Boundary.knots	boundary points at which to anchor the B-spline basis (default the range of the data). If both knots and Boundary.knots are supplied, the basis parameters do not depend on x. Data can extend beyond Boundary.knots.

Value

A matrix of dimension `length(x) * df`, where either `df` was supplied or if `knots` were supplied, **`df = length(knots) + 3 + intercept`**. Attributes are returned that correspond to the arguments to `bs`, and explicitly give the `knots`, `Boundary.knots` etc for use by `predict.bs()`.

See Also

`ns`, `poly`, `smooth.spline`, `predict.bs`, `SafePrediction`

Examples

```
require(stats); require(graphics)
bs(women$height, df = 5)
summary(fm1 <- lm(weight ~ bs(height, df = 5), data = women))
```

Regresión natural-spline (ns)

- Parte de una base de (K+p-2) polinomios n-splines (de Boor, 1978)

$$\left\{ \mathrm{N}_1(x), \ldots\ldots\ldots, N_{p+K-2}(x) \right\}$$

debido a la restricción de **"linealidad"** en las fronteras.

- **Modelo de regresión n-spline:**

$$E\left[Y / X\right] = \beta_0 + \sum_{i=1}^{K+p-2} \beta_j N_j(x)$$

- **Grado de flexibilidad:**

K = número de nodos interiores.

$df = K+p-1$ (ó $K+p-2$ sin la constante).

Generate a Basis Matrix for Natural Cubic Splines

Description

Generate the B-spline basis matrix for a natural cubic spline.

Usage

```
ns(x, df = NULL, knots = NULL, intercept = FALSE,  Boundary.knots
= range(x))
```

Arguments

x	the predictor variable. Missing values are allowed.
df	degrees of freedom. One can supply `df` rather than knots; `ns()` then chooses `df - 1 - intercept` knots at suitably chosen quantiles of `x` (which will ignore missing values).
knots	breakpoints that define the spline. The default is no knots; together with the natural boundary conditions this results in a basis for linear regression on `x`. Typical values are the mean or median for one knot, quantiles for more knots. See also `Boundary.knots`.
intercept	if `TRUE`, an intercept is included in the basis; default is `FALSE`.
Boundary.knots	boundary points at which to impose the natural boundary conditions and anchor the B-spline basis (default the range of the data). If both `knots` and `Boundary.knots` are supplied, the basis parameters do not depend on `x`. Data can extend beyond `Boundary.knots`

Value

A matrix of dimension `length(x) * df` where either `df` was supplied or if `knots` were supplied, `df = length(knots) + 1 + intercept`. Attributes are returned that correspond to the arguments to `ns`, and explicitly give the `knots`, `Boundary.knots` etc for use by `predict.ns()`.

Examples

```
require(stats); require(graphics)
ns(women$height, df = 5)
summary(fm1 <- lm(weight ~ ns(height, df = 5), data = women))

## example of safe prediction
plot(women, xlab = "Height (in)", ylab = "Weight (lb)")
ht <- seq(57, 73, length.out = 200)
lines(ht, predict(fm1, data.frame(height=ht)))
```

Comparativa: bs y ns

- Los splines naturales son menos flexibles en las fronteras (permiten reducir la varianza).

- Los splines naturales son más flexibles en el interior.

  Al poseer 2 gl menos que los B-splines, permiten introducir 2 knots interiores adicionales.

Ejemplo: Ozono en función de la temperatura (airquality)

B-SPLINE

- Realizamos una regresión b-spline con df=4.
- Esto significa que sólo se considera 1 knot (el percentil 50 de Temp).

air.bs<-lm(Ozone~bs(Temp,df=4),data=airquality)
summary(air.bs)
Coefficients:

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	3.243	15.623	0.208	0.836
bs(Temp, df = 4)1	31.004	28.336	1.094	0.276
bs(Temp, df = 4)2	-20.998	16.677	-1.259	0.211
bs(Temp, df = 4)3	110.104	23.107	4.765	5.76e-06 ***
bs(Temp, df = 4)4	79.535	19.452	4.089	8.24e-05 ***

Residual standard error: 21.91 on 111 degrees of freedom
Multiple R-squared: 0.574, Adjusted R-squared: 0.5587
F-statistic: 37.39 on 4 and 111 DF, p-value: < 2.2e-16

Natural-SPLINE

- Realizamos una regresión n-spline con df=4.
- Esto significa que se consideran 3 knots (los percentiles 25,50 y 75 de Temp)

air.ns<-lm(Ozone~ns(Temp,df=4),data=airquality)
summary(air.ns)
Coefficients:

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	12.62	11.50	1.098	0.27469
ns(Temp, df = 4)1	13.04	11.44	1.141	0.25647
ns(Temp, df = 4)2	65.81	10.81	6.087	1.68e-08 ***
ns(Temp, df = 4)3	82.38	28.20	2.921	0.00423 **
ns(Temp, df = 4)4	79.33	12.73	6.231	8.55e-09 ***

Residual standard error: 22.14 on 111 degrees of freedom
(37 observations deleted due to missingness)
Multiple R-squared: 0.5651, Adjusted R-squared: 0.5495
F-statistic: 36.06 on 4 and 111 DF, p-value: < 2.2e-16

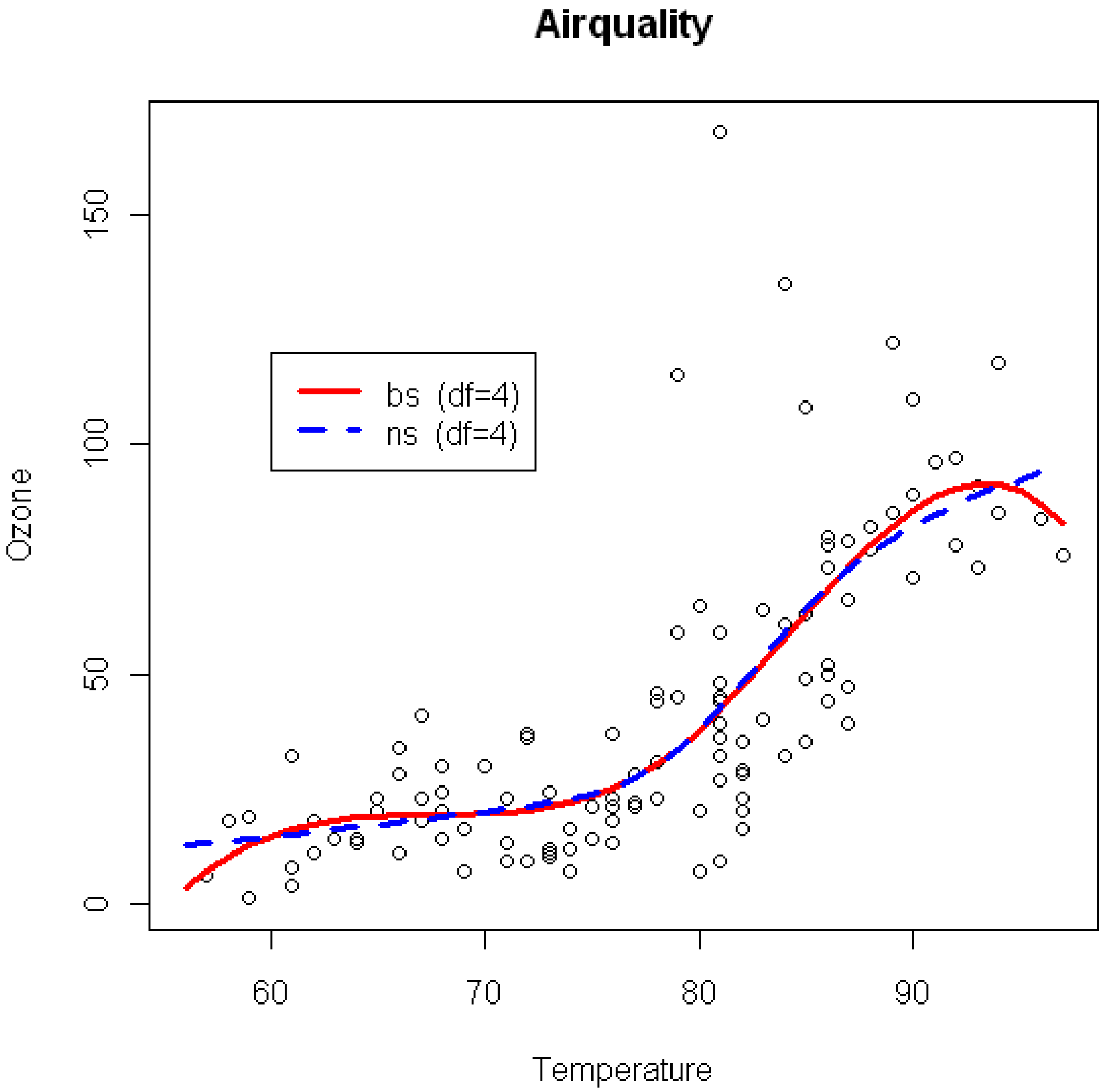
```
plot(airquality$Temp,airquality$Ozone, xlab="Temperature", ylab="Ozone",
main="Airquality")

new <- data.frame(Temp = seq(min(airquality$Temp),max(airquality$Temp),1))

lines(new$Temp,predict(air.bs,new),lty=1,col="red",lwd=3)

lines(new$Temp,predict(air.ns,new),lty=2,col="blue",lwd=3)

legend(60,120,c("bs  (df=4)","bs  (df=4)"),col=c("red","blue"), lty=c(1,2),lwd=3)
```



VENTAJAS /DESVENTAJAS DE LA REGRESIÓN SPLINE

Ventajas:

- Es una regresión **paramétrica LOCAL**.
- Puede realizarse fácilmente con R, usando **lm** ó **glm**, pudiéndose aplicar a **cualquier tipo de respuesta** (continua, binaria, poisson,…).

Desventajas:

- Debemos fijar de antemano:

    1. Nº de knots K (ó equivalentemente los df del modelo).
    2. Localización de los knots.

- Todavía no existe un criterio automático totalmente óptimo para la selección y localización de los knots.
- Si el criterio es subjetivo, el investigador debe guiarse por el conocimiento del problema en cuestión.