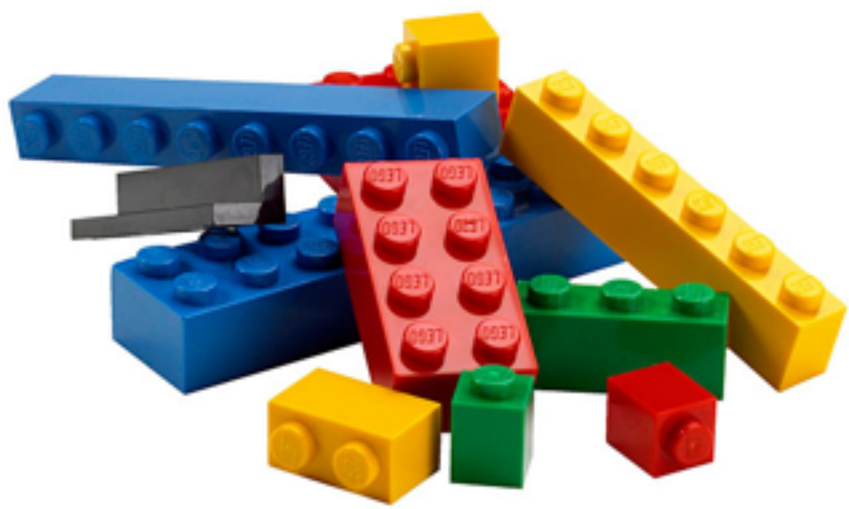
Spline Regression

$m(x) \approx$ a linear combination of basis functions $\tilde{m}(x) = \sum_{k=1}^{K} \beta_k \phi_k(x)$

$$\mathcal{S} = \left\{ \tilde{m} \;\middle|\; \tilde{m} = \sum_k \beta_k \phi_k \right\}$$



To find β_k's minimizing

$$\sum_{i=1}^{n} \{Y_i - \tilde{m}(x_i)\}^2$$

$$= \sum_{i=1}^{n} \{Y_i - \beta_1\phi_1(x_i) - \beta_2\phi_2(x_i) - \cdots - \beta_K\phi_K(x_i)\}^2$$

m

$\tilde{m}$

$\mathcal{S}$

Truncated power basis: $\{1, x, x^2, \cdots, x^p, (x-\kappa_1)_+^p, \cdots, (x-\kappa_K)_+^p\}$

$$\tilde{m}(x) = \beta_0 + \beta_1 x + \cdots + \beta_p x^p + \sum_{k=1}^{K} \beta_{pk}(x-\kappa_k)_+^p$$

$$\downarrow_\beta \sum_{i=1}^{n} \{Y_i - \tilde{m}(x_i)\}^2$$

$$y = \begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{pmatrix} \qquad X = \begin{pmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^p & (x_1-\kappa_1)_+^p & \cdots & (x_1-\kappa_K)_+^p \\ 1 & x_2 & x_2^2 & \cdots & x_2^p & (x_2-\kappa_1)_+^p & \cdots & (x_2-\kappa_K)_+^p \\ \vdots & \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^p & (x_n-\kappa_1)_+^p & \cdots & (x_n-\kappa_K)_+^p \end{pmatrix}$$

$$\beta = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{pK} \end{pmatrix}$$

$$\downarrow_\beta \ (y - X\beta)^\top (y - X\beta)$$

$$\hat{\beta} = (X^\top X)^{-1} X^\top y$$

Model: $Y_i = m(x_i) + \epsilon_i, \quad \epsilon_i \sim \cdot(0, \sigma^2)$

To minimize the sum of squares $\sum_{i=1}^{n} (Y_i - m(x_i))^2$.

Penalization (Regularization)

$$\downarrow \ \sum_{i=1}^{n} (Y_i - m(x_i))^2 + \lambda J(m)$$

(roughness penalty)

$K \uparrow$ $\Rightarrow$ Goodness-of-Fit $\uparrow$

Roughness $\uparrow$

$$\downarrow_{\beta} \sum_{i=1}^{n} \{Y_i - \tilde{m}(x_i)\}^2 + \lambda \sum_{k=1}^{K} \beta_{pk}^2$$

$$\hat{\beta}_{\lambda} = (X^{\top} X + \lambda D)^{-1} X^{\top} y$$

$$\text{with } D = \begin{pmatrix} O_{(p+1)\times(p+1)} & O_{(p+1)\times K} \\ O_{K\times(p+1)} & I_{K\times K} \end{pmatrix}$$

```
library(SemiPar)

data(calif.air.poll)
attach(calif.air.poll)
fit <- spm(ozone.level ~ f(daggett.pressure.gradient)+
                         f(inversion.base.height) +
                         f(inversion.base.temp))

summary(fit)
par(mfrow=c(2,2))
plot(fit)
```