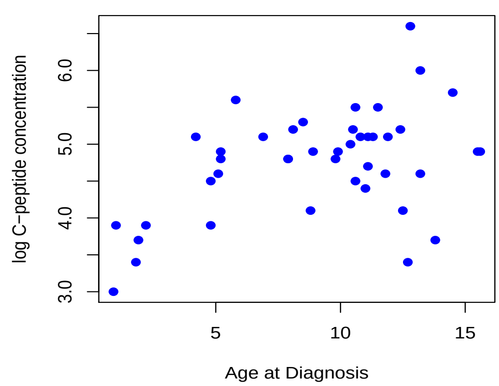
Nonparametric regression using smoothing splines

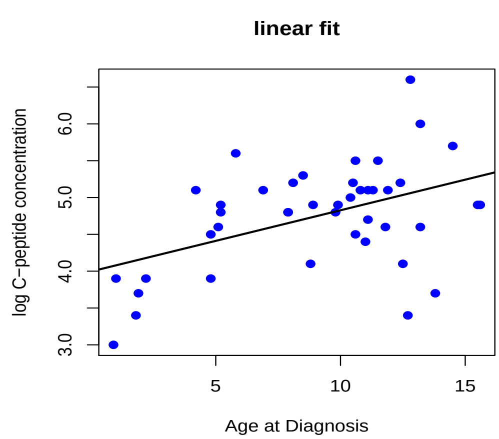
- Smoothing is fitting a smooth curve to data in a scatterplot
- Will focus initially on two variable problems: Y and one X
- Will extend to more than 2 predictors at the end
- Our model:

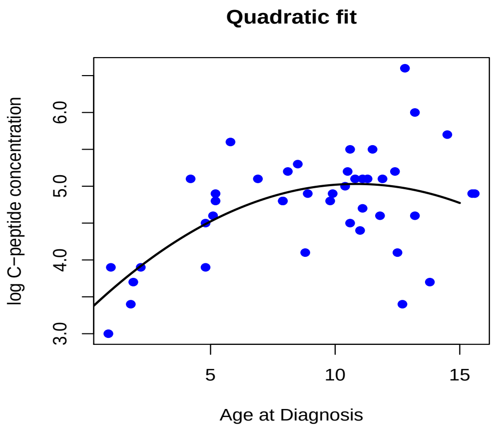
$$y_i = f(x_i) + \varepsilon_i,$$

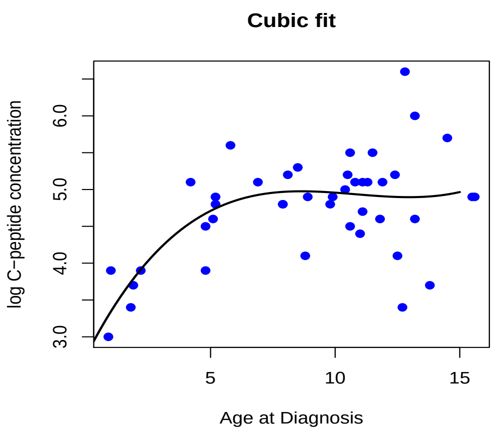
  where $\varepsilon_1, \varepsilon_1, \ldots \varepsilon_n$ are independent with mean 0
- f is some unknown smooth function
- Stat 301, 587 etc: f has a specified form with unknown parameters
  - f could be linear or nonlinear in the parameters,
  - e.g. $Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$
  - functional form always specified
- If f not determined by the subject matter, we may prefer to let the data suggest a functional form

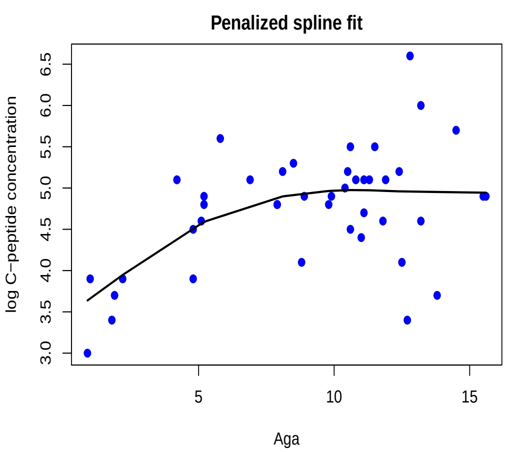
- Why estimate f?
  - can see features of the relationship between X and Y that are obscured by error variation
  - summarizes the relationship between X and Y
  - provide a diagnostic for a presumed parametric form
- Example: Diabetes data set in Hastie and Tibshirani's book *Generalized Additive Models*
- Examine relationship between age of diagnosis of diabetes and log of the serum C-peptide concentration
- Here's what happens if we fit increasing orders of polynomial, then fit an estimated f











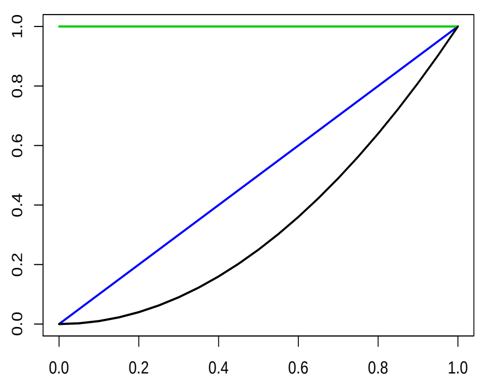
- A slightly different way of thinking about Gauss-Markov Linear models:
  - If we assume that f(x) is linear, then $f(x) = \beta_0 + \beta_1 x$
  - In terms of the Gauss-Markov Linear Model $\boldsymbol{y} = X\beta + \epsilon$,

    $$X = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix} \text{ and } \beta = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}$$
  - The linear model approximates $f(x)$ as a linear combination of two "basis" functions: $b_0(x) = 1,\ b_1(x) = x$,

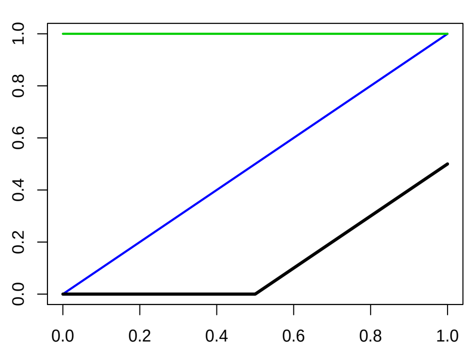
$$f(x) = \beta_0 b_0(x) + \beta_1 b_1(x)$$

- If we assume that f(x) is quadratic, then $f(x) = \beta_0 + \beta_1 x + \beta_2 x^2$.
- In terms of the Gauss-Markov Linear Model $\boldsymbol{y} = x\beta + \epsilon$,
  $X = \begin{bmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ \vdots & \vdots & \vdots \\ 1 & x_n & x_n^2 \end{bmatrix}$ and $\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix}$
- The quadratic model tries to approximate f(x) as a linear combination of three basis functions:
  $b_0(x) = 1,\ b_1(x) = x,\ b_2(x) = x^2$

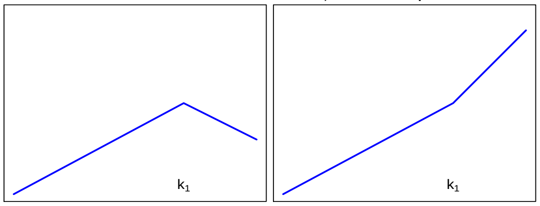
$$f(x) = \beta_0 b_0(x) + \beta_1 b_1(x) + \beta_2 b_2(x)$$



- Now consider replacing $b_2(x) = x^2$ with
  $S_1(x) = (x - k_1)^+ \equiv \begin{cases} 0 & \text{if } x \leq k_1 \\ x - k_1 & \text{if } x > k_1 \end{cases}$
  where k_1 is a specified real value.
- $f(x)$ is now approximated by $\beta_0 b_0(x) + \beta_1 b_1(x) + u_1 S_1(x)$, where u_1 (like β_0 amd β_1) is an unknown parameter.



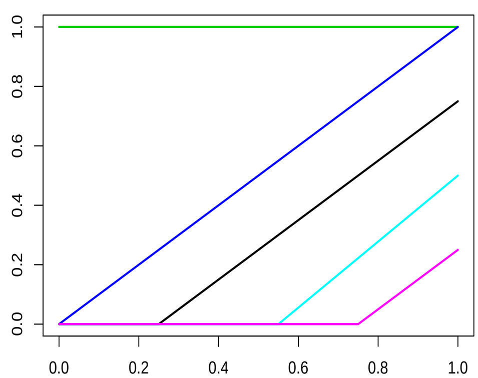
- Note that $\beta_0 b_0(x) + \beta_1 b_1(x) + u_1 S_1(x) = \beta_0 + \beta_1 X + u_1(x - k_1)^+$
$= \begin{cases} \beta_0 + \beta_1 x & \text{if } x \le k_1 \\ \beta_0 + \beta_1 x + u_1(x - k_1) & \text{if } x > k_1 \end{cases}$
$= \begin{cases} \beta_0 + \beta_1 x & \text{if } x \le k_1 \\ \beta_0 - u_1 k_1 + (\beta_1 + u_1)x & \text{if } x > k_1 \end{cases}$
- This is clearly a continuous function (because it is a linear combination of continuous functions), and it is piecewise linear.

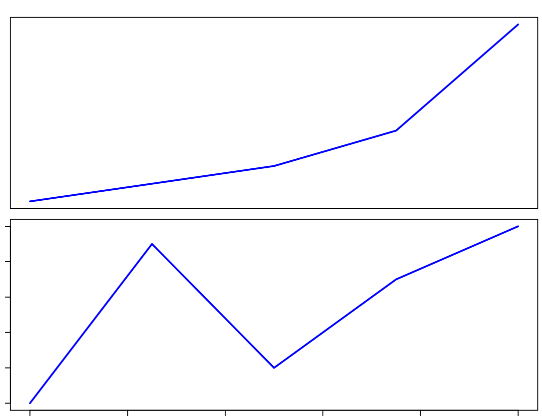


- The function $\beta_0 + \beta_1 x + u_1(x - k_1)^+$ is a simple example of a linear spline function.
- The value k_1 is known as a knot.
- As a Gauss-Markov Linear Model, $\boldsymbol{y} = X\beta + \epsilon$,

$$X = \begin{bmatrix} 1 & x_1 & (x_1 - k_1)^+ \\ 1 & x_2 & (x_2 - k_1)^+ \\ \vdots & \vdots & \vdots \\ 1 & x_n & (x_n - k_1)^+ \end{bmatrix} \text{ and } \beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ u_1 \end{bmatrix}$$

- We can make our linear spline function more flexible by adding more knots $k_1, \ldots, k_k$ so that f(x) is approximated by $\beta_0 + \beta_1 x + \sum_{j=1}^{k} u_j s_j(x) = \beta_0 + \beta_1 x + \sum_{j=1}^{k} u_j (x - k_j)^+$





- If we assume $f(x) = \beta_0 + \beta_1 x + \sum_{j=1}^{k} u_j (x - k_j)^+$, we can write our model as the Gauss-Markov Linear Model $\boldsymbol{y} = X\beta + \epsilon$, where

$$X = \begin{bmatrix} 1 & x_1 & (x_1 - k_1)^+ & (x_1 - k_2)^+ ... (x_1 - k_k)^+ \\ 1 & x_2 & (x_2 - k_1)^+ & (x_2 - k_2)^+ ... (x_2 - k_k)^+ \\ \vdots & \vdots & \vdots & \vdots \quad\quad \vdots \\ 1 & x_n & (x_n - k_1)^+ & (x_n - k_2)^+ ... (x_n - k_k)^+ \end{bmatrix}$$

  and $\beta = (\beta_0, \beta_1, u_1, u_2, ..., u_k)'$
- Estimate $\beta = (\beta_0, \beta_1, u_1, u_2, ..., u_k)'$ by OLS
- But resulting $f(x)$ usually too "wiggly".
- A "wiggly" curve corresponds to values of $u_1, u_2, \ldots u_k$ far from zero

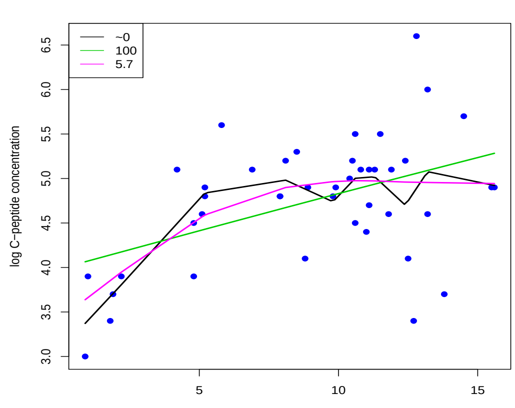
Curve	β_1	u_1	u_2	u_3	$\sum u_i^2$
Smoother	0.4	0.0	0.4	1.6	2.72
Wigglier	3.6	-6.4	4.8	-0.8	64.64

Penalized least squares: Fit + smoothness

- Usually think of fitted curve is an approximation to the true $f(x)$.
- Prefer a smoother (less flexible) estimate of $f(x)$.
- This has u_i coefficients closer to 0
- Want to find coefficients that fit the data while having small u_i.
- Statistical method: penalized least squares
- Minimizes $(\boldsymbol{y} - x\beta)'(\boldsymbol{y} - x\beta) + \lambda^2 \sum_{j=1}^{k} u_j^2$
  - Combines fit to data (1st term) and smoothness (2nd term)
  - $\lambda^2 \sum_{j=1}^{k} u_j^2$ is the penalty for roughness (lack of smoothness).
  - λ^2 is the smoothing parameter.
  - controls the emphasis on fit or on smoothness
- Details at end

Role of smoothing parameter, knots and basis functions

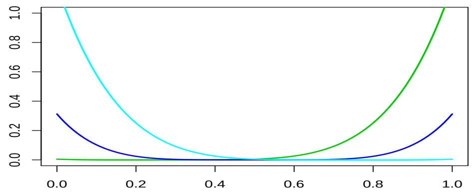
- λ^2 controls how wiggly the curve can be
  - $\lambda^2 \approx 0$, u_i's can be large $\Rightarrow$ wiggly fit.
  - λ^2 large, all u_i's $\to 0 \Rightarrow \beta_0 + \beta_1 X_i$
- knots $k_1, \ k_2, \cdots$ control where the curve bends
  - You choose where and how many
  - In practice, not very important.
  - Better to have too many than too few.
  - If too many knots, some u_i's will be 0.
- Form of the basis functions
  - linear spline function is continuous
  - but 1st and 2nd derivatives are not; they're undefined at the knots
  - curve "looks" smoother if continuous in 1st and 2nd derivatives
    - cubic regression splines
    - thin plate splines
    - And quite a few others



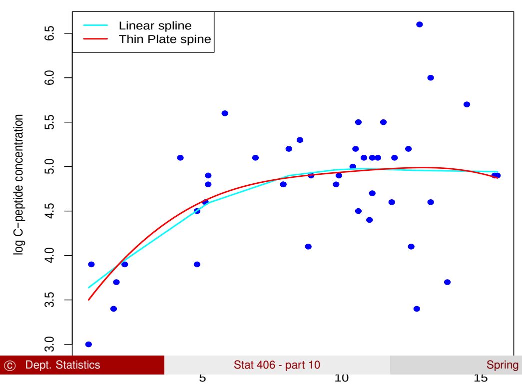
Thin plate splines

- Generalize easily to multiple X's
- The thin plate spline in concept: quadratic + spline pieces

$$f(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \sum_{i=1}^{n} u_j\, f(|x - x_i|)$$



Comparison of linear and thin-plate splines



Choosing a smoothing parameter

- How much to smooth?
  - i.e. what λ^2?
    - reminder: 0 $\Rightarrow$ no smoothing (linear or quadratic in tps)
    - large $\Rightarrow$ close fit to data points
  - Number of knots much less important
- three approaches commonly used (depending on software)
  1. Cross validation
  2. Generalized cross validation
  3. Mixed models
- Often determined by software
  - gam() in mgcv library offers 4 choices: GCV, mixed models (REML), and 2 others

Cross validation

- Same concept as in other uses we've seen
  - Assess how well a model predicts for new observations
  - Find λ^2 that minimizes cv prediction error
  - Exclude an observation, fit spline model with λ^2, predict exclude observation
  - Minimize sum of squared residuals
  - Requires a **LOT** of computing (each obs, many λ^2)
  - There is an approximation that requires a lot less computing (see details at end)

Other approaches to choosing a smoothing parameter

- Generalized Cross validation
  - Same spirit as CV, different details (see end)
  - Faster to compute; sometimes seems to work better
- Linear mixed effects model
  - Linear spline model is still $Y_i = \beta_0 + \beta_1 X_i + u_1 f(X_i, k_1) + u_2 f(X_i, k_2) + \cdots + \varepsilon$
  - Make this a mixed model by making the u_i's be random effects
  - All $u_i \sim N(0, \sigma^2)$ and independent.
  - $f(X_i, k+j)$ is still each of the J spline basis functions
  - Predictions of Y_i based on this model are identical to those using penalized least squares
  - Benefits of the mixed model approach
    - easy to add spline functions to lots of models
    - Very fast computation

Choosing number of knots

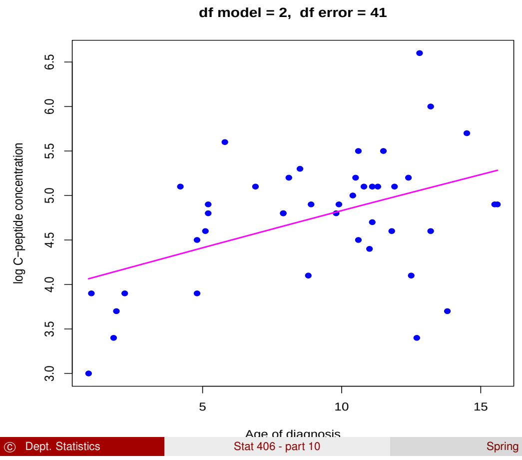
- Still need to choose number of knots (k) and their locations $k_1, ..., k_k$
- Ruppert, Wand and Carroll (2003) recommend 20-40 knots maximum, located so that there are roughly 4-5 unique x values between each pair of knots.
- Most software automatically chooses knots using a strategy consistent (roughly) with this recommendation.
- Knot choice is not usually as important as choice of smoothing parameter
  - As long as there are enough knots, a good fit can usually be obtained.
  - Penalization prevents a fit that is too rough even when there are many knots.

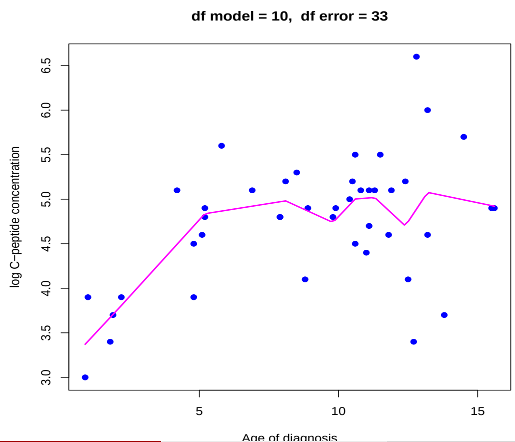
Towards inference with a penalized spline

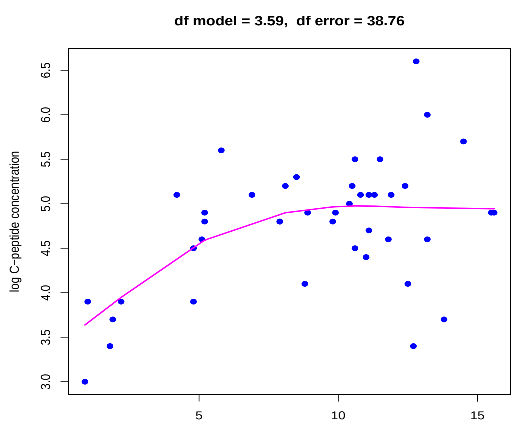
- If we want to compare models (e.g. $\mathrm{E}y = \beta_0 + \beta_1 x$ vs $\mathrm{E}y = f(x)$), need to know df for penalized spline fit
- Can do this test because
  - $\mathrm{E}y = \beta_0 + \beta_1 x$ is nested in $\mathrm{E}y = f(x)$ fit as a linear spline
  - $\mathrm{E}y = \beta_0 + \beta_1 x + \beta_2 x^2$ is nested in $\mathrm{E}y = f(x)$ fit as a thin plate spline
- If we use a penalized linear spline, how many parameters are we using to estimate the mean function ?
- It may seem like we have $k+2$ parameters $\beta_0, \beta_1, u_1, u_2, ..., u_k$.
- But fewer than $k+2$ because of penalization.
  - Actual number of parameters depends on the smoothing parameter λ^2.

Model df

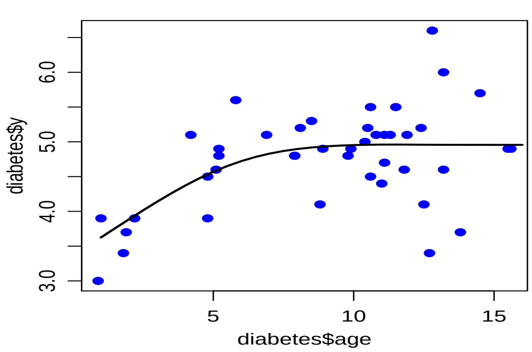
- Model df has two components: the β model and the spline basis functions
- Knowing the total model df tells you how wiggly the spline part is
  - linear spline: $\beta_0 + \beta_1 X_i$, so 1 df for that part of the model
    - Remember, intercept not counted
  - If model df = 1 or 1.1, spline model essentially a straight line
  - If model df = 2, spline model as wiggly as a quadratic
  - If model df = many more, model is very wiggly
- diabetes data: model df = 2.39







Diabetes data: cubic spline, 2.39 df



Towards inference with a penalized spline

- If we want a confidence or prediction interval around the predicted line, need to know df for error.
- And need to know error df and estimate error variance σ^2.
- Both can be computed. Lots of details (at end)
- Note: unlike usual models model df + error df $\neq$ N-1
- Diabetes data: error df = 39.01
  - Model df + error df = 2.39 + 39.01 = 41.40 (not 42 = N-1)

Extensions of penalized splines

- More than one X variable
  - Can fit either as a thin plate spline, $f(X_1, X_2)$
  - or as additive effects: $f_1(X_1) + f_2(X_2)$
  - Can combine parametric and nonparametric forms: $\beta_0 + \beta_1 X_1 + f(X_2)$
- Additive effects models sometimes called Generalized Additive Models (GAM's)
- Penalized splines provide a model for Ey
- Our discussion has only considered $y_i \sim N(\mathrm{Ey_i}, \sigma^2)$
- Can combine with GLM ideas, e.g.: $y_i \sim Poisson(f(x_i))$ or $Binomial(f(x_i))$

Details

The next slides collect mathematical and statistical details. These include:

- Finding the penalized LS estimates
- Approximation to cross-validated prediction error
- Generalized CV and mixed model approaches to choosing a smoothing parameter
- Model degrees of freedom
- Estimating σ^2 and error df

Finding the penalized LS estimate of $(\beta_0, \beta_1, u_1, ..., u_k)'$

- If we let D = diag(0,0,1,...,1) (k terms), then

$$\begin{aligned}(\boldsymbol{y} - x\beta)'(\boldsymbol{y} - x\beta) + \lambda^2 \sum_{j=1}^{k} u_j^2 &= (\boldsymbol{y} - x\beta)'(\boldsymbol{y} - x\beta) + \lambda^2 \beta' D\beta \\ &= \boldsymbol{y}'\boldsymbol{y} - 2\boldsymbol{y}'x\beta + \beta'x'x\beta + \lambda^2\beta'D\beta \\ &= \boldsymbol{y}'\boldsymbol{y} - 2\boldsymbol{y}'x\beta + \beta'(x'x + \lambda^2 D)\beta\end{aligned}$$

- Set derivatives with respect to β equal to $\mathbf{0}$
- estimating equations: $(x'x + \lambda^2 D)\beta \equiv x'\boldsymbol{y}$
- solution: $\hat{\beta}_{\lambda^2} = (x'x + \lambda^2 D)^{-1} x'\boldsymbol{y}$ for any fixed $\lambda^2 \geq 0$
- predicted values: $\hat{\boldsymbol{y}}_{\lambda^2} \equiv x\hat{\beta}_{\lambda^2} = x(x'x + \lambda^2 D)^{-1}x'\boldsymbol{y}$

Approximation to CV prediction error

- There is a quick approximation to $CV(\lambda^2)$

$$CV(\lambda^2) \approx \sum_{i=1}^{n} \left\{ \frac{y_i - \hat{f}(x_i; \lambda^2)}{1 - S_{\lambda^2, ii}} \right\}^2$$

, where $S_{\lambda^2,ii}$ is the i^{th} diagonal element of the smoother matrix $S_{\lambda^2} = x(x'x + \lambda^2 D)^{-1} x'$.

- Remember that $\hat{\boldsymbol{y}} = x(x'x + \lambda^2 D)^{-1} x'y = S_{\lambda^2} y$
- OLS: $\hat{\boldsymbol{y}} = \boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-}\boldsymbol{X}'y = P_{\boldsymbol{X}} y$
- The smoother matrix S_{λ^2} is analogous to the "hat" or projection matrix, $P_{\boldsymbol{X}}$ in a Gauss-Markov model.

Approximation to CV prediction error

- Stat 500: discussed "deleted residuals" $y_i - \hat{y}_{-i}$, where $\hat{y}_{-i}$ is the prediction of y_i when model fit without observation i.
- Can compute with refitting the model N times

$$y_i - \hat{y}_{-i} = \frac{y_i - \hat{y}_i}{1 - h_{ii}},$$

where h_{ii} is the i^{th} diagonal element of the "hat" matrix $H = P_x = x(x'x)^{-}x'$.

- $h_{ii} =$ "leverage" of observation i
- Thus, the approximation $CV(\lambda^2) \approx \sum_{i=1}^{n} \left\{ \frac{y_i - \hat{f}(x_i;\lambda^2)}{1 - S_{\lambda^2,ii}} \right\}^2$ is analogous to the PRESS statistic $\sum_{i=1}^{n} (y_i - \hat{y}_{-i})^2 = \sum_{i=1}^{n} (\frac{y_i - \hat{y}_i}{1 - h_{ii}})^2$ used in multiple regression.

2. Generalized Cross-Validation (GCV)

- GCV is an approximation to CV obtained as follows:

$$GCV(\lambda^2) \equiv \sum_{i=1}^{n} \left\{ \frac{y_i - \hat{f}(x_i; \lambda^2)}{1 - \frac{1}{n}\text{trace}(S_{\lambda^2})} \right\}^2$$

- Since trace(S_{λ^2}) = $\sum_{i=1}^{n} S_{\lambda^2,ii}$, GCV is $CV(\lambda^2)$ using the average $\frac{1}{n}\sum_{i=1}^{n} S_{\lambda^2,ii}$ instead of each specific element
- Used same way: find λ^2 minimizes $GCV(\lambda^2)$
- GCV is not a generalization of CV
- Originally proposed because faster to compute
- In some situations, seems to work better than CV, see Wahba, G. (1990). *Spline Models for Observational Data* for details
- And in very complicated situations, cannot compute H but can estimate $trace(H)$, so can't use CV but can use GCV.

3. The Linear Mixed Effects Model Approach

- Recall that for our linear spline approach, we assume the model $y_i = \beta_0 + \beta_1 x_i + \sum_{j=1}^{k} u_j (x_i - k_j)^+ + \epsilon_i$ for $i = 1, ..., n$; where $e_1, ..., e_n \overset{i.i.d.}{\sim} (0, \sigma^2)$
- Suppose we add the following assumptions: $u_1, ..., u_k \overset{i.i.d.}{\sim} N(0, \sigma_u^2)$ independent of $e_1, ..., e_n \overset{i.i.d.}{\sim} N(0, \sigma_e^2).(\sigma_e^2 \equiv \sigma^2)$
- Then we may write our model as $\boldsymbol{y} = x\beta + Z\boldsymbol{u} + \epsilon$, where

$$X = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix} \beta = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} Z = \begin{bmatrix} (x_1 - k_1)^+ & . & . & . & (x_1 - k_k)^+ \\ (x_2 - k_1)^+ & . & . & . & (x_2 - k_k)^+ \\ \vdots & & & & \vdots \\ (x_n - k_1)^+ & . & . & . & (x_n - k_k)^+ \end{bmatrix}$$

Mixed effects model

$$\boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \boldsymbol{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_k \end{bmatrix} \epsilon = \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{bmatrix}$$

$$\begin{bmatrix} \boldsymbol{u} \\ \epsilon \end{bmatrix} \sim N\left(\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \sigma_u^2 I & 0 \\ 0 & \sigma_u^2 I \end{bmatrix} \right)$$

- This is a linear mixed effects model!

Mixed effects model

- It can be shown that the BLUP of $X\beta + Z\boldsymbol{u}$ is equal to $w(w'w + \frac{\sigma_e^2}{\sigma_u^2 D})^{-1} w'\boldsymbol{y}$ where $w = [x, z]$.
- Thus, the BLUP of $X\beta + Z\boldsymbol{u}$ is equal to $S_{\frac{\sigma_e^2}{\sigma_u^2}} \boldsymbol{y} =$ (Fitted values of linear spline smoother for $\lambda^2 = \frac{\sigma_e^2}{\sigma_u^2}$))
- Thus, we can use either ML or REML to estimate σ_u^2 and σ_e^2. (Denote estimates by $\hat{\sigma}_u^2$ and $\hat{\sigma}_e^2$.)
- Then we can estimate β by $\hat{\beta}_{\hat{\boldsymbol{\Sigma}}} = (x'\hat{\boldsymbol{\Sigma}}^{-1} x)^{-1} x' \hat{\boldsymbol{\Sigma}} \boldsymbol{y}$ and predict $\boldsymbol{u}$ by $\hat{\boldsymbol{u}}_{\hat{\boldsymbol{\Sigma}}} = \hat{G} Z' \hat{\boldsymbol{\Sigma}}^{-1} (\boldsymbol{y} - x\hat{\beta}_{\hat{\boldsymbol{\Sigma}}}) = \hat{\sigma}_u^2 Z' \hat{\boldsymbol{\Sigma}}^{-1} (\boldsymbol{y} - x\hat{\beta}_{\hat{\boldsymbol{\Sigma}}})$ where $\hat{\boldsymbol{\Sigma}} = \hat{\sigma}_u^2 ZZ' + \hat{\sigma}_e^2 I$
- The resulting coefficients $\begin{bmatrix} \hat{\beta}_{\hat{\boldsymbol{\Sigma}}} \\ \hat{\boldsymbol{u}}_{\hat{\boldsymbol{\Sigma}}} \end{bmatrix}$ will be equal to the estimate obtained using penalized least squares with smoothing parameter $\lambda^2 = \frac{\hat{\sigma}_e^2}{\hat{\sigma}_u^2}$

Model df

- However, $u_1, u_2, ..., u_k$ are not completely free parameters because of penalization.
- The effective number of parameters is lower than k+2 and depends on the value of the smoothing parameter λ^2.
- Recall that our estimates of $\beta_0, \beta_1, u_1, u_2, ..., u_k$ minimize $\sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_i - \sum_{j=1}^k u_j (x_i - k_j)^+)^2 + \lambda^2 \sum_{j=1}^k u_j^2$
- A larger λ^2 means less freedom to choose values for $u_1, ..., u_k$ for from 0.
- Thus, the number of effective parameters should decrease as λ^2 increases.
- In the Gauss-Markov framework with no penalization, the number of free parameters used to estimate the mean of $\boldsymbol{y}(x\beta)$ is $rank(x) = rank(P_x) = trace(P_x)$

Model df

- For a smoother, the smoother matrix S plays the role of P_x.
- For penalized linear splines, the smoother matrix is $S_{\lambda^2} = x(x'x + \lambda^2 D)^{-1} x'$ where

$$X = \begin{bmatrix} 1 & x_1 & (x_1 - k_1)^+ ... (x_1 - k_k)^+ \\ 1 & x_2 & (x_2 - k_1)^+ ... (x_2 - k_k)^+ \\ . & . & . \\ . & . & . \\ . & . & . \\ 1 & x_n & (x_n - k_1)^+ ... (x_n - k_k)^+ \end{bmatrix} \quad D = \begin{bmatrix} \overset{0}{2 \times 2} & 0 \\ 0 & \overset{I}{k \times k} \end{bmatrix}$$

- Thus, we define the effective number of parameter (or the degrees of freedom) used when estimating f(x) to be $tr(S_{\lambda^2}) = tr[x(x'x + \lambda^2 D)^{-1} x'] = tr[(x'x + \lambda^2 D)^{-1} x'x]$

- Recall that our basic model is $y_i = f(x_i) + \epsilon_i \quad (i = 1, ..., n)$ where $e_1, ..., e_n \overset{i.i.d.}{\sim} (0, \sigma^2)$.
- How should we estimate σ^2 ?
- A natural estimator would be $MSE \equiv \frac{\sum_{i=1}^n \{y_i - \hat{f}(x_i, \lambda^2)\}^2}{df_{ERROR}}$
- df_{ERROR} is usually defined to be $n - 2tr(S_{\lambda^2}) + tr(S_{\lambda^2} S'_{\lambda^2})$.
- To see where this comes from, recall that for $\boldsymbol{w}$ random and A fixed $E(\boldsymbol{w}' A \boldsymbol{w}) = E(\boldsymbol{w})' A E(\boldsymbol{w}) + tr(AVar(\boldsymbol{w}))$

$$\text{Let } \boldsymbol{f} = \begin{bmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_n) \end{bmatrix} \text{ and } \quad \hat{\boldsymbol{f}}_{\lambda^2} = \begin{bmatrix} \hat{f}(x_1; \lambda^2) \\ \hat{f}(x_2; \lambda^2) \\ \vdots \\ \hat{f}(x_n; \lambda^2) \end{bmatrix} = S_{\lambda^2} \boldsymbol{y}$$

- Then, $E[\sum_{i=1}^n \left\{ y_i - \hat{f}(x_i; \lambda^2) \right\}^2]$

$$\begin{aligned}
&= E[(\boldsymbol{y} - \hat{\boldsymbol{f}})'(\boldsymbol{y} - \hat{\boldsymbol{f}})] \\
&= E[||\boldsymbol{y} - \hat{\boldsymbol{f}}||^2] = E[||(I - S_{\lambda^2})\boldsymbol{y}||^2] \\
&= E[\boldsymbol{y}'(I - S_{\lambda^2})'(I - S_{\lambda^2})\boldsymbol{y}] \\
&= \boldsymbol{f}'(I - S_{\lambda^2})'(I - S_{\lambda^2})\boldsymbol{f} + \text{tr}[(I - S_{\lambda^2})'(I - S_{\lambda^2})\sigma^2 I] \\
&= ||(I - S_{\lambda^2})\boldsymbol{f}||^2 + \sigma^2 \text{tr}[I - S'_{\lambda^2} - S_{\lambda^2} + S'_{\lambda^2} S'_{\lambda^2}] \\
&= ||\boldsymbol{f} - S_{\lambda^2}\boldsymbol{f}||^2 + \sigma^2[\text{tr}(I) - 2\text{tr}(S_{\lambda^2}) + \text{tr}(S'_{\lambda^2} S_{\lambda^2})] \\
&\approx \sigma^2[n - 2\text{tr}(S_{\lambda^2}) + \text{tr}(S'_{\lambda^2} S_{\lambda^2})]
\end{aligned}$$

- Thus, if we define $df_{ERROR} = n - 2\text{tr}(S_{\lambda^2}) + \text{tr}(S'_{\lambda^2} S_{\lambda^2}), E(MSE) \approx \sigma^2$

- The Standard Error of $\hat{f}(x; \sigma^2)$:

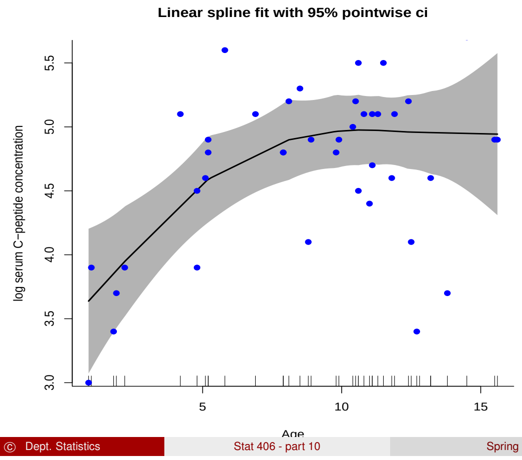
$$\hat{f}(x; \lambda^2) = \hat{\beta}_o + \hat{\beta}_1 x + \sum_{j=1}^k \hat{u}_j (x - k_j)^+$$

$$= [1, x, (x - k_1)^+, ..., (x - k_k)^+] \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \\ \hat{u}_1 \\ \vdots \\ \hat{u}_k \end{bmatrix}$$

$$= [1, x, (x - k_1)^+, ..., (x - k_k)^+](x'x + \lambda^2 D)^{-1} x' \boldsymbol{y} = \boldsymbol{S_{\lambda^2}}' \boldsymbol{y}$$

- If λ^2 and the knots, k_i, are fixed and not chosen as a function of the data, $\boldsymbol{C}$ is just a fixed (nonrandom) vector.

- Thus, $Var[\hat{f}(x; \lambda^2)] = Var(\boldsymbol{S_{\lambda^2}}' \boldsymbol{y}) = \boldsymbol{S_{\lambda^2}}' \sigma^2 \boldsymbol{I} \boldsymbol{S_{\lambda^2}} = \sigma^2 \boldsymbol{S_{\lambda^2}}' \boldsymbol{S_{\lambda^2}}$
- It follows that the standard error for $\hat{f}(x; \lambda^2)$ is $SE[\hat{f}(x; \lambda^2)] = \sqrt{MSE\ \boldsymbol{S_{\lambda^2}}' \boldsymbol{S_{\lambda^2}}}$
- If λ^2 and/or the knots are selected based on the data (as is usually the case), $\sqrt{MSE\ \boldsymbol{S_{\lambda^2}}' \boldsymbol{S_{\lambda^2}}}$ is still used as an approximate standard error.
- However, that approximate standard error may be smaller than it should be because it does not account for variation in the $\boldsymbol{S_{\lambda^2}}$ vector itself
- Ruppert, Wand, and Carroll (2003) suggest other strategies that use the linear mixed effects model framework.
- Calculate pointwise $1 - \alpha$ confidence intervals for $\hat{f}(x_i)$ by $t_{1-\alpha/2, dfe} \sqrt{Var[\hat{f}(x; \lambda^2)]}$,
where dfe is the df_{ERROR} defined a few pages ago


Linear spline fit with 95% pointwise ci
log serum C−peptide concentration
3.0
3.5
4.0
4.5
5.0
5.5
5
10
15
Age