

Module 5, part II: Penalized and Smoothing Splines

BIOS 526

Reading

- Sections 5.4 and 5.5 in Hastie et al.
- Sections 3.1 - 3.14, 4.9 in Ruppert et al.

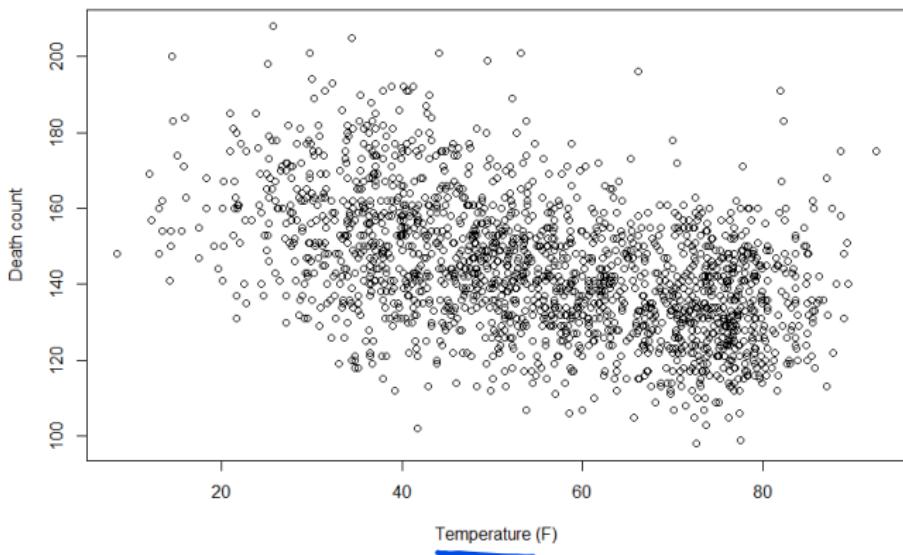
Concepts

- Constraints and penalized regression.
- Smoothing matrix and smoothing parameter.
- Generalized cross-validation to choose roughness penalty.
- Mixed models to choose roughness penalty.

Motivating Example: Daily Temperature and Deaths

- alldeaths: daily non-accidental deaths in the 5-county New York City, 2001-2005.
- Temp: daily temperature in Fahrenheit.

```
> load ("NYC.RData")
> plot(alldeaths~Temp,xlab="Temperature (F)",ylab ="Death count",data=health)
```



Regression Problem

Let y_i be the number of non-accidental deaths on day i and x_i be the same-day temperature.

We consider the nonparametric regression problem:

$$y_i = g(x_i) + \varepsilon_i, \quad \varepsilon_i \stackrel{\text{iid}}{\sim} (0, \sigma^2)$$

We can approximate $g(x_i)$ using a generalized additive model

$$y_i = \beta_0 + \sum_{m=1}^M \beta_m b_m(x_i) + \varepsilon_i, \quad \varepsilon_i \stackrel{\text{iid}}{\sim} (0, \sigma^2)$$

E.g., linear spline with 9 equidistant interior knots $\kappa_1, \kappa_2, \dots, \kappa_9$ within the observed range of daily temperature, a piecewise linear spline model is

$$g(x_i) = \beta_0 + \beta_1 x_i + \beta_2 (x_i - \kappa_1)_+ + \beta_3 (x_i - \kappa_2)_+ + \dots + \beta_{10} (x_i - \kappa_9)_+$$

Automatic Knot Selection

What if we don't know the number and locations of the knots?

Approach:

- Start with **a lot of knots**. This ensures that we will not miss important fine-scale behaviour.
- Assume most of the knots are not useful and **shrink their coefficients toward zero**.
- Determine how much to shrink based on some criteria (e.g. GCV or AIC).

Benefits:

- Knot placement is not important if the number is dense enough.
- Shrinking most coefficients to zero will stabilize model estimation similar to performing variable selection.

Penalized Spline

Consider the basis expansion:

$$g(x_i) = \beta_0 + \beta_1 x_i + \sum_{m=1}^M \beta_{m+1} b_m(x_i)$$

Constrain the magnitude of the coefficients β_j .

Consider the ridge-regression penalty:

$$\beta_2^2 + \beta_3^2 + \dots + \beta_{M+1}^2 \leq C$$

equivalently,

$$\|\beta^*\|_2^2 \leq C$$

where $\beta^* = [\beta_2, \dots, \beta_{M+1}]'$ and C is a positive constant.

Notation:
 $\|\beta^*\|_2$:
L2-norm

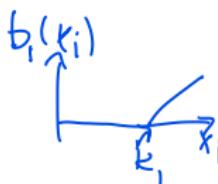
Penalties

- Ridge regression = ℓ_2 -penalty = $\|\beta^*\|_2^2$.
- Other penalties: lasso = absolute value = ℓ_1 -penalty = $\|\beta^*\|_1 = \sum_{j=2}^{M+1} |\beta_j|$.
- Ridge shrinks coefficients of vectors in b-spline basis, but does not induce sparsity.
- Ridge is easy to solve – closed form solution!
- Lasso tends to make some coefficients exactly zero. Trickier to solve. More on this later in the course.
- A small C will shrink more coefficients, as well as shrink them closer to zero.
- Our goal:

convert the two problems—number of knots and their location—into a single parameter that we can choose with GCV

Matrix of $g()$

For simplicity, consider a linear spline. Then evaluate the basis functions at each x_i , $i = 1, \dots, n$:

$$\mathbf{X} = \begin{bmatrix} 1 & x_1 & \overbrace{(x_1 - \kappa_1)_+}^{\text{basis function}} & \cdots & \overbrace{(x_1 - \kappa_{M+1})_+}^{\text{basis function}} \\ 1 & x_2 & (x_2 - \kappa_1)_+ & \cdots & (x_2 - \kappa_{M+1})_+ \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_n & (x_n - \kappa_1)_+ & \cdots & (x_n - \kappa_{M+1})_+ \end{bmatrix}$$


Then write $g(x_i)$, $i = 1, \dots, n$ in matrix form:

$$\mathbf{G} \equiv \mathbf{X}\beta, \quad g(x_i) = \beta_0 + \beta_1 x_i + \sum_{m=1}^M \beta_{m+1} (x_i - \kappa_m)_+$$

Then the residuals are $\mathbf{Y} - \mathbf{G} = \mathbf{Y} - \mathbf{X}\beta$.

Constrained formulation

We define the objective function:

$$\underset{\beta}{\operatorname{argmin}} \quad (\mathbf{Y} - \mathbf{X}\beta)'(\mathbf{Y} - \mathbf{X}\beta) \text{ subject to } \underline{\beta' \mathbf{B} \beta \leq C}.$$

Here, \mathbf{B} is a diagonal matrix with 0 and 1 entries selecting which coefficients are penalized:

$$\mathbf{B} = \begin{bmatrix} 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 \end{bmatrix}, \text{ then } \beta' \mathbf{B} \beta = \sum_{m=2}^{M+1} \beta_m^2$$

- don't penalize β_0 or β_1

Penalized formulation

This problem can be equivalently formulated as

$$\underset{\beta}{\operatorname{argmin}} \quad (Y - X\beta)'(Y - X\beta) + \lambda \beta' B \beta$$

There is a one-to-one mapping between λ and the constraint C . λ is often called the smoothing parameter, tuning parameter, hyperparameter.

note:
 $\frac{d}{d\beta} (y - X\beta)^2$
 $-2X(y - X\beta)$

For QUADRATIC FORM matrix cookbook

Closed-form solution

(won't need to derive on exam)

$$\underset{\beta}{\operatorname{argmin}} \quad (\mathbf{Y} - \mathbf{X}\beta)'(\mathbf{Y} - \mathbf{X}\beta) + \lambda\beta'\mathbf{B}\beta.$$

Differentiate wrt β and set to zero:

$$\begin{aligned}
 -2\mathbf{X}'(\mathbf{Y} - \mathbf{X}\beta) + 2\lambda\mathbf{B}\beta &= 0 \\
 -\mathbf{X}'\mathbf{Y} + \mathbf{X}'\mathbf{X}\beta + \lambda\mathbf{B}\beta &= 0 \\
 (\mathbf{X}'\mathbf{X} + \lambda\mathbf{B})^{-1}(\mathbf{X}'\mathbf{Y}) &= \hat{\beta} = (\mathbf{X}'\mathbf{X} + \lambda\mathbf{B})^{-1} \mathbf{X}'\mathbf{Y}.
 \end{aligned}$$

Closed-form solution

General form for Ridge estimator

The least squares solution is

$$\hat{\beta} = (\mathbf{X}'\mathbf{X} + \lambda \mathbf{B})^{-1} \mathbf{X}'\mathbf{Y}$$

... --- j.

for some positive number λ . Note:

- When $\lambda = 0$, $\hat{\beta}$ becomes the ordinary least squares estimate. So no penalization is present ($C = \infty$).
- When $\lambda \rightarrow \infty$, $(\mathbf{X}'\mathbf{X} + \lambda \mathbf{B})^{-1}$ becomes small, so $\hat{\beta}_j \rightarrow 0$ if $B_{jj} = 1$.

here, $\hat{\beta}_0$ and $\hat{\beta}_1$ are not penalized

but the coefficients corresponding to knots
go to 0 as $\lambda \rightarrow \infty$ (converge to simple linear regression)

Mortality and Temperature Example

Consider the death and mortality analysis. Assume 40 equidistant knots and linear splines:

$$y_i = \beta_0 + \beta_1 x_i + \sum_{m=1}^{40} \beta_{1+m} (x_i - \kappa_m)_+$$

We will penalize $\beta_{1+m}, \dots, \beta_{M+1}$ using the **B** matrix:

$$\mathbf{B} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ \mathbf{0}_{40 \times 7} & \mathbf{0}_{7 \times 40} \\ & \mathbf{I}_{40 \times 40} \end{bmatrix}$$

$$\therefore \beta' \mathbf{B} \beta = \sum_{m=1}^{40} \beta_{1+m}^2$$

Creating piecewise linear spline

We can create a design matrix with piecewise linear splines.

```
> knots = seq(range(health$Temp)[1], range(health$Temp)[2], length.out = 40+2)
> # place knots evenly on interior of the range of x
> knots = knots[c(2:(length(knots)-1))]
> X = cbind(rep(1,length(health$Temp)), health$Temp)
> for (i in 1:length(knots)) {
+   X = cbind(X, (health$Temp-knots[i])*(health$Temp>knots[i]))
+ }
> B = diag(42)
> B[1,1]=0
> B[2,2]=0
> dim (X); dim (B)
[1] 1826    42
[1] 42 42
```

Mortality and Temperature Example

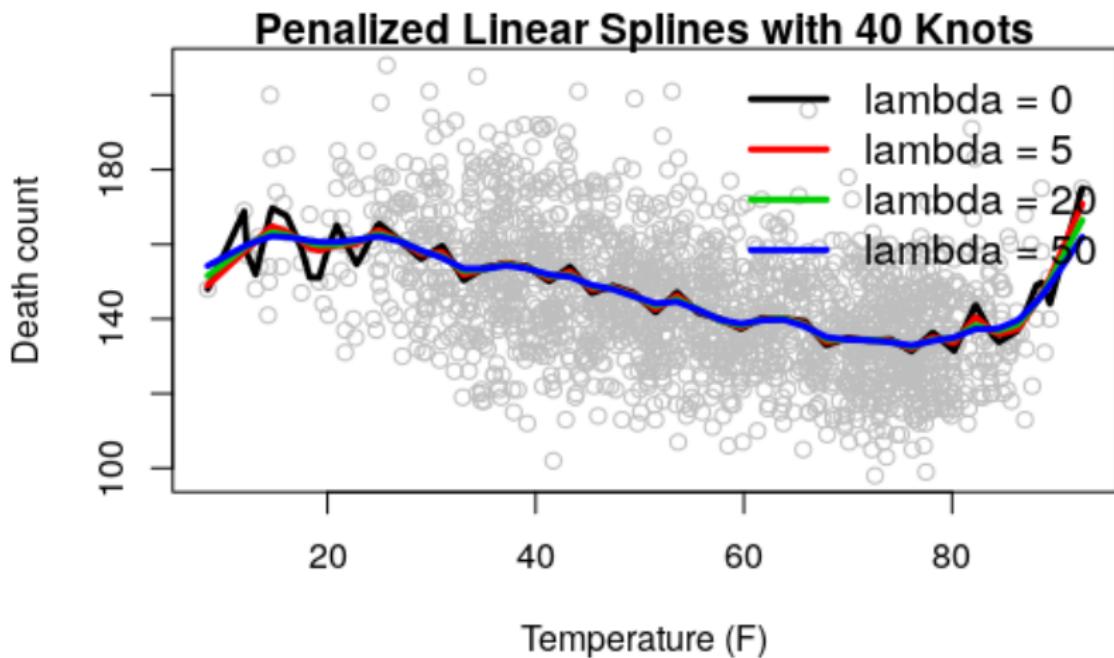
We now search through different values of λ . For each λ , we will

- Calculate the penalized $\hat{\beta}$.
- Calculate $\hat{\beta}' \mathbf{B} \hat{\beta}$.
- Calculate the fitted value $\hat{\mathbf{Y}} = \mathbf{X} \hat{\beta}$.
- Calculate the GCV using the matrix: $\mathbf{X}(\mathbf{X}' \mathbf{X} + \lambda \mathbf{B})^{-1} \mathbf{X}'$.

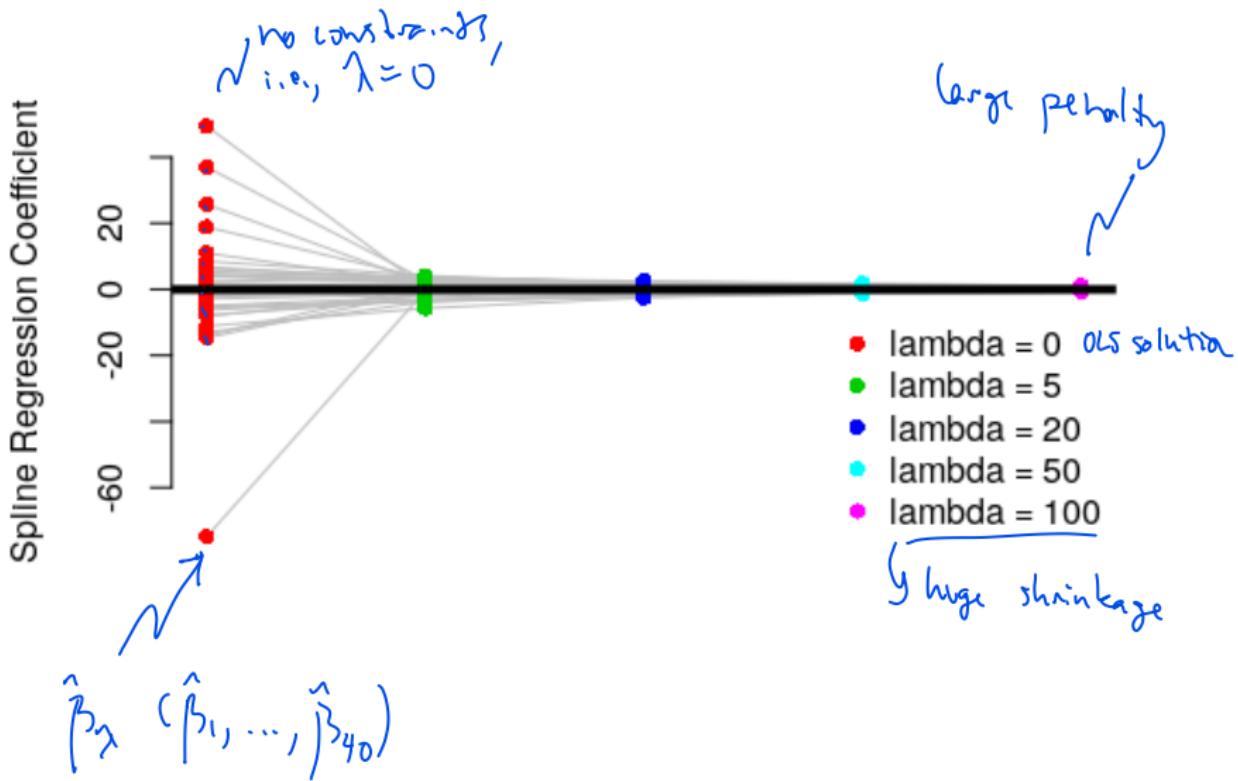
We will select the λ with the smallest GCV.

```
> Y = health$alldeaths  
  
> lambda = 0  
> beta = solve (t(X) %*% X + lambda*B) %*% t(X) %*% Y  
> H = X %*% solve (t(X) %*% X + lambda*B) %*% t(X)      ##Hat matrix  
> Yhat = X %*% beta                                     ##Fitted values  
> GCV = mean ( (Y-Yhat)^2 ) / (1- mean (diag(H)))^2  
> C = t(beta) %*% B %*% beta
```

Effects of Penalization



Effects of Penalization: Shrinkage



Shrinkage

less wiggly:

General principle:

- ↑ shrinkage → ↓ variance.
- ↑ shrinkage → ↑ bias.

\hat{g} are based on data Y

$$\begin{aligned} & \mathbb{E}[\hat{g}(x_i) - E(\hat{g}(x_i))]^2 \\ & [g(x_i) - E(\hat{g}(x_i))]^2 \end{aligned}$$

How do we determine the tuning parameter λ ?

In other words, how do we determine how much we should shrink?

Effective Degrees of Freedom

With the constraint $\beta' \mathbf{B} \beta < C$, $\hat{\beta}$ is no longer the ordinary least squares estimate.

Let $\hat{\mathbf{Y}} = \mathbf{S} \mathbf{Y}$ where \mathbf{S} is a smoothing matrix.

In ridge regression, $\mathbf{S} = \mathbf{X}(\mathbf{X}'\mathbf{X} + \lambda \mathbf{B})^{-1}\mathbf{X}'$.

Each element is shrunk towards zero. We can define an effective degrees of freedom $\underline{df_{eff}}$ as

$$df_{eff} = \text{tr}(S) . \quad (5)$$

$\sum_{i=1}^n S_{ii}$

Note:

Hence, "effective" df: effective number of parameters

in OLS, $S = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$, $SS = S$, i.e., idempotent, $\text{tr}(S) = \text{rank}(S) = df_{sp}$
in Ridge, $SS \neq S$, replace df with EFFECTIVE DF = $\text{tr}(S)$

Effective Degrees of Freedom, cont.

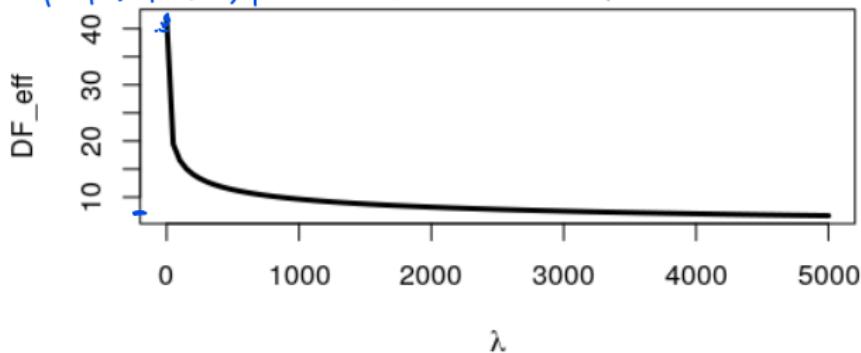
Note when $\lambda = 0$, $df_\lambda = \text{rank}(\mathbf{X}) = p$, the degrees of freedom without penalization.

As $\lambda \uparrow$, $df_\lambda \rightarrow 2$ because $\hat{\beta}_0, \hat{\beta}_1$ are not penalized

here, $\hat{\beta}_0$ has $df=42$
 $(\hat{\beta}_0, \hat{\beta}_1, \hat{\beta}_2, \dots, \hat{\beta}_{42}$ for spline coefficients)

Effective DF

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & - & - & 0 \end{bmatrix} = \mathbf{B}$$



Generalized Cross-validation Error, revisited

We previously defined GCV:

$$GCV = \frac{1}{n} \frac{\sum (y_i - \hat{y}_i)^2}{\left[1 - \frac{1}{n} \text{tr}(H)\right]^2}$$

Note that $\hat{Y} = HY$ where $H = X(X'X)^{-1}X'$.

Now we can apply GCV to **any** prediction of Y that can be written in the form:

$$\hat{Y} = SY.$$

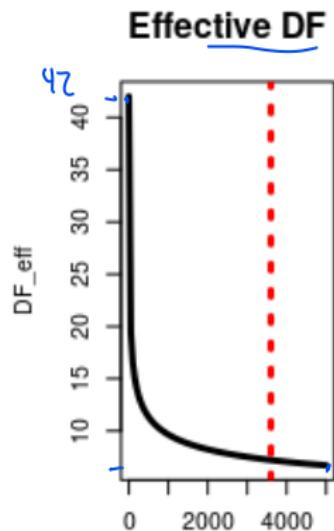
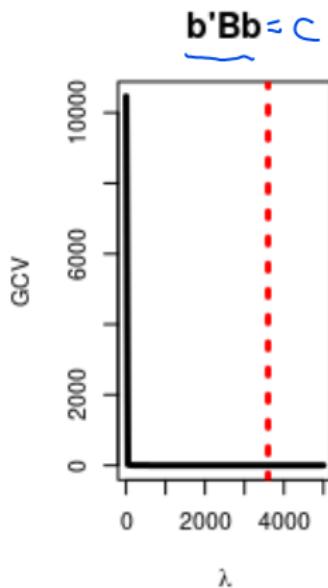
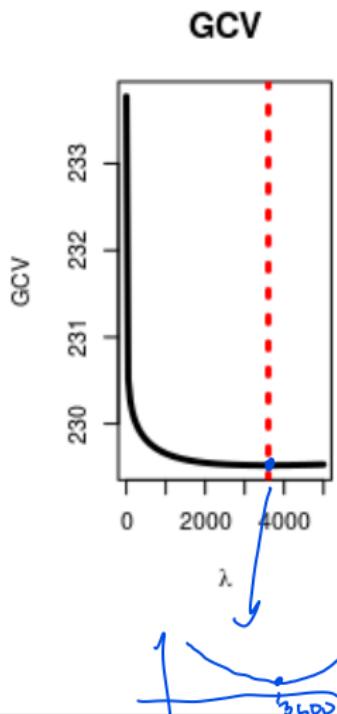
Then GCV is defined:

$$GCV = \frac{1}{n} \frac{\sum (y_i - \hat{y}_i)^2}{\left[1 - \frac{1}{n} \text{tr}(S)\right]^2}$$

This is the definition we will use hereafter.

Smoothing Parameter Selection

Penalized linear splines with 40 knots. (GCV-optimal $\lambda = 3600$)



"⁴⁷ effective number of parameters ↴ is ↑ λ ↑

Residual Error Variance Estimate

Recall our model is

$$y_i = g(x_i) + \epsilon_i \quad \epsilon_i \stackrel{iid}{\sim} N(0, \sigma^2).$$

We now have an estimate $\hat{g}(x_i)$. How about σ^2 ?

We have two options:

in mgcv::gam
edf

$$\hat{\sigma}^2 = \frac{\sum_{i=1}^n [y_i - \hat{g}(x_i)]^2}{n - df_{\text{eff}}}. \quad (6)$$

The above is a biased estimate. Some software gives you the option to use

in mgcv::gam
Ref. df

$$\hat{\sigma}_{\text{unbiased}}^2 = \frac{\sum_{i=1}^n [y_i - \hat{g}(x_i)]^2}{n - \underbrace{2\text{tr}\{\mathbf{S}\}}_{\text{df}} + \underbrace{\text{tr}\{\mathbf{SS}'\}}_{\text{df}}}. \quad (7)$$

note: in OLS, $n - 2p + p < n - p$ ✓

Variance of $\hat{g}(x_i)$

Now we can calculate uncertainty associated with $\hat{g}(x_i)$ at each x_i .

With slight abuse of notation, let \tilde{x}_i' be the row vector of basis function values for x_i .
....

The variance of $\hat{g}(x_i)$ is

$$\tilde{x}_i' = (1, \underbrace{x_i, (x_i - \bar{x}_1)_+, \dots, (x_i - \bar{x}_{40})_+}_{\text{pxl vector}})$$

$$\begin{aligned}\text{Var}(\hat{g}(x_i)) &= \text{Var}(\tilde{x}_i' \hat{\beta}) = \tilde{x}_i' \text{Var}(\hat{\beta}) \tilde{x}_i \\&= \tilde{x}_i' \text{Var}((X'X + \lambda B)^{-1} X' Y) \tilde{x}_i \\&= \tilde{x}_i' [X'X + \lambda B]^{-1} X' \text{Var}(Y) [X'X + \lambda B]^{-1} \tilde{x}_i \\&= \sigma^2 \tilde{x}_i' [X'X + \lambda B]^{-1} X' X [X'X + \lambda B]^{-1} \tilde{x}_i\end{aligned}$$

Note: you should decide whether or not to include the variance due to the intercept. If $x_i[1] = 1$, then the variance estimate of $\hat{g}(x_i)$ includes this source of uncertainty.

(sometimes it is confusing whether $\hat{g}(x_i)$ includes β_0 , we will see that mgcv::gam does not include β_0 ,

Confidence interval and prediction interval

Obtain point-wise confidence interval derived from previous expression by plugging in $\hat{\sigma}^2$ for σ^2 .

If $\lambda = 0$:

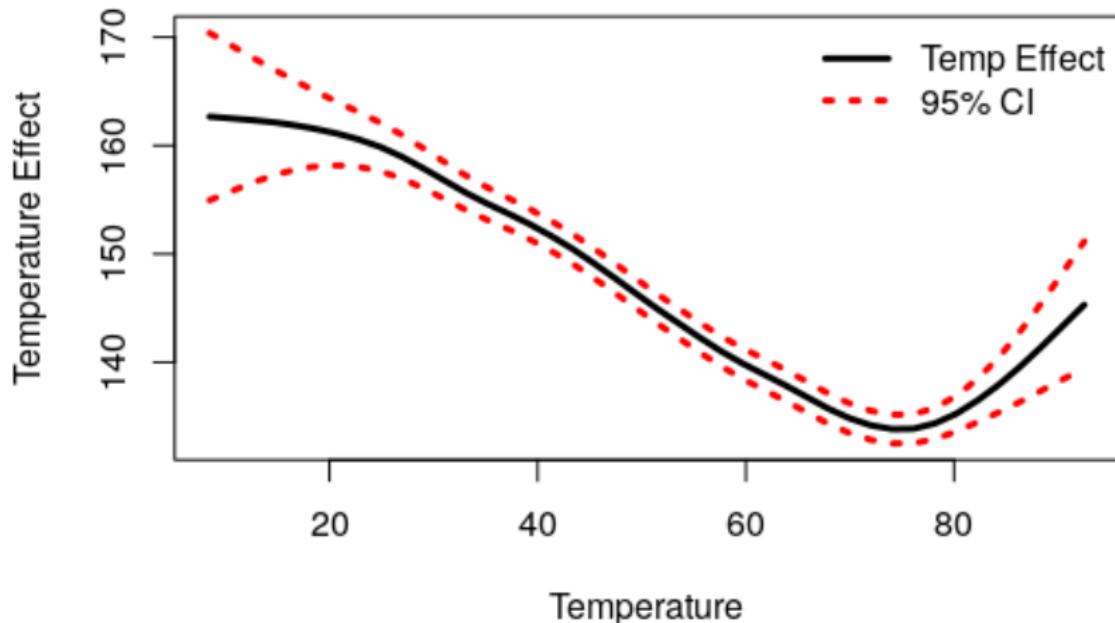
Similarly the variance for an unobserved point y_i^* with covariate x_i^* has variance new value of (y_i^*, \tilde{x}_i^*)

$$Var[y_i^*] = \sigma^2 + \sigma^2 \mathbf{x}_i^{*'} (\mathbf{X}'\mathbf{X} + \lambda \mathbf{B})^{-1} (\mathbf{X}'\mathbf{X}) (\mathbf{X}'\mathbf{X} + \lambda \mathbf{B})^{-1} \mathbf{x}_i^*.$$

includes our estimate of σ^2

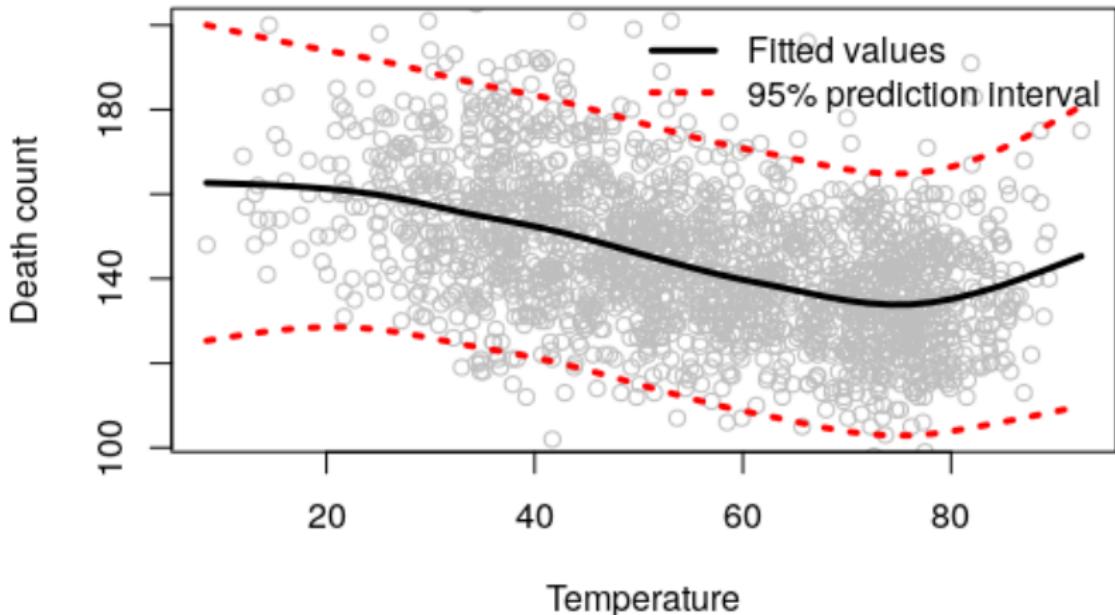
Temperature Effect on Mortality: pointwise CI

```
> Upper95.ci = Yhat + 1.96* sqrt(diag(pred.vcov))  
> Lower95.ci = Yhat - 1.96* sqrt(diag(pred.vcov))
```



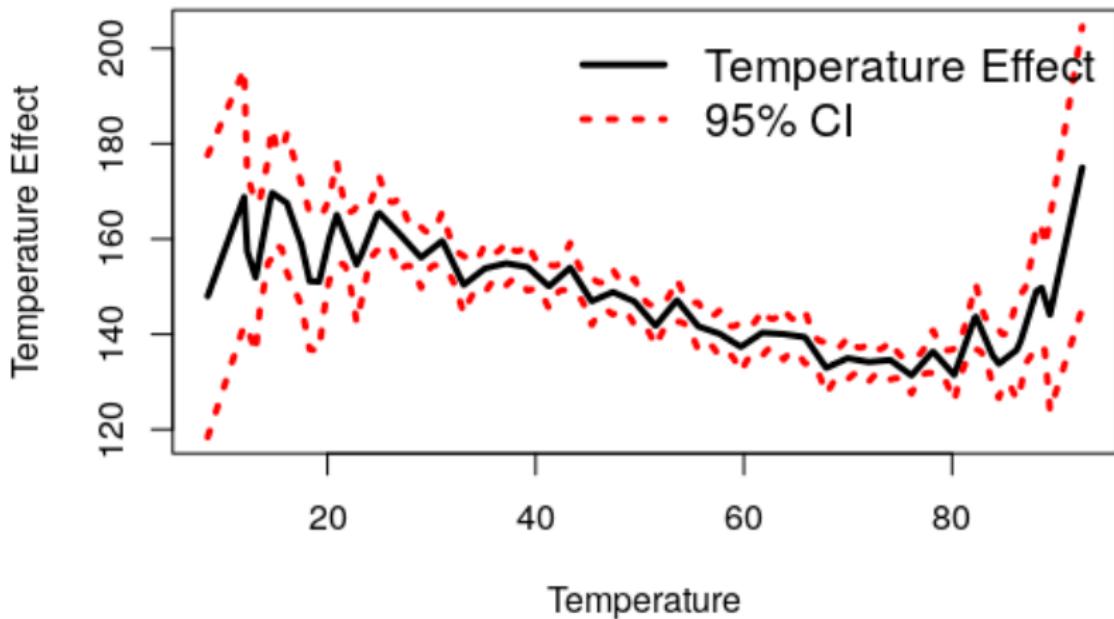
Daily Mortality Prediction

```
> Upper95 = Yhat + 1.96* (sigma1 + sqrt(diag (pred.vcov)) )  
> Lower95 = Yhat - 1.96* (sigma1 + sqrt (diag (pred.vcov)) )
```



Temperature Effect on Mortality

Compare to a model without penalization ($\lambda = 0$).



Smoothing Splines: other penalties

(good to know, will not be part of exam)

A function with large second derivatives can be interpreted as rougher, as the function is allowed to change very rapidly.

We now add a “roughness” penalty to encourage smoothness:

$$\hat{g}(x) = \underset{\substack{g \in \mathcal{G} \\ \dots}}{\operatorname{argmin}} \left\{ Y - g(x) \right\}^T \left\{ Y - g(x) \right\} + \lambda \int_a^b (g''(x; p))^2 dx$$

\mathcal{G} : twice-differentiable functions

Smoothing spline, cont.

$$\hat{g}(x) = \arg \min_{g \in \mathcal{G}} \{\mathbf{Y} - g(\tilde{\mathbf{x}})\}' \{\mathbf{Y} - g(\tilde{\mathbf{x}})\} + \lambda \int \{g''(x; \boldsymbol{\beta})\}^2 dx.$$

where \mathcal{G} is the class of twice-differentiable functions and $\tilde{\mathbf{x}} \in \mathbb{R}^n$ is the vector of x_i , $i = 1, \dots, n$.

- Note that first derivatives are not penalized.
- The second part uses the squared second-derivative that is a good measure of roughness.
- Shrinks coefficients in a cubic polynomial, causing function to change less quickly.
- λ determines the relative importance of minimizing the residual sum of squares or the roughness.

Smoothing Spline

It turns out the solution $\hat{g}(x)$ is a “natural cubic spline” (a cubic spline with linearity at the boundaries) with knots at the observed points x_i .

More generally, the objective function in (8) with penalized second derivatives is equivalent to

$$(Y - X\beta)'(Y - X\beta) + \lambda \beta' B \beta$$

basis expansion ↗ X ↗ not diagonal
special matrix B
matrix equivalent
to $g''(x_i)$

for a certain B matrix based on second moments of the basis functions, no longer diagonal; see Ruppert et al p. 75.: semi-parametric regression

The key point is that (9) is a general formula applying to different ridge-like penalties for certain B .

As before,

- for a given λ , we can estimate $g(x)$ using penalized least squares;
- search through λ to minimize GCV or another criterion.

Package mgcv in R

note: there is another package called "gamm", not as good,
we will use mgcv

The mgcv (Mixed GAM Computation Vehicle) package in R contains the gam() function to fit a large variety of smoothing splines with **automatic** smoothing parameter selection. We will examine different options throughout the class.

Default option is given in parenthesis.

- Basis functions (default: thin plate regression spline).
- Basis dimension (default: $k = 10$ with one constraint: $\sum \hat{g}(x_i) = 0$, makes max edf=9. min edf=1 (unpenalized linear term, if edf=1, results nearly equivalent to lm()). $10 \text{ parameters} - 1 \text{ constraint} = 9$ free parameters)
- Selection methods (default: GCV).
- Family (default: Gaussian).
- Standard error computation (default: Bayesian).
↳ we'll talk a little about in MS.3

Temperature Effect on Mortality

```
> library (mgcv)
> fit1 = gam(alldeaths~s(Temp), data= health)
> summary(fit1)
```

Family: gaussian

Link function: identity

Parametric coefficients:

| | Estimate | Std. Error | t value | Pr(> t) |
|-------------|----------|------------|---------|------------|
| (Intercept) | 143.917 | 0.354 | 407 | <2e-16 *** |

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Approximate significance of smooth terms:

| | edf | Ref.df | F | p-value |
|---------|------|--------|------|------------|
| s(Temp) | 6.03 | 7.2 | 80.6 | <2e-16 *** |

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

R-sq.(adj) = 0.241 Deviance explained = 24.3%

GCV = 229.47 Scale est. = 228.58 n = 1826

approximate F-statistic:
if edf=1, simplifies to
the usual f-statistic.

mgcv::gam output

Approximate significance of smooth terms:

| | edf | Ref.df | F | p-value |
|---------|------|--------|------|------------|
| s(Temp) | 6.03 | 7.2 | 80.6 | <2e-16 *** |
| --- | | | | |

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

R-sq.(adj) = 0.241 Deviance explained = 24.3%

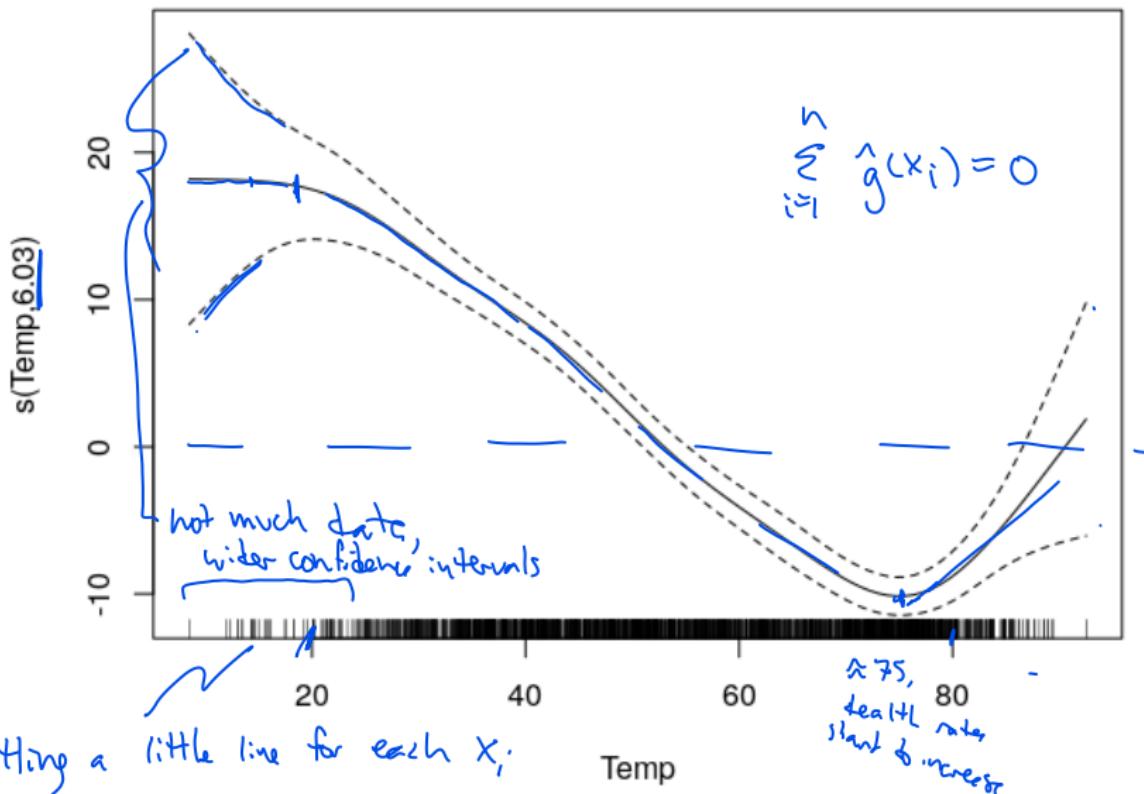
GCV = 229.47 Scale est. = 228.58 n = 1826

$$\hat{\sigma}^2$$

- edf = effective Df for $tr(\mathbf{S})$.
- Ref edf = effective Df for $2tr(\mathbf{S}) - tr(\mathbf{S}'\mathbf{S})$.
- Scale est. = estimated residual error σ^2 (using edf).
- F statistic: approximate significance of Temp. Uses Ref edf.
- Use plots to interpret $\hat{g}(x_i)$.

plot(fit)

Temperature Effect on Mortality



Checking gam

The default is $k = 10$, such that highest possible EDF is 9 (because of identifiability constraint).

```
> gam.check(fit1)
```

Function we use to get diagnostic plots

Method: GCV Optimizer: magic

- Smoothing parameter selection converged after 5 iterations.
- The RMS GCV score gradient at convergence was 7.242e-05 .
- The Hessian was positive definite.
- Model rank = 10 / 10

Basis dimension (k) checking results. Low p-value ($k\text{-index} < 1$) may indicate that k is too low, especially if edf is close to k' .

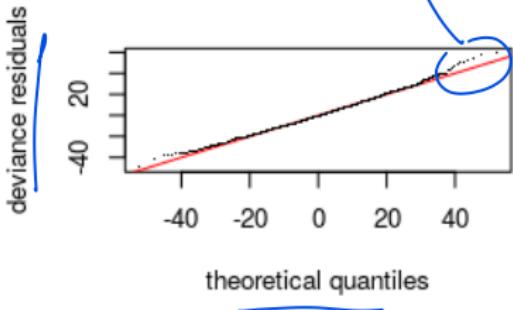
| | k' | edf | k-index | p-value |
|---------|------|------|---------|---------|
| s(Temp) | 9.00 | 6.03 | 1.02 | 0.88 |

H_0 : the basis dimension is adequate
-this test is not very useful

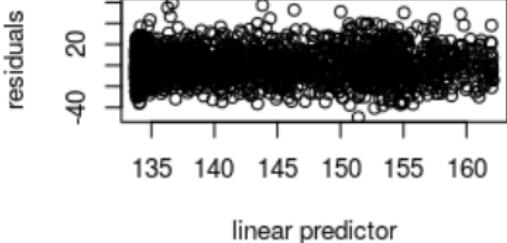
in general k' should be "notably" larger than edf, e.g. $edf < 5$; here we will check fitting model with a higher basis dimension

gam.check plots

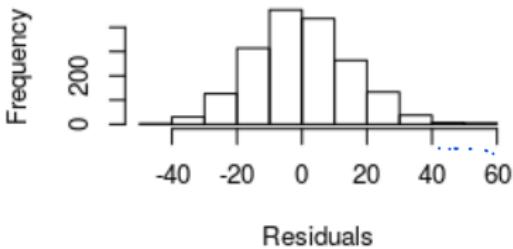
see some departure,
not too bad



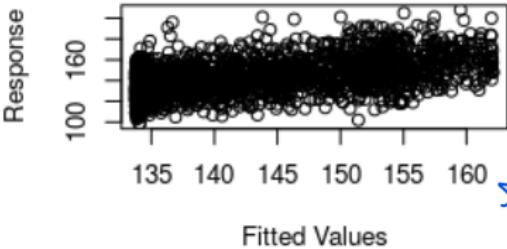
Resids vs. linear pred.



Histogram of residuals



Response vs. Fitted Values

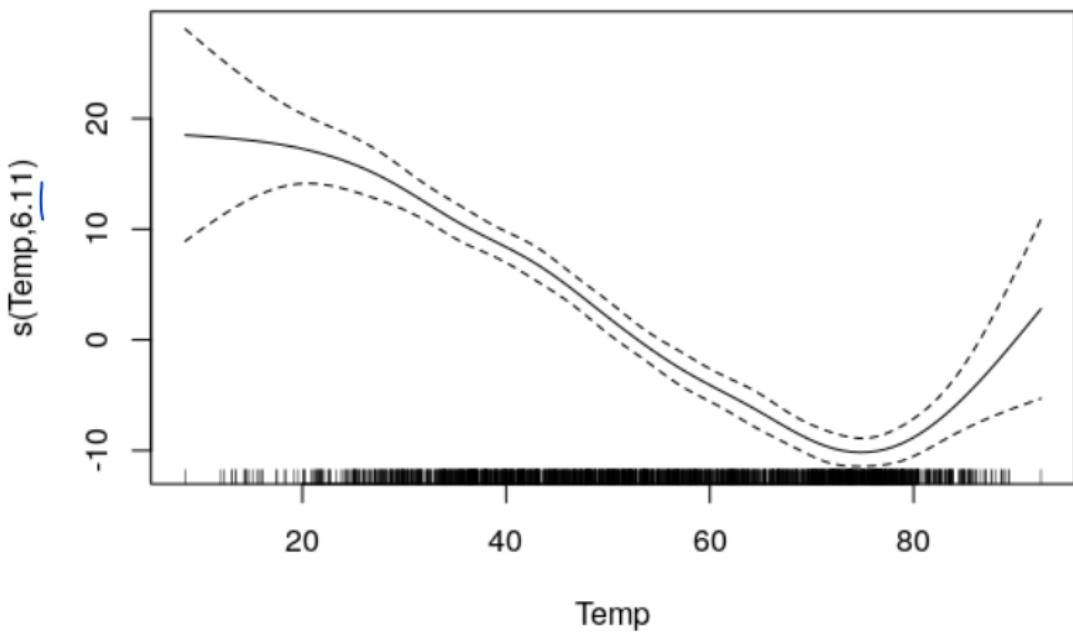


should
look
linear
if it
is
sufficient

Temperature Effect on Mortality using cubic

```
> fit.checkcubic = gam(alldeaths~s(Temp,bs='cr',k=10),method='GCV.Cp',data=health)
```

check to see if a different spline impacts things
here pretty similar to default of thin plate regression spline



Temperature Effect on Mortality

Thin plate splines with $k = 40$. *check whether k*
(sensitivity analysis to basis dimension)

```
> fit2 = gam (alldeaths ~ s(Temp, k = 40), data = health)  
> summary (fit2)
```

Formula:

```
alldeaths ~ s(Temp, k = 40)
```

Parametric coefficients:

| | Estimate | Std. Error | t value | Pr(> t) |
|-------------|----------|------------|---------|------------|
| (Intercept) | 143.917 | 0.354 | 407 | <2e-16 *** |

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Approximate significance of smooth terms:

| edf | Ref.df | F | p-value |
|---------|--------|------|-----------------|
| s(Temp) | 6.23 | 7.85 | 73.9 <2e-16 *** |

*only need a little bit, k=10
sufficient, either is okay*

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

R-sq.(adj) = 0.241 Deviance explained = 24.3%

GCV = 229.51 Scale est. = 228.6 n = 1826

Extract Useful Model Statistics

Full list see ?gamObject.

- AIC (with edf at penalized estimates)

```
> AIC (fit)
```

```
[1] 15109.62
```

- Variance-covariance matrix of the coefficients of your spline basis

```
> dim (fit$Ve) ### Frequentist's
```

```
[1] 10 10
```

```
> dim (fit$Vp) ### n Bayesian n
```

```
[1] 10 10
```

- Fitted value

```
> fit$fitted
```

↳ have frequentist coverage probabilities, so we default to using these

Penalized splines as BLUPs

- GCV may undersmooth.
- An alternative is to treat the coefficients of the truncated polynomials as random effects, and then use BLUPs.
(or whatever basis you are using)
- For concreteness, consider a linear spline:

- applies to non-clustered data
- smoothing trick

$$y_i = \beta_0 + \beta_1 x_i + \sum_{m=1}^M \theta_m (x_i - \kappa_m)_+ + \epsilon_i,$$

random slope

$$\theta_m \stackrel{iid}{\sim} N(0, \tau^2), \quad \epsilon_i \stackrel{iid}{\sim} N(0, \sigma^2)$$

• very wiggly data have larger τ^2
• very noisy data have larger σ^2

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_M \end{bmatrix} \quad \Theta = \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_M \end{bmatrix} \quad \mathbf{X} = \begin{bmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix} \quad \mathbf{Z} = \begin{bmatrix} (x_1 - \kappa_1)_+ & \dots & (x_1 - \kappa_M)_+ \\ \vdots & & \vdots \\ (x_n - \kappa_1)_+ & \dots & (x_n - \kappa_M)_+ \end{bmatrix}$$

Mixed model for estimating a penalized spline

Given τ^2 and σ^2 , we seek to minimize

$$\frac{1}{\sigma^2} \|\mathbf{Y} - \mathbf{X}\beta - \mathbf{Z}\Theta\|^2 + \frac{1}{\tau^2} \|\Theta\|_2^2,$$

which we can think of ridge regression with penalty $\lambda = \frac{\sigma^2}{\tau^2}$.

We estimate all parameters from the data using the mixed modeling tools we previously learned, and thus obtain a model-based estimate of λ .

$$\|\mathbf{Y} - \mathbf{X}\beta - \mathbf{Z}\Theta\|^2 + \frac{\sigma^2}{\tau^2} \|\Theta\|_2^2$$

if σ^2 is large, large penalty

Selecting penalty using mixed models

- In mgcv::gam, we can use the option `method='REML'`
- Often results in greater smoothing (*smaller effective degrees of freedom*)

```
> fit.reml = gam(alldeaths~s(Temp,bs='tp',k=10),method="REML", data= health)
> summary(fit.reml)
```

Family: gaussian

Link function: identity

Formula:

```
alldeaths ~ s(Temp, bs = "tp", k = 10)
```

Parametric coefficients:

| | Estimate | Std. Error | t value | Pr(> t) |
|-------------|----------|------------|---------|------------|
| (Intercept) | 143.9168 | 0.3539 | 406.7 | <2e-16 *** |

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Approximate significance of smooth terms:

| | edf | Ref.df | F | p-value |
|---------|-------|--------|-------|------------|
| s(Temp) | 5.499 | 6.665 | 86.66 | <2e-16 *** |

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

R-sq.(adj) = 0.24 Deviance explained = 24.3%

-REML = 7555.7 Scale est. = 228.66 n = 1826

Estimate the slope at a particular x_i

In linear regression $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$.

In GAMs, we have $\hat{y}_i = \hat{\beta}_0 + \hat{g}(x_i)$, and slope changes with x_i .

What is the rate of change at 40 degrees Fahrenheit?

$$\hat{g}'(40)$$

```
> # visually check whether this is consistent with the plot  
> newd <- health[1, ] # grab any row; we are going to change temperature only  
> newd$Temp <- 40 - 1e-05 # subtract some small number  
> y1 <- predict(fit.reml, newd)  
> newd$Temp <- 40 + 1e-05 # add some small number  
> y2 <- predict(fit.reml, newd)  
> (y2 - y1)/2e-05  
49  
-0.525
```

Interpretation

We interpret smoothers $\hat{g}(x_i)$ by looking at plots.

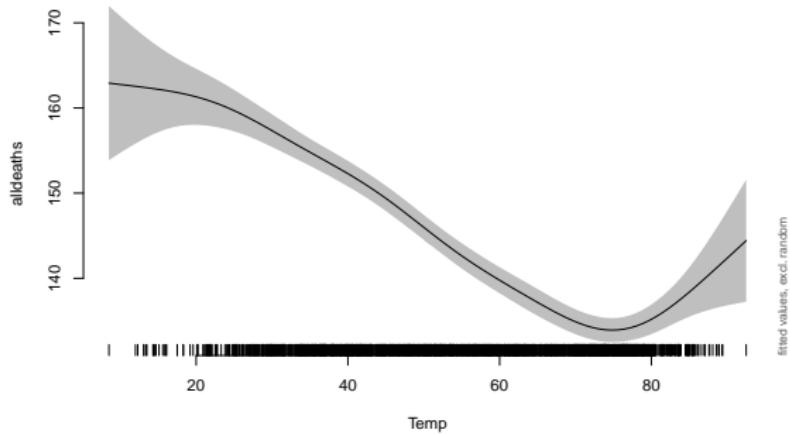
We can add some details regarding the slopes at particular x_i .

Deaths are highest at cold temperatures (< 10 degrees F) and slightly decreasing until approximately 20 degrees. Then deaths decrease at a similar rate from approximately 25 to 75 degrees. The number of deaths decreases by approximately 0.5 people / degree in a neighborhood of 40 degrees. Then the number of deaths starts to increase around 75 degrees. At 85 degrees, the number of deaths increases by approximately 0.8 for every 1 degree increase in temperature.

$$\hat{g}'(40)$$

$$\hat{g}'(85)$$

$g(x)$ Interpretation



Additive model with random intercept

GAMM: generalized additive mixed model

Recall the Nepal arm circumference dataset.

Data on 200 children collected at a maximum of 5 time points about 4 months apart.

Consider a non-linear effect of age and a random intercept:

$$\text{arm}_{ij} = \beta_0 + g(\text{age}_{ij}) + \theta_i + \varepsilon_{ij}$$

$$\theta_i \stackrel{\text{iid}}{\sim} N(0, \tau^2)$$

$$\varepsilon_{ij} \stackrel{\text{iid}}{\sim} N(0, \sigma^2)$$

$$\theta_i \perp \varepsilon_{ij}$$

Additive model with random intercept

Syntax used to denote $c(1:d)$

```
fit.gamm = gam(arm~s(age)+s(id,bs = 're'),method='REML',data=nepal)
> gam.check(fit.gamm)
```

```
Method: REML Optimizer: outer newton
full convergence after 6 iterations.
Gradient range [-4.190947e-07,-8.779523e-09]
(score 894.1436 & scale 0.2364939).
Hessian positive definite, eigenvalue range [1.077516,461.7172].
Model rank = 207 / 207
```

Basis dimension (k') checking results. Low p-value ($k\text{-index}<1$) may indicate that k is too low, especially if edf is close to k' .

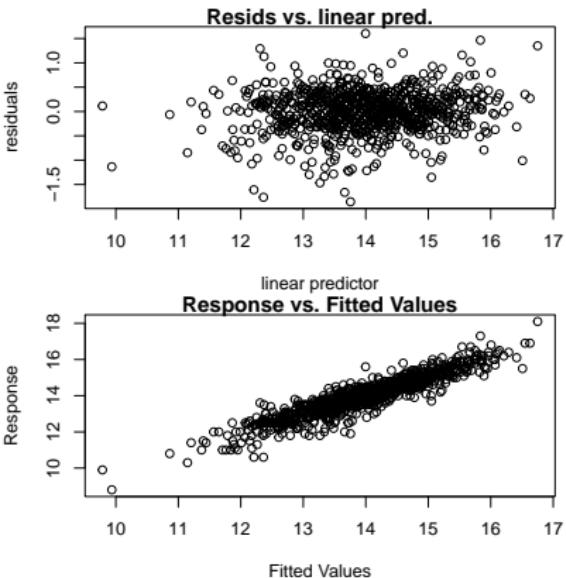
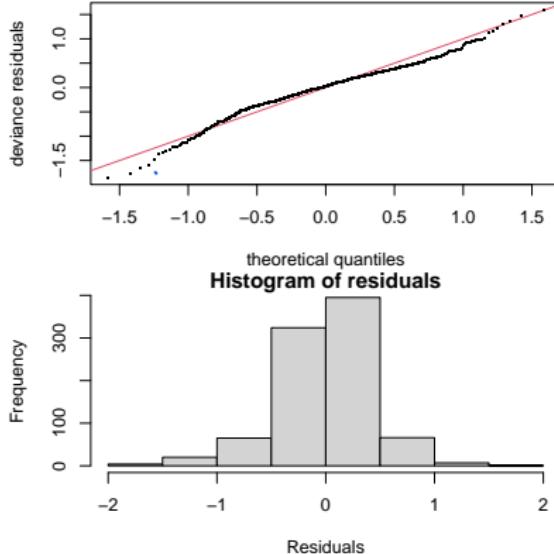
| | k' | edf | k-index | p-value |
|--------|--------|--------|---------|---------|
| s(age) | 9.00 | 7.05 | 1.03 | 0.79 |
| s(id) | 197.00 | 181.43 | NA | NA |

=#of participants - 1

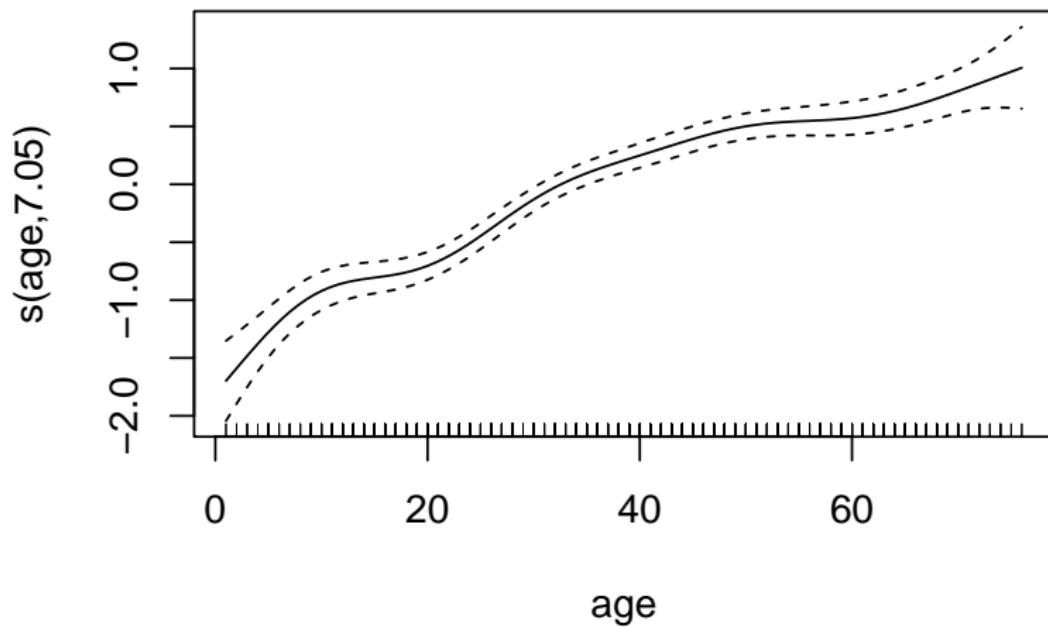
- I tend to prefer REML
- EDF somewhat close to k' . Other diagnostics okay.
- R code looks at $k = 20$ and results are similar (edf=8.5), so either this model or the one with $k = 20$ is fine.
- For random effect, k' equals number of subjects. Good to check this because will try to fit a smooth if you don't code as a factor.

\sim my treatment effects in an annoying way, but these are actually Ω_i from random intercept

Additive mixed model with random intercept



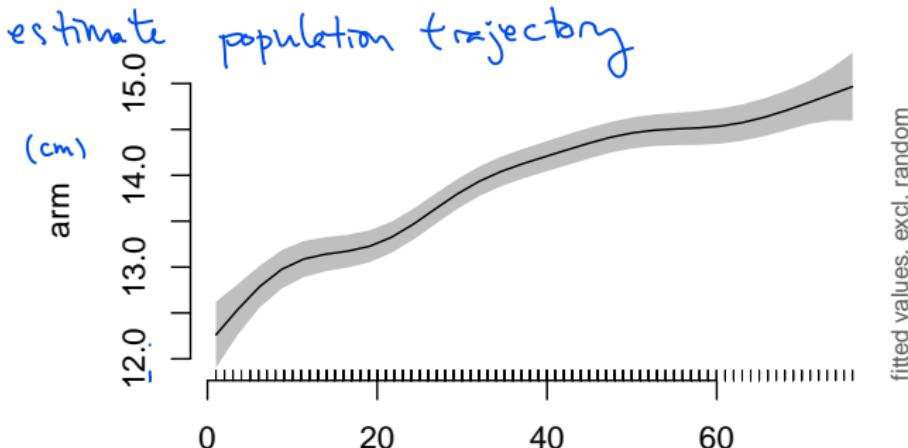
Effect of age on arm circumference



Effect of age on arm circumfenece

This plot includes the intercept:

```
> library(itsadug)
> plot_smooth(fit.gamm, view='age', rm.ranef=TRUE)
Summary:
* age : numeric predictor; with 30 values ranging from 1.000000 to 76.000000.
* id : factor; set to the value(s): 3. (Might be canceled as random effect, check below.)
* NOTE : The following random effects columns are canceled: s(id)
```



Effect of age on arm circumference

We can also plot a few of the curves+random effects.

```
myylim=c(9,18)
plot_smooth(fit.gamm.reml,view='age',cond=list(id=10),col='orange',ylim=myylim)
plot_smooth(fit.gamm.reml,view='age',cond=list(id=40),col='red',add=TRUE,ylim=myylim)
plot_smooth(fit.gamm.reml,view='age',cond=list(id=120),col='purple',add=TRUE,ylim=myylim)
plot_smooth(fit.gamm.reml,view='age',cond=list(id=50),col='turquoise',add=TRUE,ylim=myylim)
```

