Smoothing with penalized splines

A brief introduction and an illustrative application

Antonio Gasparrini

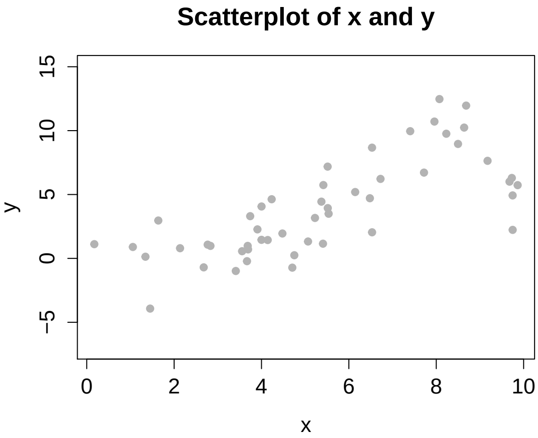
Department of Social and Environmental Health Research
London School of Hygiene and Tropical Medicine (LSHTM)

Centre for Statistical Methodology – LSHTM
29 May 2015

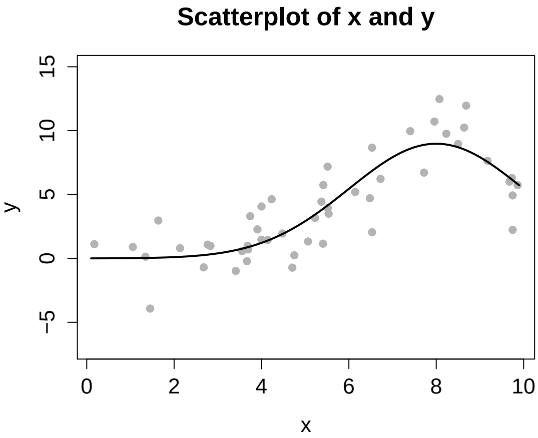
Outline

Outline

Relationships between x and y



Non-linearity



Regression models

A relationship between a predictor x and an outcome y is usually estimated through **regression models**, controlling for potential confounders

In the simple linear case:

$$y_i = \alpha + f(x_i) + \sum_{p=1}^{P} \gamma_p z_{ip} \tag{1}$$

A number of alternative options are available for representing $f(x)$, describing the relationship as a **smooth shape**

Smoothing methods

Parametric

Polynomials, fractional polynomials, regression splines

In between

Penalized splines

Non-parametric

Lowess, kernel, smoothing splines

Outline

Splines: basis representation

A spline is a numeric function composed by **piecewise-connected polynomial functions**

The advantage of using regression splines is that $f(x)$ can be represented in a **basis form**:

$$f(x_i; \boldsymbol{\beta}) = \sum_{j=1}^{d} \beta_j b_j(x_i) = \mathbf{x}^{\mathsf{T}} \boldsymbol{\beta} \tag{2}$$

where $b_j(x)$ are a series of d (known) basis transformations of x

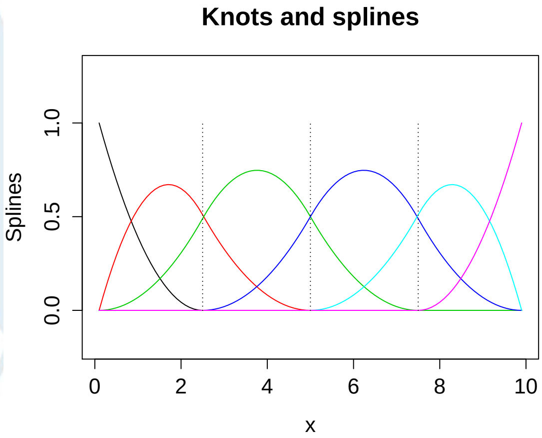
Regression splines

This type of splines allow the use of **standard estimation methods**, derived by minimizing the usual least square objective:

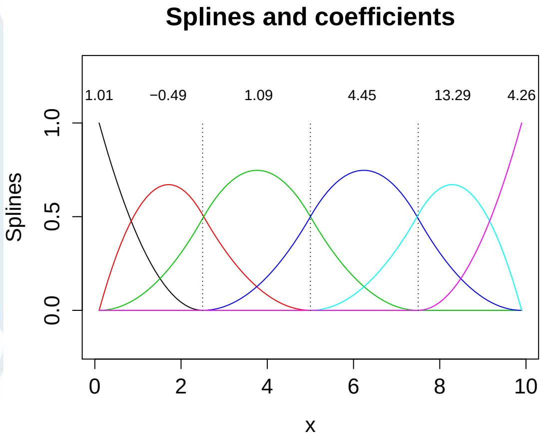
$$\sum_{i=1}^{N}\left(y-\alpha-f(x;\boldsymbol{\beta})+\sum_{p=1}^{P}\gamma_p z_p\right)^2 = ||\mathbf{y}-\alpha-\mathbf{X}\boldsymbol{\beta}-\mathbf{Z}\boldsymbol{\gamma}||^2 \quad (3)$$

Several different transformations for $b_j(x)$ (*e.g.* B-splines, natural splines), determining the mathematical properties

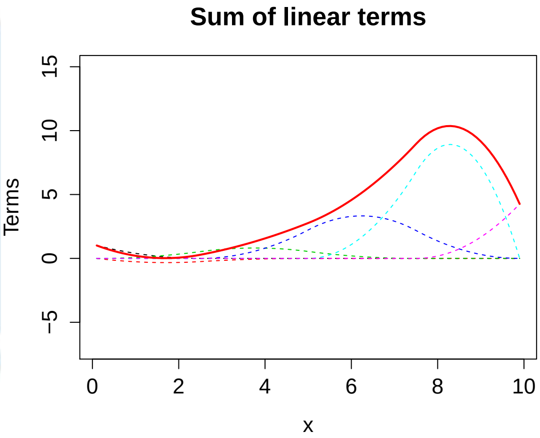
Graphical representation - I



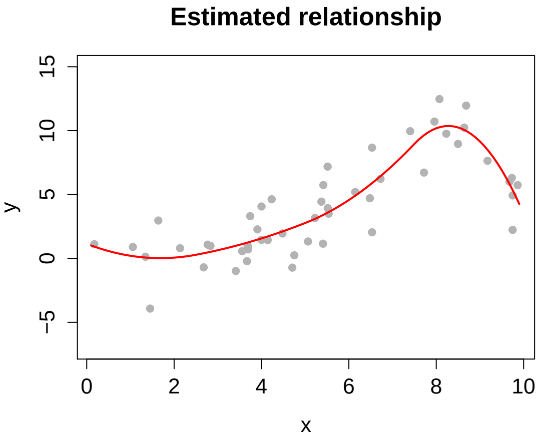
Graphical representation - II



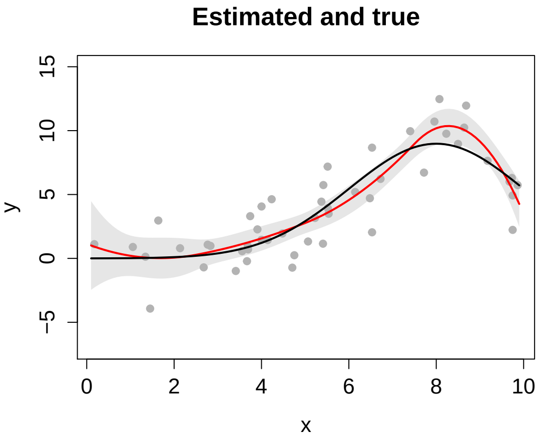
Graphical representation - III



Graphical representation - IV



Graphical representation - V



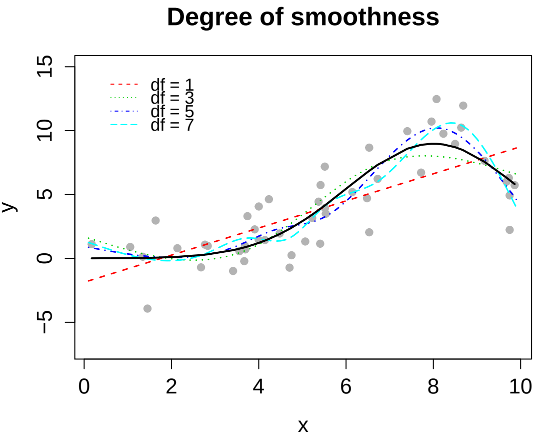
Problems and limitations

In regression splines, the **smoothness** of the fitted curve is determined by:

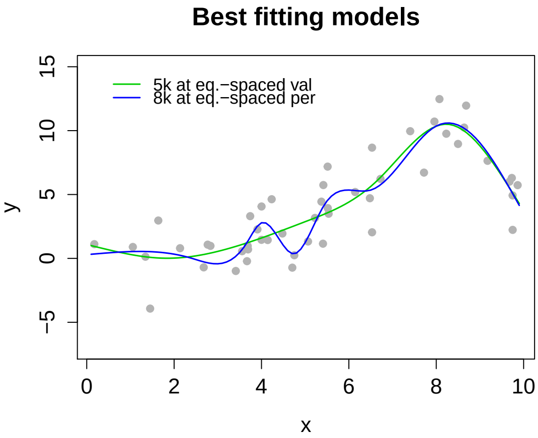
- the degree of the spline
- the specific parameterization
- the number of knots
- the location of knots

No general selection method for number and position of knots

Number of knots



Knots location



Outline

Generalized additive models

A general framework of smoothing methods is offered by **generalized additive models** (GAMs)

GAMs extends traditional GLMs by allowing the linear predictor to depend linearly on unknown smooth functions. In the linear case:

$$y_i = \alpha + f(x_i) + \sum_{p=1}^{P} f(z_{ip}) \tag{4}$$

where f are traditionally represented by non-parametric terms such as **smoothing splines** of **lowess**

Penalty

The idea is to define a flexible function and control the smoothness through a **penalty term**, usually on the second derivative

The objective in (3) is modified to:

$$\sum_{i=1}^{N}\left(y-\alpha-f(x;\beta)+\sum_{p=1}^{P}f(z_{ip})\right)^{2}+\lambda\int[f''(x)]^{2}dx \qquad (5)$$

with λ as **smoothing parameter**

Penalized splines

However, traditional GAMs are limited by **complex and computationally-heavy** estimation methods

Penalized splines offer an flexible and efficient version of GAM, based on low-rank basis transformations

The objective in (5) can be re-written in matrix terms as:

$$||\mathbf{y} - \alpha - \mathbf{X}\beta - \mathbf{Z}\gamma||^2 + \lambda\beta^{\mathsf{T}}\mathbf{S}\beta \tag{6}$$

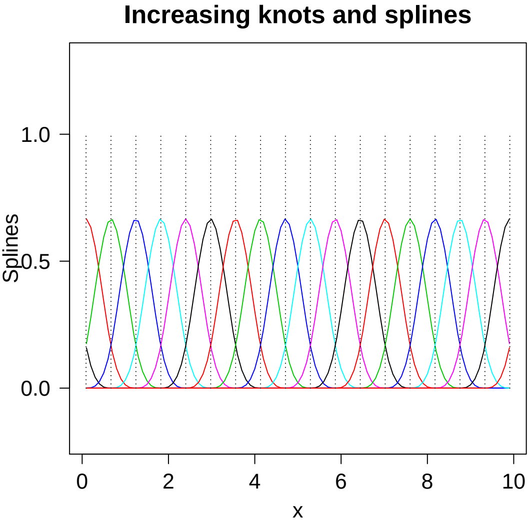
where $\mathbf{S}$ is a **penalty matrix**

Smoothers

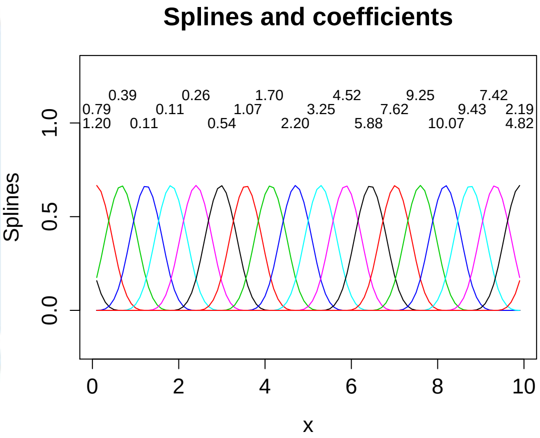
Alternative smoothers available, differing by **parameterization** and **penalty**:

- Thin-plate splines
- Cubic splines
- P-splines
- Random-effects
- Markov random fields
- Soap film smooths
- ...

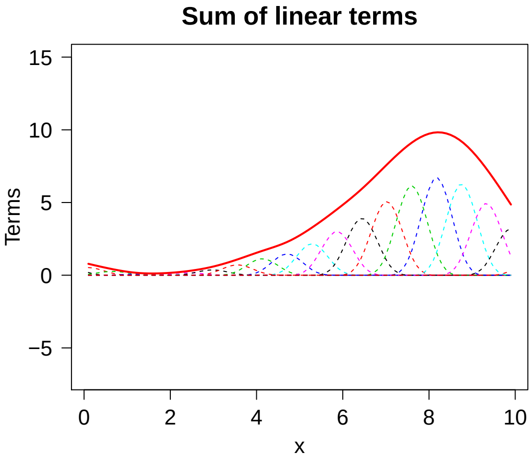
Graphical representation - I



Graphical representation - II



Graphical representation - III



Estimation

Estimation concerns **coefficients** of penalized and unpenalized terms (α, $\boldsymbol{\beta}$, γ) and **smoothing parameters** (λs)

For the former, a **penalized iteratively reweighted least squares** (P-IRLS) scheme is used

Estimation of λs is integrated through either **outer iteration** or **performance iteration**, using **GCV**, **UBRE/AIC** or **REML**

Advantages

Relatively **low-rank basis** and **simplified penalties**

Completely **parametric form**

Number and location of **knots** not critical

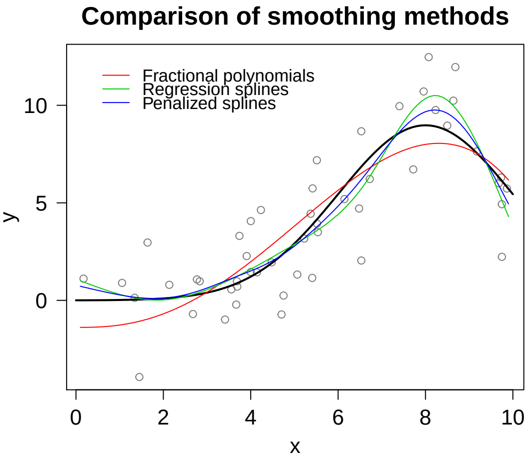
Automatic **smoothing selection**

Efficient **computational methods**

Well-grounded **theoretical framework**

Outline

Comparison


Comparison of smoothing methods
Fractional polynomials
Regression splines
Penalized splines
y
x
0
2
4
6
8
10
0
5
10

Simulations

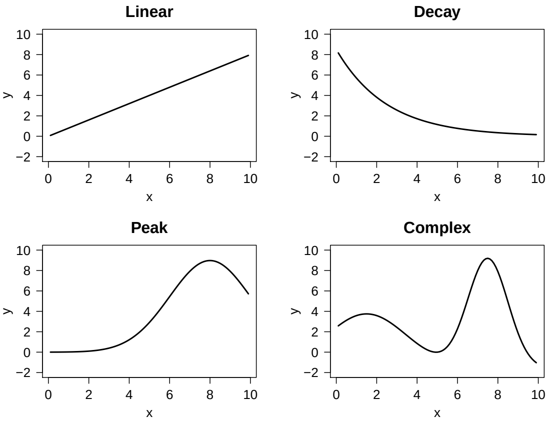
Comparing alternative methods:

- Fractional polynomials
- Regression splines
- Penalized splines

Different shapes:

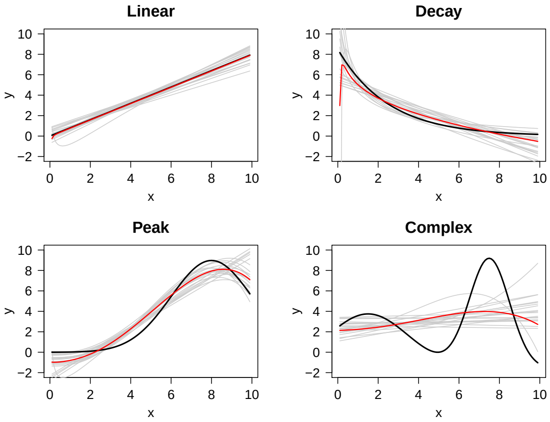
- Linear
- Decay
- Peak
- Complex

Simulated shapes



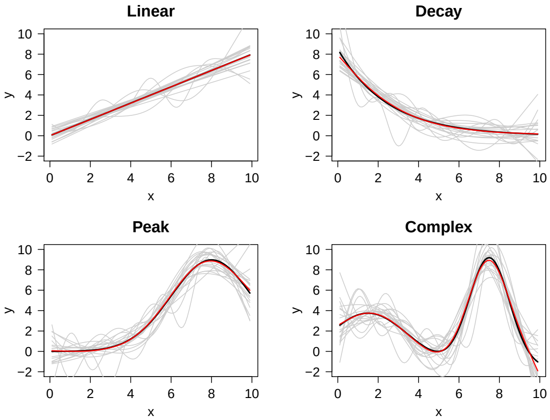
Simulation results - I

Fractional polynomials



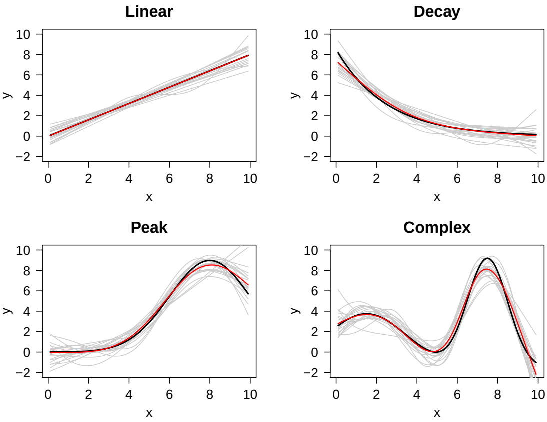
Simulation results - II

Regression splines



Simulation results - III

Penalized splines



Simulation results - IV

Statistics

	Fun	(e)df	Bias	Cov	RMSE
Linear	GLM	2.21	0.01	0.90	0.82
Linear	GAM	1.28	0.00	0.95	0.66
Decay	GLM	3.45	0.05	0.87	0.89
Decay	GAM	2.61	0.13	0.94	0.76
Peak	GLM	4.82	0.05	0.89	0.93
Peak	GAM	4.06	0.19	0.94	0.84
Complex	GLM	6.65	0.11	0.87	1.02
Comples	GAM	5.87	0.31	0.91	0.94

Outline

Distributed lag non-linear models

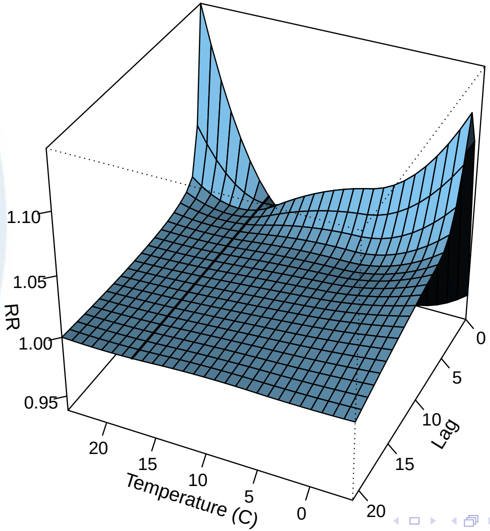
Statistical tools to model **non-linear** and **lagged** dependencies

Defined by a **cross-basis** function of lagged exposures:

$$s(x_{t-\ell_0}, \ldots, x_{t-\ell}) = \sum_{\ell=\ell_0}^{L} f \cdot w(x_{t-\ell}, \ell) \qquad (7)$$

The function is composed of an **exposure response** function $f(x)$ and a **lag-response** function $w(\ell)$

An example



Tensor product basis

Parameterized by a **special tensor product**:

$$s(x_{t-\ell_0}, \ldots, x_{t-\ell}) = (\mathbf{1}^\top_{L-\ell_0+1}\mathbf{A}_t)\eta = \mathbf{w}_t^\top \eta \qquad (8)$$

with

$$\mathbf{A}_t = (\mathbf{1}^\top_{v_\ell} \otimes \mathbf{R}_t) \odot (\mathbf{C} \otimes \mathbf{1}^\top_{v_x}) \qquad (9)$$

where $\mathbf{R}_t$ and $\mathbf{C}$ are basis matrices for x and ℓ, respectively

Penalized DLNMs

Question: what about a penalized version of DLNMs?

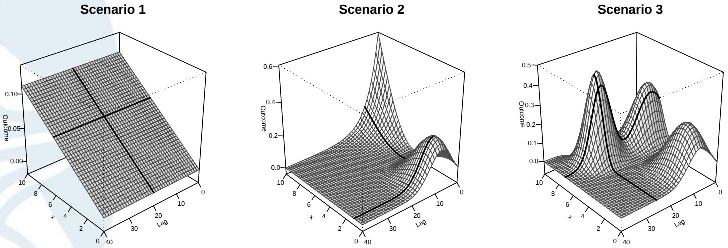
Modify objective in (6) to:

$$||\mathbf{y}-\alpha-\mathbf{W}\boldsymbol{\eta}-\mathbf{Z}\gamma||^2+\boldsymbol{\eta}^{\mathsf{T}}\left(\lambda_x\left(\mathbf{1}_{v_\ell}^{\mathsf{T}}\otimes\mathbf{S}_x\right)+\lambda_\ell\left(\mathbf{S}_\ell\otimes\mathbf{1}_{v_x}^{\mathsf{T}}\right)\right)\boldsymbol{\eta} \quad (10)$$

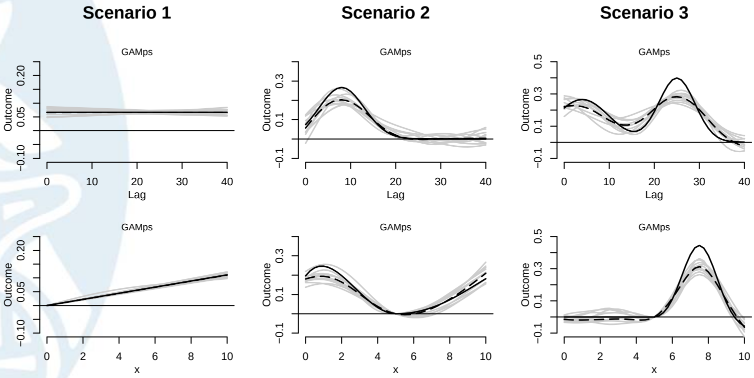
with λ_x, λ_ℓ and $\mathbf{S}_x$, $\mathbf{S}_\ell$ as **smoothing parameters** and **penalty matrices** for each dimension

LONDON SCHOOL of HYGIENE &TROPICAL MEDICINE

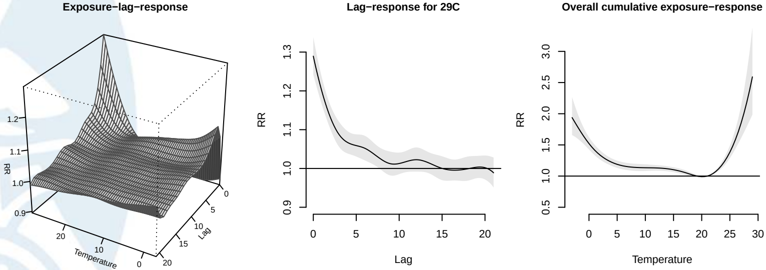
Simulated surfaces



Simulation results



An application



Outline

The R package mgcv

Collection of functions implementing GAMs with penalized splines

Written by Simon Wood, extensively documented

Example of code:

```
library(mgcv)
model <- gam(y ~ s(x,bs="ps") + z, data, family=gaussian,
method="REML")
```

The function `s` determines the spline transformations and penalties

The R package `dlnm`

Collection of functions implementing DLNMs

Example of code:

```
library(dlnm)
cb <- crossbasis(x,lag=c(10),
argvar=list(fun="bs",degree=2,knots=5,cen=0),
arglag=list(fun="ns",knots=c(3,6),int=F))
model <- glm(y ~ s(x,bs="ps") + z, data, family=gaussian)
pred <- crosspred(cb,model)
plot(pred,"3d",xlab="x",ylab="Lag",zlab="Effect")
```

Embedding `dlnm` and `mgcv`

Example of code:

```
library(dlnm) ; library(mgcv)
cb <- crossbasis(x,lag=c(10), argvar=list(fun="ps"),
arglag=list(fun="ps"))
pen <- cbPen(cb)
model <- model <- gam(y ~ cb + z, data, family=gaussian,
method="REML",parapen=list(cb=list(pen)))
pred <- crosspred(cb,model)
plot(pred,"3d",xlab="x",ylab="Lag",zlab="Effect")
```

Outline

Some comments

Penalized splines combine the **flexibility** of non-parametric methods with **stability and simplicity** of parametric smoothers

Based on **theoretically-grounded** and **computationally-efficient** estimators

Well implemented in the package `mgcv` in R

Research **still ongoing**

LONDON SCHOOL of HYGIENE &TROPICAL MEDICINE